FontLAB

FONTLAB Fontographer 5

FONT EDITOR FOR DESIGNERS –
POSTSCRIPT, TRUETYPE, UNICODE, OPENTYPE
USER'S MANUAL FOR MAC OS X

Copyright © 2005-2012 by Fontlab, Ltd. All rights reserved.

Cover illustration: Paweł Jońca, pejot.com

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written consent of the publisher. Any software referred to herein is furnished under license and may only be used or copied in accordance with the terms of such license.

Fontographer, FontLab, *FontLab logo*, *ScanFont*, *TypeTool*, *SigMaker, AsiaFont Studio, FontAudit* and *VectorPaint* are either registered trademarks or trademarks of Fontlab, Ltd. in the United States and/or other countries.

Apple, the *Apple Logo*, *Mac*, *Mac OS*, *Macintosh* and *TrueType* are trademarks of Apple Computer, Inc., registered in the United States and other countries.

Adobe, *PostScript, PostScript 3, Type Manager, FreeHand, Illustrator* and *OpenType logo* are trademarks of Adobe Systems Incorporated that may be registered in certain jurisdictions.

Windows, *Windows 95, Windows 98, Windows XP, Windows NT, Windows Vista* and *OpenType* are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

IBM is a registered trademark of International Business Machines Corporation.

Other brand or product names are the trademarks or registered trademarks of their respective holders.

THIS PUBLICATION AND THE INFORMATION HEREIN IS FURNISHED AS IS, IS SUBJECT TO CHANGE WITHOUT NOTICE, AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY FONTLAB, LTD.

FONTLAB, LTD. ASSUMES NO RESPONSIBILITY OR LIABILITY FOR ANY ERRORS OR INACCURACIES, MAKES NO WARRANTY OF ANY KIND (EXPRESS, IMPLIED OR STATUTORY) WITH RESPECT TO THIS PUBLICATION, AND EXPRESSLY DISCLAIMS ANY AND ALL WARRANTIES OF MERCHANTABILITY, FITNESS FOR PARTICULAR PURPOSES AND NONINFRINGEMENT OF THIRD PARTY RIGHTS.

User manual release 5.2 [3/2012]

Contents

CONTENTS 3

INTRODUCTION 13

Getting the most out of your Fontographer materials 14
- System requirements 15
- Support 15

BASICS 17

Setting preferences 18
- International 18
- Undo 18
- Autosaving 19
- Linespacing 19
- Editing 19
- Point display 21
- Windows and dialog boxes 21

The UPM size 22
- Ascender 22
- Descender 23
- Line gap 23
- Safe zone top and bottom 24
- x-height 24
- Origin line 25
- Baseline 25
- Basepoint 25
- Width 25

The font window 26
- Title bar 26
- Scroll bar 27
- View by pop-up 27
- Glyph slots 28

Opening windows 30

The outline window 31
- The info bar 32
- Changing glyphs in the outline window 33
- Path direction indicator 33

The bitmap window 34

The metrics window 36

Selecting glyphs in the font window 38

Viewing glyphs in the outline window 40
- Viewing and sizing the glyph 40
- Using the magnifying tool 40
- Fit in Window 41

Magnification 41
Scrolling with the hand tool 42
Moving by dragging 42

Viewing modes 43
Preview 43

Selecting and deselecting objects 45
Drag-selecting objects 46
Click-selecting objects 47
Shift-selecting objects 47
Selecting parts of a path 48

Drawing layers 50
Outline layer 50
Template layer 51
Guides layer 52
Hints layer 53

Using the palettes 54
Layers palette 54
Tool palette 54
Control point tools 59
Transformation tools 61
Constraining tools 69

Fontographer's menus 71
Undo and redo 71

Closing and Quitting 72
Closing 72
Quitting 72
Folders and paths 73

MODIFYING YOUR FONTS 75

Steps to modifying your font 76

Opening a Font 77
Existing fonts 77
Opening Fonts with Drag-Drop 78
Opening Recently Used Fonts 78
Font Formats 79
Glyphs Number 79
Opening Macintosh fonts 80
Opening Font Collection 81
New fonts 81

Changing the glyph's weight 82

Naming your font 85

Saving your work 86
Save 86
Save As 87
Reverting to the last saved version 87

Generating your font 88

Installing the font 89

Using the font 90

About font piracy 90

Creating an oblique font 91
Skewing multiple glyphs 92

Creating a fraction using composite glyphs 93

Creating a ligature 97

Changing the glyph width 99

Creating a condensed glyph or font 100

Setting the basepoint 102

CREATING NEW FONTS 105

Autotracing 107

Advanced tracing options 111
Curve fit 111
Allow curve fit errors 112
Balance lines 112
Eliminate close points 112
Make straight lines 112
Look for cusps 112
Treat nearly flat paths as straight lines 113
Find extrema points 113

Transformation options 114
Flip 114
Move 115
Rotate 117
Scale 118
Scale uniformly 119
Skew 120
Multiple transformations 121
3D transformations using the Transform dialog box 122

Guidelines 127
Setting guidelines 127
Setting guidelines from the Font Info dialog box 128
Adding new guidelines 129
Snapping to guides 131

Creating a stroked font 132
Outline versus stroked glyphs 132
Setting stroke attributes 133
End caps and joins 135
Expand stroke 136
Clean Up Paths 138

Creating calligraphic glyphs 140

Calligraphic tutelage from Judith Sutcliffe 142

Creating variable weight glyphs 148

Blend fonts to create new fonts 150
When things go wrong... 153

Jonathan Hoefler says: 155

ALTERING OUTLINES 157

Altering a logo 159
Pasting EPS outlines from the Clipboard 160

Paths and points 161
Closed paths 161
Path direction and fills 162
Normal fill 163
Even/odd fill 164
Correct path direction 164
Reverse path direction 165

Types of points 166
Curve points 166
Corner points 167
Tangent points 168
Selecting multiple points 169
Changing a point type 170
Inserting points 171
Duplicating points 172
Power duplicating 173
Removing points 174
Splitting a path 175
Splitting line segments 176
Joining points 177
Adding serifs 178
Merging points 179
Moving a point 180
Demagnified move 181
Keyboard commands to move points 181
Accurate point placement 182

Point and path preferences 184
Path display 184
Point display 186
Show and hide control points 187
Editing and placing BCPs 188
BCP principles 188
Dragging a control point's BCPs 189
Dragging a curve point's BCPs 189
Dragging a corner point's BCPs 190
Dragging a tangent point's BCPs 191
Retracting BCPs 191

Auto curvature 192

EDITING BITMAPS 195

Using the bitmap window 197
The bitmap window 198
Editing a bitmap 199
The central edit area 201
Ascent and descent values 201
Offset and width values 202
Visible layers 203
Tools in the bitmap window 204
Undo and redo 209
Richard Beatty Says: 209

Changing bitmap views 210
Enlarging using the View menu 210
Switching glyphs in the bitmap window 210
Next and previous glyph 211
Next and previous point size 211

When should you recalculate bitmaps? 212
Preserving your original bitmaps 213

METRICS – SPACING AND KERNING 215

Spacing 219

Pair kerning 222

Auto spacing 224

Auto kerning 226

The metrics window 228
Editing a Text String 230
Glyph display 231
The spreadsheet area 234

Vertical spacing & kerning 236

Importing metrics 237
Clearing kerning pairs 237

Exporting metrics 238
The Fontographer Metrics file 239
Copying widths 240

More powerful spacing and kerning commands 241
Setting widths 241
Equalizing sidebearings 242

Advanced metrics operations 243
Setting metrics 243
Assisted metrics 245
Metrics assistance 246
Kerning assistance 249
Advanced auto spacing 253
Advanced auto kerning 256

FONT INFO 261

Names 262
Name fields 263

Dimensions 265

Encoding 267
Encoding options 268

Credits 270

Licensing 272
Advanced Licensing 274

Recommendations 281

Exporting EPAR 283

PRINTING 285
Sample text 287
Sample file 290
PostScript file 291
Key map 292
Kerning pairs 293
Sample Glyphs 294
The print header 296

GENERATING AND EXPORTING FONTS 297

Before you do anything... 298

Relevant Font Formats 299
OpenType PS 299
TrueType / OpenType TT 300
Macintosh TrueType 301
Macintosh Type 1 301
Windows Type 1 302

Easy or advanced? 303

Generating cross-platform fonts 305
OpenType Options 306

Generating Macintosh fonts 307
Mac Type 1 Suitcase 308
Mac Type 1 Font 310
TrueType suitcase 311
PostScript Type 3 312
Other Type 3 Formats 314
When should you use hints? 315
A word about flex 315

Generating Windows fonts 316
PostScript Type 1 316
Symbol encoded Windows fonts 318
PostScript Type 3 318

Pack your suitcase: bitmap fonts 319
Bitmaps vs. outlines 319
Adding bitmap sizes 320
Deleting bitmap sizes 320
Bitmap format 321

Exporting files 322
Exporting EPS files 322
Exporting BDF files 323

CREATING A FONT FAMILY 325

Menu grouping and style linking 327

Families: Windows, Sun, NeXTSTEP 329

Font families on the Macintosh 330
How Style Merger works 332
Things you should know about Style Merger 334

INSTALLING AND REMOVING FONTS 335

Installing Macintosh Type 1 fonts 336
Installing Type 1 fonts in Mac OS X 336

Installing Other fonts in Mac OS X 337

Installing Windows fonts 338
Installing fonts in Windows XP 339
Installing fonts in Windows Vista/7 340

Removing installed fonts 341
Removing a Macintosh font 341
Removing a Windows font 341

OPENTYPE FONTS 343

Font Features 344
Features and Lookups 346
Scripts and Languages 346

Feature Definition Language 347
Language Syntax 348

Importing OpenType Fonts 356

Generating OpenType Fonts 357
Preparing OpenType features 358

Known Features 370
init, medi, fina and isol Features 370
Latin Features 371

EXPERT ADVICE 375

General preferences 377
International 377
Autosaving 377
Undo Settings 377
Sounds 378

Fonts preferences 379
Linespacing Auto-Calculation 379
Opening Type 1 fonts 379

Editing preferences 380
Smooth outline 380
Distances 381
Snapping 381
Paths 382

Point display 383

Windows and dialog boxes 385
Windows 385
Dialogs 385

DSIG preferences 386

Restore Defaults 387

Font blending – the technical details 388
The blending process 389

Font hinting 392
Are you still with us? 392
What is hinting all about? 392
Hinting controls 397
Autohint 398
Editing hints in the outline window 399
Removing hints 400
Making new hints 400
Selection Info for hints 401
Vertical Alignment Zones 403
Common Stems 405
What happens when Fontographer opens up PostScript Type 1 fonts 405
What happens when Fontographer opens TrueType fonts 406

Using a text editor to tweak Fontographer 5 on your Macintosh 407
Adding custom encodings 407
A special note to designers of non-Roman Macintosh fonts 410
Setting developer IDs 411
Customizing Sample Text printout 411
Character palette 412
Multiple master fonts 413
A quick overview 414

The genesis of a multiple master font 417
Planning multiple master fonts 419
Set up from file 424
Generating a multiple master font... 432

REFERENCE 435

Windows in Fontographer 1. Font window 436
View by menu 437
Glyph properties 440
Searching for glyphs 440

2. Outline window 441
Tool palette 442
Layers palette 448
Changing and hiding layers 449
Magnification 449
Switching glyphs 449

3. Bitmap window 450
Tool palette 451
Ascent/Descent/Offset/Width 453
Recalculate From outline 453
Scrolling 453
Switching characters 453
Changing point sizes 453

4. Metrics window 454
Kerning and sidebearing lines 455
Key commands to change spacing and/or kerning 455
Show kerning 456
Load Text 456

Menus 457

The Fontographer menu 457
The File menu 458
The Edit menu 461
The View menu 463
The Element menu 465
The Points Menu 474
The Metrics menu 476
The Hints menu 478
The Window menu 479

Special keys 480
Keyboard alternatives 480
The Full List of Keyboard Shortcuts 481

APPENDIX A. TIPS 485

Answers to commonly asked questions 487

APPENDIX B. AN ANNOTATED BIBLIOGRAPHY OF TYPOGRAPHY AND THE OTHER ARTS OF THE BOOK 493
Overviews of Printing Types 495
History and Development of Lettering and Letterforms 497
Type Designs from Various Periods 498
Typography 500
Book Design 502
Type Designers 503
Typeface Reference Works 505
History of Printing 506
Letterpress Printing 507
Other Book Arts 509
Bibliographies 510

APPENDIX C. GENERAL INFORMATION 513

Fontographer background 513

Bitmap background 515
Filling techniques 518

Glossary 519

INDEX 533

ACKNOWLEDGMENTS 543

Introduction

Fontographer makes it easy to create new typefaces or add your logo to existing typefaces. Fontographer's drawing tools help you create a professional-quality character in minutes and print that character on any PostScript or TrueType compatible printer. With Fontographer and your personal computer, you can create designs that match those produced by professional typographers.

Fontographer 5 generates ATM-compatible Type 1 fonts, as well as Type 3 PostScript fonts, TrueType fonts, OpenType fonts and multiple masters on the Macintosh, and Encapsulated PostScript (EPS) files. You can import EPS files directly, and use their outlines in the drawing window. You can also use metrics information from a variety of sources, and export information to those sources as well. Additionally, you can import kerning tables from Adobe Font Metrics (AFM) and other files. You can also take advantage of the PostScript graphics you create in FreeHand and Adobe Illustrator by pasting them directly into your characters.

Many dialog boxes in the program give you two options: Easy and Advanced mode – letting you have total control of the program if you want it, or allowing you to rely on its simple and automatic settings.

For advanced users who don't always want to rely on automatic hint settings, there is a menu of hinting controls. And in the Metrics arena, Fontographer lets you space and kern faster and easier than ever. With auto space, auto kern, and assisted kerning and metrics you can save yourself from having to kern and space each individual character or font separately. Fontographer can do it automatically, or you can use the same kerning and spacing information from one font, for others that kern and space similarly.

So whether you are a novice or an experienced graphic designer, Fontographer allows you to assign your characters and graphic images to any key or combination of keys, and gives you the added ability to instantly repeat and resize these images in any application.

Getting the most out of your Fontographer materials

This version of the *Fontographer User manual* is designed for Macintosh users of Fontographer 5 only. If you have Fontographer 4.7 for Macintosh you need the previous version of the manual.

We placed keyboard alternatives after certain menu commands; get into the habit of using these quick commands that our more experienced users prefer.

The *Fontographer User manual* assumes that you are familiar with the computer and that you have a working knowledge of how your system operates. If you need more information on these topics, refer to your Macintosh owner's manual.

We don't have a separate tutorial. Ours is included right in this *Fontographer User manual*. Tutorial icons are interspersed throughout the manual.

Tutorial icon

If you are a first-time user, the best way to start learning Fontographer is to use its tutorial. Ours is a novel approach to tutorials... but we think you'll enjoy this new method of introducing you to Fontographer's basics.

The tutorial icon appears next to exercises in some chapters. We've also placed icons next to text you should read before you start the exercise itself. *We strongly recommend that you work your way through the tutorial exercises in order*.

The tutorial is meant to be a guide to give you practice using some of Fontographer's standard features. It does not cover all features, nor even most of them, but should be used as a starting place, if you are unfamiliar with the program.

Once you have completed the tutorial exercises, review the rest of your *Fontographer User manual* for information that will help you plan and create your fonts.

System requirements

To run the Macintosh version of Fontographer 5.2, you must have an Intel-based Macintosh computer running Mac OS X 10.5 or later operating system with 35 MB of hard drive space, and at least 50 MB of available RAM. Fontographer 5.2 is an Intel only application.

The previous Macintosh version of Fontographer 5.1 can run both on a Power PC and Intel-based Macintosh computers running Mac OS X 10.4 or later operating system. Fontographer 5.1 is a universal binary application.

You must purchase additional copies of Fontographer in order to run more than one copy at the same time. For additional copies, contact Fontlab Ltd. at orders@fontlab.com or your nearest Fontographer dealer.

Support

For further information about Fontographer browse to the Fontographer home page:

http://www.fontlab.com/fontographer/

Use the following address to get Fontographer updates and upgrades:

http://www.fontlab.com/support/

In case of any questions or to report possible bugs in Fontographer or any other of our products browse to:

http://www.fontlab.com/support/

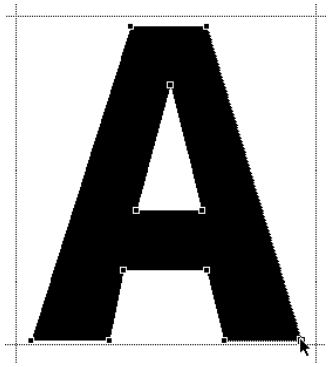
The Fontographer technical notes collection is available for download (in PDF format). Technical notes address various issues and problems concerning Fontographer 4.x:

http://www.pyrus.com/downloads/FG4Technotes.pdf

Basics

If you're a professional typographer, Fontographer offers you tools you can use to create professional typefaces. You can use Fontographer as a drawing tool, not just as a font editor. Create logos quickly and easily, just like with a drawing program, or scan any image from a book or other source and let Fontographer auto trace it. Metric tools, including automatic kerning, spacing, and metric tables help you create a consistently spaced and kerned font faster than ever before.

If you're a novice, Fontographer gives you the tools you need to quickly create and change fonts without intensive study and practice, plus the opportunity to increase your skill to a professional level.

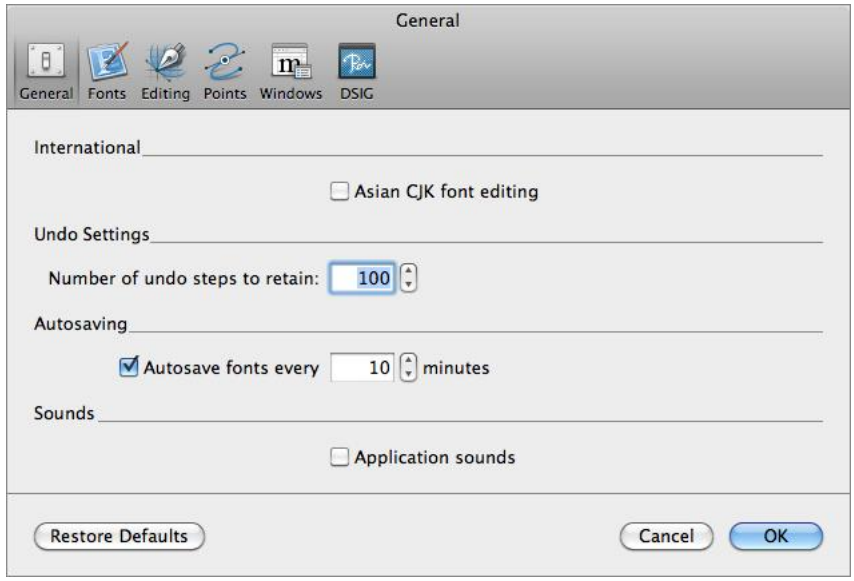


Create glyphs from scratch or by autotracing a scanned image.

1

Setting preferences

Fontographer 5 automatically places a preferences file in the user's Library/Preferences folder called *com.fontlab.fontographer.plist*. To set preferences, choose **Preferences** from the **Fontographer** menu. See "General preferences" in Chapter 13, "Expert Advice", for more information.



International

Use this setting if you are going to work with Asian CJK (Chinese, Japanese or Korean) fonts. It allows you to define **vertical glyph metrics**: the vertical advance "width" (called *vertical advance vector*) for Asian glyphs used to type in vertical direction from top to bottom.

Undo

Use the Preferences dialog to select the number of **Undo** commands you want to allow. Choosing **Undo** from the **Edit** menu will undo the last command. You can undo up to 100 commands using Fontographer's default setting or change the number of undo levels in the Preferences dialog box. The maximum number of undo levels allowed is 256. The more undo levels you allow, the more memory you use, and the less you have for the font you're editing. Fontographer will automatically throw away undo levels when there isn't enough memory. But this is only a problem for older systems with low RAM available.

Autosaving

If you want to protect yourself from system or program crashes you can use the Autosave function that will periodically save the current font.

Use the check box to activate Autosave and enter the time interval (in minutes) at which you want to save the font. The font will be saved into the same folder where your font was last saved.

Linespacing

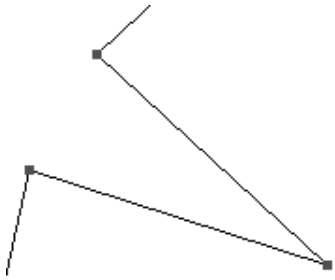
These two options define how Fontographer behaves when it needs to import, calculate and export different linespacing font values. Unless you do not need this for some special purpose leave the first option selected. This will keep linespacing in your font the same for all platforms and applications.

Editing

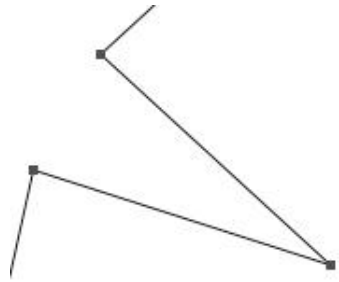
To set Preferences related to editing behavior, choose **Editing** in the header of the Preferences dialog box.

Smooth outline

This setting lets you to select between non-anti-aliased and anti-aliased rendering of the outline. It doesn't influence the generated font but only the screen preview.



smoothing is off

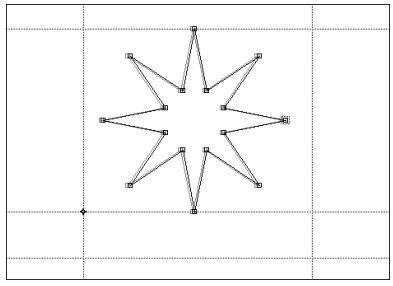


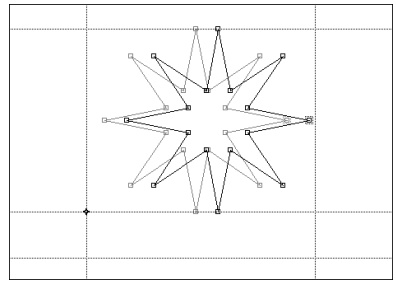
smoothing is on

Cursor-key distance

This preference setting lets you set the distance the arrow keys will move a selected object (in em units) when they are pressed. If you press an **ARROW** key while holding down the **OPTION** key, the distance the object moves is divided by 10; if you press an **ARROW** key while holding down the **SHIFT** key, the cursor distance will multiply by 10.

For example, the following images show the effect of using the **ARROW** keys to move selected objects. In the first example, the **ARROW** key is used by itself, moving the starburst 10 em units closer to the bar. In the second example, the starburst is moved with the **ARROW** key and the **SHIFT** key held down, making the distance 100 em units. Holding down **OPTION-ARROW** key divides the value by 10 – making it one em unit.

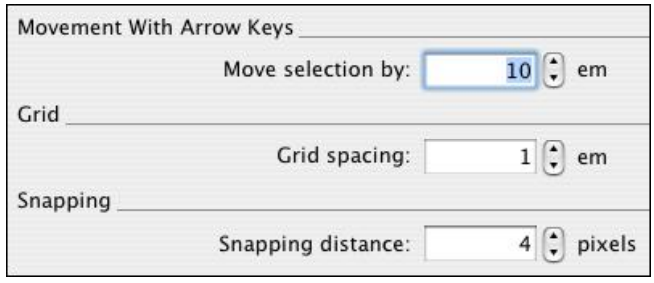




Grid spacing

There is an invisible key grid in the outline window that can be used to position glyphs. When you choose **Snap to Grid** from the **View** menu, **SHIFT-COMMAND-I**, any objects you move will automatically snap to the nearest grid.

You can change the distance between grids from one em unit to another distance. The Snap-to distance represents the distance at which an object will snap to a grid intersection. The Snap-to distance preference is entered in screen pixel units. The window's current level of magnification will affect how this operates.

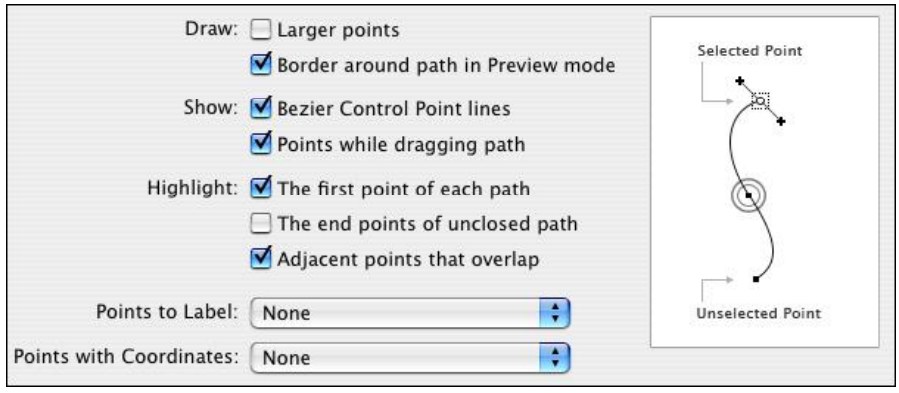


Set the cursor distance, grid spacing, and snap-to distance.

➲ ***Tip:*** Listen to your snaps by turning the application sounds on in the General preferences.

Point display

The Point display dialog box contains several options for viewing points. Decide whether you want to view large or small points, Bézier control point (BCP) lines, or just BCP points. You can highlight the path origin, or the ends of unclosed paths. You can also highlight adjacent overlapping points or points on a path you are dragging.



Labels can be shown for all points or only for selected points and BCPs (numbered or lettered in sequence). Or you can show the x and y coordinates for each point or for selected points only.

Fontographer will use the default settings unless you choose otherwise.

Windows and dialog boxes

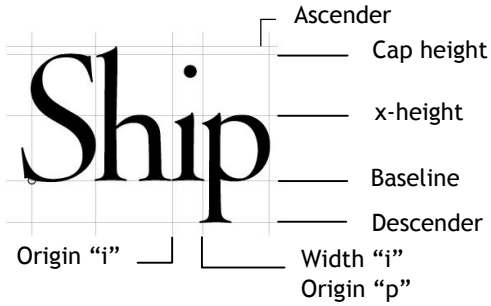
Preferences lets you set the placement of windows, remember values and positions of dialog boxes, control how glyphs fit into windows, and have palettes move with the windows.

For more information about setting any of the preferences, see “General preferences” in Chapter 13, “Expert Advice”.

The UPM size

Each glyph fits into a rectangle called an em square. An em square is usually a square the size of a capital letter “M”, which extends to the descender line so it is also called UPM (units per M). The em square received its name from the capital “M” that filled the piece of metal used to form the type body in early printing days. The font’s height is determined by the distance between the ascender and descender. All fonts are normalized inside the printer, at an em square one point (approximately 1/72 inch) high. Defining a font in this fashion allows a single outline to be scaled to any size requested. Think of UPM units as relative coordinates rather than as specific physical distance.

Refer to the figure below as you follow the overview.



Note: The ascender and the descender in “Ship” lie at the top and the bottom of the em square, respectively. This may not always be the case.

Ascender

A font’s maximum distance above the baseline is called its ascender. Fontographer automatically places an ascender guideline in the outline window at the top of the em square (UPM size). You can change it by choosing **Font Info** from the **Element** menu.

➲ ***Tip:*** Glyphs in Type 1 fonts normally should not extend above the ascender line. Any glyph that extends above the ascender line or below the descender line may have its bitmap representation vertically scaled to fit between the ascender and descender. See “When should you recalculate bitmaps?” in Chapter 5, “Editing Bitmaps”, for more information. See also Safe zone top and bottom values.

Descender

A font's maximum distance below the baseline is called its descender. Fontographer automatically places a descender guideline at the bottom of the em square (UPM size) in the outline window. The descender line lies at the lower vertical position specified when you created the font. You can change this setting by choosing **Font Info** from the **Element** menu.

➲ ***Tip:*** Usually glyphs do not drop below the descender line or they may interfere with glyphs on the next line. Sometimes, however, this is a desired effect, such as instances when vertical bars must connect from one line to another (for example, borders). In this case use the Safe zone top and bottom values described below.

You can use ascender and descender to control the UPM size in which your glyphs are drawn. Fontographer's default values are: ascender 800 and descender -200. You may change these proportions as needed. The default values provide UPM size of 1000 units.

Line gap

Line gap (or leading) is the space between the descender of the previous line of text and the ascender of the next line of text. It specifies how much space there is between lines. The term leading comes from earlier days when thin strips of lead were inserted between lines of text to provide line-to-line spacing. Currently, some applications ignore this line gap value. Fontographer's default value for line gap is 20% of the UPM value. Line gap size is not used directly by a PostScript font but is used by Fontographer when calculating the default leading for bitmap generation.

Line gap is the space between the descender of the previous line and the ascender of the following line.

Safe zone top and bottom

In addition to the ascender, descender and line gap values, there is another pair of vertical font metrics: safe zone top and safe zone bottom lines.

Safe zone top is the line above which glyphs can be cropped (clipped, trimmed) in some applications. Safe zone bottom line — the line below which glyphs can be cropped. Some text layout systems lay out text so that the safe zone bottom of one line of text is immediately followed by the safe zone top of the next line of text. Any glyphs that go beyond those lines will be either cropped in Windows GDI applications, or will be squeezed in Mac OS Classic applications. Therefore the distance between those lines is called "safe zone".

These values are calculated automatically by Fontographer in most cases.

Note: Safe zone top, ascender, descender, line gap, safe zone bottom are **linespacing values**, so they should ideally be consistent for all fonts in the font family. In OpenType fonts, the safe zone top value will always correspond to OS/2 Win Ascent value and the safe zone bottom value will always correspond to OS/2 Win Descent. In other words, the distance between OS/2 Win Ascent and OS/2 Win Descent can be referred to as the safe zone.

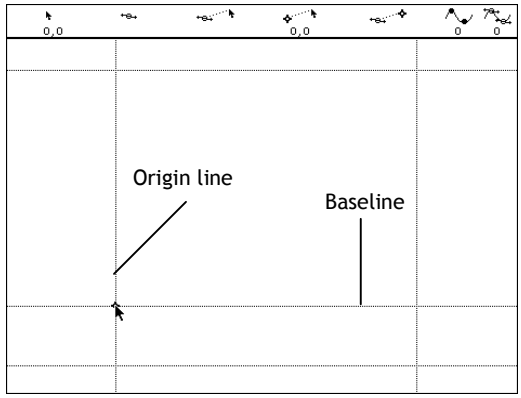
x-height

The x-height marks the top of lowercase letters such as “x” and “o.” Fontographer does not create this automatically, but you can easily create one yourself in the guides layer. You can position it anywhere, since it is only a guideline you’re creating.

➲ ***Tip:*** Faces with taller x-heights are generally perceived as larger and more readable than those with smaller x-heights.

Origin line

The origin line of the em square is always at a horizontal location of zero:



Baseline

The baseline is the line upon which capital letters sit. When printing mixed fonts on a line, all the different fonts' baselines line up with one another. The baseline position does not need to be explicitly specified, since it is always at a horizontal location of zero.

Basepoint

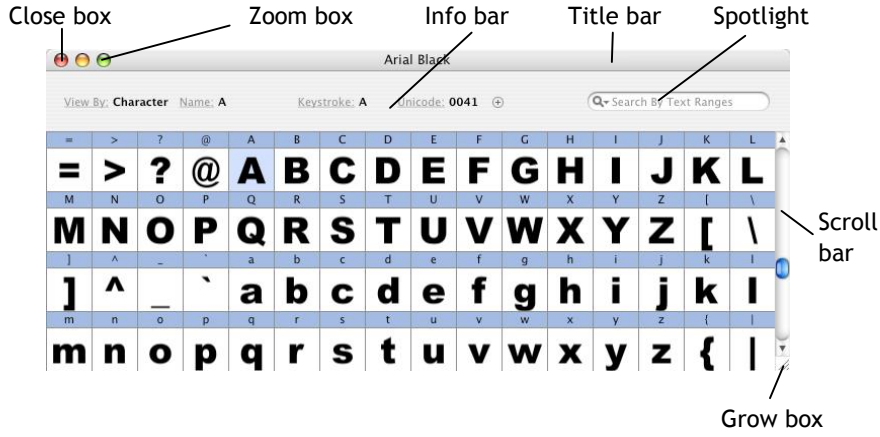
The ✦-shape at the intersection of the origin line and baseline is the basepoint.

This is a movable reference point for position measurements or as an aid to line up points. When you open a new glyph's outline window (or display a new glyph in an existing window), the basepoint is positioned at location 0,0 – at the origin point of the em square. Fontographer allows you to change the basepoint to another location by selecting a point and then choosing **Set Basepoint** from the **Point** menu.

Width

Width is a movable vertical line that specifies the width of each glyph. When the printer (or screen cursor) draws glyphs on a line, the invisible origin line of the next glyph is placed on top of the imaginable width line of the printed glyph. Widths may be zero but cannot be negative.

The font window



Fontographer's font window displays all the glyphs that make up a font. You can scroll through the sequence of glyphs with the standard scroll bar to the right.

The font window also has other standard features like close, grow, and zoom boxes.

Directly below the title bar is an info bar that displays font-related information. In the **View by** pop-up you are given a list of the ways to view font labels.

When you choose an item from the **View by** pop-up, you change the font labels within the font window display. There are thirteen different types of font labels. The character mode shows the symbols for each glyph in the Latin alphabet, while the keystroke mode shows you the keys you type to access the glyphs. In some cases the glyph symbol and the keystroke may be different.

You'll find more detailed information about font labelling information in "Glyph slots" on page 28.

You can see some of the codes used to name the selected glyph in the info bar displayed to the right of the **View by** pop-up:

Title bar

Fontographer's title bar is at the top of the font window. The name of the font appears here. You can move the font window around the screen by clicking its title bar, holding down the mouse button, and dragging the window to a new location.

Scroll bar

Standard scroll bar is drawn along the right side of the window. By moving the scroll bar, the window may be scrolled up or down to show other glyphs.

View by pop-up

The **View by** pop-up provides thirteen different ways to look at your glyphs. Each one can be selected to show a different kind of name in the font label.

- Name displays the glyph's name. Each glyph in a font must have its name.
- Character displays the system character corresponding to each slot in the font window.
- Keystroke displays the keyboard sequence used to access a glyph.
- Unicode codepoint is a value assigned to a glyph based on an international numbering system with 16-bit numeric designations for each glyph in every language used (or planned for use) in electronic information systems.
- Decimal code number is a number value assigned to a glyph based on a numbering system with a base of 10.
- Hexadecimal code is based on a numbering system with a base of 16.
- Octal code number is a number value assigned to a glyph based on a numbering system with a base of 8.
- Width view of a glyph displays the glyph's width in em units.
- Left sidebearing shows the distance from the left edge of a glyph to its origin, in em units.

- Right sidebearing displays the distance from the right edge of a glyph to its width line.

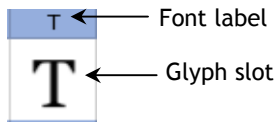
And when viewing Type 3 fonts:

- Fill tint shows you what percentage of black fills a glyph. A fill of 0 is white; a fill of 100 is black.
- Stroke tint lets you see the percentage of black in the stroked part of a glyph.
- Stroke weight displays the weight of the glyph's strokes in em units.

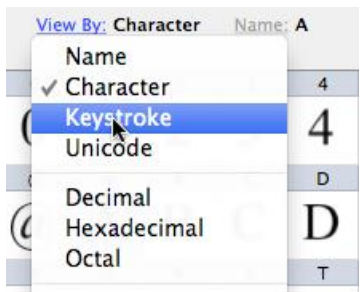
Glyph slots

Glyph slots are visible in the font window. Glyph slots can show the glyph paths. The scroll bar can be used to display additional slots not currently shown. The font label above the slot displays the code or letters that represent the glyph.

When the label is grey, the outline window contains paths, a defined width, or a scan. When the label is blue, it means the glyph has been changed.

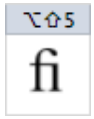


Font labels display in one of several ways. Viewing by character shows the Macintosh standard representation of that ASCII glyph code in the font label, while viewing by keystroke shows the keystrokes you'll need to press to type that glyph. Change view modes with the font window's **View by** pop-up.



Below, the Macintosh keystroke for "ellipsis" is shown. The font label displays the **Option-;** keyboard command that is needed to type it.

In this example, the keyboard command is **Shift-Option-5**.



For this glyph the keyboard command is **Shift-Option-K**.

A slot containing two asterisks signifies specific things in each mode. In the character and keystroke mode, it means that you cannot access the glyph from the keyboard. In the width, and left and right sidebearing modes, the double-asterisks tell you that the glyph is undefined. In the fill tint, stroke, and stroke weight modes the double-asterisks mean that the glyph is either unfilled, unstroked, or empty (and therefore the glyph shows no weight in em units).

With the character view, an empty, undefined glyph slot displays as two asterisks.

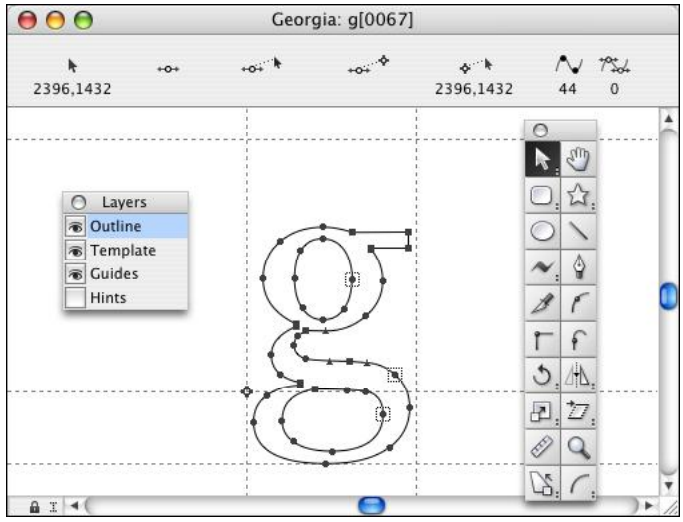
Opening windows

When you are familiar with the font window, look more closely at the glyphs you see in this window. There are three possible windows from which to view each glyph. Each window gives a different perspective of the glyph. The one you'll use the most often is likely to be the outline window, although you will also have uses for the metrics and bitmap windows.

- The outline window is where you'll do the most glyph editing. A glyph's outline reveals its filled or unfilled shape bounded by paths and points.
- The bitmap window contains a glyph image displayed in pixels. Changing this bitmapped image alters the screen font's appearance.
- The metrics window displays a filled glyph image and provides tools to modify kerning and spacing.

Any window can be accessed either from the font window or from Fontographer's **Window** menu. Fontographer lets you open several windows at once. The actual number depends on how much memory is available in your computer.

The outline window



To open a glyph's outline window:

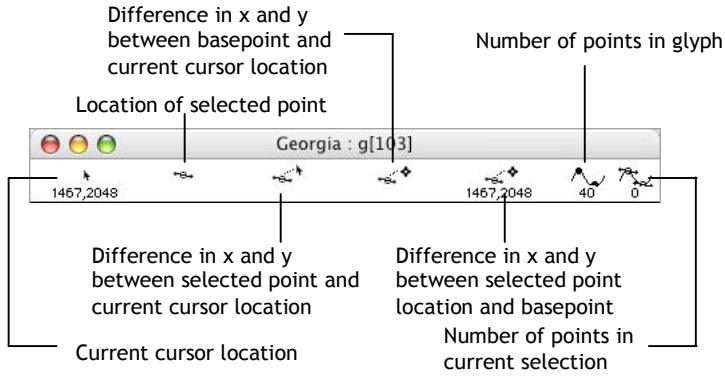
You can open the outline window in one of four ways:

- Double-click the selected glyph.
- Select the glyph and type **COMMAND-Y**.
- Select the glyph and press the **RETURN** key.
- Select the glyph and choose **Open Outline Window** from the **Window** menu.

The outline window is where most editing will take place. At the top of the window you see the title bar, which contains the name of the font and the font's glyph code. Beneath the title bar is the info bar, with the numeric coordinates of the cursor in relation to various objects or positions in the window. Each indicator represents a different distance, or the number of selected points, respectively.

Surrounding the screen image to the right and along the lower edge of the window are the scroll bars, which operate as they do in other programs. Close and grow boxes also work in the standard way.

The info bar



The info bar consists of a row of coordinate values displayed under the title bar. Each value represents the cursor's coordinates on an x and y axis in relation to a particular object or position.

For example, the first value lists the cursor's distance from the glyph's origin (in em units). As you move the cursor, the values change. This information can help if you use precise measurements in your drawings.

Click the **I** icon in the lower left corner of the outline window (next to the lock icon) to display or hide the info bar.

Changing glyphs in the outline window

Using the keyboard

You can move to another glyph's window by typing the desired glyph symbol on the keyboard. If you are moving from one outline window to another, the lock icon in the first window must be unlocked.

Lock icon

The lock icon is operated by clicking it or by pressing **Return**. This toggles the icon on or off.

The lock appears black when the glyph is locked into position. This prevents the glyph in the outline window from changing in case another key is accidentally pressed. When the lock is locked, the numeric keypad can be used to switch tools in the tool palette.

When the lock is white or unlocked, the outline window can be changed to another glyph's outline window by typing the new glyph's key.

Click the lock icon to lock or unlock the window

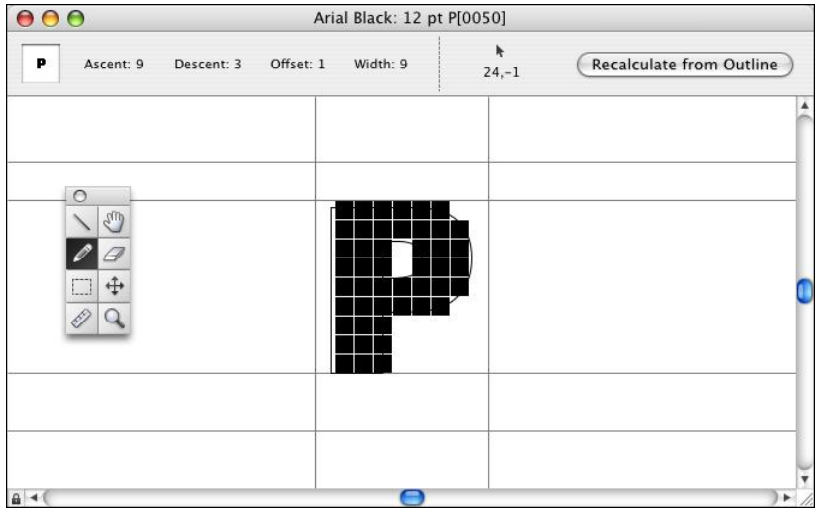
Using the View menu

When any glyph is selected in an active font window, or when any outline window is open, you can choose **Next glyph**, **Command-]**, or **Previous glyph**, **Command-[**, from the **View** menu to move forward or backward in the sequence.

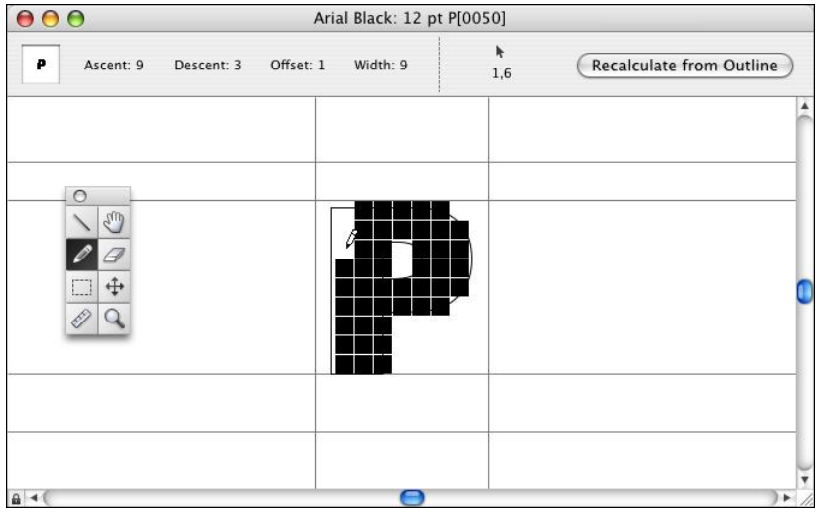
Path direction indicator

The path direction indicator tells you the direction of any selected path. You can use this feature to make sure that the paths of any glyph you create or edit are drawn in the necessary alternating pattern, from the outside to the inside of a glyph, so that the glyph will print with an accurate fill display. The arrow will point to the left or to the right, to indicate either a counterclockwise or a clockwise orientation. Clicking this indicator changes the direction of a selected path.

The bitmap window



Hand edit your bitmaps in the bitmap window.



Click with the pencil tool.

To open a bitmap window:

There are three ways to open a window for a bitmap glyph:

- Press and hold down the **Option** key while you double-click the desired glyph.
- Select the glyph and type **Command-J**.
- Select the glyph and choose **Open Bitmap Window** from the **Window** menu.

The bitmap window shows how glyphs look as they appear on the screen. However, bitmaps aren't normally used in printing your type or graphics, so time spent editing them will have no impact on the printed results. Therefore, tweaking and adjusting bitmaps may not be for you.

Actually, advances in programs related to type production have almost made editing bitmaps unnecessary. If you use modern Apple and/or Microsoft systems that use outlines for screen display, you don't need to spend time perfecting the look of the screen font.

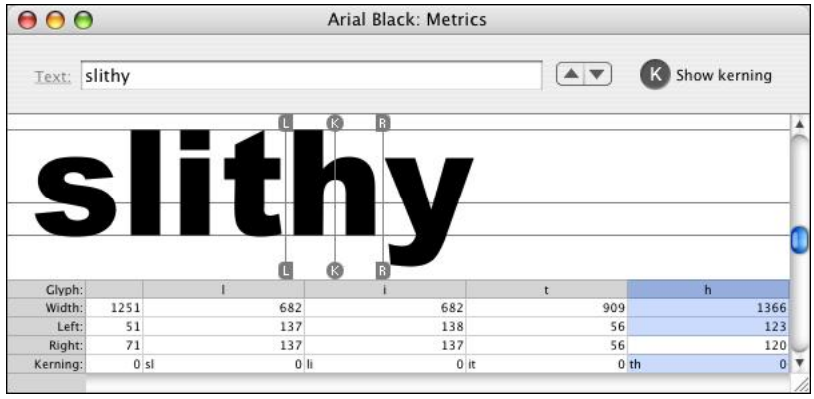
A title bar appears at the top of the bitmap window. The first element in the title bar is the point size, followed by the font name, and the glyph code and location of the bitmap being displayed. Directly underneath the title bar is the info bar, which includes the actual size view of the glyph, the cursor location, and the ascent, descent, offset, and width values for the glyph. To the right is the **Recalculate from outline** button, which allows you to recalculate the size and shape of the bitmap based on the dimensions of the outline. This gets rid of any changes you have made to the bitmap and makes it conform to the shape of the outline (any subsequent editing to the outline will affect the bitmap).

In the lower left corner you will find the lock icon for allowing or preventing access to other glyphs with the type of a keystroke. When locked, the current glyph's bitmap window is locked in place; when unlocked, the icon appears hollow and you can access other glyphs by typing their keystrokes.

You have eight tools in the bitmap window: the straight line, hand, eraser, pencil, marquee, move tool, measuring tool, and the magnifying tool.

These are explained in more detail in Chapter 4, "Altering Outlines".

The metrics window



To open the metrics window:

There are three ways to open the metrics window:

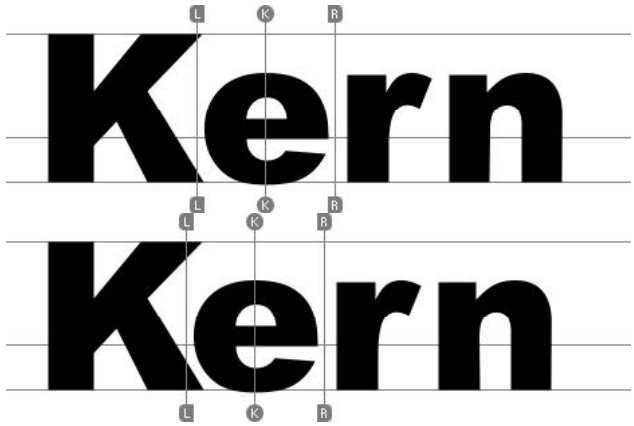
- Select a glyph in the font window and type **Command-K**.
- Select a glyph in the font window and then choose **Open Metrics Window** from the **Window** menu.
- Type **Command-K** to access the metrics window directly from the outline window.

The metrics window is where you can make manual kerning and spacing changes to your font. You can change and view the kerning and spacing values manually, by moving the sidebearing and kerning lines on the text in the screen display, or by typing in metric values to the table below the screen. Available options from the **Metrics** menu are: **Auto Space**, **Auto Kern**, and **Kerning Assistance** and **Metrics Assistance** (for advanced metrics).

The metrics window displays a standard title bar listing the font name. The scroll bars, close boxes, and grow boxes work in the usual way. Underneath the title bar you'll find the info bar – including a textbox for typing in sample text (pairs of letters, words, or phrases) for kerning and spacing within the window. Clicking the **Show kerning** icon will show the text sample kerned; when it is deselected (light) the sample appears unkerned.

Kerning is on *Kerning is off*

The **Text** link lets you display a text file in the textbox and in the window. The **Up** and **Down arrows** let you scroll through the file line by line, displaying its text in the window.



Adjust left and right sidebearings and kerning distances by adjusting guides.

Glyph:	K	e	r	n
Width:	1706	1366	909	1366
Left:	152	72	127	123
Right:	0	64	-54	120
Kerning:	0	Ke -70	er 0	rn 102

Or type the values in the spreadsheet area.

The metrics spreadsheet below the typed letters shows you the numeric values for each glyph's width, left and right sidebearings, as well as its kern distance. Glyph width refers to the horizontal distance from the origin to the width line. Sidebearings are the distances from the left and right of the glyph outline to the horizontal boundaries of the glyph's bounding box. For more information about the metrics window, refer to "The metrics window" in Chapter 6, "Metrics – Spacing and Kerning".

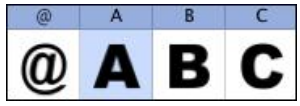
Selecting glyphs in the font window

Fontographer lets you select glyphs individually or in ranges.

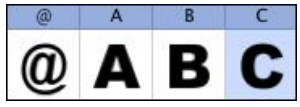
Selecting a single glyph:

Click a glyph slot with the mouse or type the glyph's keystrokes. The selected glyph slot highlights within the font window and the display automatically scrolls to show the highlighted slot.

1. Click the slot containing the "A". Notice that the entry is now highlighted.



2. Type "C" on the keyboard. The "A" becomes deselected and the "C" becomes selected.



3. Quickly type the glyph name. For example, to select the "@" glyph, type "at" on your keyboard. The "C" becomes deselected and the "at" becomes selected. Use **Shift** for uppercase names.

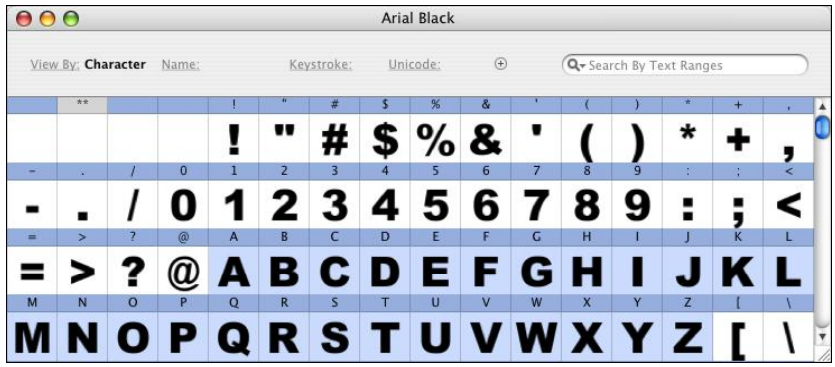
Selecting a range of glyphs:

When you apply transformations to more than one glyph, you'll probably want to select a range of glyphs within the font.

Position the pointer on the first glyph in the range and drag across the others you want to include. Release the mouse button when all the glyphs have been selected.

- Position the pointer on the "A", then press and hold down the mouse button while you drag to the "Z", and then release.

  The glyphs "A–Z" will be selected.

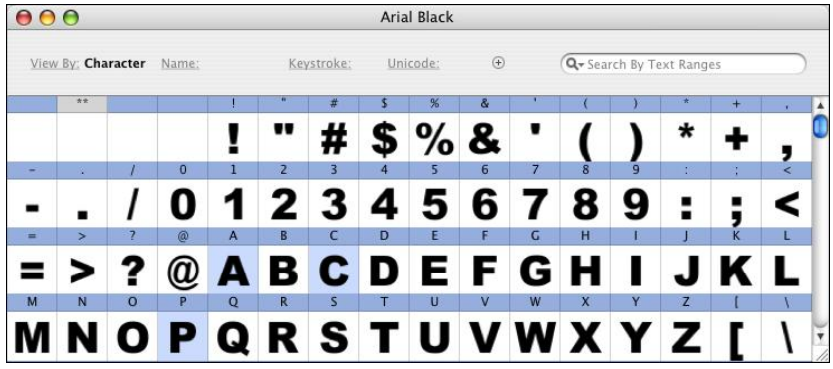


You can change the extent of the selected range, without starting over, by pressing the **Shift** key before pressing the mouse button to deselect certain glyphs or to add additional ones. But what if the glyphs you want to select are not in a continuous range?

To select discontinuous glyphs:

1. First select a glyph with the mouse, then press and hold down the **Shift** key while you finish selecting other glyphs.
2. Click the "A" and then hold down the **Shift** key and click the "P" and the "C".

   All three glyphs will be selected.



Selecting glyphs with arrow keys:

You can also select glyphs in the font window by using the keyboard's **arrow** keys. Use the **Left** and/or **Right arrow** keys to select adjacent glyphs, and use the **Up** and **Down arrows** to choose the slot above or below the highlighted glyph.

Viewing glyphs in the outline window

Viewing and sizing the glyph

When you want to zoom in/out on the glyph image in the outline window, with the default preferences settings, use the grow box in the lower right corner, or type in the appropriate key commands. You can also size images using the **View** menu and the magnifying tool.

Enlarge or zoom in on the glyph

Press **Command-Spacebar** and click in the glyph edit area. The location you click will center in the window.

Reduce or zoom out on the glyph

Press **Command-Option-Spacebar** and click the area that you want to center in the window.

The reduce command decreases the size of the glyph by zooming out from it.

Using the magnifying tool

Use the magnifying tool to enlarge or reduce an entire glyph, or just certain parts of it, in the window.

To enlarge an image:

- Select the magnifying tool from the tool palette or press **Command-Spacebar** until the magnifying tool appears.
- Click the part of the window that you want centered within the magnified screen area.
- Select the points or path to magnify with the pointer and choose a magnification menu item.
- Use the magnifying tool to drag a box around the area to magnify.

To reduce an image:

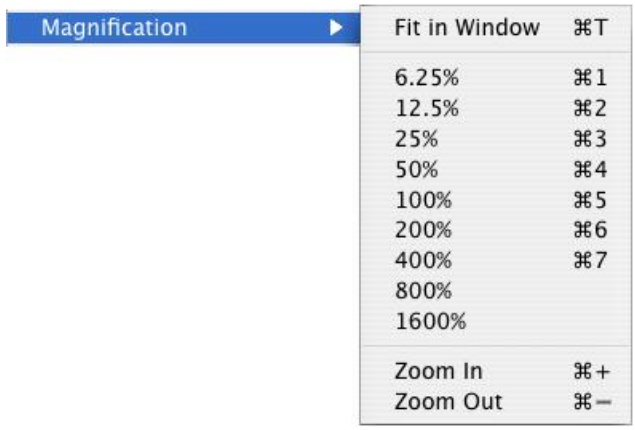
- Select the magnifying tool or press **Command-Option-Spacebar** and click the part of the window that you want to reduce.
- Select the points or path for reduction with the pointer and choose a magnification menu item.
- Select the magnifying tool and drag a box around the area to reduce while pressing **Option**.

Fit in Window

Choose **Magnification** from the **View** menu and then **Fit in Window** from its submenu, or type **COMMAND-T** to scale the glyph so the em square just fits into the window. It also centers the glyph within the window.

Magnification

Use the Magnification option to view a glyph image at various levels of magnification. You can select one of the sizes from the pop-up.



Or you can fit the glyph's em square in the window by typing **COMMAND-T**. The magnification choices are:

Magnification	Macintosh
Fit in Window	**COMMAND-T**
6.25%	**COMMAND-1**
12.5%	**COMMAND-2**
25%	**COMMAND-3**
50%	**COMMAND-4**
100%	**COMMAND-5**
200%	**COMMAND-6**
400%	**COMMAND-7**
800%	
1600%	

Note: You also can use the **Command-Plus** and **Command-Minus** keys on the additional keypad to zoom-in and zoom-out.

Note: **Command-Shift Plus** and **Command-Minus** keys on the main keypad can also be used but be careful because **Command-=** stands for **Equalize Sidebearings**.

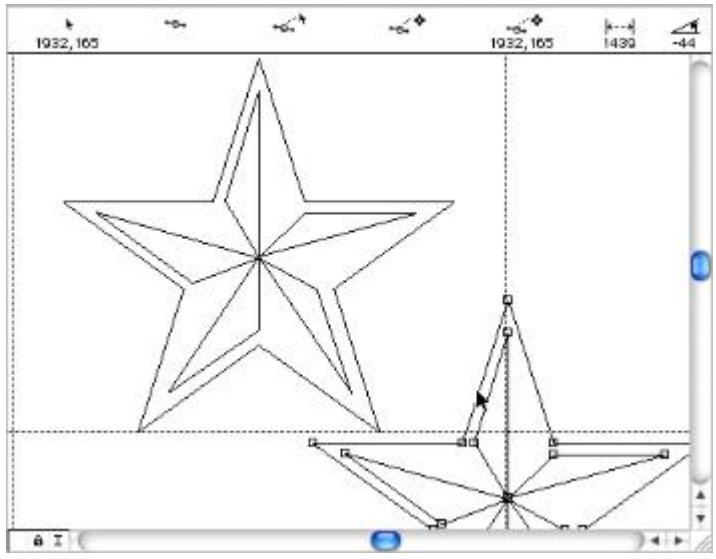
Scrolling with the hand tool

Use the hand tool to scroll the screen image in any direction. Select the tool by holding down the **Spacebar** or clicking the hand icon. Continue holding down the **Spacebar** and click the mouse button. Drag the image area in the preferred direction.

When the window is locked, and other tools are selected, you can use the **Spacebar** to select the hand tool to quickly move to a particular location within the window. When you release the **Spacebar**, the original tool will again be selected.

Moving by dragging

In Fontographer, you can scroll by dragging an object or a path outside the window area. As long as the cursor is outside the image area and the mouse is pressed down, the screen view will scroll in the direction of the movement. For example:

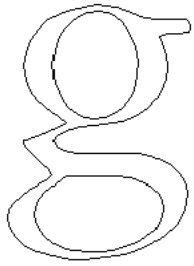


Viewing modes

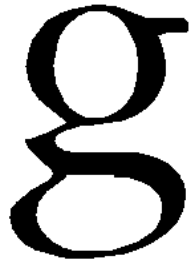
Fontographer's **View** menu provides different ways of looking at the image in the glyph window. Each mode offers a different perspective of the glyph image when you are constructing or editing a font.

Preview

When you select **Preview** from the **View** menu, **COMMAND-L**, the outlined image appears filled almost like it does when it prints. That's because you're seeing a preview of what the printed result will look like. You can also edit the glyph while you're in the preview mode.



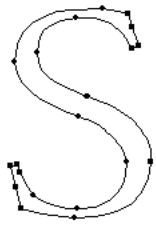
Outline



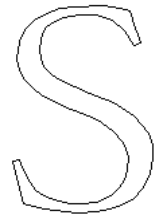
Filled glyph

Show Points

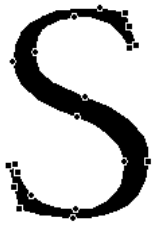
From the **View** menu you can select **Show Points**, **COMMAND-U** to display all the points in the glyph. Corner points are square-shaped, tangent points are triangular, and curve points are circular.



Points showing in Outline mode



Points not showing



Points showing in Preview mode

Points not showing

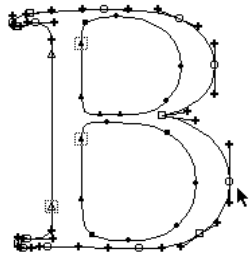
➲ ***Tip:*** To temporarily display points when they are turned off, click anywhere outside the path.

Selecting and deselecting objects

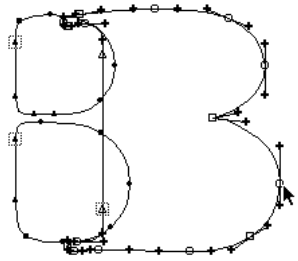
You work with objects in the outline window by selecting and altering them. To select an object, choose the pointer from the tool palette and use one of the following methods.

- Double-click a path to select the entire path.
- Use the pointer tool to drag a rectangle around the object.
- To select more than one object or path double-click the first path and then hold down the **Shift** key while selecting the others.
- Choose **Select All**, **Command-A**, from the **Edit** menu to select all paths in the glyph slot.

When you select the outline or outside path of a glyph, only that outline is selected. Other paths remain unselected.



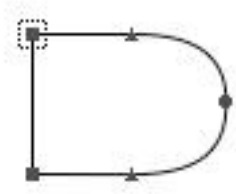
Select the outside path of a glyph containing more than one path.



Only that outline is selected.

When you select an object, the points along the path appear to be hollow. In some cases, handles extend from points. These Bézier control points or BCPs (the points that guide a Bézier curve) are control handles for changing the curve of the lines they connect.

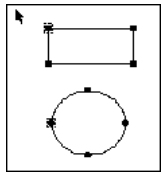
You will also notice a dotted box around one point in each path. This is the first point or origin of the path.



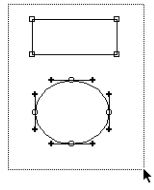
Use the Preferences dialog box to turn this option on or off. In some transformations (like font blending and multiple master fonts), the first point is aligned with initial points in other images, so knowing where each point is located gives you an idea of how the action will result. For more details about font blending, refer to "Blend fonts to create new fonts" in Chapter 3, "Creating New Fonts" and "Font blending – the technical details" in Chapter 13, "Expert Advice". Multiple master fonts are discussed in "Multiple master fonts" in Chapter 13, "Expert Advice".

Drag-selecting objects

If you want to select an object or a group of objects, position the pointer outside the area you want to select and then drag the mouse around it. A marquee appears around the selected area. When you release the mouse, the rectangle image disappears but the area remains selected.



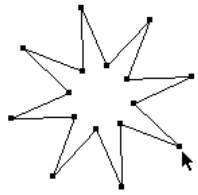
Position the pointer outside the area you want to select.

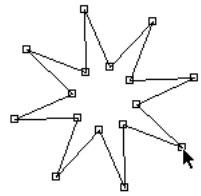


Drag the mouse around it.

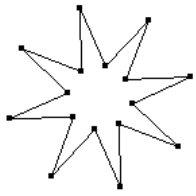
Click-selecting objects

Double-clicking an object (composed of a single path) selects it. To deselect the object, click outside the shape or press the **Tab** key.





Double-click an object... *to select it.*

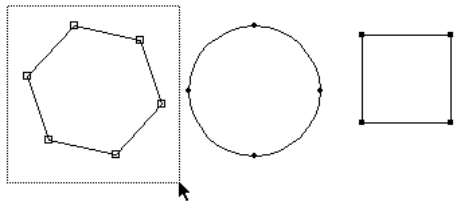


*Press the **Tab** key to deselect it.*

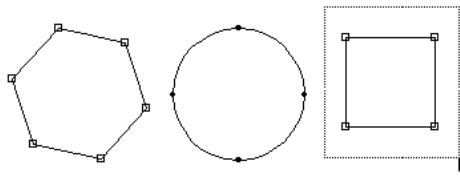
Shift-selecting objects

To select more than one path, position the pointer on the first object and double-click. Select all other objects by double-clicking them while the **Shift** key is pressed.

You can also use the **Shift** key when drag-selecting objects.



Drag around an object with the pointer tool and release the mouse.



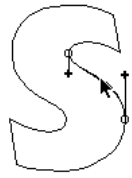
*Press the **Shift** key and drag around another object.*

Selecting parts of a path

In Fontographer, you can select part of a path by selecting points that are on it. If you want, you can select parts of paths belonging to different objects or glyphs as well.

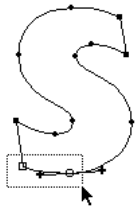
You can select a segment and its two connecting points in three different ways:

- Click once on the connecting path with the pointer, if you have selected the preference setting for this option.



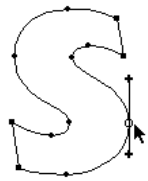
Click directly on the path.

- Drag the pointer around the segment (and its two adjoining points) and release the mouse.

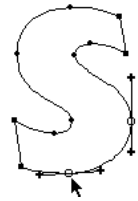


Drag around the segment and its two adjoining points.

- Click each of the two points connecting the line segment while you hold down the **SHIFT** key.



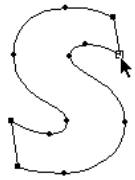
*Select a point and hold down the **SHIFT** key...*



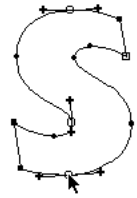
while you select another point.

To select points on different parts of the same path or on different paths:

- Click the pointer tool on a point you want to select. Press the **SHIFT** key and continue selecting points along the paths.

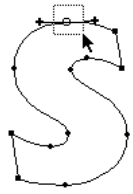


Click a point.

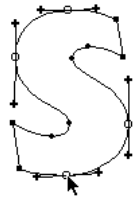


*Hold down the **SHIFT** key to continue selecting points.*

- Drag the pointer around a point you want to select. Press the **SHIFT** key and continue selecting other points.



Drag a pointer around a point.



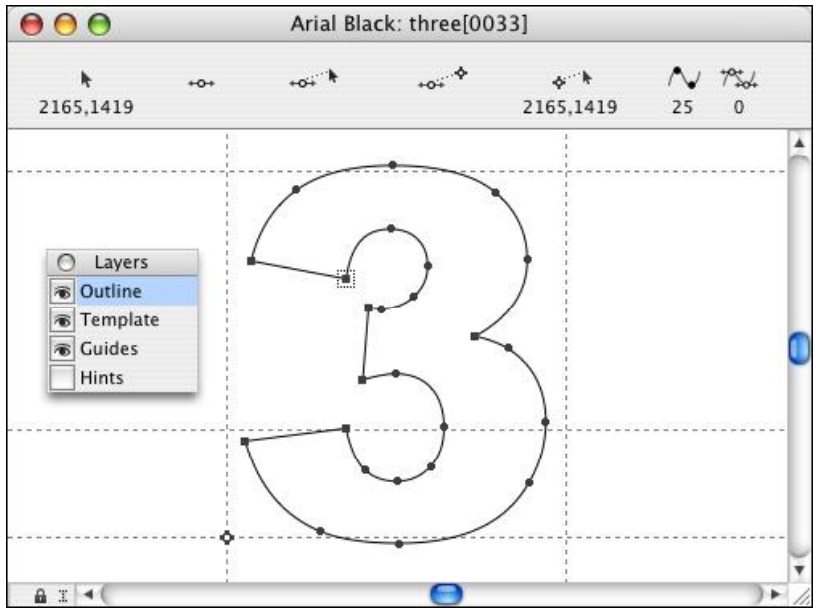
*Press the **SHIFT** key to continue selecting other points.*

Drawing layers

Fontographer has four layers. Each layer serves a different purpose in helping to construct a glyph.

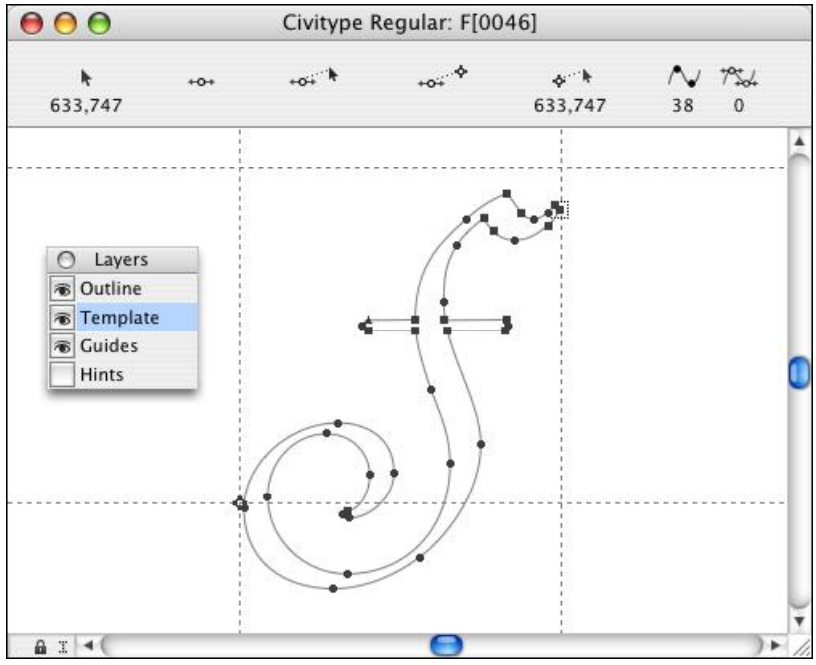
Outline layer

All changes to a font's outlines take place in the outline layer. Glyphs appear here unfilled (unless the preview option is turned on). The outline contains points and line paths of glyph images. Everything drawn in the outline layer becomes part of the final glyph.



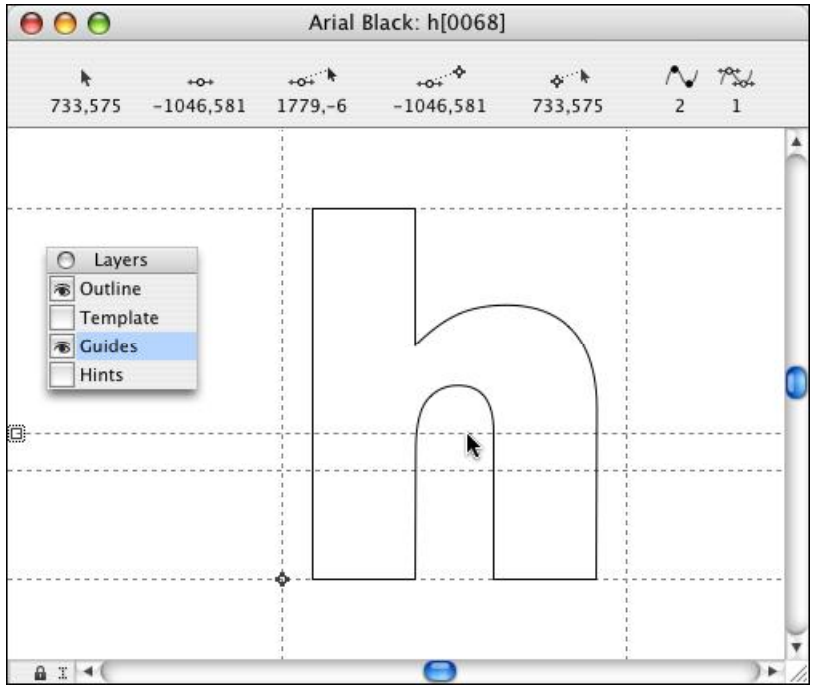
Template layer

Images in the Template layer are used as drawing references. Pasted or imported images are placed here. You can also create your own images in this layer to act as templates. Whatever is drawn here will not appear in final exported versions of your font. This layer is used strictly as a drawing aid. When you view the images in the template layer, they appear with gray outlines or fills rather than with the black outlines or fills you see in the outline layer.



Guides layer

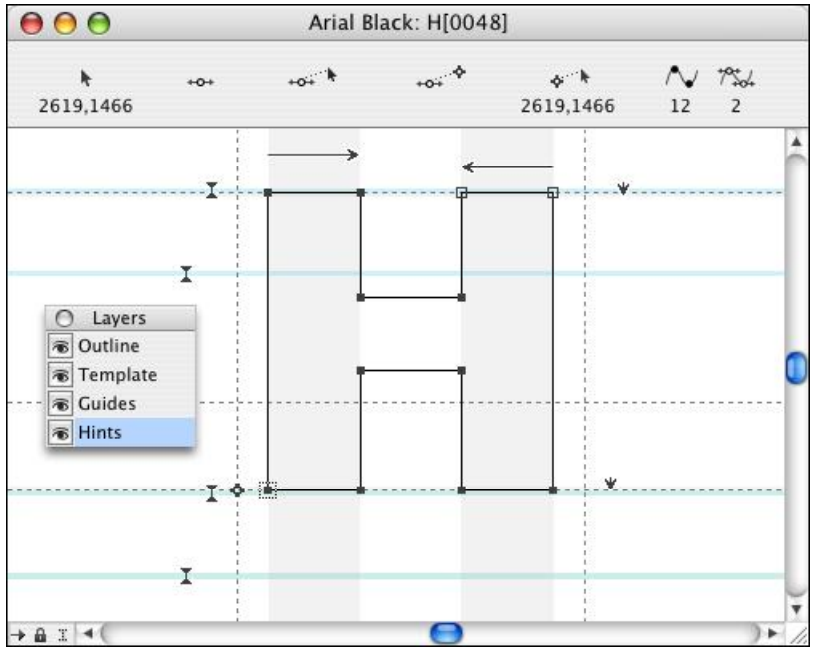
The guides layer can be used as a drawing aid. Additional guidelines can be pulled out of the origin line and the baseline. These lines serve as guidelines in glyph construction. For example, a horizontal line can be used to set the x-height. You can create multiple horizontal and vertical guidelines and other shapes with the standard drawing and editing tools.



Changing the guidelines for one glyph will change the guidelines in all the other glyph windows in the font; likewise, any edits to the guidelines in this layer will immediately appear in every other glyph in the font. However, any images created in the guides layer are not exported since this layer is just a design aid for all the glyphs. You can undo changes made in the guides layer just like you would in any other layer or window.

Hints layer

Use the Hints layer to specify how smaller sizes of a glyph will be printed when output to the screen or low-resolution printers. The object of Fontographer's hinting process is to preserve the shape of the glyphs at smaller sizes, including stem widths and other features that define the glyphs in a font.



For more information about the hinting process and the hints layer, refer to "Font hinting" in Chapter 13, "Expert Advice".

Using the palettes

Layers palette

In Fontographer, the Layers palette displays when you open the outline window. It can be moved or closed like any window, and selected or deselected for display from the **Window** menu.

Each of Fontographer's layers is used in a slightly different fashion to help make glyph construction easy. Fontographer allows drawing in any of these four layers.

To change layers, click the mouse on the name, not the checkbox, of the desired layer. The layer appears highlighted within the palette and display on the screen. To display another layer in the outline window, click the appropriate checkbox. When a layer's checkbox is selected, its contents display onscreen.

Tool palette

The tool palette contains tools for modifying outlines. The palette itself can be closed and moved just like any other window. Clicking each icon or indicator with the mouse changes the pointer to the selected tool. Tools can also be accessed by keystrokes. You can choose to display (or not display) the palette by clicking the **Show Tool Palette** option in the **Window** menu.

Pointer tool

Use the pointer tool to select elements and drag points and objects. You can select the pointer by clicking its indicator with the mouse. When the pointer is active it is shaped like an arrow.

To change any other tool in the tool palette to a pointer, press the **TILDE** (~) key (to the left of the number 1 or letter z). The selected tool is now a pointer. For a temporary change, hold down the **COMMAND** key. When you release the key, the pointer reverts back to the selected tool.

Hand tool

Use the hand tool to scroll the screen in any direction. Select it by clicking once on its icon. When the hand is displayed, clicking and moving the mouse in any direction moves the screen correspondingly. You can also access this tool temporarily by pressing the **SPACEBAR**.

Basic shape tools

Use the Basic shape tools to draw regular shapes (rectangles, squares, ovals, circles, stars, polygons, and lines). You can use other tools (control point and pen, for example) to draw these shapes, but the basic shape tools make the process easier.

To access a basic shape tool, click its icon. Shapes are drawn by selecting the appropriate tool and click-dragging in the glyph window until the shape becomes the size you prefer. Some basic shapes can be constrained (the rectangle to a square or the oval to a circle) by holding down the **SHIFT** key while dragging with the mouse.

The dialog box for the rectangle tool allows you to round the corners in the rectangle, whereas the dialog box that goes with the multigon tool lets you change the shape from a star to a multi-sided shape, and gives you the ability to change the number of sides in any shape you choose.

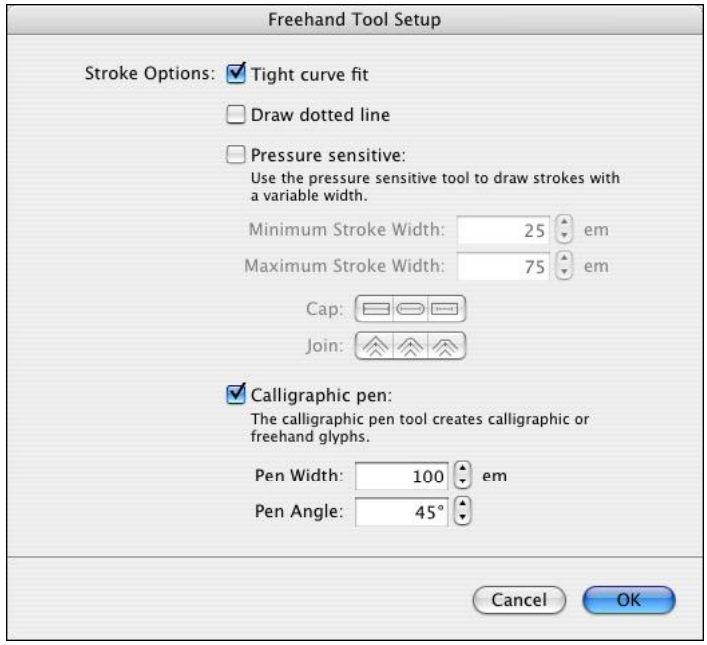
When the lock icon is in the locked position, you can use numeric keystroke equivalents to select the tools. The keystroke commands are: rectangle tool (1), multigon tool (2), oval tool (3), and straight line tool (4).

Freehand drawing tool

Use Fontographer's freehand drawing tool to draw open paths. You can create a closed path by overlapping beginning and ending end points. For more information about paths, refer to "Paths and points" in Chapter 4, "Altering Outlines". The freehand tool can be used directly with the mouse or with pressure-sensitive pens and digitizing tablets.

Double arrows within the icon box of any tool signify that an additional dialog box will appear when you double-click the tool.

The Freehand Tool Setup dialog box offers additional options for either a calligraphic or a variable-weight pen.



Calligraphic pen tool

The calligraphic pen tool creates calligraphic or freehand glyphs.

You can create calligraphic glyphs with the same basic strokes that you use to create them with a real calligraphic pen. Click the **Calligraphic pen** checkbox in the Freehand Tool Setup dialog box to turn the freehand tool into a calligraphic pen.

Pressure sensitive tool

Use the pressure sensitive tool to draw strokes with a variable width. Select the **Pressure sensitive** option in the dialog box, or select the **Pressure sensitive** option and the **Calligraphic pen** option to draw calligraphic strokes with a variable width.

When the lock icon is in the locked position, the numeric keystroke equivalent for choosing the calligraphic/freehand tool is 5.

Pen tool

The pen tool is a multipurpose tool similar to those in Adobe FreeHand and Adobe Illustrator. The pen tool combines the capabilities of the tangent, corner, and curve points, so you can draw paths without having to switch tools. Select the pen tool by clicking its icon.

This tool places points depending on the actions of the mouse. If you click with the mouse, the pen tool places a corner point. If you click and drag the mouse, the pen tool will place a curve point where you click. Dragging does not move the position of the point, but has the effect of changing the shape of the curved path between the curve point and any adjoining points.

Knife tool

Use the knife tool to cut paths. When you click the indicator, the pointer becomes a knife and is ready for use. To use the knife on a path, drag it across the path. A point will appear. To separate the path, choose the pointer tool and drag the selected point and release.

Points as well as paths may be split with the knife tool (although it is sometimes easier to use the **Split Points** command on the **Points** menu). To split a point, click it and then choose the selection pointer. You can drag the split point away from the original.

You can also delete a segment between points by **Option**-clicking the segment.

With the lock icon in the locked position, access the knife by typing 7.

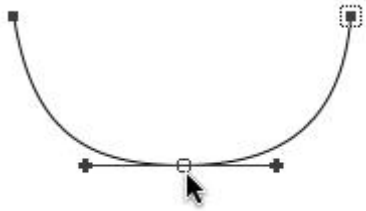
For more information on splitting paths using the knife tool, refer to "Splitting a path" in Chapter 4, "Altering Outlines".

Control point tools

Fontographer's control point tools are the mainstay of font construction. Use them, the basic shape tools, or the pen tool to create the shape you want your paths to take.

Curve point tool

The curve point tool is used to smoothly join curves to other curves.

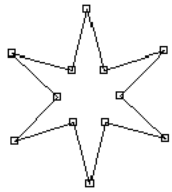


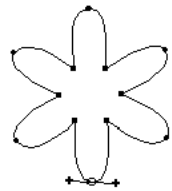
A curve point determines that any adjoining line segments will be curves, regardless of the type of points they are attached to. This is different than the way a corner point operates – the shape of its attached segments depends on the connecting points beside it in the path. Select the curve point tool by clicking its indicator. To place a point, click once in the edit area with the pointer tool.

When the lock icon is in the locked position, the numeric keystroke equivalent for the curve point tool is 8.

Corner point tool

The corner point tool can be used in several ways. You can join corner points to other corner points to create angles in polygons, squares, and triangles in glyph stems; or you can join corner points to curve or tangent points to create gentle curves or cusps.



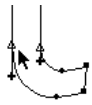


The shape of line segments extending from corner points is determined by points connected to the corner point.

When the lock icon is in the locked position, the numeric keystroke equivalent for choosing the corner point tool is 9.

Tangent point tool

The tangent point tool is used to connect straight lines to curves with a smooth tangent join.



Tangent points can also be used to connect straight line segments together. The tangent point tool can be selected by clicking its indicator.

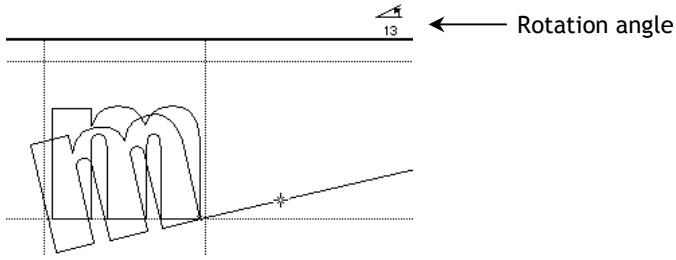
When the lock is in the locked position, the numeric keystroke equivalent for the tangent point tool is 0.

Transformation tools

Use the transformation tools to transform individual glyphs or selected contours in several different ways, with the help of dialog boxes and measurements.

Rotate tool

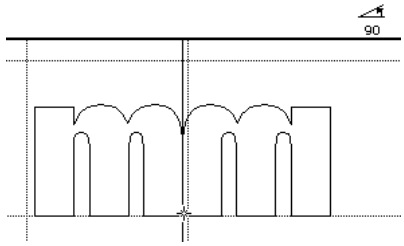
The rotate tool is used to rotate objects.



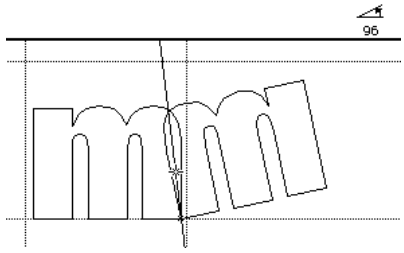
Holding down the **Shift** key while you rotate confines the rotation to 45-degree increments.

Flip tool

The flip tool creates a mirror image of the object and positions it in 45-degree increments as you drag. The angle of reflection is displayed in the info bar.

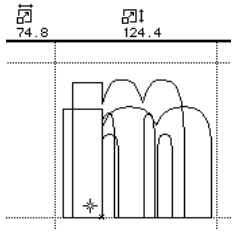


You can drag and flip freely by holding down the **Shift** key as you use the flip tool.

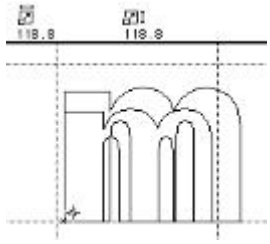


Scale tool

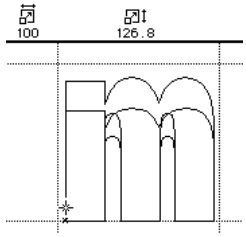
The scale tool increases or decreases the size of the object. Depending on the horizontal or vertical direction you drag, you can condense and/or stretch the object. The scale values are displayed in the info bar.



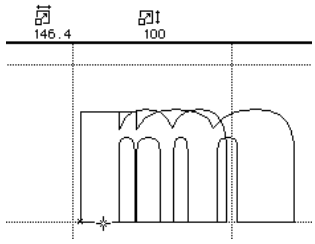
If you hold down the **Shift** key and drag at a 45-degree angle (up or down), you can scale proportionately.



If you hold down the **Shift** key while dragging straight up or straight down (vertically), the glyph scales vertically without affecting its horizontal size.



If you hold down the **Shift** key and drag horizontally, the vertical size won't be affected.

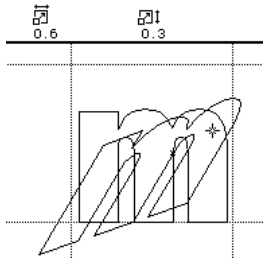


Double-clicking it brings up the transform dialog box with Scale selected as the first transform.

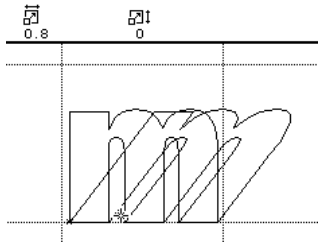
When you **Option**-double-click the scale tool, the Transform dialog box appears with Scale Uniformly selected as the first transformation.

Skew tool

The skew tool alters the horizontal and/or vertical angle of the object.



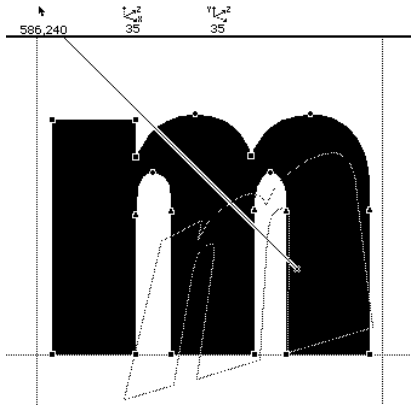
To skew in 45-degree incremental angles, hold down the **Shift** key while you skew.



The small double-click indicators at the lower right corner of each tool's icon indicate that double-clicking the tool brings up a dialog box. Use the transform dialog box to transform objects or one, several, or all glyphs in the font precisely.

Perspective tool

With the perspective tool, you can create letters that zoom off the page toward you. It works in conjunction with 3D rotate to apply three-dimensional rotations to two-dimensional objects while still maintaining perspective.



Before you use the perspective tool, it's a good idea to set up the perspective point. **Option**-double-click the perspective tool icon in the tool palette to bring up the Perspective Setup dialog box.

In this dialog box you will set the perspective distance and choose a perspective point of basepoint, center of selection, glyph origin, or mouse click.

Changing the Perspective Distance/Point will not change the appearance of the glyph in the outline window. The Perspective Setup changes the perception of the tool, not how the program views the image.

After completing setup and clicking **OK**, you'll need to select the perspective tool and actually apply the desired transformation to the glyph in your outline window.

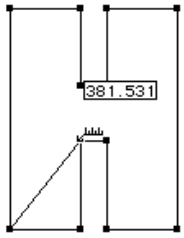
Note: When a 3D transformation has been applied to a two-dimensional object, the object becomes two-dimensional again. The object itself does not maintain three-dimensional coordinates. To do multiple 3D transformations while maintaining 3D coordinates in between each transformation you must use the transform dialog box, not the tool palette.

For more about using the perspective tool, applying 3D transformations, and navigating the Transform dialog box, refer to "Transformation options" in Chapter 3, "Creating New Fonts".

Measuring tool

The measuring tool measures distances between any two areas in the outline window. It can also help you make sure that certain distances are the same (for example, the widths of stems or serifs in glyphs like H and M).

To measure, click the first point to be measured and drag to the second point. The distance displays in em units.



Magnifying tool

The magnifying tool zooms in and out your image with a click of the mouse. Select the tool by clicking its indicator. When you want to magnify a portion of the screen, click and drag a box around the area with the tool and the selected area will magnify.

You may also change magnifications by clicking (rather than dragging). Position the magnifying tool over the area you wish to magnify and click the mouse.

To zoom-out the image, or demagnify, press the **Option** key and click the mouse. The indicator displays a minus sign within the magnifying tool to show it is reducing the image's size.

The magnifying tool can be temporarily accessed from any other tool by holding down **Command-Spacebar**. To demagnify the view, hold down **Option-Command-Spacebar**.

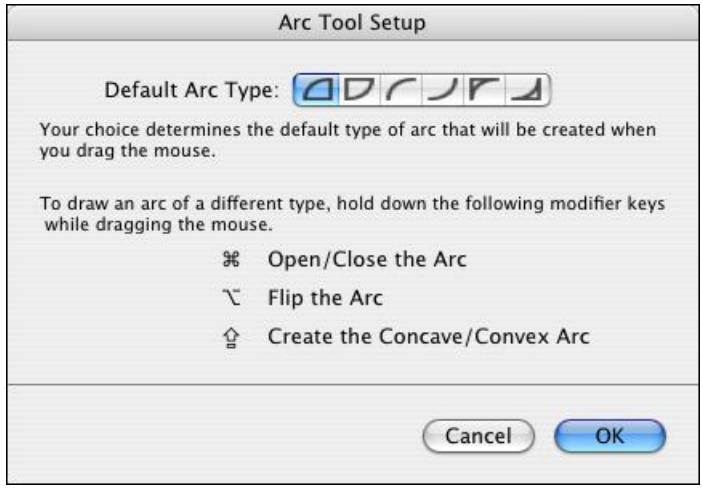
Note: You also can use the **Command-Plus** and **Command-Minus** keys on the additional keypad to zoom-in and zoom-out.

Arc tool

Use the arc tool to create one fourth of an oval and arc-like shapes. It appears in the tool palette as one of the following icons:

This arc tool icon (and the objects you draw with it) changes according to the arc types you select in the Art Tool Setup dialog box.

If you double-click the arc tool icon, the Arc Tool Setup dialog box appears.



Arc Types

By default the arc tool creates closed paths. Two examples of closed path arcs are and .

Open path arcs are and .

Arcs like and are called concave arcs.

Keyboard shortcuts

You can create different types of arcs without changing them in the dialog. Modifier keys can be applied to change the type of arc being created. The arc tool creates the type of arc that is opposite of the default type of arc when you press the modifier keys. These modifier keys should be pressed while dragging the mouse to create a new arc.

- Hold down the **Command** key while creating a new arc to toggle between creating an open and a closed arc. With the open arc selected, hold down the **Command** key to create a closed arc.
- Hold down the **Option** key to flip the arc both horizontally and vertically.
- The **Caps Lock** key toggles between creating a convex and a concave arc.

Note: Unlike the **Command** and **Option** keys, the **Caps Lock** key does not need to be pressed while dragging the mouse.

These modifier keys are also mentioned in the Keyboard Shortcuts section of the Arc Tool Setup dialog box for the convenience of the user.

The **Control** key is the demagnified move modifier key. By holding down the **Control** key when you create a new arc, you can increase/decrease the size of the arc in one em-unit increments.

Constraining tools

When you want to constrain the action of a tool to 45-degree increments, use the tool normally while holding down the **Shift** key. Each tool's constraint mode may operate with slightly different results.

Here is a list of how various tools react while constrained:

To use the rectangle tool to draw a square, press the **Shift** key and drag the mouse.

To use the oval tool to draw a circle, hold down the **Shift** key and drag the mouse.

To use the arc tool to draw a quarter circle, hold the **Shift** key down while moving the mouse at approximately a 45-degree angle (up and right, up and left, down and left, or down and right.)

Moving the mouse along 90-degree angles while holding down the **Shift** key makes a straight line appear. This is probably not what you want.

Any path or object moved with the pointer tool can be constrained. Select the object or path and begin dragging it; then press and hold down the **Shift** key to constrain the movement of the selected object.

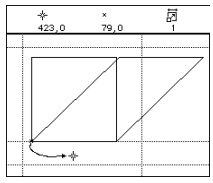
Use the **Shift** key with corner, curve, or tangent point tools while dragging points to constrain their movement. In addition, if the **Shift** key is held down when placing new points with these tools, the new point is automatically aligned (or constrained to) the previous point.

Each of the transformation tools can be constrained by pressing the **Shift** key after clicking the selected point of reference.

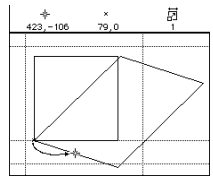
Note: The flip tool is automatically constrained, because you will most often want to flip paths and objects in exact 90-degree increments. Hold down the **SHIFT** key to deconstrain the flip tool.

If you use the **SHIFT** key with the measuring tool, you can constrain its measurements.

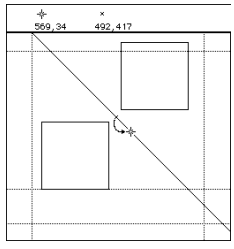
Note: In the bitmap window, the measuring tool is automatically constrained, because you will usually want to measure pixels straight across, or up and down. So, when you're in the bitmap window, hold down the **SHIFT** key to deconstrain the measuring tool.



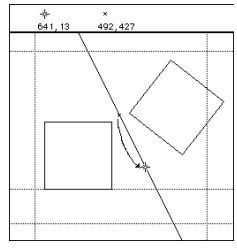
*Constrain skewing by using the **SHIFT** key.*



*Use unconstrained skewing without the **SHIFT** key.*



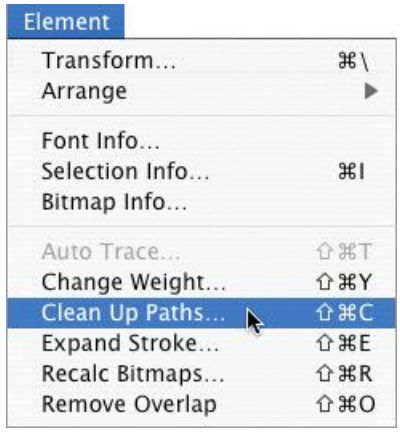
*This rotation is constrained by holding down the **SHIFT** key.*



*It is rotated unconstrained without the **SHIFT** key.*

Fontographer's menus

When you open Fontographer, a set of menus appears at the top of your screen. Each menu displays the various kinds of commands you can use to perform actions such as opening and closing files; saving, importing, or exporting font files; editing, viewing, transforming, and autotracing files; and more. The menus are the heart of the program – every major action is listed here.



If you drag down in any of these menus with the mouse, a list of commands is presented. You choose them by dragging the mouse pointer to the command that you want.

Undo and redo

Whenever you are in a glyph's outline window and you want to undo a command or a series of commands, you can choose **Undo** from the **Edit** menu or type **COMMAND-Z**.

To redo what you've undone, choose **Redo** from the **Edit** menu or type **COMMAND-SHIFT-Z**.

The default setting for the number of commands you can undo or redo is 100, but this number can be changed (up to 256) by resetting the preferences. See "Setting preferences" on page 18, or in "General preferences" in Chapter 13, "Expert Advice".

Closing and Quitting

Closing

To close any window, click the close box in the top left corner of the window:

Select **Close** from the **File** menu or type the keyboard alternative, **Command-W**.

Quitting

To quit Fontographer, choose **Quit** from the **Fontographer** menu or type **Command-Q**.

A generic Close dialog box appears whenever you close a file or quit the program, asking you whether or not you want to save your changes.

If you select **Save**, you will save the file including any changes made since the file was last saved.

If you choose **Don't Save**, the file reverts to the last saved version before quitting; you won't save any changes since the file was last saved.

Choosing **Cancel** cancels the **Quit** command and lets you continue working in Fontographer.

Fontographer offers new Quit dialog boxes for greater efficiency when quitting with multiple, unsaved documents open.

- **Discard Changes** quits without further interruption, leaving all documents unsaved.
- **Cancel** cancels the **Quit** command and lets you continue working in Fontographer.
- **Review Changes** is the default button. This choice prompts you for each open database (by name), in the order of opening. You'll have the option of saving, cancelling, or not saving changes to each one.

When only one open document is unsaved, the generic Close dialog box displays.

Folders and paths

Recent applications from Fontlab Ltd. use a new folder structure for storing their **data files** such as encoding or codepage definitions, glyph generation recipes, text samples for metrics and kerning, mapping tables, Python macros etc. Fontographer 5 looks for data files in four different folders.

Shared default data folder

typically, Macintosh HD/Library/Application Support/FontLab

This folder holds files that are commonly used by all recent Fontlab Ltd. applications: Fontographer 5, FontLab Studio 5, TransType SE/Pro, BitFonter 3, FogLamp, SigMaker 3, with more to come. In each respective subfolder, codepage definitions, encoding definitions, glyph-to-Unicode mapping files and some special data files are stored. Only Fontlab Ltd. applications and applications from registered Fontlab Ltd. developer partners should place their files there. This is to rule out conflicts between the user's customized files and default files.

Shared user data folder

typically
Macintosh HD/Users/Your Username/Library/Application Support/FontLab

This folder has exactly the same structure as the folder discussed above and can store any files customized by the user. Any file placed in the respective location within that folder will override the corresponding file placed in the shared folder. Please put your customized files in this folder.

Application default data folder

typically Macintosh HD/Library/Application Support/FontLab/Fontographer 5

This folder holds files that are only used by Fontographer 5. In each respective subfolder, metrics, kerning and other text strings and additional encodings are stored. Only Fontographer 5 application should place its files there. This is to rule out conflicts between the Fontographer specific files and files for other apps.

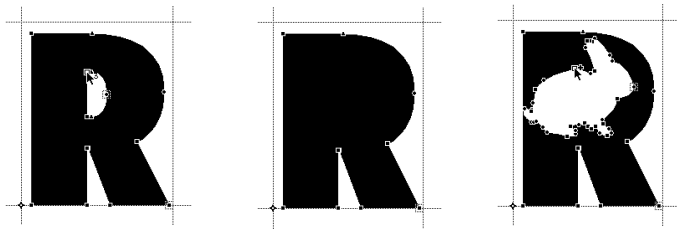
Application user data folder

typically
Macintosh HD/Users/Your Username/Library/Application Support/Fontographer 5

This folder has exactly the same structure as the folder discussed above and can store any files customized by the user. Any file placed in the respective location within that folder will override the corresponding file placed in the application default and shared folders. Please put your customized files in this folder.

Modifying Your Fonts

Imagine being able to create completely new fonts without ever drawing a thing, making new small caps versions, and new oblique typefaces – without drawing a line, placing a point, or manipulating a curve. This is just what Fontographer lets you create: completely new fonts by modifying your existing fonts.



It's easy to create new fonts, modify existing fonts, and add graphics to your fonts.

You've invested a lot of money in the typefaces you own. Although many talented people create their own from scratch, the easiest way to create a completely new typeface is by modifying the fonts you already have. Fontographer makes it so easy to modify your existing typefaces that you almost don't have to think about it. This chapter gives you some quick ways to make modifications that will encourage you to create typefaces of your own. However, fonts, like other software, have licenses that determine what you are legally allowed to do. Check your font EULA first.

2

Steps to modifying your font

1. Open a font.

Note: If it's a PostScript font, it's a good idea to import the metrics by choosing **Import** from the **File** menu to avoid kerning and spacing faux pas, and other unattractive results. Macintosh users might want to import the bitmaps too. For more details, refer to "Import" in Chapter 14, "Reference".

2. Modify it; for example, you can simply change the weight.
3. Save the file (optional).
4. Generate an installable font.
5. Install the font.
6. Put it to work.

Use one font like...

Arial Roman

and the Change Weight command to get...

Arial Black

or Scale it to create...

ARIAL SMALL CAPS

or Skew it to create...

Arial Oblique

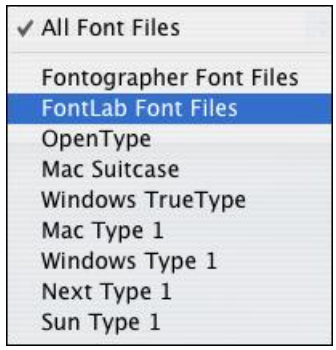
Opening a Font

Existing fonts

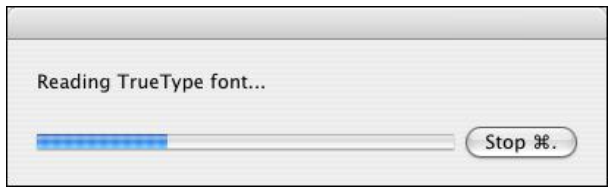
To open an existing font, choose **Open** or **COMMAND-O** from the **File** menu. A file selection dialog box appears that works in the standard fashion. Once the selection dialog box is open, you can select a font file by clicking its name and then **Open**, or simply by double-clicking its name.

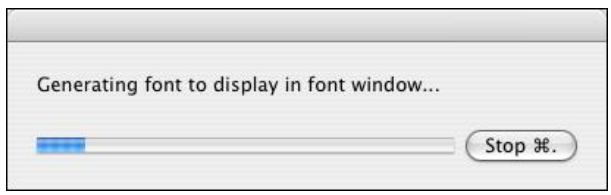
Fontographer 5 can open font files of the following formats: Mac Suitcase (without extension or .dfont), Mac Type 1 (PostScript), TrueType/OpenType TT (.ttf), TrueType Collection (.ttc), Windows Type 1 (PostScript) and Windows Multiple Master (.pfb), Unix/ASCII Type 1 (.pfa), OpenType PS (.otf), Fontographer font files (.fog), FontLab Studio font files (.vfb).

If you want to list only fonts in a particular format, select that format in the Format popup menu:

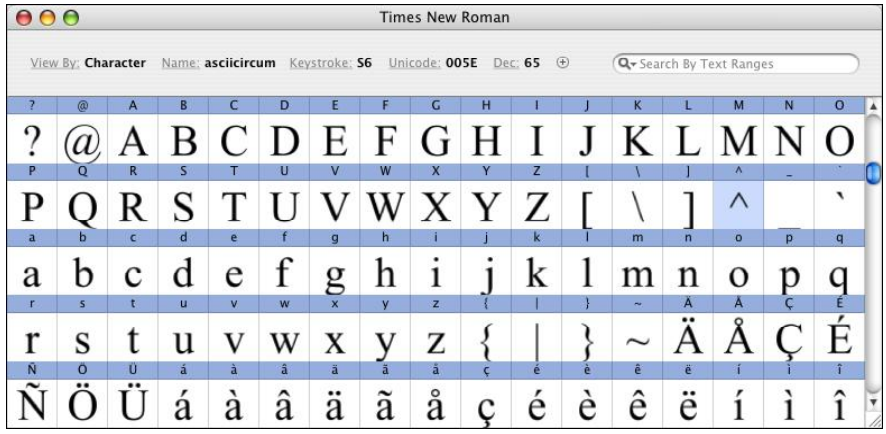


One or more progress dialog boxes will appear before Fontographer displays the Font Window. To cancel progress dialog boxes, type **COMMAND-PERIOD**.





Fontographer's progress dialog boxes keep you informed of the program's status...



until Fontographer displays the Font Window.

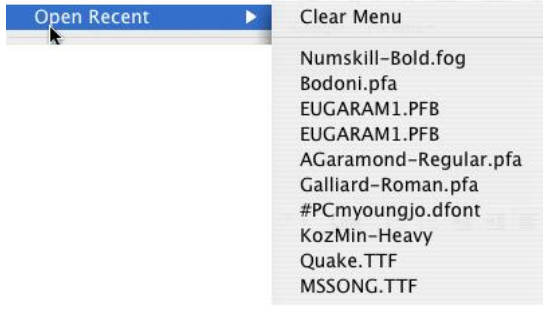
Note: You can open as many fonts as you like depending on the amount of memory you have available.

Opening Fonts with Drag-Drop

An easy way to open fonts in Fontographer is to drag-drop font files from Finder. Even if Fontographer is not running, you can drag-drop files onto its icon on the desktop or in the Dock to run Fontographer with those fonts open.

Opening Recently Used Fonts

All fonts that you recently opened in Fontographer are added to the list of the most recently used font. This list is used in the **File>Open Recent** menu:



Next time you want one of them, just select the font file in the menu and Fontographer will open it.

Font Formats

The **.fog file format** used in Fontographer 5 is fully **backwards compatible**, so Fontographer 5 can open any .fog file created in Fontographer 3.x and 4.x. The format is also **cross-platform**-compatible so .fog files saved from the Windows version can be opened in the Macintosh version and vice versa. In addition, the format is largely **upward compatible**. This means that a .fog file saved from Fontographer 5 can be opened in Fontographer 3.x or 4.x. Only those elements of the format that were supported by the old version will be retained and some information may change slightly. However, the most important elements of the font such as key Font Info entries, glyph outlines and kerning pairs will be retained.

For example, .fog files saved from Fontographer 5 for Windows can be opened in Fontographer 4.7 for Macintosh and vice versa, with as much as possible information retained.

Fontographer 5 also opens .vfb files saved from FontLab 2.5-5 (but not 2.0). If you have fonts saved in a proprietary format of another application and would like to open these fonts in Fontographer, the best way is usually to create a Windows-compatible Type 1 font from your other application and open the Type 1 font in Fontographer. If you wish to move your .fog files created in Fontographer 3.5-5 to FontLab Studio, you can use the **Fontlab Font File** command in the **File>Export** menu. Fontographer 5 can now convert .fog files into FontLab Studio-compatible .vfb files directly, retaining not only outline information but also mask layers, guidelines, background bitmaps etc.

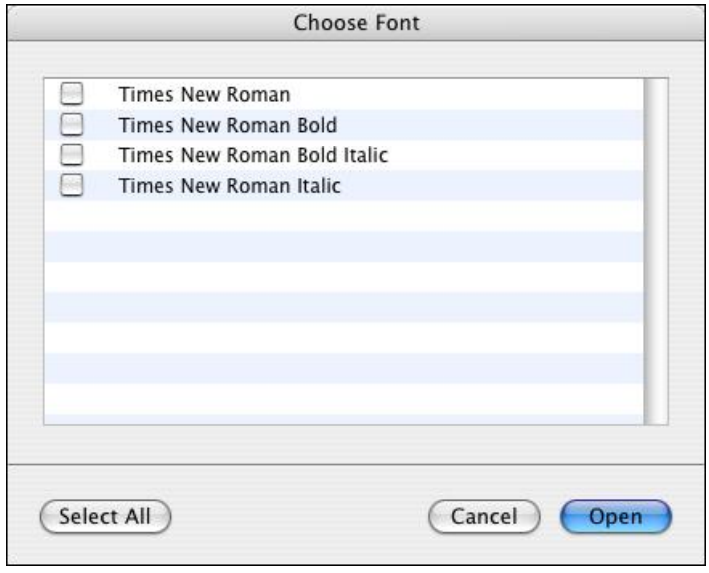
Glyphs Number

Fontographer has a restriction on the number of glyphs in the font – 32 767. Fontographer won't open the font if it contains more glyphs. If you need a font editor that supports more glyphs (up to 65,535), Fontlab Ltd. offers AsiaFont Studio, a multibyte font editor that is the "bigger brother" of Fontographer.

Opening Macintosh fonts

A Macintosh Type 1 (PostScript) font always consists of several files: one file for every typeface in the family containing outlines plus one suitcase for a family containing metrics, bitmaps and other important font information. Fontographer 5 can now open both parts including suitcases at one step. If you choose to open suitcase you then do not need additionally import metrics or bitmaps later.

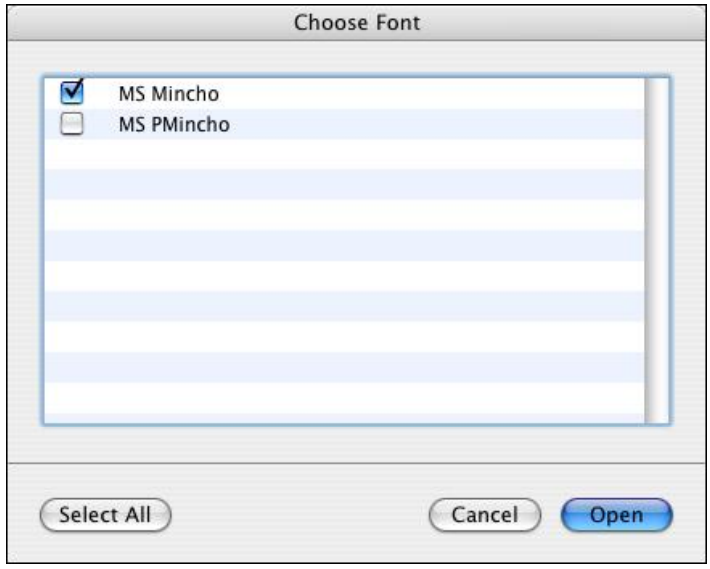
If the selected suitcase-based font family contains more than 1 font face Fontographer opens another dialog allowing you to choose particular fonts to open:



Select only fonts you need and click on **Open**.

Opening Font Collection

When you select a TTC (TrueType Collection) font for open, you are presented with a special **Choose Font** dialog box:



The list of fonts in a collection has checkboxes to let you select which fonts should be opened. You can check those of them that you want to be opened. Click on the **Select All** button to switch on every checkbox.

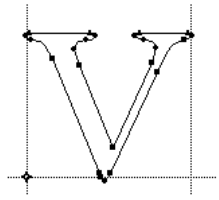
When you are ready click on **Open** to start importing TTC font.

New fonts

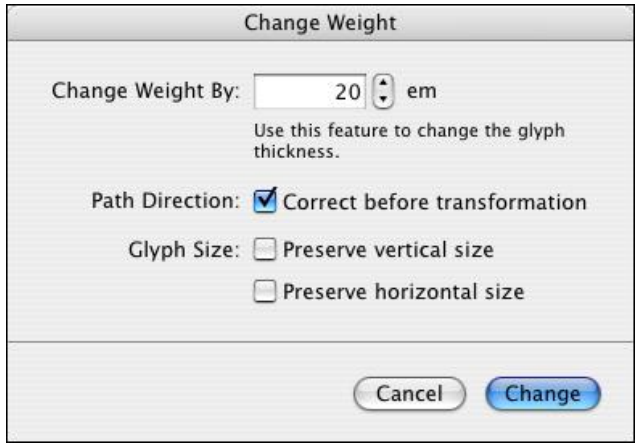
To open a new font, choose **New Font** from the **File** menu. Fontographer opens a new untitled font window. From this window you can begin to create a Fontographer database file from which you will be able to generate a usable font. For more information about creating fonts, see Chapter 3, "Creating New Fonts".

Changing the glyph's weight

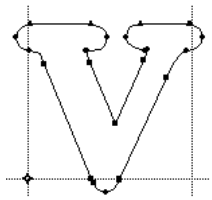
You can quickly create a heavier or lighter version of one glyph, several glyphs, or the entire font by using Fontographer's **Change Weight** command.



Open any glyph's outline window.



*Enter a value in the **Change weight by** box.*

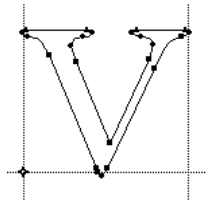


Fontographer changes the weight of the glyph.

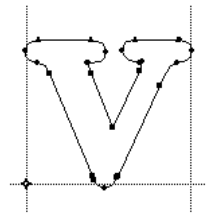
To change weight:

1. Go to the Font window and double-click the "v" to open it.
2. Choose **Change Weight** from the **Element** menu.

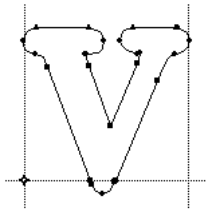
   The Change Weight dialog box appears.
3. Enter 30 in the **Change weight by** text edit box and click **Change**. Fontographer increases the weight of the "v" by 30 em units.



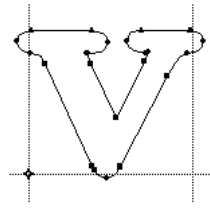
Original character.



With the Don't change vertical size option checked.



With the Don't change horizontal size option checked.



With both the Don't change horizontal size and Don't change vertical size options checked.

Fontographer also gives you the option of changing the weight of your glyph or entire font without affecting the vertical or horizontal size of the glyph.

Go back to the "v" and select **Undo Change Weight** from the **Edit** menu to undo the changes you performed in the last exercise. Repeat the exercise above, but check the **Don't Change Vertical Size** option before you click **Change**.

Look at the difference in the two options. You can try the option with the **Don't Change Horizontal Size** option next.

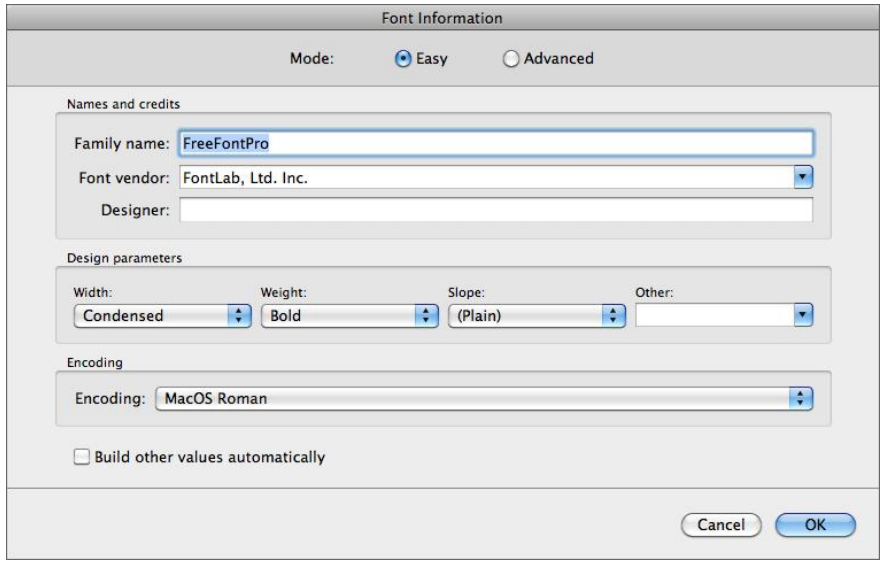
Now try the exercise with both options checked.

You can now apply the desired weight to the entire font or just to selected glyphs. From the font window, use the pointer tool to click and **Shift**-click the desired glyphs. (You can choose them all by choosing **Select All** from the **Edit** menu.) Then repeat the procedure above to apply the selected changes to the desired glyphs.

➲ ***Tip:*** You can't undo global changes in the font window, so it's a good idea to try your changes out on one glyph first. We recommend testing out your weight changes on one glyph so you can undo and redo it until it's the weight you want. Once you've decided on the amount to increase or decrease the weight of your font, you can select all the glyphs in the font window and with one command, apply that amount to the entire font.

Naming your font

You can name your font by choosing **Font Info** from the **Element** menu. The Font Information dialog box appears. In the Easy mode only the **Family name** and **Design parameters** are available. For this exercise, name your font family something simple like *MyFont*.



*Choose **Font Info** from the **Element** menu to name your font.*

If your font is condensed, bold, italic or have some other style parameters use pop-ups in the **Design parameters** section of the dialog. Choose appropriate width, weight and slope. For example, choose *Bold* as weight and *Italic* as slope if your font is bold-italic. If you want to add customization to your style name, then use the pop-up list **Other**. The **Family name** and **Design parameters** are enough for Fontographer to build all other names automatically.

For more information about naming your fonts refer to Chapter 7, "Font Info".

Be sure to name your font before you save your database file and generate a font. Otherwise your fonts will end up with unusable names like "Untitled Bold Italic", and you'll have to start over.

Note: Checking the **Build other values automatically** option will replace all font information (visible in the Advanced mode) with new automatically calculated values. So be careful with this option if you open an existing font that you are not going to change much.

Saving your work

You save Fontographer database files using the **Save** or **Save As** commands on the **File** menu. The database file is where Fontographer stores all the parts needed to construct any font. Just like you save documents in Microsoft® Word, or graphics in Adobe Photoshop, the database is where you save your fonts in Fontographer.

Note: Saving a font only saves the database file. To use the font, you'll have to generate it (see "Generating your font" on page 88 and Chapter 9, "Generating and Exporting Fonts" for more information about font generation).

Save

1. Choose **Save** when you create a new font. The Save dialog box appears with a highlighted text box for typing in the name of your new database font file.
2. Your database font file can be named something other than the name given to your font. The database font file contains all the information about your font.
3. Saving an existing font will save changes you have made to a file since the last time you saved it.

Note: Name your font in the Font Info dialog box prior to saving a database for the first time, or you'll end up with a name like "Untitled-1".

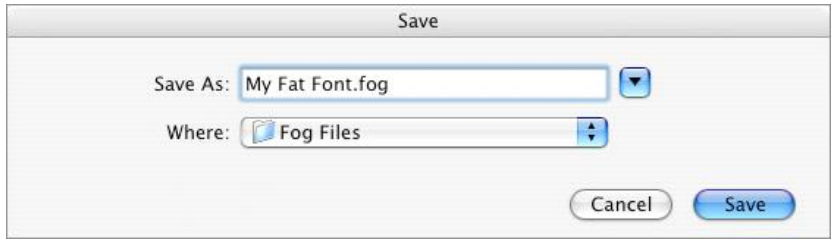
The standard file saving dialog box appears. You can name your databases anything you like, because there's no relationship between the name of the actual font you'll use in your programs and the name of the database itself.

Tip: Using **Save As** is a quick shortcut for changing database names.

Note: You can skip this step if you don't want to keep your database file for future reference, and go directly to "Generating your font" on page 88.

Save As

Choose **Save As** when you want to make another copy of a currently open font file. This will save the changes you made, without overwriting the original file. You can save the file in its current directory or you can save it in another location listed in the dialog box.



*Choose **Save As** from the **File** menu.*

You may create and name a new folder in which to store your fonts.

Reverting to the last saved version

Reverting to the last saved version can be done whenever a font is open and changes have been made to it. Choose **Revert to Saved** from the **File** menu to go back to the last saved version.

Another way to get rid of edits to the last saved version is by clicking the close box and then clicking the **Don't Save** button.

Generating your font

After you save the file, it's time to generate an installable font. You must do this if you want to use the font in another application besides Fontographer.

Fonts are composed of different files that you will need to install before you can use the font. For more about installing fonts, refer to Chapter 11, "Installing and Removing Fonts".

Choose **Generate Font Files** from the **File** menu.

The Generate Font Files dialog box offers a number of options (including the ability to generate fonts for several computer platforms), but for the purposes of this exercise you'll use the easy mode. Choose the platform you're generating fonts for and select OpenType TT/TrueType for the format. We've typed some commonly used bitmap sizes in our example and you can do the same. However, bitmaps are only necessary if you'll be using a Type 1 (PostScript) font on the Macintosh.

The **Change** button gives you the option of generating your fonts directly into a specific folder. This saves you the extra step of moving files into folders later.

The **Overwrite existing files** option lets you replace an existing file (that has the same name) with a new file. If you don't choose this option (and have a font with the same name), Fontographer will create a new font with the same name followed by a number (2, 3, etc.). For more information about font generation options, refer to Chapter 9, "Generating and Exporting Fonts".

Installing the font

Since installing fonts is different depending on the platform and operating system you're using, we can't really cover this in a quick how-to here. If you need more information about installing fonts, refer to Chapter 11, "Installing and Removing Fonts" or your system's User Manual.

Using the font

Once you've installed the font, go to the application of your choice, type some text, and select your font (just like you would any other font) from the **Font** menu.

This is the way the font used to look.

This is the way it looks now as a fat font.

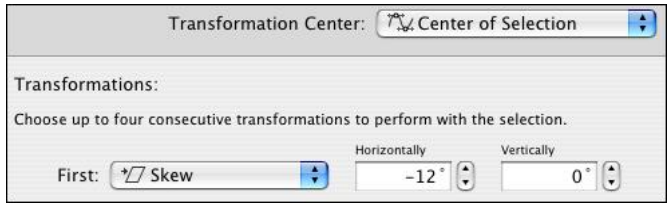
About font piracy

"Although editing your existing fonts is fine, pirating fonts or using pirated copies of other artists' type design work and reselling it is not. All big and small design studios are filled with artists who take pride in their creative contributions. Appropriating their typefaces, duplicating them, renaming them, and offering them for sale is unethical. Even if you aren't the one involved in the pirating, using fonts you know have been pirated is unethical as well".

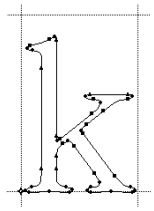
—Joe Treacy
President & Director of Typography, Treacyfaces Inc.

Creating an oblique font

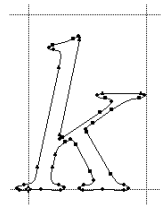
By using Fontographer's skew feature, you can create your own oblique font. You can consider this an easy way to make an oblique typeface. Actually, it's like cheating since an oblique font is just a right-slanted version of a Roman typeface; a true italic typeface has redesigned glyphs that compliment the face. But, skewing is a really easy way to create a new typeface that can add emphasis to your text.



*Choose **Skew** from the **Transform** dialog box.*



Original glyph



Skewed glyph

Again, Fontographer allows you to skew one, several, or all glyphs at once. As in our previous example, we recommend that you try out your modifications on one glyph before you apply the transformation to the entire font.

Follow the steps given in the following exercise to open your font.

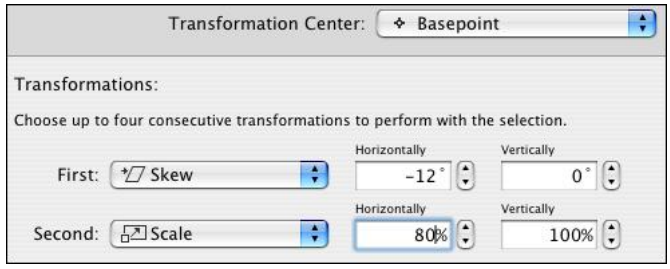
To skew a glyph:

1. Go to the Font Window and double-click the glyph "k" to open it.
2. Choose **Transform** from the **Element** menu.
3. Drag down in the First transformation pop-up until you've selected the **Skew** option and made sure the other pop-ups say: "Do nothing".

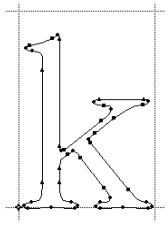
   Fontographer defaults to a horizontal skew value of -12 degrees (the appropriate angle for an oblique font).
4. Click the **Transform** button, and Fontographer skews the "k". Fontographer gives you the option of applying other transformations at the same time you skew the glyph.

To apply more than one transformation:

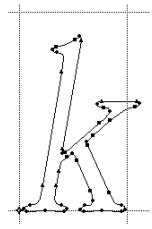
1. Double-click the “k” and select **Undo Transform** from the **Edit** menu to undo your last move.
2. Select **Skew** from the Transform dialog box and enter -12 degrees Horizontal (leave Vertical at 0).



Choose more than one transformation at once.



Original glyph



Transformed glyph

3. Then select **Scale** and enter “80” in the Horizontal text box.
4. Select **Basepoint** from the *Center transformations around* pop-up.
5. Click **Transform**.

   Fontographer skews and condenses the “k” at the same time.

Try some of the other transformation options to see what effects they have on your glyph.

Once you’ve finished trying out all the options, you can apply the transformation to several glyphs, or the entire font, by selecting groups of glyphs in the font window.

Skewing multiple glyphs

You can skew, scale, flip, or move either a single glyph or a range of glyphs. Select more than one glyph in the font window by holding down the **Shift** key while clicking glyphs. Select a range of glyphs by dragging through the glyphs. In this way you can apply transformations to one, several, or all glyphs.

Creating a fraction using composite glyphs

Have you ever checked how many fractions are in your font? If you have, you know that most commercial fonts have a small number of fractions, if any. Historically, if you wanted to type the fraction 3/8, you had to type the 3, the forward slash, and the 8, and even then the fraction didn't look good. Fontographer makes it extremely easy to add composite glyphs made up of more than one glyph combined in a single glyph slot to your font. So you can now have traditional fractions in all your fonts.

To create a fraction:

1. Go to the Font window and double-click the "3" to open it.

Note: We created this fraction in the"3" slot for illustrative purposes. You will probably want to create your fractions in unused glyph slots.

2. Click the "8" glyph slot in the Font window.

   (It's not necessary to open the Outline window to copy the glyph) Then select **Copy Component** from the **Edit** menu.

3. Click the "3" outline window and choose **Paste** from the **Edit** menu.

   Fontographer pastes the number 8 on top of the 3.

4. Select everything in the Outline window by choosing **Select All** from the **Edit** menu.

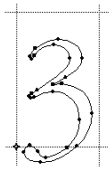
5. Go to the **Transform** menu, select **Scale Uniformly** as your first transformation, enter 60%, and make sure all the other transformation pop-ups say "Do nothing". Fontographer scales both glyphs to 60% of their original size.

6. Position your pointer away from the glyphs, and click nothing to deselect everything (or simply press the **Tab** key, which always deselects everything). Then click the mouse on the outline of the 8.

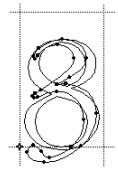
   A box will appear around the number 8 (this represents the composite glyph's bounding box). Composite glyphs do not show the points you normally see. (You cannot edit points in a composite glyph unless you first choose **Decompose Component**.)

7. Drag the box containing the 8 toward the lower right corner.
8. Double-click the path or any point on the number 3 to select all of it, and then move the 3 toward the top left corner.

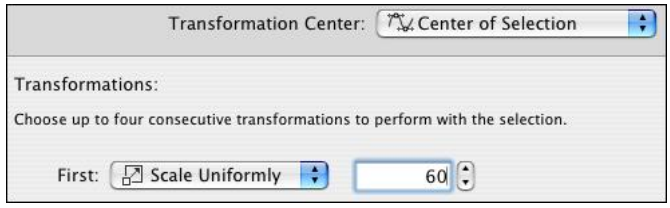
   You can create the divisor line by copying the forward slash into your glyph. You can also draw the divisor line if you prefer. However, it is often much easier to use existing glyphs to create parts.
9. Select the forward-slash glyph in the font window.
10. Choose **Copy** from the **Edit** menu.
11. Paste the forward-slash glyph into the 3-glyph slot.



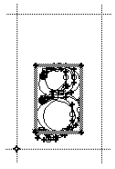
Open a glyph.



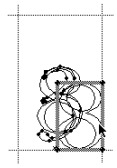
Paste in another glyph using the ***Copy Component*** command.



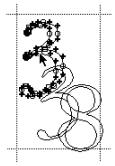
Choose **Scale** *from the Transform dialog box.*



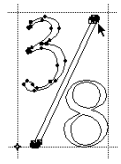
Fontographer scales both glyphs.



Move the eight toward the bottom right corner and the three toward the top left corner.



Move the three toward the top left corner.



Copy and paste the forward slash glyph into the outline window.

If you like, you can choose ***Preview*** *from the* ***View*** *menu to preview your new glyph.*

Make any change to the original glyph.

Since it's a reference glyph, the change is made in any glyph that references it.

To change a reference glyph:

1. Open the Outline window for the actual glyph 8.
2. Modify some part of it.

   You'll see any changes you make to the original number 8 reflected in the denominator you created in your fraction glyph.

➲ ***Tip:*** If you ever decide to change one of the denominators or numerators in your fraction after you've created the whole set, you'll realize the advantage of reference glyphs. If the entire set uses reference glyphs, all the fractions in the set will be updated automatically when you modify the original elements.

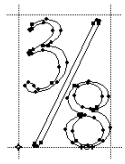
✎ ***Note:*** Because a composite glyph references a drawing rather than duplicating it, all the instructions required for creating that shape in the printer need only appear once, thus saving memory and processing time.

Unlinking a reference glyph

Fontographer also lets you remove the link from any composite glyph. This gives you access to the points in the glyph as well as removing the link to the original glyph.

To unlink a reference glyph:

1. Click the fraction you created (in the 3 glyph slot).
2. Choose **Decompose Component** from the **Edit** menu. As you can see in the illustration, the glyph's points are now visible and you can move them individually or together as a group.



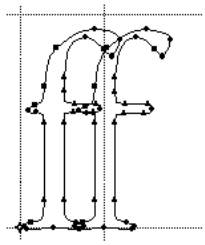
*Choose **Decompose Component** to remove any reference to the original glyph.*

Creating a ligature

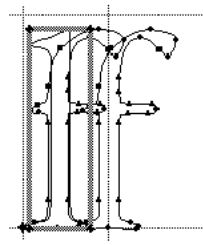
A ligature is a glyph made up of one or more glyphs. Most commercial fonts have some commonly used ligatures like æ, and œ. However, Fontographer makes it easy to create ligatures of your own without drawing a thing.

To create a ligature:

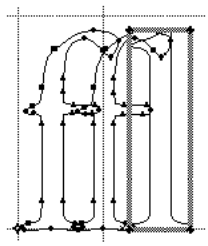
1. Open the outline window for the “f”.
2. Choose Select All and then choose Duplicate to create another “f”.
3. Move the new “f” to the right.
4. Go back to the font window.
5. Copy the “l” into the same window using the **Copy Component** command from the **Edit** menu.
6. Move the referenced “l” to the right of the second “f”.
7. Choose **Decompose Component** from the **Edit** menu.
8. Choose **Remove Overlap** from the **Element** menu.



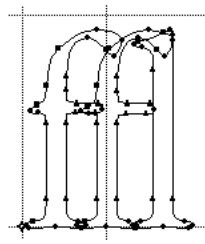
Duplicate the “f”.



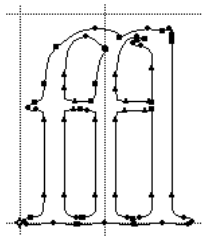
Copy a reference glyph into the window.



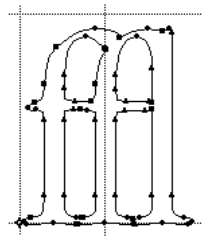
Move it into place.



Choose ***Decompose Component****.*



Then choose **Remove Overlap**.



Remove any extra points.

Changing the glyph width

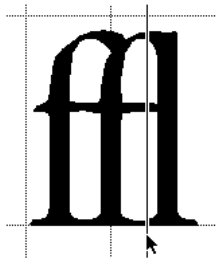
You probably noticed the vertical line that runs through the second “f”. This is the glyph width line. Width is a moveable vertical line that specifies the width of each glyph. When you print a line of text, the origin line of the next glyph is placed on top of the width line of the current glyph. Since you changed the contents of this particular outline window, it’s important that you change the width as well.

To change the glyph’s width:

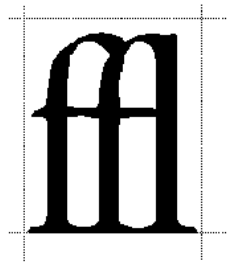
1. Make sure you still have the outline window for the “f” open.
2. You can choose **Preview** and turn off **Show Points** from the **View** menu to get a better look at the glyph if you like.
3. Move the width line with the selection pointer.

Note: We created this ligature in the “f” slot for illustrative purposes. You will probably want to create your ligatures in an unused glyph slot.

In some word processing and page layout programs, you can set the preferences to automatically substitute curly quotes for straight ones, or the fl ligature if you type “fl”. For the substitutions to occur, you’ll need to be sure to use an Adobe encoded font.



Drag on the width line...



until it’s in the position you want.

Tip: It’s sometimes easier to move and scale a reference glyph, since you don’t have to worry about selecting all the points. You can unlink the reference once you’ve moved the glyph into the position you want.

Creating a condensed glyph or font

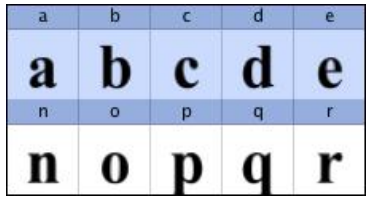
Fontographer has the ability to modify glyph images to produce interesting special effects. You can create these effects in either the outline window (on one particular glyph) or the font window (on the whole font). For example, you can create an oblique font by selecting all the glyphs in the font window and skewing them -12 degrees (like we did earlier in this chapter). Or you can create an extended font, by increasing the horizontal scaling factor of the font. Your options are limitless.

You can also create a condensed font by scaling the glyph 80% horizontally. Condensed versions of a font are the same height as their counterparts but are narrower to fit into a more compact space.

Note: It is not necessary to have the outline window open to modify more than one glyph at time.

To create a condensed glyph:

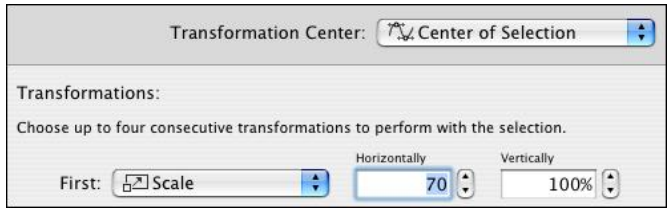
1. Click the Font Window to make it active, and then press and hold down the mouse button while you drag through the glyphs “a” through “e”.



   Click the glyphs you want to scale.

2. Choose **Transform** from the **Element** menu.

   The Transform dialog box appears.

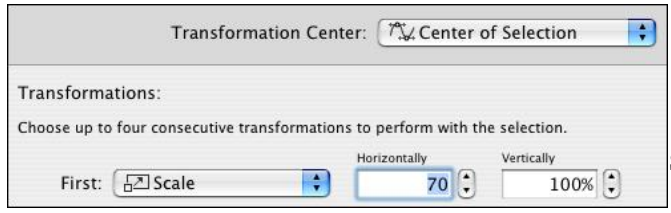


   Select Transform from the Element menu and then enter the amount to scale.

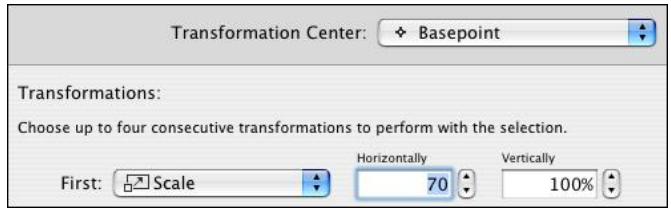
As you can see from the illustrations on this page, Fontographer scales the glyphs horizontally without changing their height.

You can apply the transformations in one of four ways:

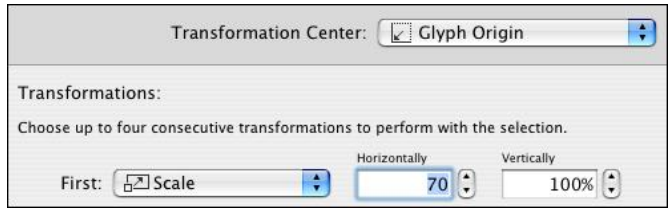
Original glyph



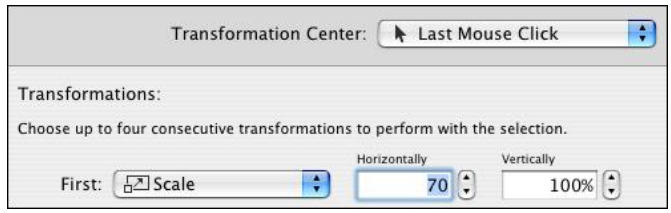
Scaled from the center of the selection



Scaled from the basepoint



Scaled from the glyph's origin

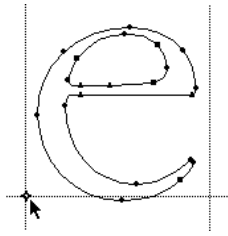


Scaled from the last mouse click

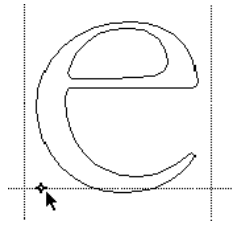
Setting the basepoint

The baseline is the line upon which the letters sit. The baseline position is always at a vertical location of zero. The dot at the intersection of the origin line and baseline is the basepoint.

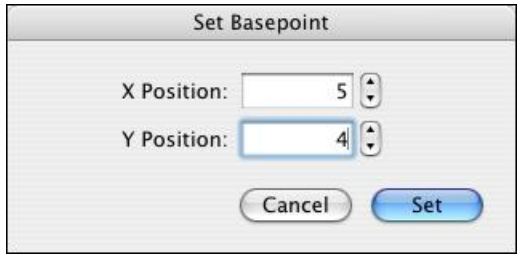
The basepoint is used to accurately and quickly align points and glyphs. The basepoint is generally at the glyph's origin (where the origin line and the baseline intersect at 0,0); however, you can position the basepoint anywhere. Fontographer allows you to set each glyph's basepoint differently. It can be moved as needed by selecting the pointer tool and dragging it to a new location, or by entering a specific horizontal and vertical location.



Select the basepoint.



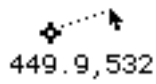
Drag it to a new location.



You can also enter location values in the Set Basepoint dialog box.

For precise numeric entry, choose **Set Basepoint** from the **Points** menu. To reset the basepoint back to the origin point, choose **Reset Basepoint** from the **Points** menu.

If you select one point and choose **Set Basepoint**, Fontographer will move the basepoint to that point. However, if you select more than one point, Fontographer will set the basepoint to the center of selection.



In the position display in the outline window, distance from the basepoint is continuously updated as the pointer moves within the drawing area. This onscreen measuring tool makes it easy to measure glyph parts. Just place the basepoint on a reference point of the glyph and watch the position display as you move the pointer. Horizontal or vertical alignment of points is very easy to check; set a basepoint on one point, then drag the other point until the horizontal or vertical delta is zero.

Creating New Fonts

Imagine being able to digitize an old fashioned typeface and transfer it to your documents, or that logo you created with a pen or pencil before you'd ever heard of a graphics program. What about the beautiful effects that are created with calligraphy fountain pens? Wouldn't it be great if you could use all of these in your word processor? With Fontographer, any of these scenarios is simple.

3

Fontographer takes what used to be possible with only pen and pencil, and puts it into the hands of the desktop designer.

Calligraphic and graphic fonts by Judith Sutcliffe

Hebrew fonts and classic fonts by Dennis Ortiz-

Fun and wacky fonts by Paul Sych

Autotracing

The bitmap option is one of Fontographer's most advanced features. Autotracing is probably most useful for tracing scanned images. Suppose you have an existing character, logo, or image that you want to assign to a keystroke. On the Macintosh, you can scan your image, save it in PICT format, and place it into your Scrapbook or Clipboard. Then you can paste your character into the outline window (where it will be used like a background template) and let Fontographer autotrace the image.

Note: Trying to autotrace bitmaps will not give good results. See Chapter 5, "Editing Bitmaps".

To paste an image into the Template layer:

It's remarkably easy to paste an image into the Template (or background) layer. Select an image from the *Scanned images* file in the *Sample files* folder in your Fontographer folder. In this example, we use a scanned Vivaldi "f".

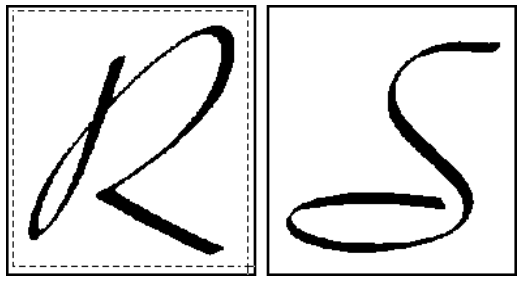
- On the Macintosh, copy the image to the Clipboard and paste it into the outline window.

Fontographer will automatically paste the image into the Template layer where it will be shown as a dimmed image.

Note: Images pasted into the Template layer are automatically sized to fit the font UPM size. If you hold down **SHIFT-OPTION**, the image will automatically scale to fit between the ascender and baseline.

Tips: All your characters should be the same size before you copy them to the Clipboard. If you neglect to do this, your background images will appear in varying sizes, and you will have to spend time making adjustments. Therefore, make all size adjustments in the scanning or drawing program of your choice, and try to drag-select (marquee) each character with the same size box (marquee area) before you paste the image into the Template layer.

To make sure you copy the image in at the same size, open your scan in a program that can preserve the full resolution of copied bitmaps in its Clipboard. Draw a box around the largest of your characters (the “M” or “W” would be a good choice). Use this box to define the area you’ll be marqueeing and then copying into Fontographer. You could even draw lines to indicate the baseline and width marks if you like.



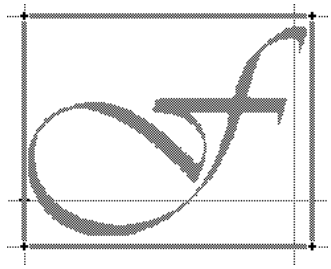
Marquee around an area just inside the bounding box.

To move a Template image:

1. Click the Template layer to select it.

2. Click the template image with the selection pointer.

   A gray bounding box appears.

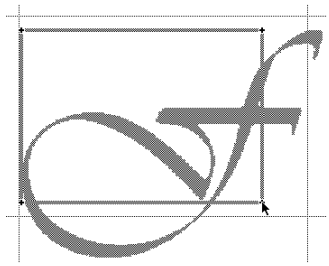


3. Move the image into place by positioning the pointer inside the image, then dragging it to a new location.

To resize a Template image:

1. Click the template image to select it.
2. Choose **Transform** from the **Element** menu and Scale uniformly 90 percent.

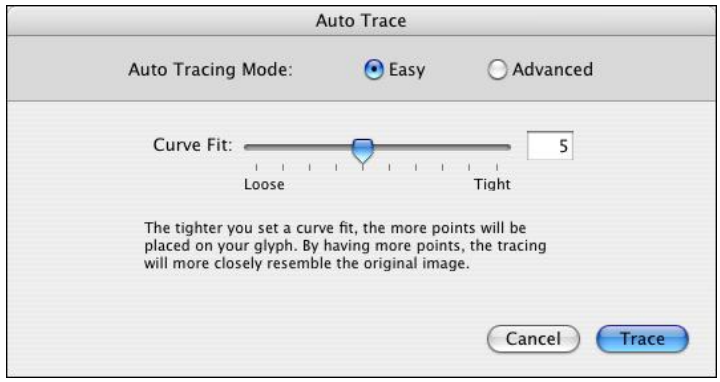
Note: Dragging the handles interactively scales the background image as well.



Tracing an image

Once you have your image sized and positioned the way you want, you can trace it.

1. Make sure you are in the Outline layer.
2. Choose **Auto Trace** from the **Element** menu. The Auto Trace dialog box appears. You have two options: Easy and Advanced.

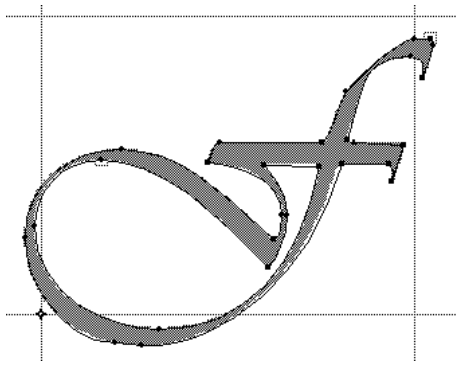


3. Choose **Easy** and keep the default **Curve fit** of "5".

   The tighter you set a curve fit, the more points will be placed on your character. By having more points, the tracing will more closely resemble the original image. However, too many points will consume unnecessary memory without appreciably improving the accuracy of your path. It is better to use as few points as possible to get the shape you desire.

Note: If the image is jagged, a tight trace setting will try to make all those jaggies mean something. The better your scanned image, the tighter you can trace.

When the Tracing progress dialog box finishes generating, you will have a completely traced character in the outline window.



You can cancel the tracing operation at any time by clicking the **Cancel** button or by typing **COMMAND-PERIOD**.

➲ ***Tip:*** When you are through tracing, you can select the background image from the Template layer and delete it. Background images you save with the font can consume disk space very quickly so you should remove them once you are through tracing.

Advanced tracing options

Fontographer's Advanced tracing mode offers some specific options.

Curve fit

Choosing an item from this pop-up will set all the other controls in the dialog box to the recommended settings for Loose, Normal, or Tight fit. Try changing the value of this pop-up a few times and see how the other controls respond. It might give you an idea of how each slider affects the fit tightness.

If you change any of the other controls in the dialog box, the **Curve Fit** pop-up will automatically switch to Custom to indicate that you have customized the values. Once you have customized the settings you can always go back to Loose, Normal, or Tight by changing the **Curve Fit** pop-up back to one of these settings. You can switch back to Custom as well. Normal is generally the best all-purpose selection. Tight would be a good selection for more intricate designs, while Loose would be good for characters with straight angles (such as block letters) and poor quality scans. Choose Custom if you want to set the Curve fit options manually.

Allow curve fit errors

This control will have the largest affect on your tracing results. A low value means Fontographer will allow fewer curve fit errors, and you'll get a very tight trace with more points. A higher value means Fontographer will allow more errors, and you'll get a loose trace with fewer points.

Balance lines

This control will have very subtle, almost unnoticeable affects on your tracing results. A low value means it will do almost nothing. A high value means Fontographer will attempt to align lines when necessary. For instance, it might try to align the left and right parts of the crossbar in a "T" character.

Eliminate close points

This control can help eliminate redundant points (points that are almost on top of each other). A low value means that almost no points will be eliminated, and the shape of the path will be the most accurate. A high value means that it will eliminate as many points as necessary, but it may slightly alter the shape of the curve in order to do so.

Make straight lines

This control has very subtle effects. It determines how straight a curve should be before it is turned into a straight line. This will never turn extremely curvy paths into straight lines; however, curves that appear to be almost straight to begin with may be slightly modified so that they are perfectly straight. A low setting for this control means almost no curves will be straightened. A high value will cause more curves to be straightened.

Look for cusps

When Fontographer traces an image, it often finds places where two paths join at a sharp angle. A join of this type is called a cusp, and Fontographer will always place a corner point at such a location. The **Look for cusps** control determines how lenient Fontographer is in finding cusps, and thus it will have an effect on how many corner points are used in the tracing results. Setting this control to a low value means it will find very few cusps, and the results won't have many corner points. Setting the control to a high value means it will find many cusps, and the results will have more corner points.

Treat nearly flat paths as straight lines

This check box is similar to the **Make straight lines** control; however, it differs in a subtle way. The **Make straight lines** control can help straighten any curves that are nearly flat. However, the **Treat nearly flat paths as straight lines** checkbox only straightens curves that are nearly flat, and that only bend to one side. For example, this control can straighten a C-shaped curve that bends to the left of the straight line, but it can't straighten an S-shaped curve that bends to either side of the straight line. If you think the difference between these two controls is too subtle for your needs, then we recommend that you ignore this control and just use the **Make straight lines** control.

Find extrema points

You should probably leave this checkbox turned on. It will make sure that points are always placed at extrema points in the tracing results, and this is recommended for Type 1 (PostScript) and TrueType fonts.

Transformation options

Fontographer's transformation options are located under **Transform** in the **Element** menu. Any of these can be applied from the font window or outline window. When used from the font window, you can apply a transformation to one, several, or all glyphs. On the other hand, in a glyph's outline window, you can only apply the transformation to the selected points. If there are no selected points, the transformation applies to the entire glyph.

Flip

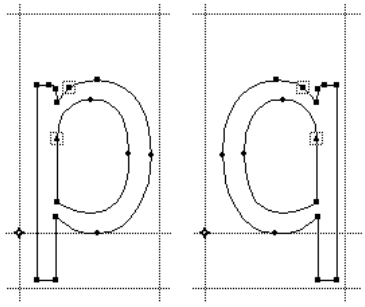
Use the Flip transformation option to flip the image to the opposite side of an imaginary horizontal or vertical line.

You access the **Flip** pop-up via the **Transform** menu or by double-clicking the Flip tool in the tool palette.

To flip selected items horizontally:

1. Select a glyph.
2. Choose the **Flip** pop-up as the first transformation.
3. Click the **Horizontal** radio button.

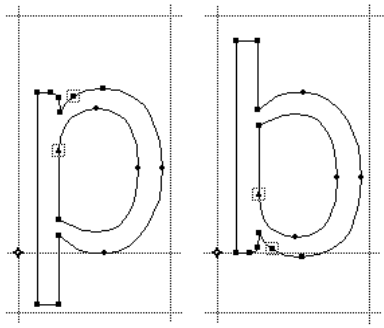
4. Click the **Transform** button.



Change the "p" to a "q" using the Horizontal Flip transformation.

To flip selected items vertically:

1. Choose the **Flip** pop-up as the first transformation.
2. Click the **Vertical** radio button.
3. Click the **Transform** button to apply the vertical flip.



Change the "p" to a "b" using the Vertical Flip transformation.

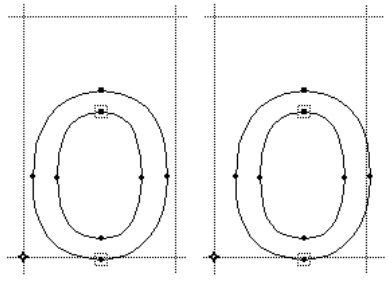
Move

The Move transformation option can move whole glyphs, or a specific point a specified horizontal and/or vertical amount.

To move selected items horizontally:

1. Choose the **Move** pop-up as the first transformation.
2. Enter a value in the Horizontal text box.

3. Click **Transform** to move the image.



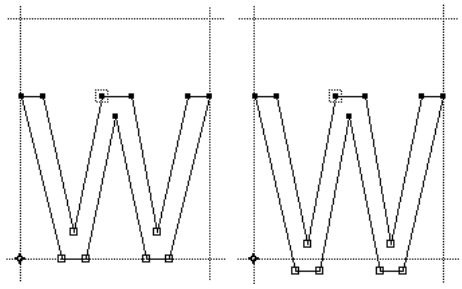
Move the "O" to the right using the Move transformation.

Note: Entering a negative horizontal value moves the image to the left.

To move selected items vertically:

1. Choose the **Move** pop-up as the first transformation.
2. Enter a value in the Vertical text box.

3. Click **Transform**.



Move the lowermost four points on the "w" using the Move transformation.

Notes: Entering a negative vertical value moves the image down in the window.

Move an image horizontally and vertically by entering values in both text boxes.

Rotate

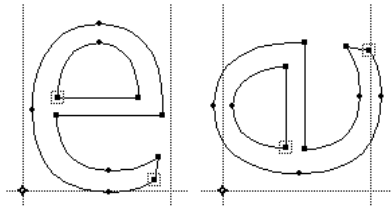
Use the Rotate transformation option to rotate the selection a specified number of degrees. Selected points rotate around the reference point by a specified angle. Positive angles indicate a counterclockwise rotation, while negative angles specify a clockwise rotation.

You access the **Rotate** pop-up via the **Transform** menu or by double-clicking the Rotate tool in the tool palette.

To rotate a selected item:

1. Choose the **Rotate** pop-up as the first transformation.
2. Enter a rotation angle in the text box.

3. Click **Transform**.



Rotate the "e" using the Rotate transformation.

Scale

There are two Scale transformation options: Scale and Scale uniformly. Both of these options are used to increase or decrease the size of an image by a specific scale factor.

Use the Scale transformation option to scale horizontal and vertical attributes independently of each other. You'll find this feature useful when you want to create condensed and extended versions of a font, since you can apply the scaling transformation to the entire font.

You access the **Scale** pop-up via the **Transform** menu or by double-clicking the Scale tool in the tool palette.

To scale a glyph vertically:

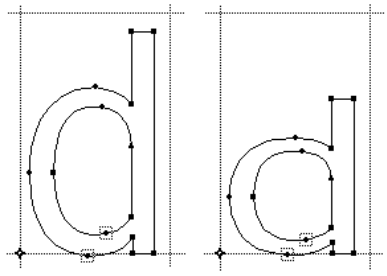
1. Choose the **Scale** pop-up as the first transformation.
2. Enter a vertical scaling value.

3. Click **Transform**.

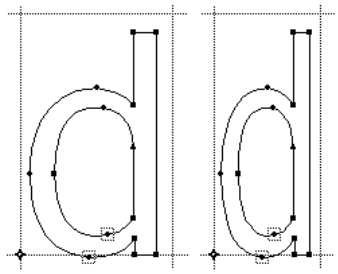
To scale a glyph horizontally:

1. Choose the **Scale** pop-up as the first transformation.
2. Enter a horizontal scaling value.

3. Click **Transform**.



Use scaling to create vertically scaled...



or condensed versions of your glyph or font.

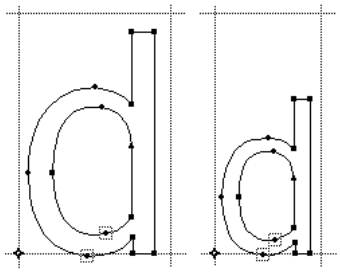
Scale uniformly

This transformation option scales the entire image uniformly. If you select a glyph and enter a scale factor of 50%, the image will be reduced to 50% of its original size (both horizontally and vertically). Doubling the size of the image would require a scale factor of 200%.

To scale uniformly:

1. Choose the **Scale** uniformly pop-up as the first transformation.
2. Enter a scaling value in the text box.

3. Click **Transform**.



Scale the "d" uniformly.

Skew

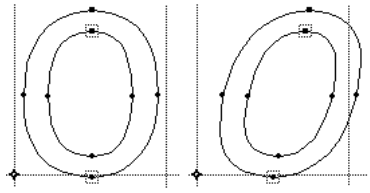
The Skew transformation option applies a slant to the image. Negative values slant the glyph to the right, positive values to the left. Vertical skewing can be used to create oblique glyphs. Italics are normally skewed vertically by 12 degrees.

You access the **Skew** pop-up via the **Transform** menu or by double-clicking the Skew tool in the tool palette.

To skew selected glyphs:

1. Choose the **Skew** pop-up as the first transformation.

2. Enter a skew value in either the Horizontal or Vertical text box.
3. Click **Transform**.

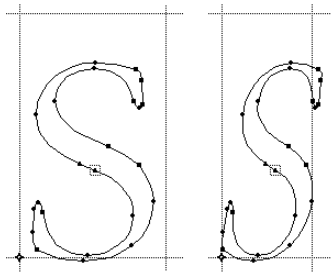


Skew the "o" with the Skew transformation.

Multiple transformations

There are times when you'll probably want to do more than one transformation at once. Use Fontographer to apply up to four transformations at one time to one glyph, or to the entire font.

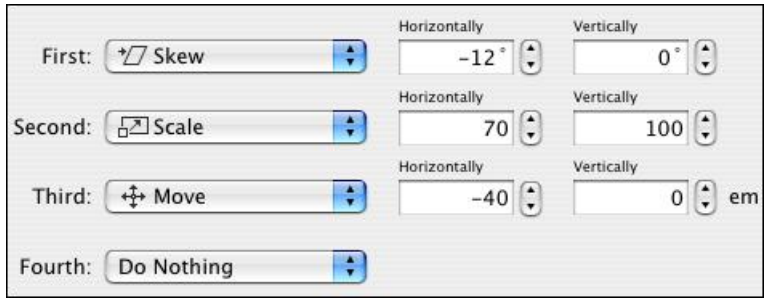
Suppose you want to create a condensed oblique font and move it closer to the baseline to compensate for the skew angle. It's easier than you might think.



Scale, skew, and move a glyph (or even the whole font) with one command.

To apply multiple transformations:

1. Select a glyph.
2. Choose **Transform** from the **Element** menu.



3. Select up to four transformations.
4. Enter the transformation values.
5. Click **Transform**.

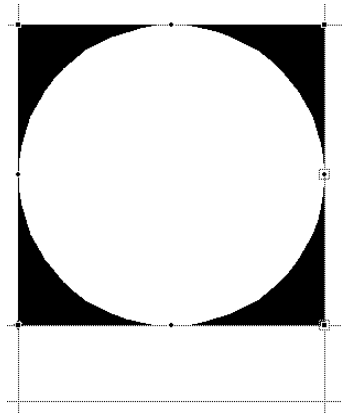
3D transformations using the Transform dialog box

Using the Transform dialog box to apply 3D transformations is simpler if you use the tools to set up the transformation. Double-clicking the Perspective tool will bring up the Transform dialog box ready to apply a 3D rotation transformation. **Option**-double-clicking the selection tool will bring up the Transform dialog box ready to apply a 3D move transformation. Let's do an example of a 3D rotation. In our example, we will draw and then transform a square/circle.

To use the Scale tool:

1. Draw a square and a circle (holding down the **Shift** key to constrain the tools) that start at the origin point and extend to the descent.
2. Drag the width line on top of the rightmost point on the circle.
3. Choose **Correct Path Direction** from the **Element** menu.

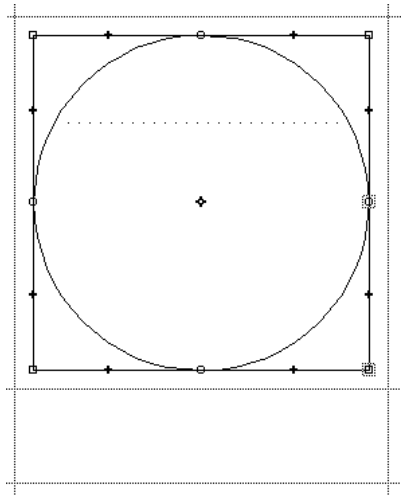
   The glyph should look like this with **Preview** selected from the **View** menu or by pressing **Command-L**:



4. Choose **Select All** from the **Edit** menu.
5. **Option**-double-click the Scale tool to bring up the Transform dialog box with Scale uniformly as the first transformation.
6. Choose Center of Selection from the Center transformations around pop-up at the top of the dialog box.

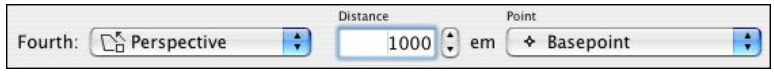
7. Type 90 into the text edit field and press **Return** or click **Transform**.

   The glyph should look like this in Outline mode:



To use the Perspective tool:

1. **Option**-double-click the Perspective tool to bring up the Perspective Setup dialog box.



2. Set the Distance to 1000 and the Point to Basepoint and press the **Return** key or click on **Transform**.

   This tells the Perspective tool that the image you see in the outline window is being viewed as if you are 1000 em units away from the basepoint.

3. Choose **Select All** from the **Edit** menu.

4. Choose **Set Basepoint** from the **Points** menu. Your basepoint just moved to the center of the square/circle, which is defined as the perspective point in this example.

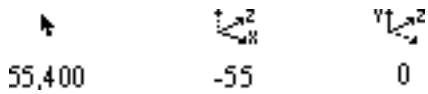
5. Choose **Copy** from the **Edit** menu.

   You will paste this copy later in this example.

6. Click and hold down the mouse on the origin line (the line that extends from the bottom of the window to the top if the window along the left side of the glyph).

7. Drag the mouse to the right while holding down the **Shift** key.

   As you drag the mouse, you will notice that the information bar looks something like this:

8. Continue dragging to the right until the number underneath the symbol reads -90.

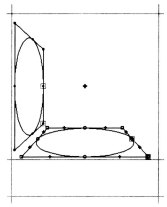
   You have just rotated the selected points by 90 degrees in the XZ plane. You can also think of it as rotating around the Y axis.

9. Choose **Paste** from the **Edit** menu. You can now apply the next transformation to the copy of the original.

10. Click down with the mouse on the baseline.

11. Now drag the mouse up while holding down the **Shift** key until the number under the symbol reads 90.

The glyph in the outline window should now look something like this:



To apply 3D transformations using the Transform dialog box:

Let's do an example of a 3D rotation. To make things simple, you will start where you left off using the Perspective tool. For this example to work, the square/circle being transformed should already be copied into the Clipboard.

1. Choose **Paste** from the **Edit** menu.

2. Choose the Perspective tool from the tool palette.

3. Click down with the mouse on the width line and release the mouse button immediately. The mouse click will be used as the center of the transformation in the Transform dialog box.

Note: Any tool can be used for Step 3, but if you use the Selection tool, the selection will be lost.

4. Double-click the Perspective tool.

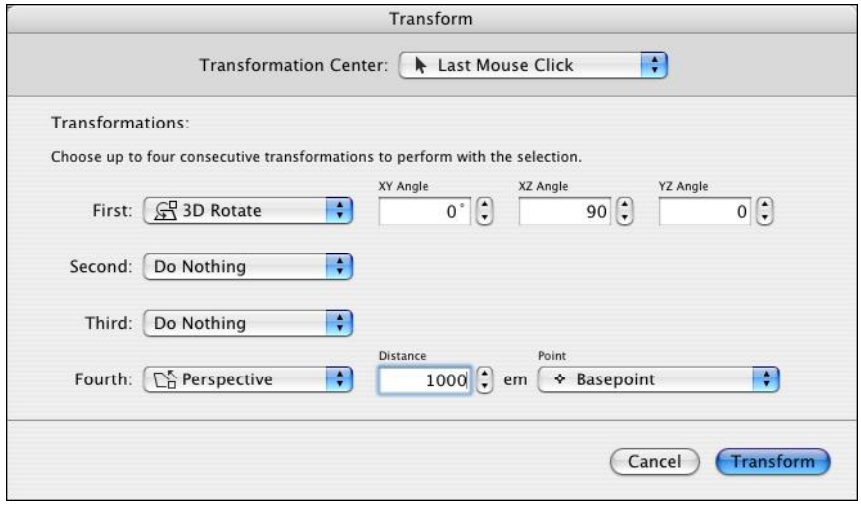
   The Transform dialog box will be brought up ready to apply a 3D rotation.

5. Choose Last Mouse Click from the **Transformation Center** pop-up.
6. Set the XY angle to 0, the XZ angle to 90, and the YZ angle to 0.
7. Set the Perspective **Point** pop-up to Basepoint.

Notes: By default the **Distance** is set to the em square. In our example the em square is 1000.

If you wish to set up additional transforms in this dialog box, then make sure you do the perspective transform last. Any transforms that occur after the perspective transform will not have a three-dimensional appearance.

When you are done, the Transform dialog box should look like this:



8. Press the **RETURN** key.

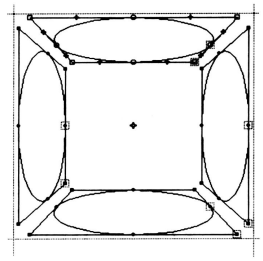
To create the top of the cube:

You have probably noticed that a three-dimensional box or cube is being created. The previous set of steps created the right side of cube. The next set of steps will create the top side of cube.

1. Choose **Paste** from the **Edit** menu.
2. Click down on the ascent line.
3. Double-click the Perspective tool.
4. Set the XZ angle to 0 and the YZ angle to -90.

5. Press the **RETURN** key.

   If you have been following our example from the beginning, your glyph in the outline window should look something like this:



To do a 3D Move:

Let's continue where we left off, creating a disjointed cube. For this example to work, the square/circle you are transforming should still be copied into the Clipboard.

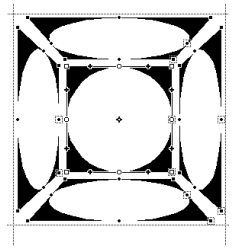
You will now create the back side of the cube by moving the selected points backward (by 800 em units) into the third dimension.

1. Choose **Paste** from the **Edit** menu.
2. **OPTION**-double-click the selection tool to bring up the Transform dialog box.

   The dialog box will come up showing Move and Perspective, ready to do a three-dimensional move.
3. Set both the first transformation (at the top of the dialog box) and the Perspective transform (at the bottom of the dialog box) to Basepoint.
4. Change the Horizontal and Vertical text edit fields to 0, and change the Depth edit text field to 800.

Note: If a Perspective transform does not follow the Move transform, the Depth field will not be available.

5. Press the **RETURN** key or click on **Transform**.
6. Select **Preview** to view your completed glyphs:



Guidelines

Setting guidelines

The Guides layer is used to construct drawing guidelines that are common to every glyph of the font (such as the x-height line). The Guides layer is similar to the Template layer, except it is drawn in light gray or green and appears behind every glyph of the font. Guides are purposely drawn in a lighter color so they can be distinguished from the outline and template images.

Guidelines may be edited or created from any glyph's outline window. Change to the Guides layer by clicking its name in the Layers palette or by typing "g" when the lock icon is in the locked state.

Change to the Guides layer by clicking it or by typing "g"...

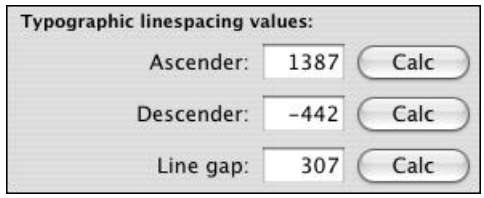
when the lock icon is in the locked state.

With the Guides layer active, you can edit or draw guidelines with the standard drawing and editing tools.

Changes made to the Guides layer will appear in every glyph in the font. You can undo changes made to the Guides layer just like you would in any other layer.

Setting guidelines from the Font Info dialog box

Ascent and descent are already defined and can be reset by choosing **Font Info** from the **Element** menu and typing the values in the Ascender and Descender text boxes.



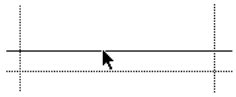
Note: Most word processing and page layout programs now support line gap (leading), but a few may still ignore the information you enter in the Line gap field.

Adding new guidelines

Additional guidelines can be set two ways in the Guides layer.

To create a guideline:

1. Select the Guides layer.
2. Use the selection tool to drag vertically from the baseline or horizontally from the origin line.



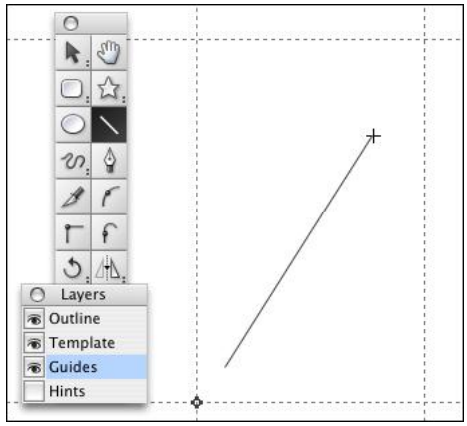
Drag from the baseline...

or origin line to create a guideline.

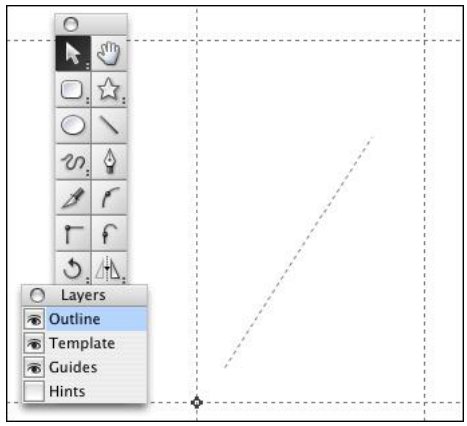
➲ ***Tip:*** Remove these guidelines by **Option**-clicking them, or use the pointer tool to drag them back into the origin or baseline from which they originated.

You can also create a guide by drawing it with any of the drawing tools in Fontographer.

1. Select the Guides layer.
2. Click the drawing tool of your choice to draw a guideline.



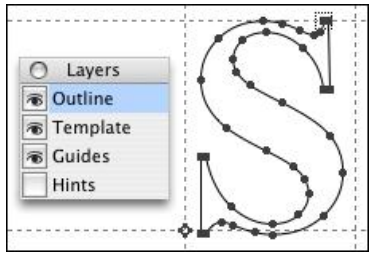
Drag your own guide in the Guides layer...



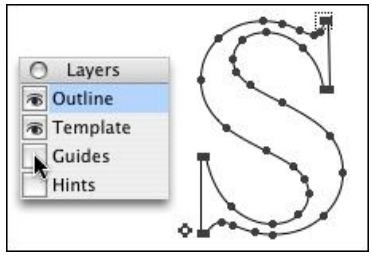
and it will appear in the Outline layer of every glyph in the font.

Hiding guidelines

You can hide the guides by clicking the Guides layer checkbox to turn it off or (when the lock icon is locked) by typing **Option-G**.



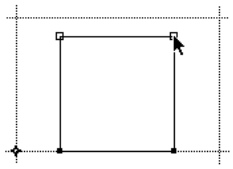
Turn guides on...



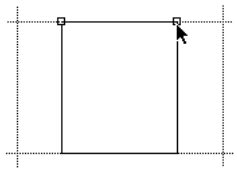
or off.

Snapping to guides

Choose **Snap to Guides** from the **View** menu. When points are within a predetermined distance (you set this value in the **Fontographer > Preferences > Editing**) from the guideline, they snap or align to that guideline.



Select the point and move toward the guideline.



The points snap to align with the guideline.

Note: The **Snap to Guides** option snaps only to baseline, origin, width, ascent, descent, and guides you pull out from the baseline or origin. It does not snap to guides you draw.

Creating a stroked font

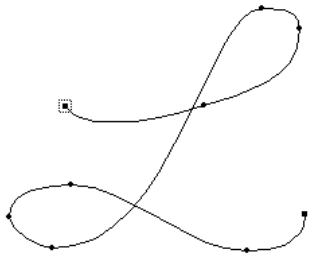
Outline versus stroked glyphs

Up until this point in the manual, we have been discussing outline glyphs. You create outline glyphs by drawing a path around the glyph's perimeter. Stroked glyphs are constructed by drawing just the centerline path. PostScript draws these glyphs by sweeping a pen along the path. The pen has a width called the stroke weight, which is defined in em units. As PostScript sweeps along the path, it paints a line that is so many units wide.

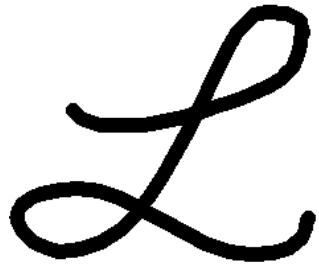
Any glyph that is constructed entirely of equal weight strokes can be drawn as a stroked glyph. For example, we drew the letters in this Fontographer logo as stroked glyphs. You must generate these as Type 3 fonts, since neither TrueType nor Type 1 fonts allow stroked glyphs.

FONTOGRAPHER

Fontographer gives you complete control over the type of pen you'll use to create your stroked font. In addition to its width, you can specify its appearance and behavior where segments join. Also, since some people prefer drawing with a pen, Fontographer makes it easy to change the stroked glyph into an outline glyph (or font).



Draw glyphs like this with the freehand tool. This glyph's stroke weigh is 40 em units.



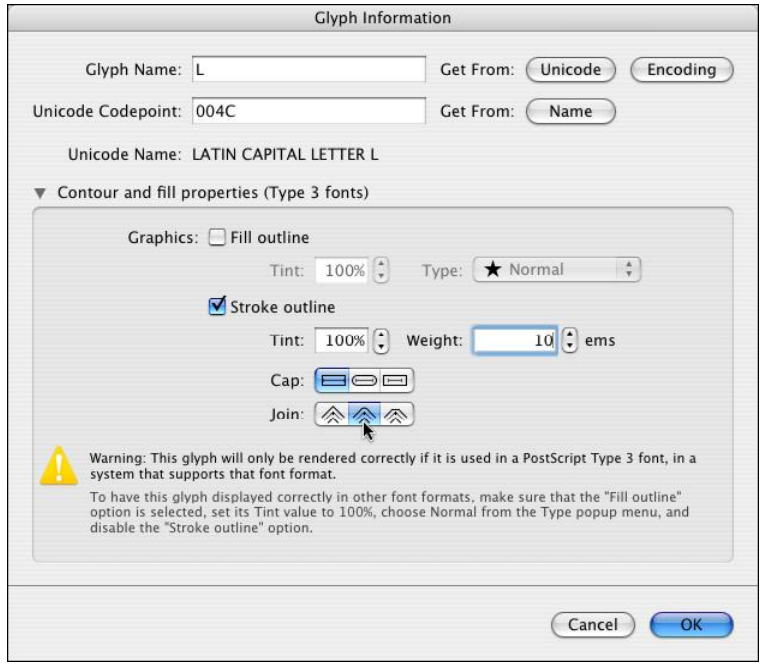
Preview the glyph with or without the points.

Setting stroke attributes

Before you can create a stroked glyph, you will need to change the attributes of the glyph from filled to stroked.

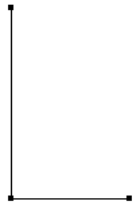
To change a filled glyph to stroked:

1. Choose **Selection Info** from the **Element** menu.
2. Turn off **Fill** and turn on **Stroke**.
3. Leave the Tint set at 100%.
4. Enter a pen Weight.

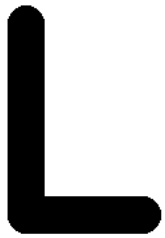


You'll notice that Fontographer has two pop-ups for **Cap** and **Join**. In our example, we use Round on both since we want the ends of the glyphs to be rounded. Each of these options is discussed after this example.

5. Click **OK** and get ready to draw a glyph.
6. Choose a drawing tool or one of the control point tools.
7. Draw an "L".
8. Turn on Preview to see what your glyph actually looks like.



Draw with a corner point tool...



to create round capped glyphs...

or square capped glyphs.

End caps and joins

There are three types of end caps: butt, round, and square.

- Butt end caps stop right at the end point of the line.
- Round end caps project a semicircle out from the end point. This semicircle has a diameter equal to the stroke weight and center point at the end point.
- Square end caps project out one half the stroke weight in the direction of the path.

There are three types of line joins: miter, round, and bevel.

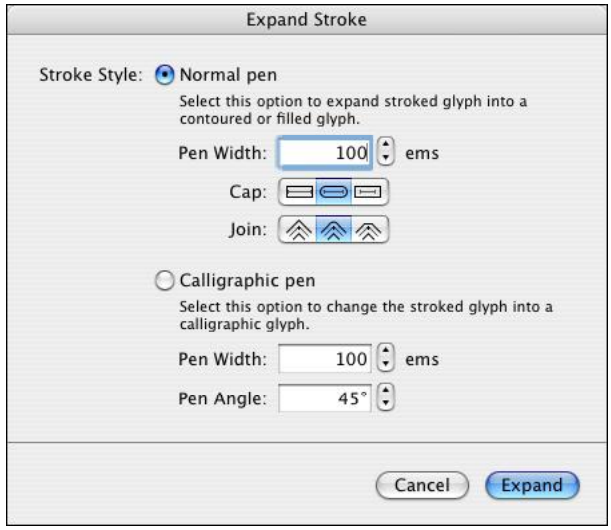
- Miter joins extend the outer edges of the lines until they meet at an angle, like the corners of a picture frame.
- Round joins draw a circle with a diameter equal to the stroke width at each bend.
- Bevel joins are drawn as if the joining segments were stroked with butt cap ends and the resulting notch filled with a triangle.

Expand stroke

The **Expand Stroke** command is used to expand stroked glyphs into contoured (outline) or filled glyphs.

To expand the stroke width:

1. Choose **Expand Stroke** from the **Element** menu.
2. Click **Normal Pen**.



3. Enter a Pen width value.
4. Click **Expand**.

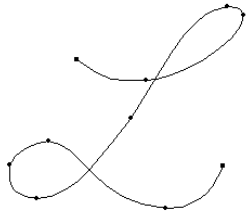
Fontographer automatically changes the stroked glyph into an outline glyph.

Change a stroked glyph... *into an outline glyph.*

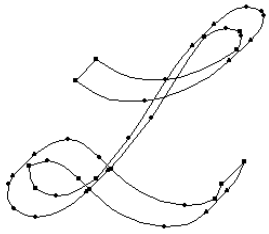
The **Expand Stroke** command can also be used to automatically change your stroked glyph into a calligraphic glyph.

To change a stroked glyph into a calligraphic glyph:

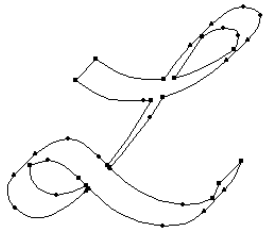
1. Choose **Expand Stroke**.
2. Click **Calligraphic Pen**, enter a value, and click **OK**.
3. Choose **Selection Info** and change the glyph to a Filled glyph.
4. Choose **Remove Overlap** from the **Element** menu.



Draw a stroked glyph with the freehand tool.



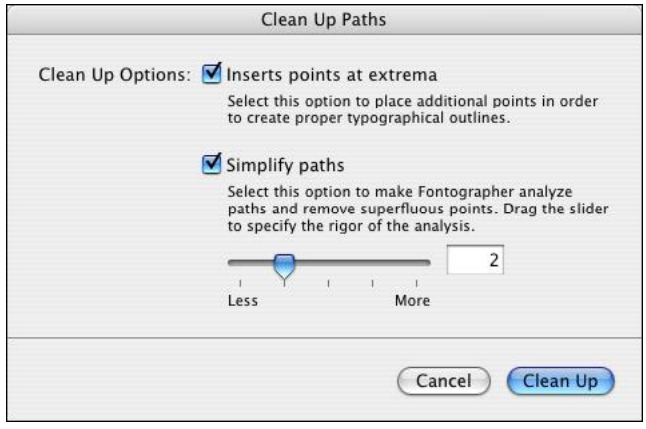
Change it into a filled glyph.



Remove any overlapping areas.

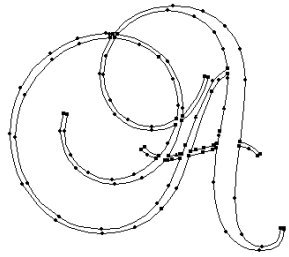
Clean Up Paths

One of the most revolutionary features in Fontographer is called **Clean Up Paths**. This incredible command automatically improves the quality of your outlines by removing unnecessary points. Fontographer will try to change the path as little as possible; less will change the outline as little as possible – more will remove more points and thus, change the path more. And if you have fewer points, your printing time will be faster as well.

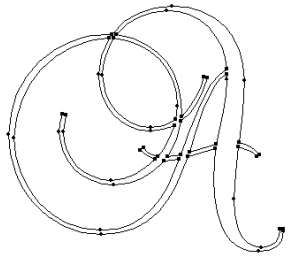


There are two different uses for this dialog box: one is to put points where they belong for proper typographical outlines. This is called putting points at the extrema. If you bring up the dialog box and only have the Insert points at extrema box checked, that's all Fontographer will do. Simplify paths will remove points it judges to be superfluous. The slider control adjusts the ratio between getting rid of a lot of points (and changing your path a little), and getting rid of fewer points and maintaining the integrity of the path.

We believe you should spend your time creating wonderful designs instead of worrying about point placement and the mechanical details of Bézier path construction. When your glyph is through, just choose **Clean Up Paths** from the **Element** menu or apply the command to the entire font directly from the font window.



Clean Up Paths *changes the glyphs with 114 points…*



to one with 53 points, without noticeably changing the outline.

Note: This feature is useful if you're converting TrueType fonts to PostScript.

Creating calligraphic glyphs

One of the most notable features of Fontographer is the freehand drawing tool that you can use directly with the mouse or with a pressure-sensitive pen and digitizing tablet. Additionally, you can use the freehand drawing tool as either a calligraphic pen or a variable-weight pen.

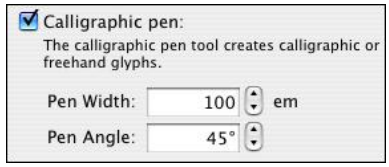
Note: If you use a pressure-sensitive pen and tablet, make the appropriate tablet settings before you draw your glyphs.

To use the calligraphic pen:

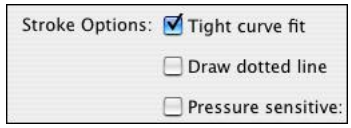
1. Double-click the freehand drawing tool.

   The Freehand Tool Setup dialog box appears.

2. Click the **Calligraphic pen** checkbox.



3. Make sure the **Pressure sensitive** option, the **Tight curve fit** and the **Draw dotted line** Stroke options are all turned off.



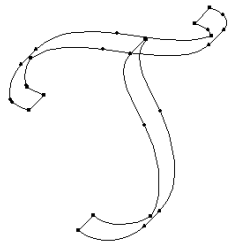
4. Enter a Pen width of 100 em units.

   For the purposes of this exercise, you should leave the Pen angle set at 45 degrees since that is one of the recommended angles for calligraphic glyphs.

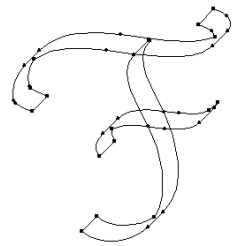
5. Click **OK**.

   The freehand tool icon will change to a calligraphic pen icon.

6. Press and hold down the mouse button while moving it around on your desk or mouse pad, or use a digitizing tablet to draw a calligraphic glyph.



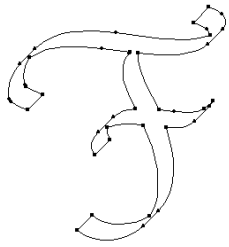
Use the calligraphic pen...



to draw as many strokes as you like.

7. Choose **Remove Overlap** from the **Element** menu.

   Fontographer removes the overlapping area, and your calligraphic glyph is complete.



Remove any overlapping areas and clean up paths.

8. Choose **Preview** from the **View** menu and turn off **Show Points** to view your glyph without points and filled.

Turn off points and then preview your new glyph.

Calligraphic tutelage from Judith Sutcliffe

Editor's note: When we asked Judith Sutcliffe for some tips on creating calligraphic characters, we never dreamed she'd give us so much information. Loving calligraphy the way we do, and since she's one of the best calligraphers we know, we included it here (in full)... we think you'll agree that it's worth keeping and reading.

alligraphy is not the same thing as type. Generally, type is carefully structured, straight-backed, and neatly drawn. Calligraphy is often looser, more graphically expressive and flowing, and it more closely reminds us of the instrument with which it was drawn. You can use Fontographer to simulate a flat-nibbed pen or a Chinese brush, any of the variety of instruments with which people have written with calligraphic panache over the centuries.

Start by taking a look at the past and present of western, eastern or Middle Eastern calligraphy. Your local library or bookstore's graphics section will have books with samples of the work of ancient and modern calligraphers.

Try working with a calligraphic pen or brush. You don't have to be a whiz at it. You just need to pay attention to the limitations of the medium. Try holding a wide, flat-nibbed pen at a 45-degree angle and making vertical, horizontal and angled lines as well as circles. You will quickly see how the glyphistic shapes of western calligraphy are achieved.

That knowledge of how the position of the pen affects the line of ink that flows from it is what you need to know to draw calligraphy with Fontographer. You need to internalize the mechanics of the pen in your head. So that when you draw an "O" you know that the pen will issue a wide curving line on the right-side downstroke, but will taper to near nothingness as you pull the stroke to a close at lower left. Because the pen has a precise width, the stroke will have a predictable variance in size.

Most basic strokes are made without turning the pen, but more advanced strokes do involve a twist of the wrist and pen, and if you carefully study the calligraphy manuals, you will find these little secrets. Also, most letters are made with two or more strokes joined together invisibly. An “O” is made from a downward left stroke and a downward right stroke, not one stroke all the way around. These conventional stroke combinations are shown in most calligraphy manuals and are easy to learn. Remember, you don’t have to be a calligrapher; you just have to learn to think like a calligrapher’s pen.

If you are interested in oriental calligraphy, dip a Chinese brush in ink and press the soft tip to paper, pushing the bristles about half way down and to one side and then gradually lifting it and tapering off to the other side. Note the shape of the ink stroke. The movement of the brush and the look of the marks it leaves are considerably more complex than the workings of the western pen. However, Chinese calligraphy involves a graphic language of a very small number of simple stroke shapes combined into more complex glyphs. There is a particular way to make dots, horizontal lines, vertical lines, lines angled left, lines angled right, corners and endings of lines. You can find these in books on Chinese calligraphy.

You can also draw with a pen or brush roughly, without paying much attention to any calligraphic tradition, but the instrument will still leave glyphistic marks. Those are what you want to remember.

Turning to Fontographer, you might wish to start by copying some calligraphic hand that you admire. There are three ways to do this. One is by scanning the original in and auto tracing it. For the instructions on that method, refer to “Autotracing” on page 107.

The second way is without a drawing tablet: Choose a couple of letters that are most glyphistic of the hand you are going to recreate. You might start with the lowercase “l” and “o” as they contain the basic straight and rounded strokes on which the rest of the alphabet will be based. One way to do this quickly is to use corner points for every point you place. That will rough out the letter for you. Then go back and change the points ruling what should be curves to curve points. Adjust, check the black image in Preview, and readjust. If you draw an “o” that you think works, copy it to the background (Template layer) of “c”, “d” and other rounded glyphs and construct them over the pattern. That will give you a consistency of form that a calligrapher works for years to achieve in eye and hand coordination. The same goes for your “l” and all the ascenders of the lowercase alphabet. You must, of course, keep your stroke widths very similar, as it has all been written with the same pen.

But – don’t be too perfect, or you’ll be making something more akin to a text typeface. Let every rounded form vary just a little bit from every other one; let the ascenders lean a little, but not so much that it is obvious. This is a subtle way to add life to your font.

Continue building glyphs until you have enough to write a word in the metrics window and to print out some word-like gibberish from the print sample window. Take the printed proof and look at it upside down. What's your first impression of the weight of the letters? Turn it right side up and look again. How do the letters look together? Do they look related in their stroke widths, sizes, and leanings? If not, try to pick out the offending glyphs and rework them to fit into the family a bit better. How's the spacing? Move the margins in the metrics window until you get a pleasant spacing, particularly in smaller sizes.

The third method is with a drawing tablet: You'll be doing the same thing as described above, but you'll have the aid of Fontographer's wonderful, automatic, electronic calligraphic pen. In your hand it looks like the cordless pen of your drawing tablet, but on screen it draws like a calligraphic pen or a Chinese brush. If you have any natural or trained calligraphic talent, you will find a drawing tablet extremely useful because you can whip out a calligraphic shape with one swoop of that pen. You will want to experiment a bit at the outset with the various nib widths and slant variations available. And you will want to try the calligraphy pen alone, the pressure pen alone, and the combination of the two. For imitating western calligraphy you will use the calligraphic pen with or without the pressure mode. Try it both ways and see which produces most easily the shapes you have in mind.

Now here's one difference that practicing pen calligraphers will need to curb at the outset. Fontographer glyphs are usually made in one continuous outline. So make an "O" in one fell swoop, not two separate ones. Pretend you're writing on really slick paper and your pen doesn't catch. Go all the way around. If you don't, you'll have to patch the two sections together and that's a drag. (It's easy, of course, using the Remove Overlaps command, but save yourself time by not creating multiple parts in the first place.)

Editor's comment: Although it might seem from this text that Fontographer comes with extra digitizing hardware, it doesn't. To use an actual digital pen, you will have to get one from a vendor.

Since this is spontaneous drawing, you might consider sketching each letter several times in succession across the glyph window and then picking the best one to keep.

This is only the beginning. Unless you are one of the world's best calligraphers, you are not going to whip out 26 perfect calligraphic letters on the first try. No problem. Do the best you can. Decide which ones are the right proportions, the best style, and then do minor alterations on the others to bring them in line. Using Fontographer commands, scale them up or down; rotate when necessary, and so forth. If a stroke is too narrow, grab points on one side and pull to widen it. You can't do this in ink but you sure can in Fontographer.

Sometimes when you sketch a letter, the Fontographer outline that appears will have more points than are necessary. You want the fewest points possible, so prune out the excess (using **Merge Points** or **Clean Up Paths**). Also check to be sure that you have the path direction correct (clockwise on outer outlines). If your glyph has overlapping parts, do a **Select All** and **Remove Overlap**. If that doesn't work, make a copy of the outline and put it in the Template layer, then in Outline rework the overlapping section, following the original calligraphic shape. If there's something that doesn't please you, just grab the points and adjust. It's not ink; you can tweak it until you get it right.

Assuming you've got a lowercase glyph that's looking good, you can get a little fancier for your uppercase glyphs. Maybe you will add a few flourishes. Remember that the margin and kerning adjustments leave you great leeway in how you place each lowercase letter in combination with each uppercase letter. Use automatic kerning to make them fit just the way you want.

➲ ***Tip:*** You can also add a thin space key to your font, with considerably less space than you have on your **Spacebar**. A thin space can be used to adjust the spacing between two letters that are leaning on each other just a teeny bit. It's very handy, especially with calligraphic letters with an excess of flourishing strokes. Assign the thin space to a handy key such as the vertical line key or the backward slash or to the nonbreaking space key.

Let's say you have now worked out a basic alphabet, and it is looking good when you do some proof printing of various letter combinations. But when you print two "l"s together, they look mechanical and wooden. Here's where the fun begins, and the complications start. Make yourself a nice calligraphic double "l", with one letter a little taller than the other. Then when you're setting type, you can do a search and replace command and drop a hand-tooled double "l" in for every two twin "l"s. Alternate glyphs look really neat and give the look of authenticity to calligraphic typesetting. You can create as many double letters as you want. You can create nice combinations of "Th". You can give your font 15 different "a" glyphs, if you feel in the mood. That's the creative part.

The complex part comes in with the decision on just where in the world of keyboard glyph positioning do you put a double "l". There are no standards and no rules. If you are the only person who is going to use the font, you can do anything you like, as long as you make yourself a map so you can find that double "l" six months from now.

But if you're going to sell the font, you will do some brow furrowing, because you will probably be supplying both Mac and Windows versions of your font, and glyphs maps differ considerably between the two platforms and also within the two platforms. The safe way out is to not put anything in the upper ASCII positions (numbers 128 to 255) except standard position international accent glyphs. Put extras into a separate font on the uppercase/lowercase keys, even though it is much less convenient than having all the alternates in one font.

Or, if you're creating fonts in Macintosh original format, put the alternates wherever it's most convenient for Macintosh users. Then make a separate font and put the upper ASCII glyphs into it, on uppercase/lowercase keys and supply both fonts for Windows users.

Editor's comment: Since this was originally written the advent of the OpenType font format has solved most of these problems.

Another aspect of calligraphic font making arises if you decide to make a script face – one in which all letters in a word appear joined, as if written in one continuous hand. It is quite possible to create a font of script letters that will appear when printed out to be written as a continuous line, but it is not easy. The general principle is that you must design a standard shape for both incoming and outgoing strokes and use them as part of every glyph. The margins of glyphs need to be set so that the outgoing stroke of one letter overlaps the incoming stroke of the following glyph. This takes some careful experimentation with and slight manipulation of each glyph's incoming and outgoing strokes. But once you get it to work right, it will look quite natural, especially if here and there you leave an incoming stroke off. You will want to avoid kerning as much as possible and should design an alphabet set that needs very little.

For Chinese or Japanese calligraphy or to give an oriental flavor to a western alphabet, try a cordless pen and drawing tablet with Fontographer set to pressure-sensitive pen only. It's very quick and sensitive and will take a little getting used to, but with some practice you will be able to construct Chinese glyphs with only minor need for point adjustments. If you are quite serious about working on a Chinese font, you will probably want to work out a library of the basic strokes and copy from that storehouse when building new glyphs. The **Remove Overlap** command will be very handy. You can always adjust each new glyph for balance and style.

Judith Sutcliffe: The Electric Typographer, January 18, 1993

- ***Tips:*** For Gothic-type letters, use round caps and joins for your pen settings.
- To get square, poster-type letters, use square caps and joins.
- For Roman and Italic alphabets, use square caps and joins and use the calligraphic pen option.
- Most sources recommend that you hold your pen at a 45-degree angle when you create calligraphic glyphs. The Speedball textbook (20th edition) recommends that the height of your lowercase letters (and ascenders and descenders) be five pen widths high. They also recommend that you fit your strokes together so that overlaps won't show in your finished letters. The **Remove Overlap** command will take care of that for you.

  Editor's note: The drop cap at the beginning of this article is from Judith's "Uncle Fats" collection.

Creating variable weight glyphs

Variable weight glyphs give the effect of being drawn with a brush. That is, they can have wide and thin areas. Fontographer's freehand drawing tool can be used directly with the mouse or with a pressure-sensitive pen and digitizing tablet to give you these effects.

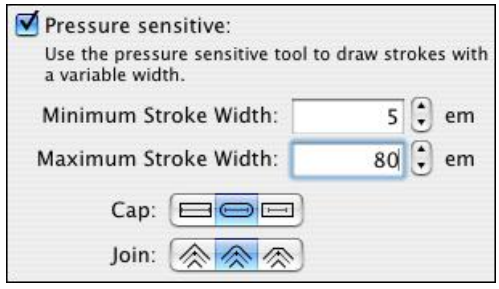
Note: If you use a pressure-sensitive pen and tablet, make the appropriate tablet settings before you draw your glyphs.

To use the pressure-sensitive pen:

1. Double-click the freehand drawing tool.

   The Freehand Tool Setup dialog box appears.

2. Click the **Pressure sensitive** checkbox.



3. Make sure the **Calligraphic pen** and other options are turned off.
4. Enter a minimum and maximum stroke width (like the ones we've set here).

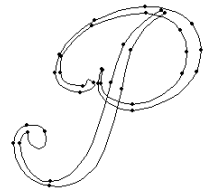
   Try experimenting with different values, and line caps and joins to get different line effects.

5. Click **OK**.

   The freehand drawing tool icon changes to reflect the Pressure sensitive setting.

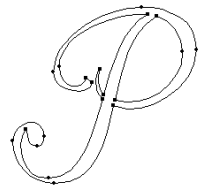
   If you are using a pressure-sensitive pen, go to step 6. If you are using a mouse, skip to step 8.

6. By applying varying amounts of pressure as you draw, you can create Script glyphs with thick and thin areas like our example:



Draw a glyph.

7. Choose **Remove Overlap** and then **Clean Up Paths** from the **Element** menu. Fontographer removes the overlapping area, and your glyph is complete.



*Remove overlapping areas and choose **Clean Up Paths**.*

If you are using a mouse instead of a pressure-sensitive pen, follow these instructions:

8. Press and hold down the mouse button while moving it around on your desk or mouse pad to draw a glyph.

9. Press the **RIGHT** and **LEFT ARROW** keys while you move the mouse button to see the different effects you can create.

The **LEFT ARROW** key (or the number "1") reduces the stroke width (down to the minimum stroke width you specified in the Freehand Tool Setup). The **RIGHT ARROW** key (or the number "2") increases the stroke size (up to the maximum stroke width you specified in the Freehand Tool Setup dialog box).

Turn off points and preview your glyph.

Blend fonts to create new fonts

Experimentation is at the heart of creating fonts. And Fontographer's **Blend Fonts** option is the ultimate vehicle for creativity.

Blend Fonts is very much like FreeHand's blending of one object to another. It helps if your fonts are similar in glyphistics. Imagine that your fonts are at opposite ends of a one-dimensional line segment and that you are creating a new font that is some percentage of the way between them (or beyond them). You can edit those intermediate versions as you like, and quickly produce a family of weights, say, from just two master designs like Extra Light and Extra Bold.

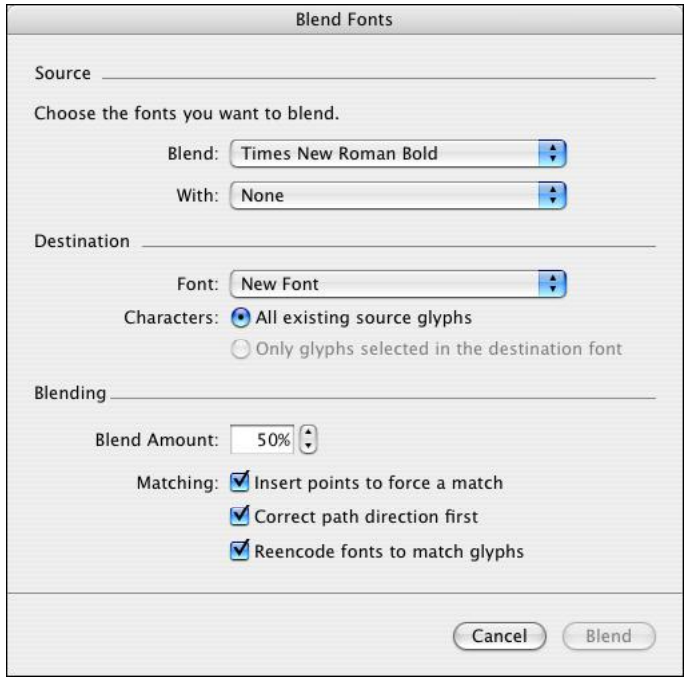
To blend fonts:

1. Open a font.

   Font blending needs the least amount of attention when you blend between similar fonts. In this exercise, we use two from the same family: Times New Roman and Times New Roman Bold.

2. Choose **Blend Fonts** from the **Element** menu.

   The Font Blend panel appears.



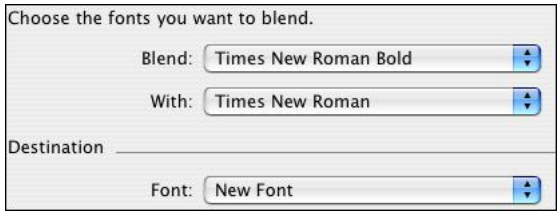
The font you opened in step 1 automatically appears as the Blend source font. Source fonts are the fonts you will base your third font upon.

If you have any other fonts open when you choose **Blend Fonts**, they will automatically appear in the **Blend** and **With** source fonts pop-ups.

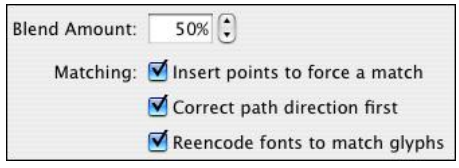
3. Choose a source font for the **With** position by clicking its pop-up and selecting Other.

   A standard dialog box appears that lets you choose fonts directly from your drive, a directory, or a folder.

Note: You can also choose New font to open an Untitled font for your first font and then choose both your source fonts. If you use this option, you'll also have the option of selecting the Untitled font as a destination font.



4. Enter a **Blend amount** of 50%.



5. There are three other settings below the **Blend amount**; leave them turned on for this exercise.

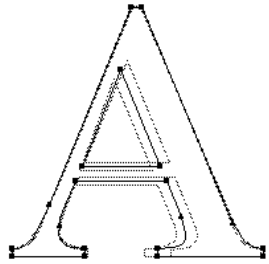
Note: If you leave the option set at **Correct Path Directions First**, Fontographer sets the rightmost point as the origin point. If you want your origin points to remain in the same place (in your new Blended font), you should leave this option deselected.

6. Click the **OK** button.

   Fontographer automatically creates a completely new Untitled font based on the values you set in the **Blend amount** text box.

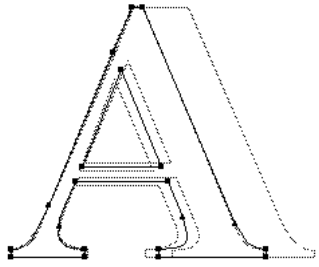
The two source fonts are now dynamically linked to the new font. If you open any outline window in the new font, you will see both source outlines in the Template layer (see illustration below).

Any changes you make to the source font's outline will show in the Template layer, but will not change the new blended font's outline unless you choose **Blend Fonts** again.



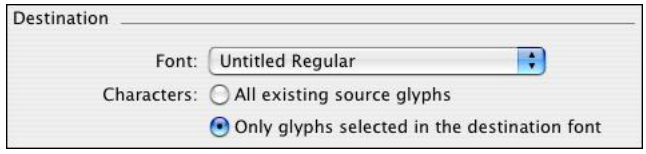
To change selected destination glyphs:

1. Make some changes to a glyph in one of your source fonts.



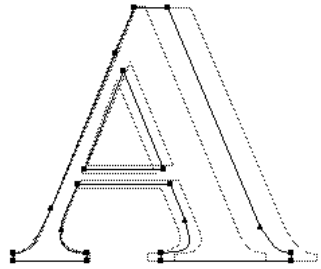
The change is reflected in the Untitled font glyph's Template layer.

2. Click the **Only glyphs selected in the destination font** radio button.



3. Click **OK**.

   Fontographer will automatically reblend the glyphs.



4. Turn off the Template layer and **Show Points** and then choose **Preview** from the **View** menu to view your new glyph.

   You can continue making minor tweaks to the blended glyph (or font), or simply save it as a new variation and go on to another variation with a different interpolation percentage.

   Closing the new font unlinks the source fonts.

Note: Every font you open needs 200-300K of RAM, so you will need to be generous with memory if you are going to do lots of interpolation (font blending).

When things go wrong...

Doing the above example using Times New Roman and Times New Roman -Bold, you most likely encountered an error message after the blend attempt that read, "Could not complete your request because some glyphs didn't match during blending. Consult your log file for details about the unblended glyphs". When you open the text file *Fontographer.log*, you will see the list of error(s). Here are some examples of possible errors:

Char. #1	Char. #2	Path #	Error
290	290		different # of paths
303	303	1	different types of paths
741			glyph missing in font 2

The message “different # of paths” refers to the glyph decimal location in the fonts. So glyph 290 in Font 1 and glyph 290 in font 2 don’t match because they have a different number of paths. This means that, for example, your lowercase “g” in font 1 has three paths, but in font 2 it only has two. Or the Ccedillas (ç) don’t match because in one font you used remove overlap, causing the glyph to have one path, but you forgot to do it in font 2, so the ç still has two paths.

The message “different types of paths” occurs when a glyph such as ç is a referenced glyph in one font, but is an actual outline glyph in the other. You can correct this by using the **Decompose Component** command on the referenced glyph. The path # is the number of the path that the error message applies to, and it doesn’t check any further than the first mismatch.

The last message, “glyph missing in font 2” means that although one font has this glyph, the other does not, and Fontographer can’t blend what isn’t there. If the two fonts do not have the same number of glyph slots, your blended font will have the same number of glyphs as the font with the fewest glyph slots.

After consulting your log file, you can go back and correct the errors and then blend again – blending only the selected glyphs if you wish – to finish your font blend.

Just remember, the key to success using **Blend Fonts** is that your glyphs in both fonts must have the same number of paths and the same types of paths, and each font must have the same number of glyph slots.

For information about the blending process, refer to “Font blending – the technical details” in Chapter 13, “Expert Advice”.

Jonathan Hoefler says:

In 1990, *Sports Illustrated* commissioned my studio to design six typefaces – a set of six sans serifs in differing widths (Champion Gothic Heavyweight, Middleweight, Welter-weight, Lightweight, Featherweight, and Bantamweight). These fonts were drawn in Adobe Illustrator and manufactured in Fontographer.

In 1992, the rights for these typefaces bounced back to me, and I began selling them retail. Several art directors called to say how much they liked the six fonts, but expressed hope that I would someday extend the range to include a super-condensed typeface. Given that no one was willing to underwrite the design of a seventh font (whereas *Sports Illustrated* had paid for the considerable development of the first six faces), it seemed unlikely that I would ever have the opportunity to revisit the fonts.

Until 1993, when Fontographer added the interpolation feature (now called Font Blend).

Using interpolation, I could quickly create intermediate fonts between the Bantamweight and Featherweight types; more importantly, using extrapolation (using a blend amount greater than 100 percent), I was able to carry the changes from the wider Featherweight font, through the narrower Bantamweight font, into a new typeface, a super condensed sans serif I dubbed Champion Gothic Flyweight.

By caricaturing the ways in which the Bantamweight and Featherweight fonts differed, the new extrapolated font highlighted some of the design flaws in the original two faces, which I was able to easily correct. But most importantly, this technique enabled me to create a new font in just minutes, rather than weeks.

Altering Outlines

Self-expression knows few boundaries with Fontographer. The only limits are your own imagination and skill – the raw materials, the drawing tools and layers, offer you all you need to start creating typefaces. The outline window is the place to test your creative wings. Add serifs to a sans-serif font, create geometric designs, or import your favorite illustrations from other PostScript drawing programs. If you want to learn more about the basics of font production, make sure you read the sections entitled “Typography” and “Type designers” in Appendix B, “Bibliography of Typography and Allied Subjects”.

4

The tools in Fontographer's outline window let you alter graphic images or font characters in different ways. You can move points or paths, duplicate points, merge points, insert points, remove them, or drag them. You may want to alter your outlines based on other images placed in the Template layer of the window, either using copied images from other characters or scanned images for tracing. Refer to Chapter 3, "Creating New Fonts", if you'd like more information about tracing a scanned image.

Create your own logo like the wordmark Paul Sych created for Coca-Cola's transit posters and television advertisements.

Altering a logo

This quick-step exercise shows you how to import an image from a drawing program into a glyph slot in Fontographer so you can access it with a keystroke. You will make some changes to the logo, and then change its width by scaling the image. For practice, import any Encapsulated PostScript (EPS) image created in a drawing program like Adobe FreeHand or Adobe Illustrator® or use the art file, *Torch logo.eps*, provided in your Fontographer folder.

1. Open a font in Fontographer, and from the Font window select and open a glyph's Outline window.

   You can delete the glyph outline from its outline window if you need to, by choosing **Select All** from the **Edit** menu and then pressing the **BACKSPACE** key.

2. Select **Import** from the **File** menu and then select **EPS** from the submenu.

3. Select the PostScript file named "Torch logo.eps" from the Fontographer/Sample Files folder, or choose your own FreeHand or Adobe Illustrator EPS file.

   Fontographer automatically imports the image into the glyph's outline window. The image will be scaled to fit between the glyph's baseline and ascent lines.

   If you choose **Preview** from the **View** menu, you can see that the fills are automatically transferred to the typeface.

Note: The **Import EPS** option accepts only outline path data (for example, points). It ignores bitmap PICT data, TIFFs, text, and special effects (for example, graduated, radial, and tiled fills).

Tip: If you have trouble getting a single glyph logo to display on screen and/or print, you may need to break the glyph into parts that can be placed into multiple keystrokes.

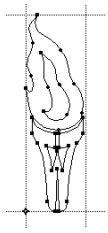
Pasting EPS outlines from the Clipboard

You can also paste FreeHand and Illustrator images directly into the glyph edit window without saving the file as an EPS file.

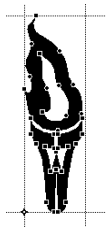
- While you are in FreeHand or Illustrator, select the graphics you wish to copy, then press the **Option** or **Control** key and select **Copy** from the **Edit** menu.
- In Fontographer, select a glyph slot and paste in the image.

The path data will be pasted into your glyph so it fits between the ascent and the descent. Holding down the **Option** key while pasting the EPS file or bitmap image will retain the image's size at the moment you copied it.

Pressing **Shift-Option-Paste** key will fit the EPS file or bitmap image between the baseline and the ascent.



Import your EPS images directly into the outline window.



You can view the file in the Preview mode with points turned on...

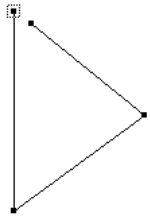
or with points turned off.

Paths and points

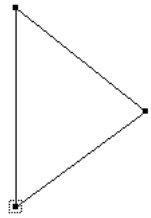
Paths are made of straight or curved line segments that are either connected or unconnected. Glyphs like the lowercase “i” and “j” are composed of two separate non-overlapping paths: the dots and the stems. When you join the end points of a path, you’ve closed that path.

Note: All typographically correct paths need to be closed.

If you already know how paths work, you can skip this section and go to “Types of points” on page 166.



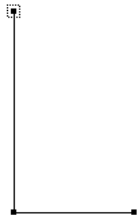
Open path



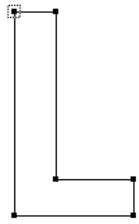
Closed path

Closed paths

Closed paths can be filled; open paths can’t. PostScript always closes paths and fills images unless you specifically command Fontographer to do otherwise. To turn glyph fill off, choose **Selection Info** from the **Element** menu and deselect **Fill** in the glyph Information dialog box. If you want to have outline and path connections without a fill (this is a glyph made up of stroked lines), click the **Stroke** checkbox. Basically, you should know that 99% of the time you’ll be using closed paths. For more about stroked glyphs, refer to Chapter 3, “Creating New Fonts”.



Open path



Closed path

Path direction and fills

A glyph with an open path remains unfilled by PostScript, but glyphs with closed paths are filled. PostScript automatically closes paths unless you specify otherwise. Open paths are lines in which the final point does not reconnect to the initial point in the path and are basically worthless unless you're creating a Type 3 font. A closed path includes an end point that reconnects with the first point in the path.

➲ ***Tip:*** To keep a path open in PostScript printing, choose **Selection Info** from the **Element** menu to access the glyph Information dialog box. You will find options to turn both the fill and stroke on or off. You should turn fill off – thereby eliminating both the fill and the path. Click the stroke item checkbox to restore the path only.

Note: This option only works correctly with Type 3 fonts.

In the glyph Information dialog box (which displays when you choose **Selection Info** from the **Element** menu), you can select one of two types of fills – Normal or Even/odd. The standard PostScript filling technique is called a winding number fill, which is the Normal fill. The Even/odd fill operates differently (See "Even/odd fill" on page 164 of this chapter).

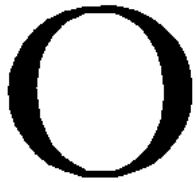
When you choose **Selection Info**, you will only get the glyph Information dialog box if no points are selected or more than one is selected. When one point is selected, the Point Information dialog box will appear.

Normal fill

You should use the Normal fill type for PostScript Type 1 and TrueType fonts.

Note: The Normal fill relies on the outside path being described as clockwise and the next inside path being described as counterclockwise, hence the term winding number.

This results in normally filled glyphs, like in the “o” below.

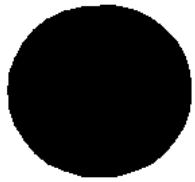


The outside path of this glyph is going in the clockwise direction; the inside path is counterclockwise so the inside of the glyph appears transparent.

The current path direction, clockwise or counterclockwise, is shown in the **Element** menu. Click a control point and choose **Clockwise** or **Counterclockwise** from the **Path** menu to change path direction. You can also change direction via the path direction indicator in the outline window. Path direction is defined only on closed paths.

Path direction indicator

Technically, it doesn’t make any difference whether the outside path is clockwise or counterclockwise, but for the sake of consistency between Fontographer’s fonts and the proper operation of automatic hints, we recommend that the outer paths should be clockwise and the inner paths counterclockwise.



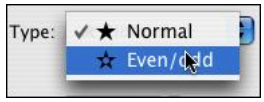
If the inside path of the glyph is changed to a clockwise direction (the same direction of the outside path), everything in the glyph becomes filled.

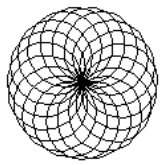
Even/odd fill

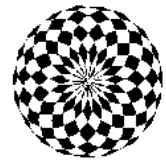
The other filling technique is called an even/odd fill. You should use the even/odd fill type for PostScript Type 3 fonts or for modifying the behavior of the **Remove Overlap** command. An outside path is not filled until it crosses a path. The area that it crosses is filled; the next area the path crosses becomes unfilled, and so on.

In the previous illustration of the "o", even/odd filling would give the desired result even if both paths were clockwise.

➲ ***Tip:*** You can change a normal fill to an even/odd fill by choosing **Selection Info** from the **Element** menu and then choosing **Even/odd** from the **Type** pop-up.







Original illustration *With a normal fill* *With an even/odd fill*

Correct path direction

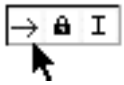
Path directions can be automatically corrected – you can tell Fontographer to examine all the path directions and reorder them if necessary. Path directions must be correctly set for proper filling of glyphs. Choosing **Correct Path Direction** from the **Element** menu tells Fontographer to examine all the selected glyphs and, if necessary, automatically adjust their path directions.

Outer paths will be set to clockwise, inner paths to counterclockwise. When required Fontographer automatically, but temporarily, reverses these; for instance, when generating Type 1 fonts.

✎ ***Note:*** You can stop this operation at any time by typing **Command-period**.

Reverse path direction

To reverse path direction, select the path to be reversed and click the path direction indicator with the mouse (in the lower left corner of the outline window), or choose the opposite direction from the **Element** menu – either clockwise or counterclockwise. Reversing path direction will alter the fill in your glyph.



Click the path direction indicator.

Change this glyph...

to this glyph.

Types of points

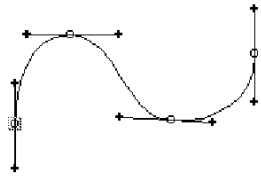
Fontographer uses three different types of points: corner points, curve points, and tangent points. Don't be confused by the different point types – like control handles, they're nothing more than different ways of working with the same basic element. The shape of glyphs is determined by the kinds of points used to construct them. You control the shape of the line segments in a path by either manipulating two control handles attached to each point or directly manipulating the path itself.

Certain principles operate in using the corner, tangent, and curve points. Once you understand these principles, creating and altering glyphs is easy.

Curve points

When you place a curve point or convert another type of point to a curve point, Fontographer automatically extends two control handles from the point to create a smooth curve between the preceding and following points on the path. The shape of the lines that extend from both sides of a curve point will be an arc.

The curve point tool is used to create curve points and join curves to other curves smoothly. The curve point tool can be selected by clicking its indicator. When the lock icon is in the locked position, the numeric keypad equivalent for choosing the curve point tool is 8.

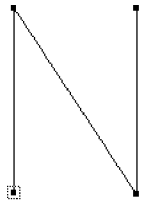


- ***Tips:*** Click the existing path with the curve point tool to add curve points.
- When you select a curve point, Fontographer displays the point as a hollow circle. You can change an existing point to a curve point by selecting **Curve Point** from the **Points** menu.
- You can use either the **Option** to drag a control handle out of a control point.

  Basically, if a series of curve points is connected, the points will display an arc that takes the most graceful and efficient route in maintaining the line through the sequence of points. In mathematical terms, the slope of the curve is continuous through the point.

Corner points

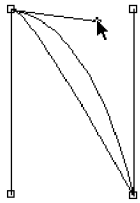
The corner point tool is used to join straight segments to curved segments or to other straight segments at an angle, or to connect two curve segments at a cusp. The corner point tool can be selected by clicking its indicator. When the lock icon is in the locked position, the numeric equivalent for choosing the corner point tool is 9.



➲ ***Tip:*** Click the existing path with the corner point tool to add corner points.

When you select a corner point, Fontographer displays the point as a hollow square. You can change an existing point to a corner point by selecting **Corner Point** from the **Points** menu.

➲ ***Tip:*** You can drag the control handles out of the corner point to make the corner point behave similarly to a curve point, by holding down the **Option** key while you drag out of the point.



Tangent points

The tangent point tool is used to connect straight lines to curves with a smooth tangent join. Tangent points may also be used to connect straight line segments together. You can select the tangent point tool by clicking its indicator. When the lock icon is in the locked position, the numeric keypad equivalent for choosing the curve point tool is 0.

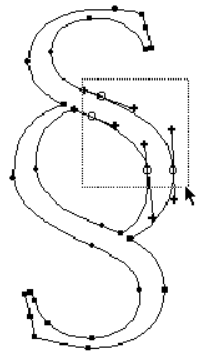
When you select a tangent point, Fontographer displays the point as a hollow triangle. You can change an existing point to a tangent point by selecting **Tangent Point** from the **Points** menu.

➲ ***Tip:*** Click the existing path with the tangent point tool to add tangent points.

Selecting multiple points

In Fontographer, you can select a group of points or paths in a variety of ways:

- Position the pointer tool outside the area of points you want in the selection, press down the mouse button, and drag to the opposite corner of the group of points or paths you want to include.



  A dotted-line box surrounding your selection appears as you drag. Release the mouse button when you're through selecting points.

- You can select an entire path by double-clicking any point in that path or on the path itself.
- To select any combination of points, just press the **Shift** key and select each point individually with the pointer tool.
- Select all the points by choosing **Select All** from the **Edit** menu.

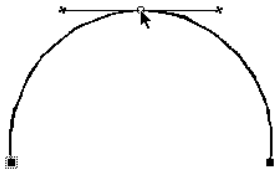
Changing a point type

You can convert any point into another type of point at any time. To change a point's type, select the point or group of points, and choose the new point type from the **Points** menu, or press the **Command** or **Control** key plus the appropriate numeric equivalent. The need to change point type arises in cases where the shaping of a line requires different attributes than those offered by the currently selected point type.

To change point types:

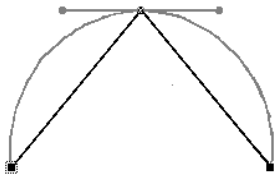
There are two ways to change point types:

1. Select the point with the pointer tool, by clicking it once.



2. Then choose the point type you'd like to change it to from the **Points** menu.

   The point changes to the new point type. The checkmark beside the point type in the **Points** menu indicates the selected point type.



Note: The **Undo** command will switch the point back to its previous state.

Inserting points

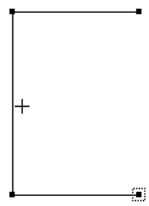
In general, with font glyphs and other graphic images, the fewer points you include the more graceful the image. On the other hand, there are situations when you need to add points to get more control. One professional typographer – Judy Sutcliffe – recommends roughing out glyphs by drawing them in an outline form with corner points, and then returning and substituting other kinds of points where needed. With whatever process you use for building images or glyphs, there will be instances where you want to add points to a path, so you can more easily control the path's shape.

To insert a point:

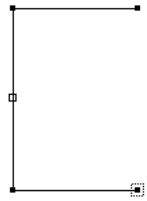
1. Select the appropriate control point tool (or the pen tool).
2. Position the pointer on the figure where the new point should be inserted (on top of a line or a curve).

   If the pointer is positioned some distance away from the line or curve, a new path is started; or if the current path is active (open), it is continued.
3. Click the figure.

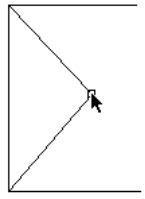
   The point is inserted into the figure at that position.



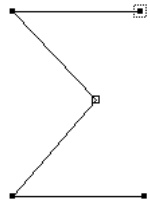
Click a path...



to place a new point.



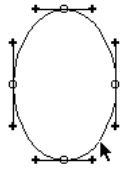
Drag the point...



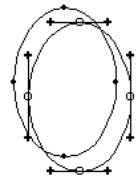
to a new location.

Duplicating points

By choosing **Duplicate** from the **Edit** menu, you can copy selected points and paths. The duplicated points will offset slightly (down and to the right) from the original outline and are selected.

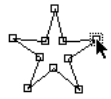


Select a path and choose **Duplicate** *from the* **Edit** *menu.*



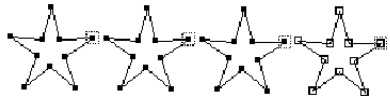
A duplicate path appears.

➲ ***Tip:*** If you use the Duplicate command, and move the first duplicated path immediately after the new image appears, all future duplicates will be spaced the same amount of distance from each other.



Select a path.

Duplicate the path and move it to a new position.

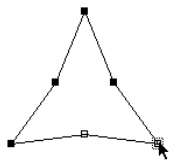


Each subsequent duplicated image is spaced the same distance.

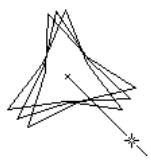
Power duplicating

By combining the **Clone** and **Duplicate** items in the **Edit** menu you can create some pretty interesting images. Follow our example below to create a spiral glyph, and then try some combinations of your own.

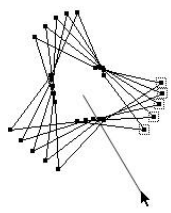
1. Create any shape.



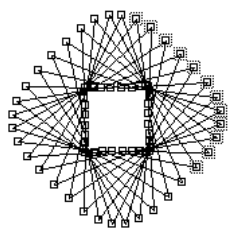
2. Choose **Clone** from the **Edit** menu.
3. Use the rotation tool on the cloned image.



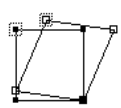
4. Select **Duplicate** from the **Edit** menu.

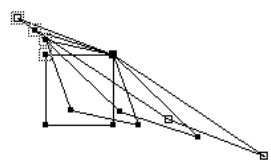


5. Repeat step 4 to make as many duplicates as you like.

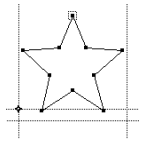
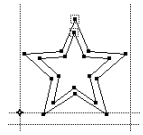
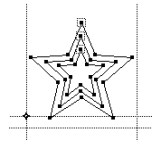


Try this with some of the other transformation tools to see what effects you can create.





Create a square. *Clone and then skew it.* *Duplicate it as many times as you like.*

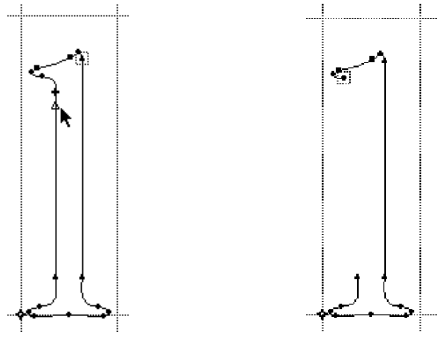




Create a star. *Clone and reduce it.* *Duplicate it as many times as you like.*

Removing points

Anytime you want, you can delete a point in a path by selecting the point and pressing the **DELETE** key. Fontographer removes the point and opens up the path. Sometimes, you will want to split a path to create two open paths. However, most of the time you'll probably want to remove points without breaking the path – this is called merging points. We tell you how to do both in this section.

To remove points within a path:

- Select the points and choose **Clear** from the **Edit** menu.
- 
- Select the points and press the **DELETE** key.

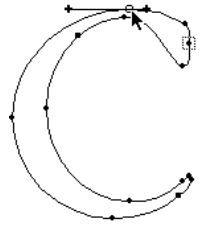
This removes active points, opening the path up if it was closed, or splitting it if it was open and the selected point was not an end point.

Splitting a path

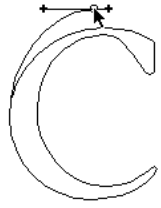
Sometimes, you'll need to split a path at a point to create either two open paths or, if you're working with a closed path, to create an open path. Simply click a point and then select **Split Points** from the **Points** menu. Fontographer splits the selected point into two separate points. Both of the new points are selected after Fontographer splits the point. The point connected to the start of the original path (according to the path's direction) will be under the point connected to the end of the original path.

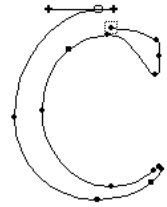
To split a path:

1. Click the point you want to split into two points.



2. Choose **Split Points** from the **Points** menu. Fontographer splits the selected point into two points. The second point is hidden under the top point.
3. Press the **Tab** key to deselect any selected points.
4. Choose the selection pointer from the tool palette.
5. Click the split point and move it.

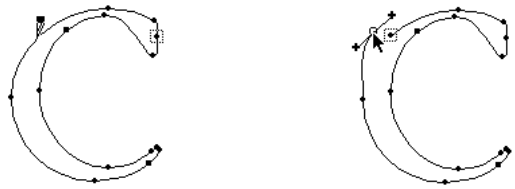




➲ ***Tip:*** You can split more than one point at a time by holding down the **Shift** key while you click the points and then choosing **Split Points** from the **Points** menu.

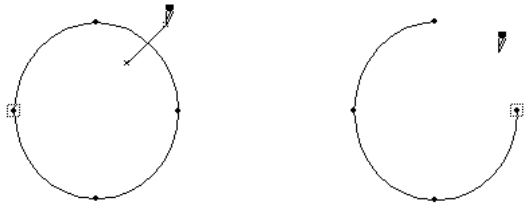
Splitting line segments

If you want to split a path by splitting a line segment (rather than by splitting the path at a point), select the knife tool and drag across the path where you want to split it. Fontographer splits the path, creating two new points where the knife tool crossed (or intersected) the path.



Drag the knife tool across the path to split the path.

➲ ***Tip:*** You can drag across lots of paths at once and split them all.



Hold down the ***Option*** *key while you drag the knife tool across the path to remove the path or line segment.*

Joining points

When you want to connect a path to another by joining points, as in instances where you are joining serifs and stems, we recommend using the Frankenstein approach – pasting your image together by joining points. This can be especially useful if you want to copy the stems from a serif typeface and add them to your sans-serif typeface.

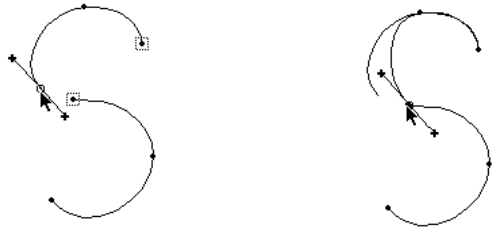
Note: Never mistake this method for **Merge Points**, found in the **Points** menu. Although related, the **Merge Points** command essentially removes any selected points on a path without breaking the path.

To join points from open paths:

- Drag an end point of one of the paths over an end point of the other path.

Fontographer joins the paths.

In the example on the right, the curve point changed to a corner point since Fontographer always tries to maintain the original shape of the path.



Drag an end point from one path over the end point of another path...

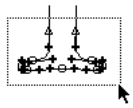
to join the paths.

Adding serifs

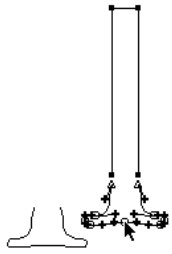
For those instances where you might want to combine the serifs from a serif typeface with your sans-serif typeface:

1. Drag around an area to select the points you want to copy (in your serif typeface).
2. Choose **Copy** from the **Edit** menu.
3. Choose **Paste** from the **Edit** menu to paste the points into the new glyph (in your sans-serif typeface).
4. Drag the selected path, until the points you want to merge cover their coordinate points on the other path or paths.
5. Release the mouse button.

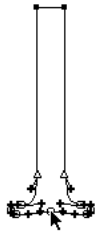
   Fontographer automatically joins the points and unites the paths.



Drag around an area to select the points and then choose **Copy** from the **Edit** menu.



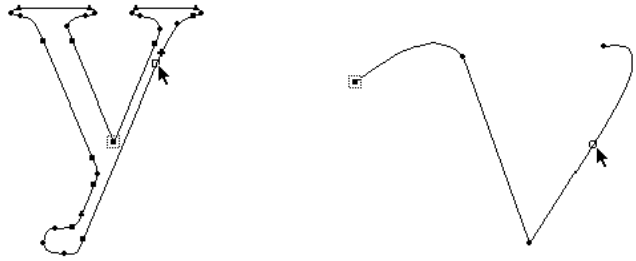
Drag the points over the points on the other path.



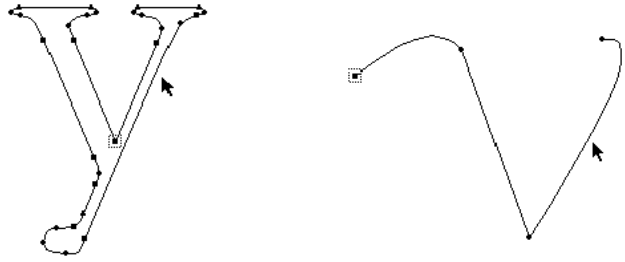
Fontographer automatically joins the points and paths.

Merging points

There will be many times when you'll want to remove excess points from a glyph. If you use the methods described in the previous sections, you'll be left with a broken line segment or an open path. Merging points simply removes the point from the path and connects the point on either side of the deleted point with a single line segment. This is handy for removing unnecessary points.



Select the point to remove, and then choose ***Merge Points*** *from the* ***Points*** *menu.*



Fontographer removes the point.

Merging points removes the active point, but joins the points on either side, so closed paths remain closed and open paths are not split into two pieces.

See "Clean Up Paths" in Chapter 3, "Creating New Fonts", for more automatic ways of removing unneeded points.

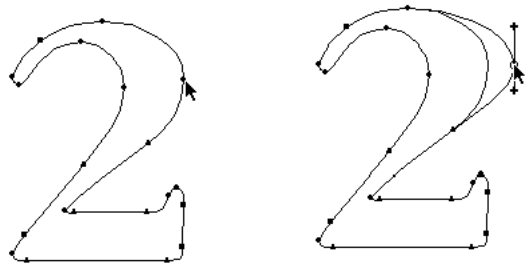
Moving a point

Moving a point or a group of points can be done in a couple of ways. Use the ARROW keys to move the point a certain number of em units in any of four directions, or drag the point with the selection pointer.

To move a point with the mouse:

1. Click the selection tool, position it on the point to be moved, and click.
2. Without releasing the mouse button, drag the point to the new location.

   When you use the selection pointer to move a point that is connected to other points, you can see the line segments change as you drag. If the point is connected to other points, the connecting lines or curves are shown during the move, so you can see the effect of the move as it happens.



Select a point *and drag.*

While you're moving the point(s), the position indicators at the top of the window are continuously updated to show the actual position and distance of the pointer from the basepoint.

➲ ***Tip:*** To hide points while dragging (to see a clearer image of the outline) go to **Preferences** in the **Fontographer** menu and deselect **Show points while dragging paths** in the Point display dialog box. This also improves speed performance during dragging because only selected points are drawn.

Demagnified move

Most drawing programs require zooming to a more detailed view to draw intricate designs. Sometimes, however, this causes a loss of overall perspective. Fontographer has a unique capability, called a demagnified move, which allows for very precise point placement. A demagnified move constrains cursor movement to one-tenth the distance specified in the Preferences, at full resolution.

To use demagnification:

- Hold down the **Control** key while dragging a point.

If your **Preferences** under the **Fontographer** menu are set to the default of 10 em units, Fontographer will move the selection in one-em-unit increments.

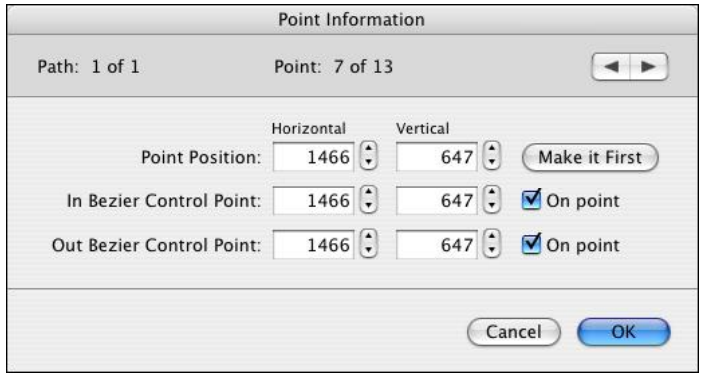
Keyboard commands to move points

- Click a control point to select it; and use the **Left**, **Right**, **Up**, or **Down arrow** keys to move the point by one em unit.
- Click the control point to select it, and hold down the **Shift** key while using the **arrow** keys to move the point by ten times the specified cursor distance.

To move a point by more than one em unit, which is the default preference setting for cursor editing behavior, select **Preferences** from the **Fontographer** menu. Type in the preferred distance in the cursor textbox.

Accurate point placement

For accurate BCP and point placement, select a control point and choose **Selection Info** from the **Element** menu. Type in whatever coordinates you like. You can move the Point Information dialog box anywhere on your screen to allow an unobstructed view of the points you are setting.



Any changes you make in the Point Information dialog box are immediately reflected in the glyph outline window. In addition to using the **Tab** key to move through the fields, additional key commands are available for the Next and Previous Points operations. These commands are shown on the **Next** and **Previous** buttons.

➲ ***Tip:*** The neatest way to get accurate point placement is to turn on **Show coordinates** for selected points in the Point display section of the Preferences dialog box, step through each point by choosing **Next Point** and **Previous Point** from the **View** menu, and use the **arrow** keys to nudge points.

To select the next point in the path:

- Click the ▶ button.

To select the previous point in the path:

- Click the ◀ button.

To make a point the first point in a path:

1. Select a point.
2. Click the **Make First** button in the Point Information dialog box.

You can undo resetting of the first point by pressing the **Cancel** button or choosing **Undo** when you're back in the outline window.

Pressing **Cancel** will undo everything you did in the Point Information dialog box and restore the glyph to its original state.

Choosing **Undo** from the **Edit** menu will also undo everything you did in the Point Information dialog box, because selecting **Undo** is considered one action.

Note: The **Make First** option is designed for those rare instances when you need manual point ordering, for example, when creating fonts for font blending (interpolation). For more detailed information about reordering points and font blending refer to "Blend fonts to create new fonts" in Chapter 3, "Creating New Fonts" and "Font blending – the technical details" in Chapter 13, "Expert Advice".

To retract BCPs into their point:

1. Click the **On Point** checkbox for the incoming and/or outgoing BCP in the Point Information dialog box.

   The BCPs will move (or disappear from view) into their point.

2. Click the checkbox again to deselect it; the BCPs will reappear and return to their previous coordinates.

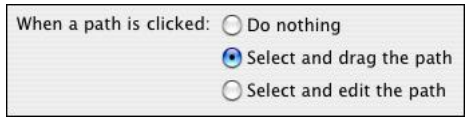
   You can also retract BCPs in the outline window with a menu command. See "Retracting BCPs" on page 191 for more information.

Point and path preferences

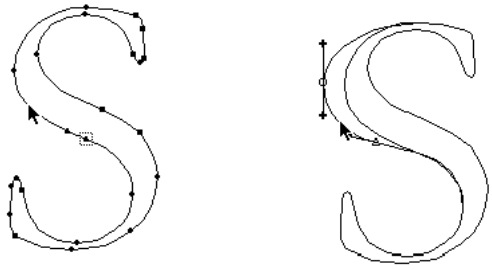
Path display

You can control the way paths appear by setting options in the Preferences dialog box. Choose **Preferences** from the **Fontographer** menu. Then select Editing in the header of the dialog box.

Next choose the radio button relating to the path behavior that best describes how you want paths to act when you select them.



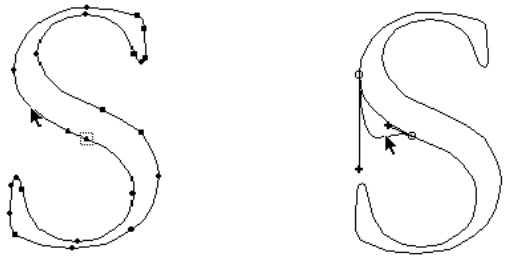
- You can drag paths as a whole. To choose this option, click the **Select and drag the path** radio button.



In this case, when you click a path with the selection pointer, you select the two adjacent points and any segments extending between and on either side of them.

- •Or choose the **Select and edit the path** radio button to move a segment lying between two points (and leave those points in place) when you click the segment and drag with the mouse.

➲ ***Tip:*** Using **Option** while dragging on a path alternates between the two options.



Instead of having to manipulate control handles to regulate the shape of the segment, you can drag the line in any direction.

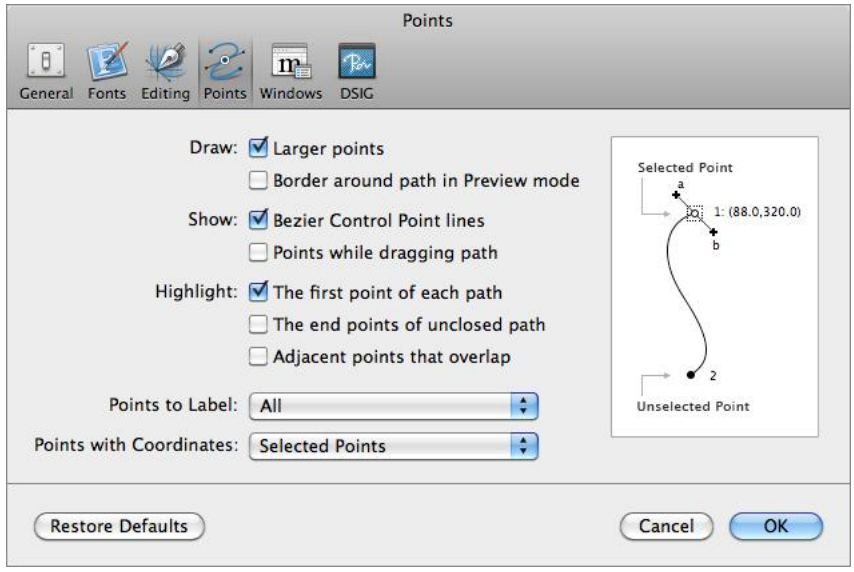
The outline of the original position of the line stays on the screen as you drag, enabling you to go back to your starting place. Of course, you can also undo these actions by choosing **Undo** from the **Edit** menu.

Choosing the **Do nothing** option results in no action taking place when you click a path with the mouse. In order to move the path, you will have to move the control point or its BCP handles.

Point display

To select the type of point display that you want to see when you edit your glyphs or graphic images, choose from the possibilities found in the Preferences dialog box. First choose **Preferences** from the **Fontographer** menu. Then select Point Display from the pop-up to display the options for viewing points.

Click each of the options to see the differences you'll get when you edit. For more information on each of the display options, refer to "Point display" in Chapter 13, "Expert Advice".

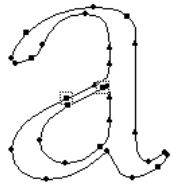


Show and hide control points

Depending on the type of work you are doing, you may or may not want to view your image with the points showing. The **Show Points** item in the **View** menu lets you show or hide control points. When you are dealing with very small paths and points that may overlap, the screen can get a little full, and in those instances you can either reduce the point size or turn off the display of points. This is important for premium WYSIWYG in the Preview mode.

To show control points:

- Choose **Show Points** from the **View** menu to turn on the display of points.



To hide control points:

- If Show Points is already checked, choose **Show Points** from the **View** menu to deselect it.

➲ ***Tip:*** To make points temporarily appear when **Show Points** is turned off, click anywhere outside the path.

Note: To change the size of the points from larger to smaller, or the reverse, turn on the **Draw Using Larger Points** Point display item in the Preferences dialog box.

Editing and placing BCPs

Fontographer gives you considerable control over the shape of curved segments. This is done by adjusting the position of the Bézier Control Points (nicknamed BCPs because Fontographer's curved segments are Bézier curves). Bézier curves define complex shapes with a minimum number of points.

All points include one or more BCPs that occasionally seem to hide inside the control point. Most corner points are constructed so that BCPs remain inside the point. Curve points ordinarily work the other way around. You can think of these points as knobs with handles on the ends.

Pull the BCP out of the control points and use them like levers to control the shapes of the extending line segments. Moving the lever adjusts the angle of the curve. The length of the lever determines the degree and depth of the curved segment. When a point has two BCPs – as in corner and curve points – one handle controls the incoming segment, and the other controls the outgoing segment.

The default preference setting displays the BCP lines when a control point is selected, although you can change this in the Point display area in the Preferences dialog box.

BCP principles

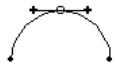
- BCPs control the shape of Bézier curves between points.
- Selecting a control point activates that point's Bézier points.
- BCPs that lie within control points may not be visible.
- There are two BCPs associated with each control point.

Note: If you are using a tool that is capable of inserting points, and you are trying to select a point's BCP by clicking within the normal control point, you may accidentally place another control point instead.

Tip: If a BCP handle is not visible, you can press the **Option** key to drag the BCP out from the control point.

Dragging a control point's BCPs

The BCPs for each of the three different kinds of points behave in different ways. A curve point's BCPs align in a 180-degree angle, whereas a corner point's BCPs move independently of one another. The tangent point's BCPs move only along the line of its slope.



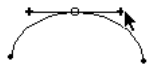
Curve point

Corner point

Tangent point

Dragging a curve point's BCPs

A curve point and its BCPs lie on the same segment. Each BCP (and its curved segment) is affected by the movement of the other BCP. Dragging one of a curve point's BCPs in any direction will result in the other BCP moving to maintain the straight line.



Click the BCP and drag.

As you drag the BCP left or right, the other BCP moves along with it.

➲ ***Tips:*** Holding down the **SHIFT** key while you drag a BCP constrains the movement to the horizontal and vertical axes, or to 45-degree increments between the two.

➲ Holding down the **SHIFT-OPTION** keys while dragging a BCP constrains the BCP movement to the slope of the curve.

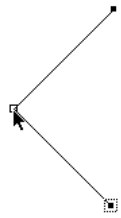
Note: If a BCP and a control point lie on top of one another, any attempt to select the BCP will select the control point instead. In other words, the point is selected before the BCP is recognized (control points have selection priority over BCPs). To get around this problem, select a point, hold down the **OPTION** key, and then drag to move the BCP off of the point.

Dragging a corner point's BCPs

A corner point's BCPs generally lie on top of that same corner point.

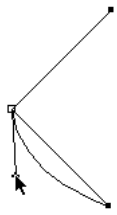
To select an incoming or outgoing BCP:

1. Hold down the **Option** key before clicking the control point.



Moving or adjusting one of the corner point's BCPs does not move the other BCP or its attached segment.

2. Hold down the **Option** key, click the point, and drag out from the control point.



➲ ***Tips:*** Press the **Shift** key and drag to constrain BCP movement to the horizontal and vertical axes, or to 45-degree increments.

➲ Hold down the **Shift-Option** keys to constrain the movement of the BCPs to the slope of the control point's tangent.

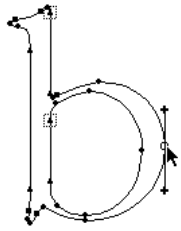
Dragging a tangent point's BCPs

A tangent point's BCPs always constrains to the tangent (that is why they are named tangent points). Consequently, a tangent point's BCPs can never be moved away from the tangent line. Moving the BCP will not change the slope of the curve. The **Option** key can be used to select the BCPs that lie on top of a tangent point.

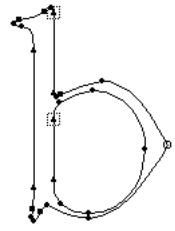
➲ ***Tip:*** If you want to extract the outgoing BCP from a tangent point, you will need to drag out the incoming BCP with the **Option**, then drag out the outgoing with the **Option**, and drag the incoming BCP back onto the point.

Retracting BCPs

As we discussed earlier, you adjust Bézier control handles by selecting the point they're attached to, then dragging the handle. You can also retract the control handles. This is useful when you have a corner point with unnecessary BCPs.



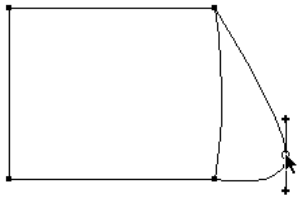
Select the control point and then choose ***Retract BCPs*** *from the* ***Points*** *menu.*



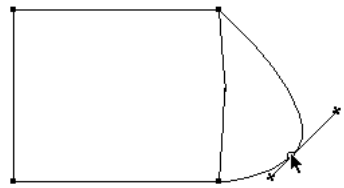
Fontographer retracts the BCPs into their point and changes the curve accordingly.

Auto curvature

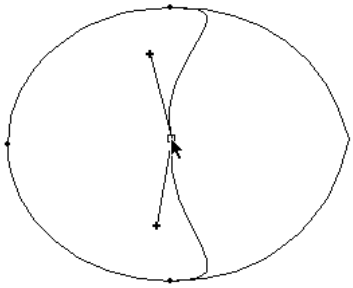
The **Auto Curvature** option instructs the curve point tool where to put the BCPs and how far to extend their handles. You can control auto curvature from the **Points** menu. Auto curvature is active primarily on curve points, but corner points can also have auto curvature if their adjacent points are curve points. Auto curvature allows a point that's being moved to automatically update the angle and length of BCPs. If the adjacent points are also set for auto curvature, their BCPs will also update automatically as the point is moved. This means that when a point is moved, you will no longer have to adjust BCPs after moving a point, nor will you need to select the adjacent points and re-edit their BCPs. A BCP with auto curvature active will look different from BCPs without it. Instead of a square +, the BCPs will look more like an x.



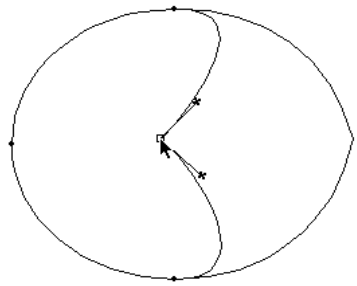
This curve point doesn't have auto curvature.



This curve point does.



This corner point doesn't have auto curvature.



This corner point does.

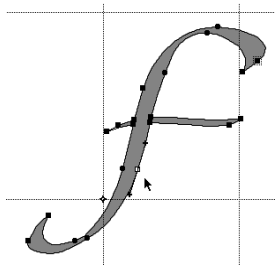
When you use the curve point tool to draw a new path, the curve points will default to have auto curvature on. To toggle it off, select **Auto Curvature** from the **Points** menu or move a BCP. When you edit a BCP, the **Auto Curvature** option turns itself off.

When you insert a curve point on an existing path, auto curvature will be automatically off, otherwise it would distort the path. If you want to turn it on, select **Auto Curvature**.

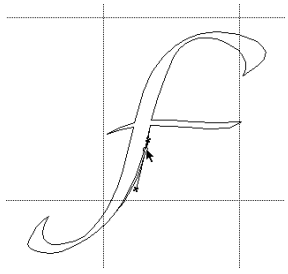
One of the most important uses for auto curvature is in tweaking paths. If you've copied a bitmap image into the template layer and autotraced it, auto curvature can make your work easier.

Here is a template image that was traced.

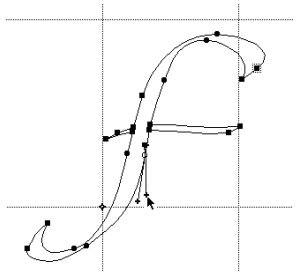
1. Move the curve point so that it will be at the extrema (the outermost edge of any curve).



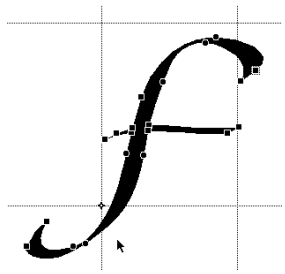
2. Select the point.
3. Select **Auto Curvature** from the **Points** menu.
4. Move the point, and the BCPs update automatically.



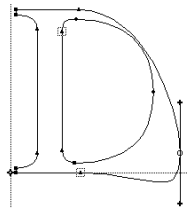
5. After you move the point, move the BCP with the **Shift** key to make it vertical (and conform to the template bitmap).



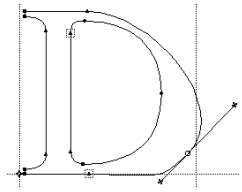
This will turn off auto curvature.



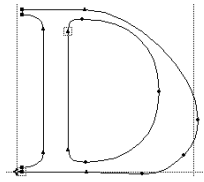
What else can you do with auto curvature? Suppose you want to make a Beer Belly font. Grab the east-most point of the bowl of letter D and pull it out and down. In the bad old days, before you could control auto curvature, it would look like this:



With the **Auto Curvature** option turned on, it will look like this:



Choose **Clean Up Paths** from the **Element** menu, and there you have it: a D with a prominent Beer Belly.



Auto curvature is not an exact science. You may want to tweak the BCPs slightly once you get the point in the place where you want it. But it saves you the effort of constantly having to tweak BCPs, only to decide later that you like it elsewhere. After you're done moving all the paths, we also recommended that you choose **Clean Up Paths** to put the extrema at the extremes so that the glyph (and font) will hint correctly.

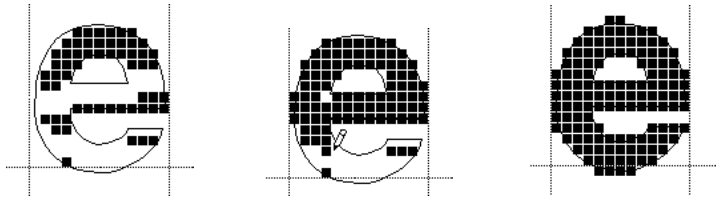
Editing Bitmaps

There are several reasons why you probably won't have to worry about bitmaps. The first is modern system software that improves the screen appearance of PostScript Type 1 fonts. The second is TrueType, the outline font format developed by Apple and Microsoft, which doesn't need bitmaps at all, since the font's outlines are used for the screen display.

5

The built-in support of Type 1 fonts in Mac OS and Windows and invention of TrueType has almost made bitmaps a thing of the past. So we recommend that you spend your time designing good outlines, and let your bitmaps take care of themselves.

However, there are some reasons why you might want to edit bitmaps. Perhaps you'd like to create grayscale Type 3 fonts, or you're creating professional fonts for older systems, or for small electronic devices that still use bitmap fonts. Or maybe you just like playing with pixels.



Create bitmap characters by turning pixels on and off.

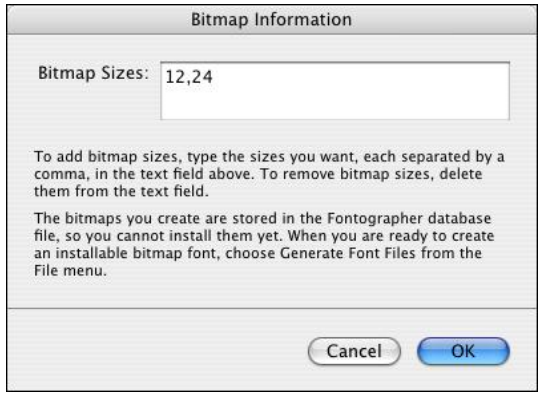
Using the bitmap window

You can use the bitmap window as a tool to improve the look of a font on the screen. This window is composed of a series of square dots, called pixels, which can be turned on or off to produce images. When the dots are turned on they appear black, and when they are turned off they become white.

To open a bitmap window:

1. Select a glyph by clicking it in the font window, opening a glyph's outline window, or clicking a glyph in the metrics window.
2. Choose **Open Bitmap Window** from the **Window** menu.

   Bitmaps must be created before you can view them. If you don't have any bitmaps for the font yet, Fontographer will prompt you to create some. You can type in whatever sizes you'd like in the Bitmap Information dialog box, which will appear automatically if you try to open the bitmap window without first creating bitmaps.

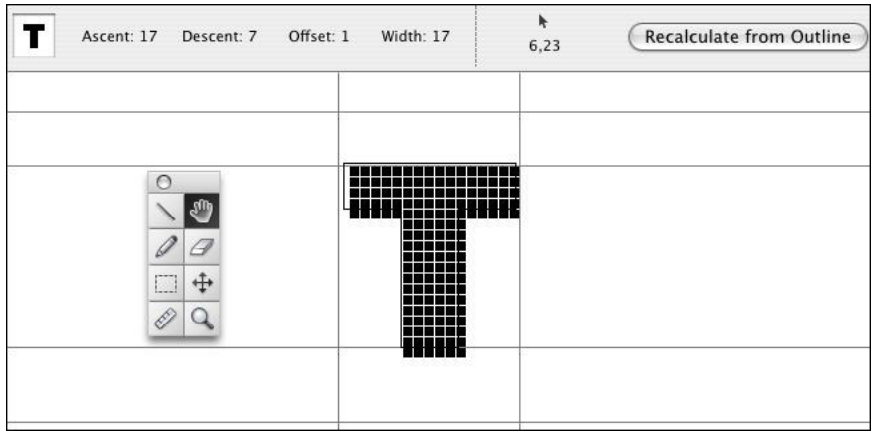


   You can open multiple windows so that different glyphs (or even different point sizes of a single glyph) can be worked on at the same time. This is especially helpful when you are editing a glyph with several bitmap sizes.

➲ ***Tip:*** To close all open bitmap windows at once, **Option**-click the close box for any of the open windows. Any open outline or metrics windows will remain open.

The bitmap window

The bitmap window includes a title bar that shows the name of the font, the point size of the glyph, and the glyph itself. The toolbox contains a pencil, eraser, hand tool, marquee selection tool, magnifying tool, measuring tool, straight line tool, and a move tool.



The lock icon in the lower left corner has two functions. When the lock is open, you can change to a different glyph simply by typing the keystroke(s) of the new glyph. Having the lock icon closed, prevents the glyph from being changed to another glyph, if you accidentally press a keystroke.

➲ ***Tip:*** The **Enter** or **Return** key toggles the state of the lock on and off.

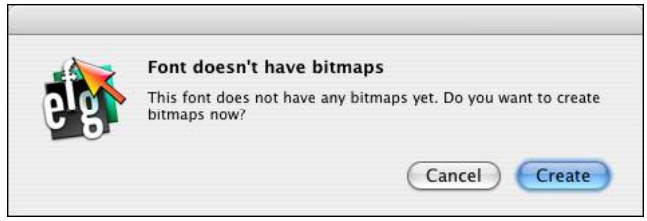
The info bar at the top of the bitmap window shows you the glyph's ascent, descent, offset, and width values, as well as listing the horizontal and vertical location of the cursor in pixels. These features give you the ability to precisely measure distances when you create a glyph. An actual point-size image of the glyph is displayed in the top left corner.

The **Recalculate from outline** button allows you to recompute a new pixel image based on the glyph's outline. This button is useful when you have edited a particular outline glyph and don't want to recalculate the entire bitmap font. For more information, refer to "When should you recalculate bitmaps?" on page 212.

Editing a bitmap

To make changes to a bitmap:

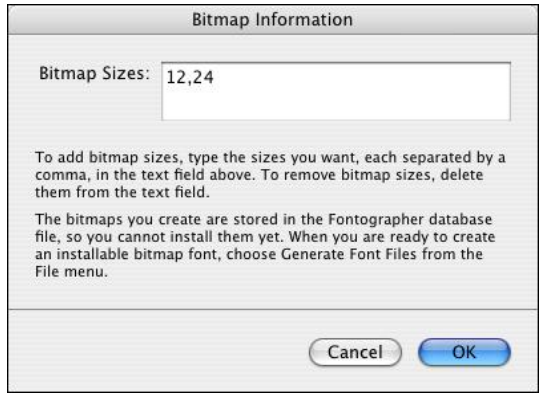
1. Select the "A" in the Font window.
2. Choose **Open Bitmap Window** from the **Window** menu.



If you are opening a font for the first time, you probably won't have any bitmaps associated with it. Fontographer will ask you if you want to generate any.

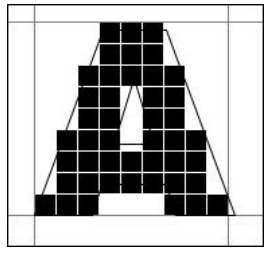
3. Click **Yes**.

The Bitmap Information dialog box will appear. Fontographer automatically defaults to include the 12 and 24-point bitmap size. You can add to or change the sizes in this list if you like.

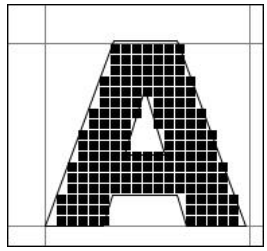


4. Click **OK**.

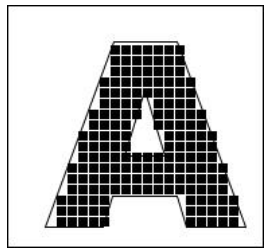
Fontographer generates the bitmap files. The bitmap window of "A" appears.



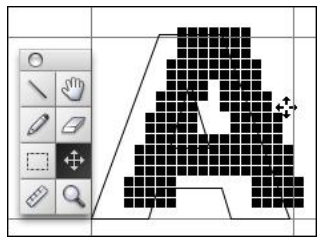
5. Choose **Next Point Size** from the **View** menu to view the next larger point size available.



6. Type **Option-G** to turn off the guidelines, if you find them distracting.

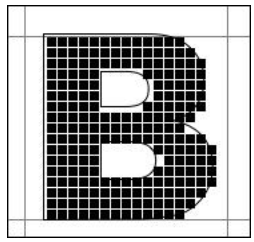


7. Select the bitmap and select the move tool to move the bitmap.

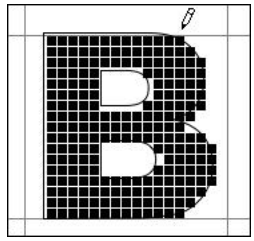


8. Choose **Next Glyph**.

   The glyph “B” appears.



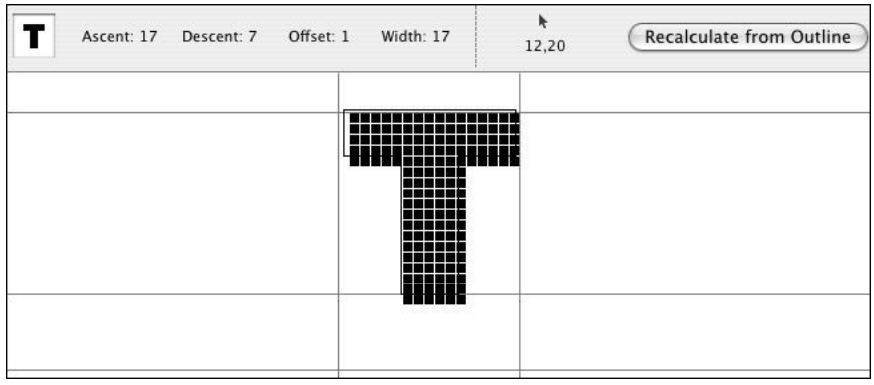
9. From this view decide which bits to change to improve the glyph's looks, and then click with the pencil tool to remove or add them.



That's all there is to editing your bitmaps. The majority of the time Fontographer creates perfectly acceptable bitmaps, so you only need this option if you're a real perfectionist.

The central edit area

The central edit area is the space where the bitmap image appears. This is where all your bitmap drawing and editing takes place. The bitmap image includes pixels, which are simply dots that may be highlighted to create a pattern on the screen. Behind the pixel image is the outline of the glyph to guide you in choosing the pixels that you want to turn on or off.



The Info bar shows the offset and width values for the glyph. The offset value will change to correspond to any changes made to the bitmap glyph's offset. The width value is the glyph's advance width in pixels and cannot be changed in the bitmap window. Changing the width in the outline window will update the bitmap window if you press the **Recalculate from outline** button.

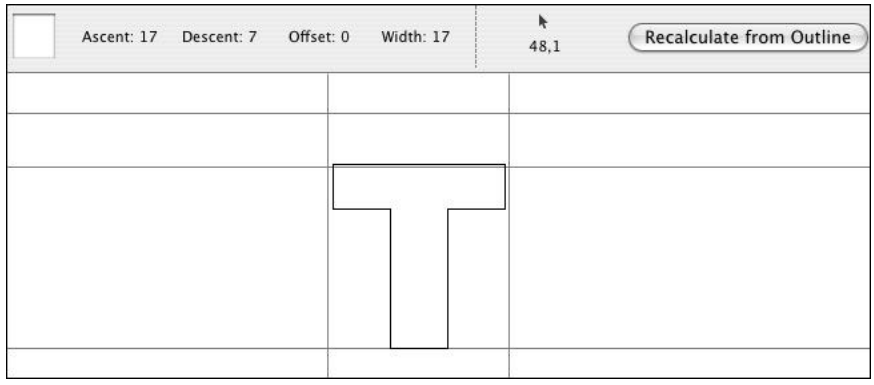
Ascent and descent values

The ascent is the number of points between the baseline and the top boundary of the em square. In the font pictured here, the ascent is 17 pixels from the baseline. Consequently, the descent is the amount of pixels between the baseline and the lowermost boundary of the em square. The descent for this point size is 7 pixels.

The bitmap window in Fontographer 5 includes maximum ascender and descender lines. These lines represent the limits for bitmap placement based upon the outlines for all existing glyphs. You'll find these guides valuable when you are manually creating bitmaps. The maximum ascender line appears only when a glyph's outline extends below the descender line. However, if you have selected to preserve line spacing (in the Recalculate Bitmaps dialog box), the maximum ascender and descender lines will not appear for that glyph; they will be the same as the normal ascender and descender lines.

Offset and width values

The offset is the distance in pixels from the leftmost black pixel to the origin line. This value changes as you edit the bitmap glyph. If there are no points between these two, the offset is zero.



There are 0 pixels between the origin and the left edge of the letter "T" pictured above. The width indicator shows us the width of the glyph in pixels: The "T" is 17 pixels wide.

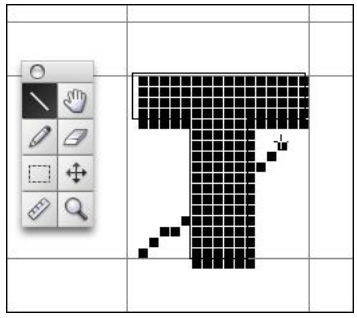
Visible layers

You can choose to display or hide guidelines in the bitmap window by typing **Option-G**. The guidelines you will see are the ascent, descent, and baseline of the font. Use them in the bitmap window as a drawing aid.

To display or hide the Outline layer, type **Option-O**. The outline is visible behind the bitmap, and you can use it as an aid in bitmap editing, but no changes to the outline can be made in this layer. Changes to the outline glyph will, however, display in the bitmap window.

Tools in the bitmap window

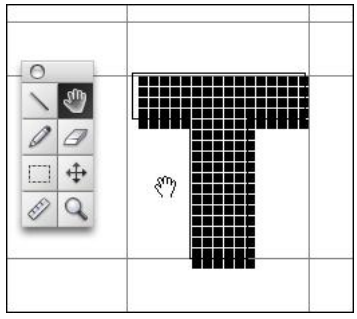
There are eight tools in the bitmap window's tool palette. The straight line tool, directly beneath the close box, allows you to draw straight bitmapped lines by holding down the mouse button and dragging. Hold down the **Shift** key to constrain the line tool to draw horizontal lines, vertical lines, or 45-degree angle lines.



When the lock icon is in the locked position, access the straight line tool by typing **1**.

➲ ***Tip:*** Holding down the **Option** key causes the line to be drawn from the center – the point where you clicked. This applies to other tools as well.

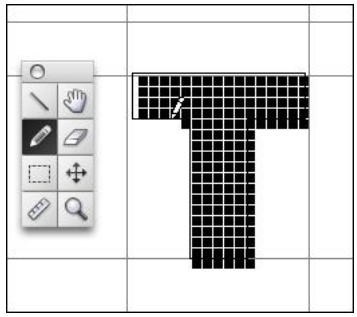
The hand tool is used to move the screen in any direction by clicking the window and dragging. The hand tool scrolls the entire glyph window. Move the bitmap image around to an optimum screen position before you begin to edit it. You'll need to do this if your glyphs are too large to fit into the central edit area.



When the lock icon is in the locked position, you can access the hand tool by typing **2**.

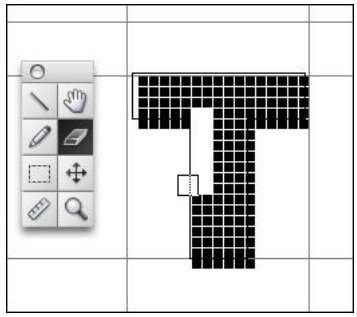
➲ ***Tip:*** You can temporarily change any of the bitmap tools to the hand tool by pressing the **Spacebar**.

Use the pencil tool to place or erase pixels with a click of the mouse. When you place the pencil above the pixel and click, the bit will change from black to white or vice versa. You can either drag or click with the pencil. Clicking draws just one dot. Dragging produces a black or white series of dots. Holding down the **Shift** key while dragging constrains drawing to a 45-degree line, a vertical line, or a horizontal straight line.



When the lock icon is in the locked position, you can access the pencil tool by typing **3**.

The eraser tool removes any pixels it touches. You can also use the pencil tool to erase pixels one-by-one, but using the eraser tool can be more effective for larger areas. Holding down the **Shift** key while dragging constrains the eraser to a vertical or horizontal straight line.



When the lock icon is in the locked position, you can access the eraser tool by typing **4**.

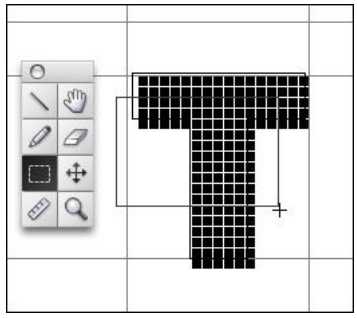
➲ ***Tip:*** Double-click the eraser tool to erase all pixels in the edit area.

The marquee selection tool is used to select and move pixels. If the marquee is present, pasted bits will be scaled to fit within it. You can paste bitmap images into the marquee area. To deselect the marquee area, click anywhere outside the central edit area or press the **Tab** key.

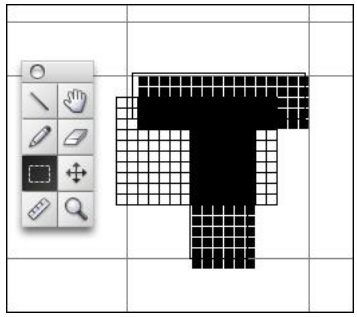
When the lock icon is in the locked position, you can access the marquee selection tool by typing **5**.

➲ ***Tips:*** You can use the marquee selection tool with the standard **Cut**, **Copy**, and **Paste** commands to move or copy pixels between glyphs.

➲ To enclose all the bits automatically, choose **Select All** from the **Edit** menu when the marquee tool is selected.



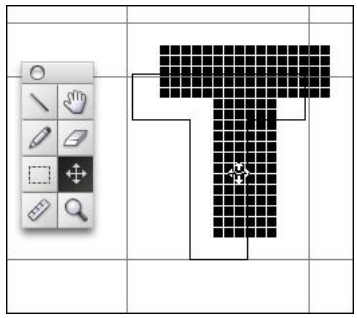
Use the marquee like a selection tool to outline an area to be moved.



The part of the bitmap you select will display as pixels on a grid. This section can be moved to another location on the bitmap.

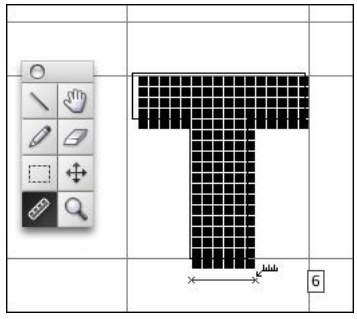
The move tool lets you move the bitmap glyph away from its outline in any direction.

When the lock icon is in the locked position, you can access the move tool by typing **6**.



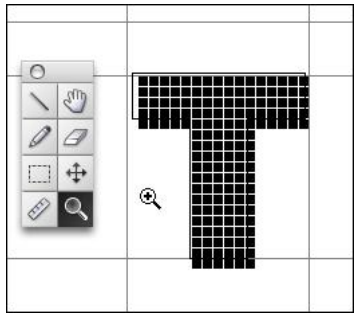
You can use the measuring tool to measure distances between pixels. When you position it and click the mouse and drag to another place, the measured distance in pixels will appear within a rectangular box. The measuring tool is automatically constrained to 45-degree angles, horizontal or vertical straight lines. Hold down the **Shift** key to move it freely.

When the lock icon is in the locked position, you can access the measuring tool by typing **7**.



The magnifying tool can increase the screen size of the bitmap by 2, 4, 8, or 16 times its actual size. When the tool appears on screen, click the mouse and release to show a magnified version. Clicking again increases the magnification unless there isn't a larger size, in which case an empty magnifying tool will appear.

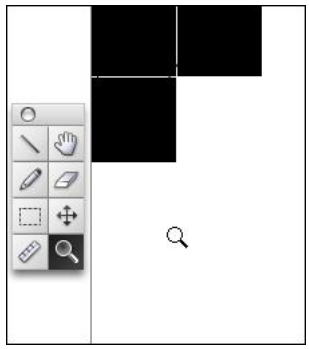
To reduce the size of the bitmap, hold down the **Option** key. The magnifying tool will display a minus sign.



When the lock icon is in the locked position, you can access the magnifying tool by typing **8**.

➲ ***Tip:*** Temporarily get the magnifying tool from any other tool by holding down **Command-Spacebar**. To reduce the image, hold down **Command-Option-Spacebar**.

When you can no longer enlarge the image, the magnification tool will display nothing in its center.



Note: You can also change the magnification of the character by choosing **Magnification** from the **View** menu and selecting the desired view size from the pop-up.

Undo and redo

There are a few ways to undo and redo changes made to the bitmap. If you want to remove an added point, just click it with the pencil tool or choose **Undo** from the **Edit** menu. To redo the change choose **Redo**. The default setting for number of Undo levels and Redo levels is 100, but you can change this from the **Preferences** option in the **Fontographer** menu.

Richard Beatty Says:

If the lack of absolute smoothness on the screen pains some users of Type 3 typefaces, you can make screen fonts (bitmaps) at all the regularly used sizes and hand-edit them to reduce the jaggies to the limits of the monitor screen, just like ATM did for Type 1 typefaces.

Changing bitmap views

Enlarging using the View menu

You can enlarge the bitmap image by selecting **Magnification** from the **View** menu.

In the submenu, you can choose a magnification level for the display of the bitmap image. When you choose **100%**, you are selecting the actual size. The other choices offer you the bitmap at 2, 4, 8, and 16 times its actual size. The corresponding command keys are as follows:

Magnification	Macintosh
Fit in Window	**COMMAND -T**
100% (Actual size)	**COMMAND-1**
200%	**COMMAND-2**
400%	**COMMAND-3**
800%	**COMMAND-4**
1600%	**COMMAND-5**
3200%	**COMMAND-6**
6400%	**COMMAND-7**

To enlarge using the magnifying tool:

- Select the magnifying tool and click once on the screen on the area you want to enlarge in the window.

You can magnify the image to 2, 4, 8, 16, 32 or 64 times the actual size. When you can no longer magnify the image, the tool will display nothing in its center.

To reduce the bitmap image with key commands:

1. Press **COMMAND-OPTION-SPACEBAR** and then click the mouse. The magnifying tool displays with the minus indicator.
2. Continue clicking to reduce the image.

Switching glyphs in the bitmap window

There's a small lock in the lower left corner of the bitmap window. If the lock is in a locked position, it locks the current glyph into that window. This prevents a glyph from being changed in case you accidentally press a glyph key. It also lets you use shortcut commands like **OPTION-G** to hide the guidelines.

If the lock is unlocked, you can switch to a different glyph by typing that glyph's keystroke(s).

➲ ***Tip:*** You can toggle the lock on and off by pressing the **RETURN** or **ENTER** key.

Next and previous glyph

To display the next sequential glyph, choose **Next Glyph** from the **View** menu or use the shortcut **OPTION-RIGHT ARROW** To select the previous glyph choose **Previous Glyph** from the **View** menu, or press **OPTION-LEFT ARROW.**

Next and previous point size

To show the next point size of a bitmap, choose **Next Point Size** from the **View** menu, or press **COMMAND-OPTION-DOWN ARROW**. The next point size will appear only if you have chosen various sizes from the Bitmap Info dialog box in the **Element** menu and you're not already at the largest point size available.

Note: Any new bitmap window you open will default to show the point size of the current window.

To change to a previous point size:

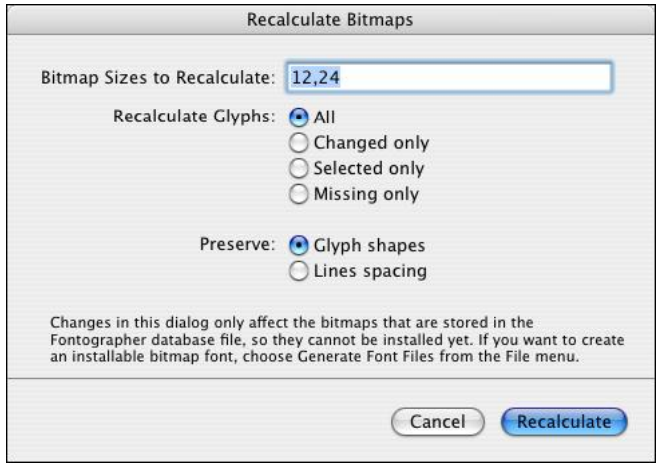
- Choose **Previous Point Size** from the **View** menu or press **COMMAND-OPTION-UP ARROW.** The next available smaller point size will automatically replace the glyph in the bitmap window.

To change to the next point size:

- Choose **Next Point Size** from the **View** menu, or press **COMMAND-OPTION-DOWN ARROW**. The next available larger point size will automatically replace the glyph in the bitmap window.

When should you recalculate bitmaps?

Recalculate bitmaps from the outline when you have edited bitmap glyphs and you want to start over. Also, when editing small point sizes, you can use this to see the results of hint editing or metrics changes. If parts of the font have changed, you can recalculate just those changed letters.



Preserving your original bitmaps

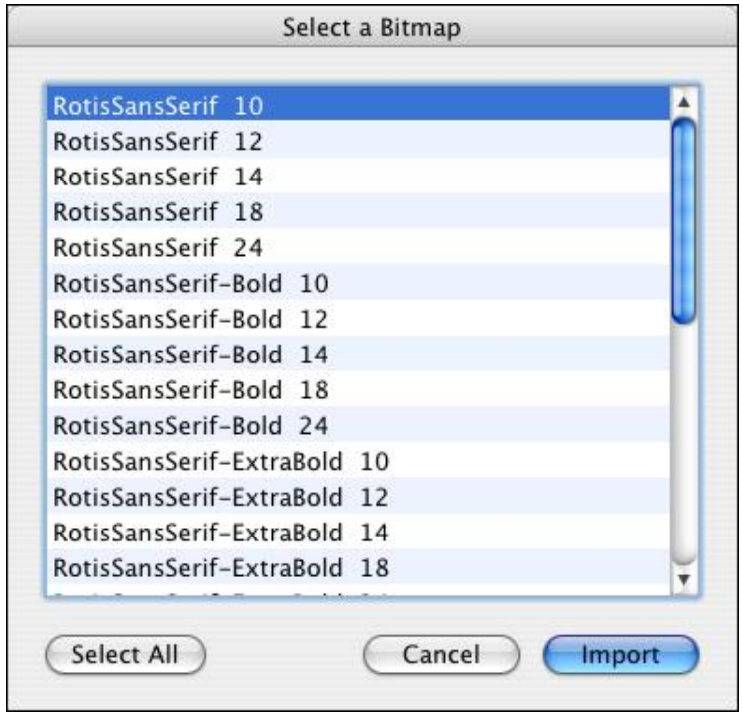
Even though you may have modified an existing font's outlines, you can still keep the original font's bitmaps if you like. This is useful in those instances where you've added some special glyphs to your font (fractions or accents, for example). You don't want to regenerate all the existing bitmaps, since commercial fonts already have hand-edited bitmaps in them anyway.

Like everything else it does, Fontographer offers you an easy way to do this: import any other bitmap font file.

1. Choose **Import** from the **File** menu in any of Fontographer's windows, and then choose **Bitmaps** from the pop-up.

   Fontographer presents a standard file dialog box that lets you find the bitmap font file (Macintosh suitcase) containing the bitmaps you want to import into this database. Then, if more than one bitmap is present, you will be given a choice of which bitmaps to import.

2. Click the file and choose the sizes and styles of bitmap.



   Fontographer will read the bitmap screen fonts from the file and store them in the currently open and selected database, so you'll be able to generate these bitmaps into a bitmap font file anytime you generate a new font.

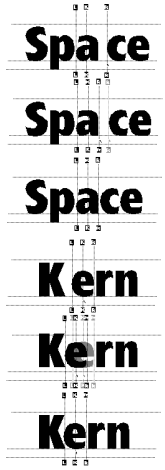
✎ ***Note:*** You won't see the change in the font window, since the representation there is simply a rendition of the outline glyph.

3. Now you can go back and hand-edit your added glyphs if you like.

✎ ***Note:*** The import of bitmaps (fonts) to the template layer is a tool we provided to use as a guide for the size and placement of your glyphs, not to provide glyphs to autotrace. Bitmap fonts contain very little information, compared to an image copied in from a paint or drawing program. When traced, bitmaps will produce a correspondingly poorer image. This is because there is not enough pixel information in the bitmap font glyphs for the algorithm to use in its calculations.

Metrics – Spacing and Kerning

Font metrics is the term used to describe how letters are spaced when they are typed. If you're producing a logo or graphic font that relies mostly on shapes rather than words, you might not need to use either kerning or spacing information. Or if you're a novice just playing around with fonts and aren't yet ready to pursue the finer details, you might delay reading this section. Professional type designers usually do use metrics when they create a font. There are a few good reasons for doing this.



Use the metrics window to manually or automatically adjust the spacing and kerning.

6

The untrained eye may not notice kerning and spacing in printed text, but any reader will experience text as more difficult to perceive if it has not been well kerned and spaced. We know that our brains like the type to coalesce in meaningful groups – and the more clearly defined the meaningful groups are in the visual field, the quicker we'll be able to read the information. When groups (words on the page) do not hang together very tightly, the brain has to work harder to see them in meaningful ways. It would be pretty hard to read this page if: **Itwasprintedlikethisinsteadofwiththecorrectspacing.**

In this case the brain has a daunting group-making job to do. Think of how you feel when reading the unspaced sentence above. When you read poorly kerned and spaced text, you get a toned-down version of that same sensation. So, if you are going to design fonts, you will probably want to pay attention to the metrical details, out of courtesy to the people who may use them.

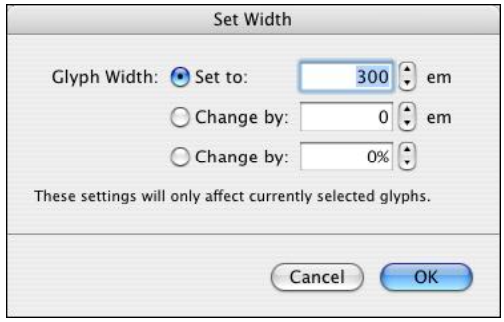
Font metrics is the term used to describe how letters are spaced when they are typed. This is easy to imagine if you remember how type used to be set. When characters were carved on the end of a piece of metal, each character's width was the width of the piece of metal on which it was carved. Setting type was a matter of laying these pieces of metal down next to each other; therefore, the spacing was determined exactly by the width of each piece of metal. To influence the metrics, you could space the characters further apart by wedging little thin strips of metal between them. You could even squeeze the characters closer together by filing down parts of the metal type.

Fundamentally, character spacing was determined by how wide the metal pieces were. And that's still pretty much what we mean by spacing: it's mostly determined by how wide the characters are. Letters are set down one after another, each new character's position determined by the width of the previous character. In other words, each new character starts where the previous one left off. One of the most impressive features of Fontographer is its ability to let you control the metrics of electronic fonts, either letter by letter or the entire font, manually or automatically.

The next exercise shows the power of Fontographer's **Auto Space** command. You can auto space an entire font at once directly from the font window. However, in this exercise, you'll work in the metrics window so you can see Fontographer auto space interactively on screen. In addition, we are going to ask you to purposely mess up the spacing of your glyphs, so you can see just how effective our auto spacing is, even in a worst-case scenario.

To automatically space a font:

1. Open one of your fonts.
2. Choose **Select All** from the **Edit** menu.
3. Choose **Set Width** from the **Metrics** menu.
4. Set the width as illustrated below:



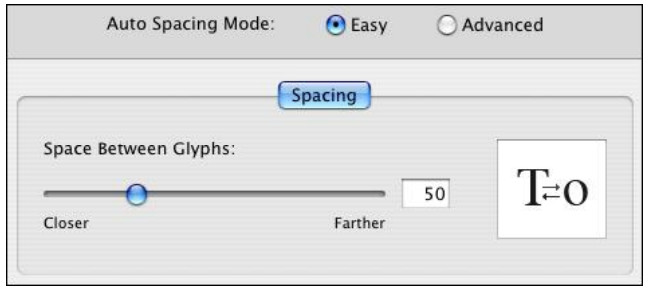
Fontographer will automatically set the spacing of the glyphs to be a uniform width of 400 em units, producing an awful, monospaced font.

5. Click any glyph to deselect all.
6. Choose **Open Metrics Window** from the **Window** menu.
7. Type the word “Space”.

Rather than manually setting different widths for each monospaced glyph, you can auto space the font.

8. Choose **Auto Space** from the **Metrics** menu.

   The Auto Space dialog box appears.



   The dialog box has an Easy and Advanced mode.

9. Choose Easy by clicking the **Easy** radio button.

10. Enter a value for spacing the glyphs. This is the result of a value “30” entered in the text edit box.

    You can adjust the spacing even further from the metrics window or apply auto spacing again (with a different value) until you get the look you like.

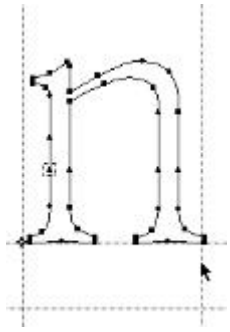
Fontographer also has auto kerning, which works in the same easy manner as auto spacing. The rest of this chapter covers both auto spacing and auto kerning in more detail.

Spacing

Since metal type pieces don't actually exist in the computer, electronic type is not bound by the same kind of physical limitations that used to apply to type production. In fact, you even have to tell the computer how wide each glyph is supposed to be, so it will know how much to move over before placing the next glyph. That's what the handy width guide in Fontographer's outline window is for: you should place the width guide where you want the next glyph to begin with respect to the one you are modifying.

➲ ***Tip:*** Usually, you should set the width of each glyph to be pretty close to the parts of the outline closest to the right.

✎ ***Note:*** If you have not defined the width of all of your lowercase glyphs, your cursor may not behave correctly in some word processing programs. This is because some programs determine cursor width by averaging the width of all of the lowercase glyphs in a font.



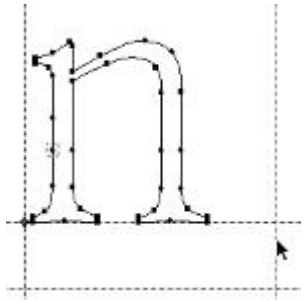
font

If you set the width of the "n" like this...

the letter spacing results in this.

Determining proper widths for each character in a font is a difficult task, because each letter can appear alongside any other letter. You can imagine that a spacing value for "W", which looks good for "Wo", may not look as good for "Wh". Therefore, finding the "right" values to use is a matter of looking at lots of examples, and making some tradeoffs. This is sometimes easier said than done: If your font has 200 characters in it, for instance, there are 200 x 200 = 40,000 different character combinations to consider.

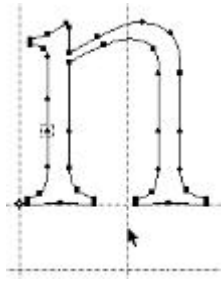
Some letters can never really be spaced correctly for all possible combinations. For example, the “T” usually presents some problems.



fon t

If you want a wider space given to the “n” and you change the width to this...

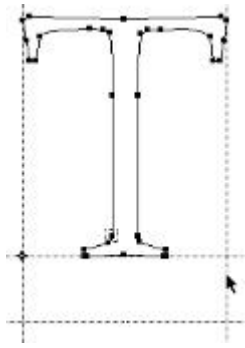
your result will look like this.



font

Of course, if you set the width too tight...

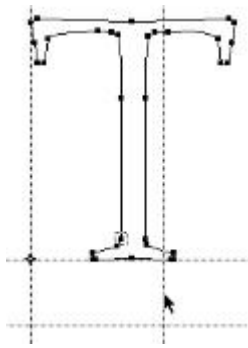
the computer will happily pile letters on top of each other, like this.



To

If you set the width of the “T” like this...

some combinations of letters might seem too loose.



To

Make the width narrower… *to fix the problem.*

Now, of course, you have a new problem.

Th TT

Look at this… *or this.*

Clearly, you can fix some of the spacing problems some of the time, but never all of the spacing problems all of the time. Which brings us to the subject of kerning.

Pair kerning

Use pair kerning to get around those sorts of spacing difficulties. With a properly kerned font, you can actually get perfectly wonderful glyph spacing all of the time, with a bit of extra work up front.

Pair kerning, or informally, just kerning, is a way of overriding a glyph's spacing in certain circumstances. To do kerning, you must first have a properly spaced font. So, in this example, you would pick a width for "T" which works best for most cases. Since overlapping glyphs look pretty ugly and should almost always be avoided, you will pick a width that fixes "Th" and "TT", and leaves "To" a little loose. Once the spacing has been determined, you then look at all the glyph combinations which cause trouble for the spacing value you picked. Typical problem glyph pairs involving "T" are "Ta", "Te", "To", "Tu", "Tw", "Ty", and so on. Those glyph pairs are prime candidates for kerning pairs. You can create different kerning pairs for each of those cases, and individually adjust the spacing for each pair.

Here's how it works: Let's say the optimum width for "T" is 825 em units. That means that the "T" is always 825 em-units wide, no matter what glyph follows it, except for the ones we define as kerning pairs. For example, the "T" in "Ta" could pretend to have a width of 780, so the "a" gets tucked underneath a bit. For "Te", we would start with the same value and modify it, if necessary.

Another way to say it is that in the presence of kerning pairs, widths of glyphs can change depending upon what the next character is (that is, the widths are contextually sensitive).

➲ ***Tip:*** You can create as many kerning pairs as you like, but for reasons we'll get into later, you should strive to have as few as you can get away with. You should never have sets of kerning pairs between one letter and every other letter (for instance, having all the kerning pairs "Ta", "Tb", "Tc",... all the way to "TZ"). If all those kerning pairs are necessary to make the spacing look good, that's a clue that you chose the wrong width for the "T". By giving the "T" a better width value, you would be able to do without some of those kern pairs. This give and take between width settings and kerning pairs is part of what makes typography both an art and a skill.

There is an upside to kerning pairs, which we have discussed, and a downside. Many programs do not support kerning. There is a rule of thumb to use about kerning and whether programs do it: typically, if the suggested retail price of a piece of software is $400 or over, it will support kerning. Otherwise, it may not. There are exceptions to this rule of course, but it is usually correct. Programs that do page layout (Adobe InDesign®, Quark XPress, and so forth) almost always use kerning; the same is true for the higher end graphics programs. Some word processors, spreadsheets, databases, and cheaper graphics programs still do not support kerning. The point is that a font must look halfway decent without any kerning in it at all, because you will probably want your font to look good both in your word processor (without kerning) and in your page layout programs (with kerning).

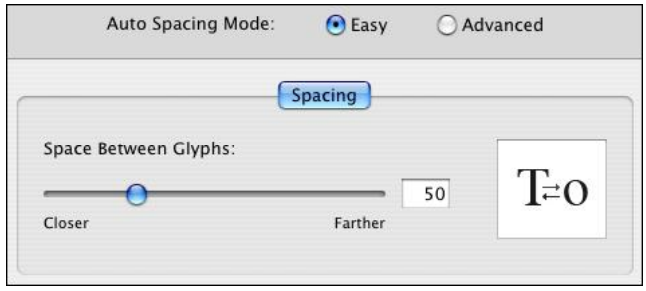
The other drawback to pair kerning is that the more kerning pairs you make for your font, the bigger and slower it becomes. Granted, it won't make the font tremendously slower or bigger, but it can become noticeable in extreme cases. Most commercial font vendors ship their fonts with between 100 and 4000 pairs per font. 100 pairs is certainly too few (except for a monospaced font, in which case it is too many). 4000 pairs is clearly (to us, anyway) a sign that the font was poorly spaced and had to be compensated for by overkerning it. A happy medium might be somewhere between 400 and 1500 pairs (for small Latin fonts), depending upon the typeface and nature of the particular type designer. For example, some people like to kern all the characters in the font, even pairs like ª*f*, or º¢, rather than just the most commonly used ones, which will naturally result in more kerning pairs.

Now that you know the basics of font metrics, let's cut right to the good stuff: auto spacing and auto kerning (located in the **Metrics** menu).

Auto spacing

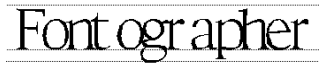
Auto spacing is one of the most incredible features in Fontographer. Auto spacing is the process by which Fontographer will examine your entire font, and give each character the best possible spacing value that it can. It is a tremendously valuable, timesaving command. Some of our famous typographical customers, while they sort of sniffed at the idea of the computer algorithmically creating widths and sidebearings for them, nevertheless allowed that they would certainly use auto spacing at least as a starting point.

Auto spacing comes in two flavors: easy, and advanced. Easy is what the casual user should choose: its parameters have already been set to do the best job for typical fonts. If you are really into spacing, you can switch to the **Advanced** mode, where you have the opportunity to change lots of values and parameters.



The only control you can modify in the **Easy** mode is the one that determines how close (or how far apart) the spacing should be.

Here's an example: We opened up the font TFHabitat and demolished the spacing by setting each letter's width to 450.



As you might expect, it looks kind of rough; however, this is what you get after having drawn a typeface without setting any spacing.

Now, rather than manually and painstakingly setting different widths for each character, we'll simply have Fontographer auto space the font. This is the result of an Easy mode auto space, with the value 60 entered using the slider:

That's a little loose for our taste; let's auto space again, this time with a value of 25:

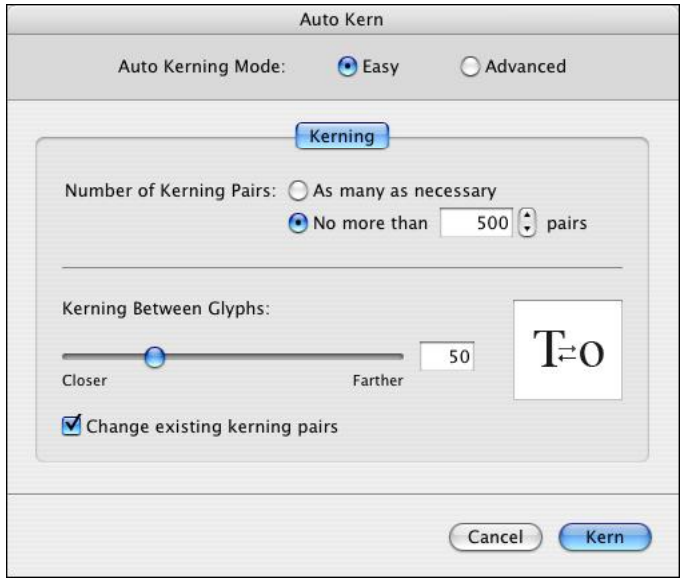
As you can see, auto spacing is as easy as choosing a value, and seeing if you like the result. Any glyphs that have spacing you don't like can easily be tweaked in the metrics window.

Auto spacing, as you can probably imagine, is not the fastest operation you can perform on the computer. It is very calculation-intensive and may seem a bit slow. Nevertheless, it is at least one hundred times faster than we could have set the spacing by hand.

There is a lot more to auto spacing; its extra controls are in the Advanced mode, which is covered a little later on. We would like to reiterate, however, that the Easy mode is perfectly adequate for probably 90% of our Fontographer customers. Don't feel obligated to deal with the Advanced mode unless you want to; it usually isn't necessary.

Auto kerning

Since we have taken the pain out of spacing a font, you might assume we did the same for kerning a font, and you'd be right. Auto kerning does a very good job of creating kerning pairs automatically for your fonts. Like auto spacing, it has two modes: Easy and Advanced. Easy-mode auto kerning has a bit more to specify than auto spacing, but it is still pretty simple.



To auto kern in the Easy mode:

1. Choose **Auto Kern** from the **Metrics** menu.
2. First, decide how many kerning pairs should be created. You can either choose "As many as it takes", or you can set an upper limit on how many should be created.

   This is a personal preference – some people like to have only 100 or 200 pairs in a font; others don't care. We recommend limiting the number of pairs to 1000 or so.
3. Next, choose a setting for how close together you want the kerning pairs to be.

   This control is a lot like the one in auto spacing, only for kerning.
4. Finally, use the **Change existing kerning pairs** checkbox to change existing kerning pairs.

This is useful for when you have manually set some special kerning pairs, and want Fontographer to do the rest without changing your work.

➲ ***Tip:*** If you want to apply auto kerning successively with different parameters; to get the best result, be sure to choose **Clear Kerning Pairs** from the **Metrics** menu before **Auto Kerning** to erase the pairs from the previous auto kern.

Here is an example of some potential kerning pairs, as seen in the metrics window in an unkerned state.

Now we'll apply auto kerning, with a slider value of 60, which yields this:

Fontographer's auto kerning has created a To, Wa, Yo, and AV pair. Remember, these are only the glyphs that are currently displayed in the metrics window, there are other kerning pairs. Note that Fontographer did not create an oW, aY, or oA kerning pair. That is because in the Easy mode, Fontographer tries to kern only the more useful pairs; you don't usually see a lowercase letter immediately followed by an uppercase letter, so it didn't create a bunch of unnecessary kerning pairs. (This behavior may be overridden. The controls for this are located in the Advanced mode of the Auto Kerning dialog box.)

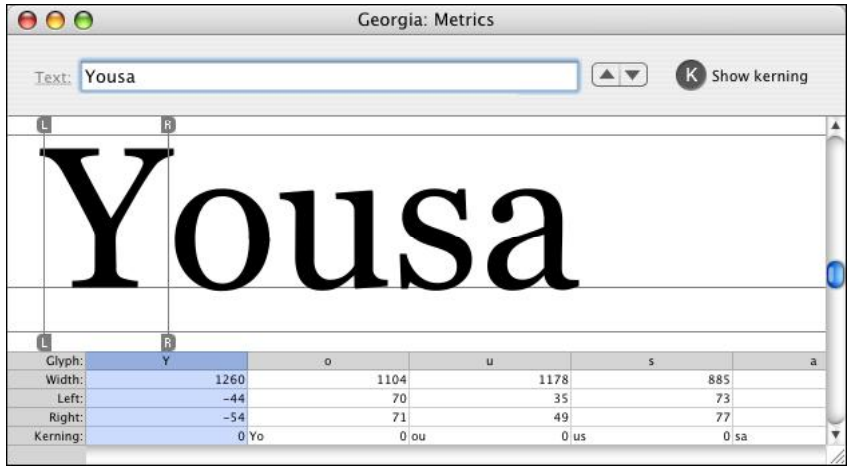
Just for comparison's sake, here are the resulting kerning pairs for a slider value of 20:

With Fontographer's auto kerning, you can try different values and see what you get, just like with auto spacing. Also like auto spacing, you'll find a ton of hidden ability, conveniently tucked away in the Advanced mode. The controls in the Easy mode are suitable for 90% of our customers. Don't feel obligated to wade through the Advanced mode parameters unless you really feel compelled.

✎ ***Note:*** To get the best results, you should auto space your font first and then auto kern it.

The metrics window

Since most metrics issues involve how glyphs look in relation to one another, you might expect a way to simultaneously view glyphs from within Fontographer. There is, and it is cleverly called the metrics window. To invoke it, simply choose **Open Metrics Window** from the **Window** menu. The contents of the metrics window should look something like the one below.



You enter text in the box at the top of the window labelled **Text**. You can either type in text directly or paste in lines of text from some other source. There is a theoretical maximum of 255 glyphs; however, the practical maximum is probably considerably smaller than that. Basically, the fewer glyphs there are in the window, the snappier Fontographer's performance is going to be. In our experience, most people never put in more than about ten glyphs at a time.

If you select some glyphs in the font window, and then open the metrics window, it will appear with those selected glyphs automatically showing in the window. To view even more glyphs at once, you can, of course, open many different metrics windows at the same time.

The large area in the middle of the window is called the glyph display area, and that's where the actual glyphs are shown. They are all positioned next to each other according to the spacing information set up in the font. Kerning pairs are also shown in this area, if there are any.

The rather complicated area at the bottom of the window is the spreadsheet area. This is where all the exact kerning and spacing values are shown. You can enter numbers directly into these cells or merely look at them to see what the values are. You can also cut, copy, and paste values between the cells.

In the upper right section of the window you see a checkbox marked **Show Kerning**. When checked, the metrics window character display will show the effects of any kerning pairs defined for that font. When deselected, it shows the characters without kerning applied to them.

Kerning is on *Kerning is off*

➲ ***Tip:*** This is a great way to judge how your font will look in different environments – with a quick mouse click, you can flip back and forth between modes, and see how the text will look in a word processor without kerning as compared to in a page layout program.

The **Text** link, also in the left corner of the window, is an alternate way to enter text samples into the window. If you click on the **Text** link, a standard file dialog box comes up, which allows you to choose a standard text file. After you have chosen a file, the metrics window will display the first line of that file.

You can navigate from line to line by using the little up and down arrows located to the left of the **Text** link. This allows you to come up with particularly illuminating files of text samples and run your fonts through them quickly and easily, without having to type in all kinds of stuff over and over.

Editing a Text String

In Fontographer 5 you may use Fontlab glyph-access notation to access glyphs that have no characters mapped to the current keyboard layout.

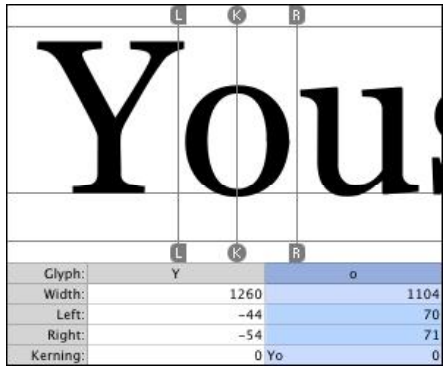
Fontlab Text String Notation:

Character	Meaning of the following text
/	Glyph name follows the slash: /A Follow the name with another '/' to continue entering glyph names or enter a *space* after the glyph name to continue entering ANSI characters: /Acaron/Adieresis BCDEF /dollar Note that glyph names are case sensitive
//	'/'
\	Unicode codepoint of the glyph in hex format may be preceded with 'u' \0445\0448\u0446 BCDE Note that Unicode codepoint in format XXXX is case-insensitive
\\	'\'

According to the above table there are four methods to see glyph 'A' in the metrics window: A/A\0041\u0041

Glyph display

The fun thing about the metrics glyph display is that all the attributes of the glyphs can be modified interactively with the pointer tool.



The “L” guide is for adjusting the left sidebearing (the distance from the glyph origin to the beginning of the actual glyph outline). The guide itself is situated at the glyph origin. Simply click anywhere on the line with the mouse button and drag to change this value.

The “R” guide does the same thing, but for the right side-bearing (the distance from the rightmost part of the glyph outline to the width of the glyph). The guide itself is situated at the glyph width. Click anywhere on the line and drag to change this value.

The “K” guide is for creating and adjusting kerning values. In Fontographer, you kern the currently selected glyph with the previous glyph. That’s why, when you select the first glyph in the metrics window, no kerning guide appears: there is no previous glyph with which to kern. To create a new kerning pair, or to modify an existing pair, simply click anywhere on the line and drag it around until the kern pair looks right.

As you drag these indicators around, you can watch their values change interactively in the spreadsheet area.

Fontographer’s multiple levels of undo and redo are active for each glyph in the window. To move through a particular glyph’s set of undo levels, select that glyph and then choose **Undo** from the **Edit** menu.

➲ ***Tips:*** OPTION-dragging on the “L” bar keeps the left sidebearing constant as you move the glyph; OPTION-dragging on the “R” bar keeps the right sidebearing constant.

➲ In order to get the feel of these bars, you have to try it. Some people like to adjust sidebearings directly, by simply moving the bars; others like the indirect fashion of using the **Option** key, which actually adjusts the opposite sidebearing from the one being modified.

Of course, moving the glyph itself around between the sidebearings is very straightforward. Simply click any glyph and move it left or right. If you hold down the **Shift** key, the glyph movement is constrained to up and down with respect to the baseline. If you hold down the **Option** key, you can move it anywhere you want: over a bit, up off the baseline, and so forth.

A good technique for manually creating kerning pairs is to enter the left glyph, "T" for example, and enter a glyph like "a" for the second glyph. Select glyph "a" and the kerning line appears. Click the kerning line and move the mouse to adjust the kerning. When you have kerned Ta to your liking, you can choose **Next Glyph** from the **View** menu, and the metrics window will update itself to show Tb. You probably don't want to kern that one though, so you can leave it alone and just choose **Next Glyph** again until you see another pair you think needs kerning. This is an easy way to move quickly through all the combinations of glyphs.

Once you have made lots of kerning pairs, you can step through them all with the **Next Kerning Pair** and **Previous Kerning Pair** commands, also found in the **View** menu. This way, you can see all the pairs you've kerned and ignore those you didn't kern.

➲ ***Tip:*** Using **Next Kerning Pair** and **Previous Kerning Pair** is a good way to learn from other type creators: you can import other fonts' kerning tables (described later in this chapter) and step through all their kerning pairs.

If a glyph is selected, you can choose a new glyph simply by typing a different letter. However, be sure you do not have a blinking insertion cursor somewhere else in the window. If you did, you would enter a new value into that field. You can easily tell if typing a new glyph will change the current selection by looking at the glyph column in the spreadsheet. If it is selected, as in the previous metrics window illustration, that means you can change glyphs in this way. The general rule of thumb is that what you type from the keyboard will replace any selected cell's contents with that keystroke. If there is a blinking cursor, on the other hand, the new keystroke will simply be inserted into the existing cell contents at the location of the blinking cursor.

- ***Tips:*** **Option-G** and **Option-H** are exceptions to the above rule. **Option-G** will toggle the horizontal guidelines on and off. **Option-H** will toggle the sidebearing and kerning lines on and off. This is to allow people to view their glyphs with as little clutter as possible.
- If you are editing in the spreadsheet area, **Option-G** and **Option-H** will simply give you those glyphs rather than changing the visual states of the guidelines. To get out of the spreadsheet area, click an empty area of the glyph display.
- Holding down the **Spacebar** in the glyph display area will temporarily give you the hand tool. Holding down **Command-Spacebar** will give you the magnifying tool. **Option-Command-Spacebar** will allow you to reduce the image. The different magnifications in the **View** menu also work in the metrics window. See Chapter 14, "Reference", for more information on magnification.

The spreadsheet area

The spreadsheet area has numeric displays for all the entities in the glyph display section: the glyph, width, left sidebearing, right sidebearing, and kerning values.

To activate a cell:

1. Click it, and it will become highlighted.
2. You can delete cell contents, paste in new values, or type in new values.
3. To have a newly entered value take effect, be sure to press the **Return** or **Enter** key.

   If you simply move to a width cell, for instance, and type in a new number, nothing will happen until you: press **Return** or **Enter**, click another cell, move to another cell, change to a different glyph (previous/ next), or click in another view.

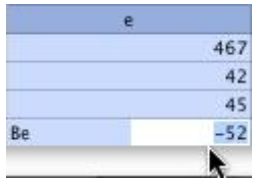
To move from cell to cell:

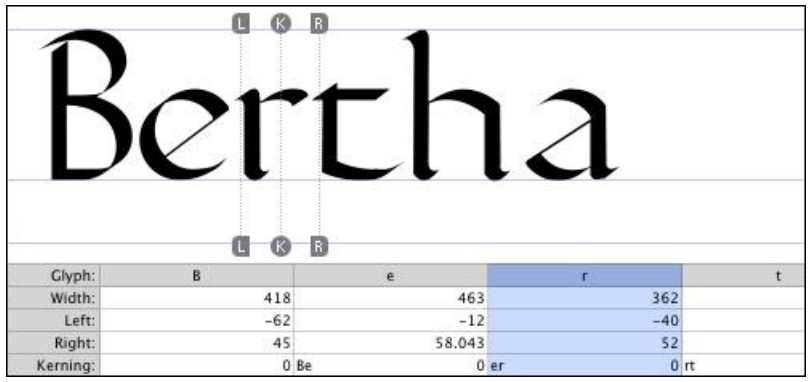
- You can use the **arrow** keys or the **Tab** key.
- You can simply click a new cell with the pointer. **Shift-Tab** moves backward through the fields, just like it does in dialog boxes.

➲ ***Tip:*** To nudge values up and down, you can use the **arrow** keys: **Option-Up** or **Option-Right** moves the values up in increments of one; **Option-Down** or **Option-Left** decrements the values by one; **Shift-Option-Up** or **Shift-Option-Right** increments the values by ten; **Shift-Option-Down** or **Shift-Option-Left** decrements the values by ten. Use these shortcuts to tweak the existing values without constantly re-entering numbers.

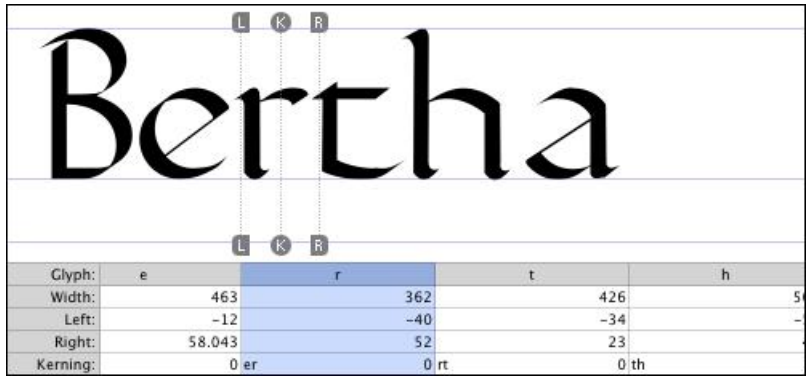
Sometimes, you will be examining a font whose glyphs are wider or narrower than the cell widths in the spreadsheet area, and so the glyph and its spreadsheet view will no longer be vertically aligned. This doesn't affect the functioning of the spreadsheet; however, it can sometimes look confusing. To correct this, simply **Option**-click the glyph in the glyph display area, and its corresponding spreadsheet column will adjust itself to be directly underneath that glyph.

➲ ***Tip:*** It's easy to remove individual kerning pairs by using the spreadsheet area. Simply double-click the kerning cell that displays the kerning pair you want to get rid of, erase the value by pressing the **Delete** key or entering zero, then press the **Return** key or **Enter** key and the kerning pair will be gone.





Here, the numbers pertaining to the "r" have drifted away from the visual representation of that letter.

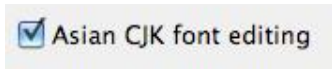


After ***Option****-clicking the letter, the spreadsheet has moved itself so the two areas align.*

One important thing to remember about the spreadsheet area is that you don't have to use it. Some people are much more visually oriented, and they get dismayed by fields of flashing numbers. That is why we made sure everything is available interactively. If you don't want to fiddle with tables of numbers, feel free to completely ignore the spreadsheet. On the other hand, be aware that some of our more production-oriented font development people have told us that they can create between 400 and 500 kerning pairs an hour, with their hands never leaving the keyboard. So see which method works best for you, and don't worry about the other.

Vertical spacing & kerning

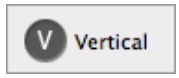
If you need to work with an Asian CJK font and edit vertical glyph metrics and kerning then go to the Preferences dialog and turn on this option:



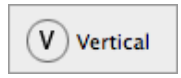
When you open the Metrics window next time you will notice the glyphs are turned on their sides:

This “vertical” mode allows you to edit Height, Top and Bottom instead of Width, Left and Right values.

Clicking on the **Vertical** button will switch the mode back to horizontal.



Vertical is on



Vertical is off

When an OpenType font file is generated in CJK mode, Fontographer will generate a ‘vkrn’ feature similar in style to the ‘kern’ feature, but based on the vertical kerning data.

Note: The Auto Space, Auto Kern, Kerning Assistance, Metrics Assistance, Set Metrics, Set Width, Equalize Sidebearings functionality only applies to horizontal metrics and kerning. The Clear Kerning command applies both to horizontal and vertical kerning.

Importing metrics

When you open up an existing Type 1 font, Fontographer automatically includes its spacing information, because it is located in the PostScript file. So the glyph widths and offsets that appear in the metrics window are the actual values from the font. However, Fontographer does not automatically load the font's kerning table. This is because the kerning information is found in a different file, and Fontographer has no way of knowing where that file might be. In the case of Macintosh Type 1 fonts you may choose to open them by opening the suitcase file. In this case Fontographer reads all the necessary spacing and kerning info.

Kerning tables are stored in various places: in the bitmap suitcase (Macintosh) and in the PFM file (Windows). Kerning and spacing information can be found in other places as well: AFM files, other Fontographer databases (.fog), or in the Fontographer Metrics files (.met) described later in this chapter. When you have opened the outline font, you can choose **Import**, then **Metrics** from the **File** menu. That command brings up a standard file dialog box. Simply select the file that contains the kerning tables, and Fontographer will open that file and apply those kerning pairs to your font. Be sure to do this when modifying existing typefaces; otherwise, unless you create them yourself, your font will not have any kerning pairs in it. OpenType fonts often have embedded kerning information. Fontographer will read in any existing kerning data as it opens an OpenType font.

Clearing kerning pairs

If you import the kerning from the wrong font, either by accident or because you were curious about how Helvetica might look with Times-Roman spacing, you can easily undo this experiment by choosing **Clear Kerning** from the **Metrics** menu. This command will remove all kerning pairs from the font you are editing including vertical kerning in CJK mode. Then use the **Import Metrics** command again.

➲ ***Tip:*** Deleting individual kerning pairs is best done in the metrics window. Just highlight the cell that contains the kerning value you want to delete, and either press the **Delete** key or enter zero, and press the **Return** key or the **Enter** key. To delete lots and lots of kerning pairs (but not all), your best bet is to export the kerning into a Fontographer Metrics file, delete a bunch with a text editor, and reimport them back into your font. This procedure is discussed on the next page.

Exporting metrics

Sometimes you want to export a font's metrics to a file. There are a couple of reasons for doing this. For example, you might want to experiment with a number of different sets of metrics to see which fits the best. With the ability to export the current metrics information, you can save off what you have, choose **Clear Kerning Pairs** from the **Metrics** menu, and then try importing a bunch of stuff. If that doesn't work out, you can import the metrics you exported and be right back where you started.

Another reason to export metrics is just to see them all. Some people like to see all the width values in a big list, as well as all the kerning pairs and kerning values. These lists can be printed out and compared with one another; you can even edit the lists and reimport the metrics information.

Exporting metrics is easy: just choose **Export**, and then **Metrics**, from the **File** menu. This will cause a standard file dialog box to appear, along with a number of choices for the kind of file to create: an AFM, PFM, or Fontographer Metrics file (.met). With Fontographer, you can, of course, import metrics from all the file types that it can create.

The Fontographer Metrics file

The Fontographer Metrics file is a standard text file any word processor can read. The reason for having a file like this is that it is more pleasant to deal with than AFM or PFM files. People who like seeing their metrics information in a big long list can do so via this file. They can also edit width and kerning values and even create new kerning pairs by typing them in and entering kerning values. Then they can import these metrics back into their fonts.

A Fontographer Metrics file looks like this:

```
FogMetricsFile
1000 em square
%  Output   character  spacing
17  670  width
18  790  width
19  58  width
20  620  width
( )  250  width
(!)  271  width
(“)  309  width
(#)  664  width
($)  585  width
(%)  736  width
(&)  748  width
...
A  733  width B  649  width C  679  width D  729  width E  575  width F  536  width
G  708  width
...
252  437  width
253  500  width
254  440  width
255  405  width
256  280  width
%  Output   kerning pairs
A C  -78  kern
A G  -99  kern
A o  -49  kern
A v  -113  kern
A w  -141 kern
A y  -141 kern
```

Glyphs for which there are no standard letters (like the first 20 glyphs or glyphs over 128) are indicated by their character number. So a line like **254 440 width** means that character number 254 has a width of 440 em units.

Copying widths

The **Copy Widths** command in the **Edit** menu is a quick way to select a bunch of glyphs in the font window, and then copy and paste their widths over a selection of other glyphs, without altering anything else in those other glyphs.

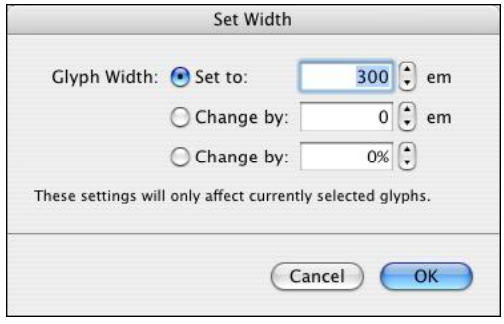
More powerful spacing and kerning commands

Up until now, we've been discussing mostly manual ways of adjusting and creating metrics. Fontographer has some really useful automatic ways to do metrics. The metrics commands, located in the **Metrics** menu, range from completely easy and automatic to very technical and extremely powerful. Or, to use a different analogy, it can be the difference between flying a kite and flying the space shuttle; it sort of depends upon what you want to do.

Setting widths

The **Set Width** command is a straightforward and easy way to adjust the widths of lots of glyphs at once.

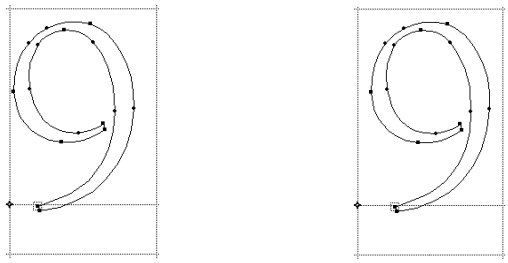
The **Set Width** command works with the current selection. So if you are in the font window, simply select all the glyphs you want to change, and then bring up the Set Width dialog box. You can simply replace each glyph's width with a new one, or you can increase or decrease the existing widths. This is an easy way to take an entire font, and quickly make the spacing five percent looser, for example.



Equalizing sidebearings

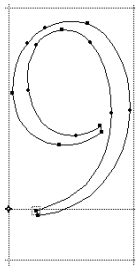
Sometimes you will want a glyph's left sidebearing to equal its right sidebearing, for the sake of vertical alignment. Numbers frequently appear in tabular form, and it's a lot nicer when the columns of numerals line up. In addition, some programs (like Adobe FreeHand) can vertically align arbitrary text.

Equalizing the sidebearings is easy: choose **Equalize Sidebearings** from the **Metrics** menu.



*Choose **Equalize Sidebearings** from the **Metrics** menu to make this glyph (left) look like this (right).*

➲ ***Tip:*** Holding down the **Option** key when you select **Equalize Sidebearings** will make the sidebearings equal by adjusting the width, rather than moving the outlines around between the origin and width.



*Holding down the **Option** key when you choose **Equalize Sidebearings**, gives this result.*

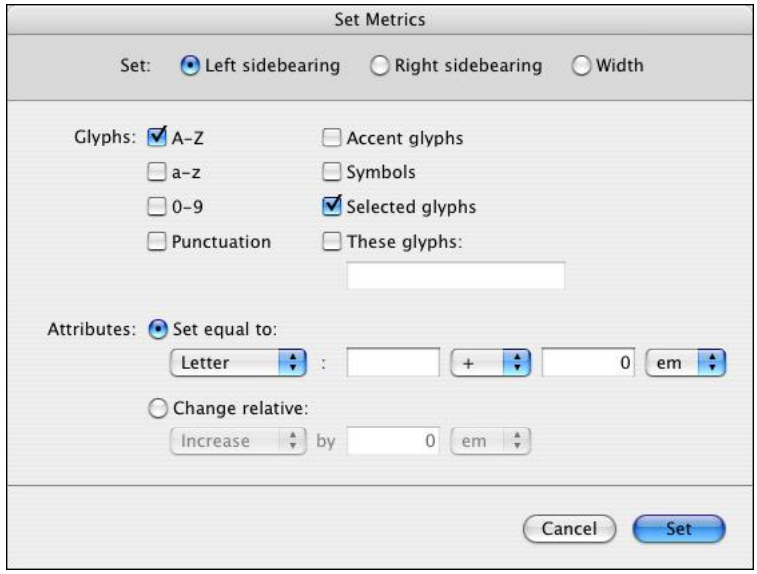
➲ ***Tip:*** If you have some points selected, Equalize Sidebearings will center just those points between the origin and the width, rather than the entire glyph. This makes for a useful centering command, but be aware that the glyph's sidebearings will probably not be equalized in this case.

Advanced metrics operations

Fontographer really tries to create a civilized shell for the rather technical problems posed by font metrics. The **Set Width** command, the metrics window, and the Easy modes of the **Auto Space** and **Auto Kerning** commands are examples of that. However, Fontographer also has an advanced metrics area, where we bring out the heavy-duty stuff. These commands are, by and large, not particularly intuitive, unless you already have a solid background in typography. The vast majority of our users will be able to do everything they need to do just splendidly by using the other tools. So we really advise most beginning users to ignore this. But if you insist, feel free to explore the cutting edge in computerized font metrics. Consider yourself forewarned.

Setting metrics

The **Set Metrics** command is similar to the **Set Width** command. In addition to the width, you can set left and right sidebearings as well. You can apply values to specific sets of glyphs, and there are lots of controls for incrementally increasing and decreasing various fields. Here is what the Set Metrics dialog box looks like:



At the top of the dialog box, you choose the characteristic you wish to modify: either the left or right sidebearing, or the width.

Next, you choose which glyphs you wish to apply the changes to. We have found that **Selected glyphs** is what we normally choose. So, you can go to the font window, select just those glyphs you want to adjust, and then choose only **Selected glyphs** in the **Set Metrics** command.

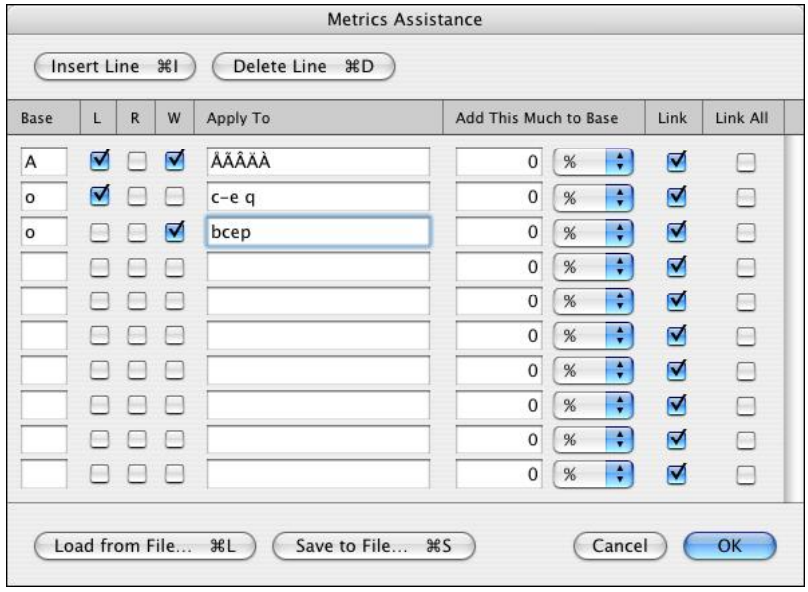
Finally, you choose what changes to make. The first line is for setting absolute values. You can set glyph's attributes equal to those of a different glyph, or equal to a specific value. You can then add or subtract either an absolute value or a percentage. For example, you could set the left sidebearings of Å, Ã, and Ä equal to the left sidebearing of A, plus five percent.

The second line is for relative changes. This line allows you to increase or decrease the selected attribute by either a number of em units, or by a specific percentage.

Note: You can destroy your font metrics in no time at all with **Set Metrics**. In software interface design, there is a tradeoff between really powerful commands and really safe ones. Since we're in the advanced section now, all the commands err on the side of being frighteningly powerful. For this reason, we advise you to save (or save as) your font before doing lots of **Set Metrics** commands, so you can revert if something doesn't work out.

Assisted metrics

Assisted metrics are halfway between completely manual metrics (set width and set metrics) and fully automatic metrics manipulation (auto space and auto kerning). Assisted metrics are for those typographers who don't quite trust the computer to do all their metrics creation for them, but don't want to have to do it all manually.

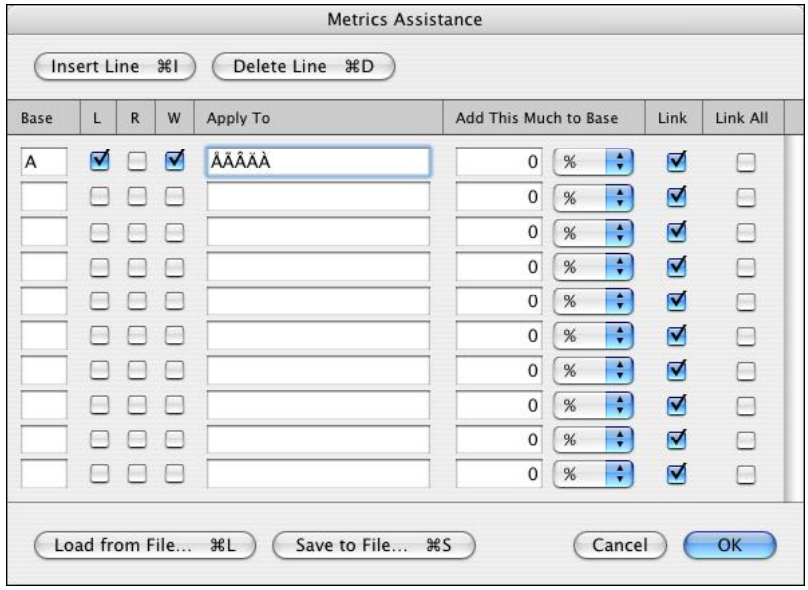


What we call assisted metrics, other people might refer to as equivalence classes. An equivalence class is a rule that dictates what other glyphs' attributes should be. These rules can involve sidebearings, widths, and kerning pairs. Equivalence classes are like programming languages for fonts. That's why not everybody likes them.

The best way to explain all this is by example, which we'll do now.

Metrics assistance

Metrics Assistance can be found under the **Metrics** menu. Here is how the Metrics Assistance dialog box looks, with one equivalence class:



Each line of this dialog box can have its own equivalence class. When you have entered more than ten classes, a scroll bar will appear along the right side of the box, which will allow you to create more classes. You can have lots of equivalence classes; they are limited only by available memory.

The **Insert Line** and **Delete Line** options allow for intuitive grouping when relating glyphs to one another.

In the left-hand column, you choose the base glyph. This is the glyph whose attributes will determine the values applied to the other members of that particular class.

Next, you choose what attributes the equivalence class is going to govern.

Then, you choose the other members in the class: these are the letters whose values are going to be set according to those of the base glyph.

Now you can optionally set up some difference (in value) to be applied to the base glyph's attributes. For instance, you could create an equivalence class that means certain glyphs' widths will be equal to a base glyph's width, plus 10 percent.

Next, you specify whether the class should be linked. If the **Link to base** checkbox is checked, whenever the values in the base glyph change, the corresponding values in all the other glyphs in that class will be automatically updated.

Finally, you can link everything together. This is similar to the **Link to base** checkbox, but **Link all** means that if any of the glyph's controlled attributes change, all other glyph's attributes will be automatically updated (including the base glyph).

Using these definitions, let's explore the sample equivalence class:

Our base glyph is "A". That means the attributes of the "A" will determine what the other glyph's attributes are. We chose to control both the left sidebearing and the width.

Note: You can control any one attribute, or any two attributes, but not all three: if you think about it, you can imagine that if the sidebearings are controlled, the overall width can't be because some of the glyphs have different widths. So all three of those characteristics can't be maintained. If you try to check all three options, one of the previously checked ones will become deselected.

The other members of the equivalence class are "Å, Ã, Â, Ä, and À". This is a fairly typical class. All the members of it are related rather obviously to the base glyph. Other common classes are ones that link "E" with "Ê, Ë, É", and so on.

There is nothing added to the base value. "A" is linked to all the letters "Å, Ã, Â, Ä, and À", but not vice versa.

When this class is applied, by accepting the dialog box, the left sidebearings and widths of "Å, Ã, Â, Ä, and À" will all be made equal to the left sidebearing and width of the base glyph "A". In addition, if you should ever edit the "A", either by changing its width or moving the outline around so that the left sidebearing changes, all those other glyphs will update automatically.

You can get pretty imaginative in figuring out which glyphs should be linked to which others. For instance, perhaps you want to link the left sidebearing of "B" to "D, E, F, H, I, K, L, M, N, P, R, and U". Many other glyphs share a similar right sidebearing.

Once you have set up all the equivalence classes imaginable, you probably won't want to do so again. That's why there is the big **Save to file** button.

Note: In order for Fontographer to be able to read the file in again when you want to use it next, the saved file must contain the suffix ".meq". This stands for metrics equivalents. This allows you to save off all the equivalence classes to a file. The **Load from file** button allows you to read them back in. This way, when you create a new font, you can use the equivalence classes you have already created, and gain a significant head start.

Of course, the real power of equivalence classes lies in the fact that once set up, you can change the values of the base glyphs. Therefore, if you really get into it, it is possible to set up enough classes so that you can completely determine the widths and sidebearings of the entire font by manually setting those characteristics for some small set of base glyphs.

Removing equivalence classes is quite straightforward: simply select the base glyph, delete it, and click **OK**. If you open the dialog box again, you will see that the class has been erased.

Sometimes you will see constructs like this: a d–m z in the **Apply to** these glyphs field. This happens when you had originally entered something like "adefghijklmz". Fontographer will detect contiguous glyphs and abbreviate them with a hyphen. This can frequently save some space, because sets like "abcdefghijklmnop qrstuvwxyz" appear much shorter as "a–z". If you want, you may also use the "a–z" convention when entering glyphs.

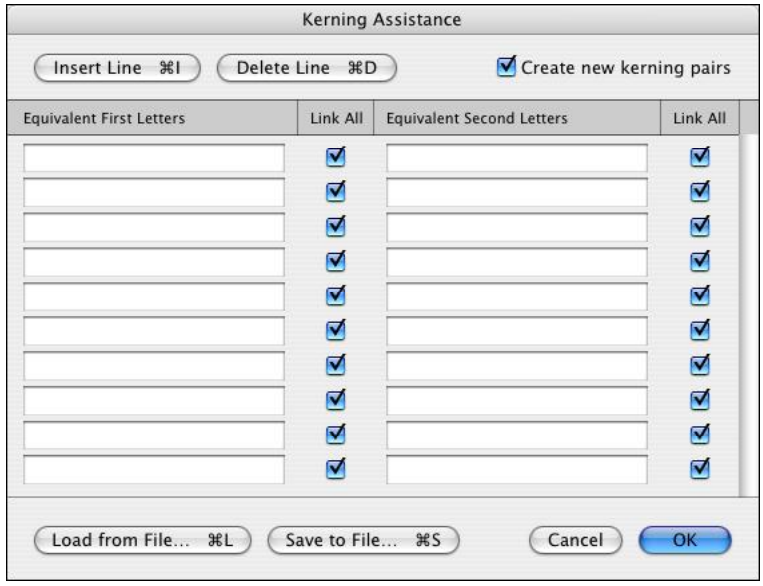
The same base glyph can be involved in different classes. For instance, one class might connect the left sidebearing of the "A" with a bunch of other glyphs, while another class could connect the width of the "A" to some different glyphs. If you need to connect "A" with more glyphs than will fit in the **Apply to** these glyphs field, it is perfectly legal to make another class, with the same base glyph and same characteristics, to continue the first class. Of course, base glyphs for some classes can be included in the **Apply to** these glyphs parts of other classes, but watch out – this can get confusing really fast.

If you include the base glyph in the **Apply to** these glyphs field of that same class, it will be recognized as a circular reference and automatically removed when the dialog box is accepted. If you set up mutually exclusive classes, which is legal to do but not recommended, the class furthest down in the dialog box will be the one that takes precedence.

Kerning assistance

Kerning assistance is a lot like metrics assistance and is best explained by example. However, if you haven't read about metrics assistance yet, you are strongly urged to do so before reading about kerning assistance.

Kerning Assistance is found in the **Metrics** menu. Here is how the Kerning Assistance dialog box looks, with a sample kerning equivalence class already entered:



Just like in the Metrics Assistance dialog box, the **Insert Line** and **Delete Line** commands allow for intuitive grouping when relating glyphs to one another.

The first column of equivalence classes are those glyphs that kern the same when they are the first character of a kern pair. (In other words, they will each get all the same kerning companions that the others have – for instance, if the font has the two kerning pairs "Te" and "Wy", it will have the four pairs "Te, Ty, We, and Wy" after the above class takes effect.) In this example, it has been declared that all kerning pairs that begin with "T" automatically have equivalent counterparts for "W", and so on. When more than ten classes have been entered, a scroll bar will appear to the right of the column to allow for the creation of additional classes.

The second column of classes are those characters that kern the same when they are the second character in a kerning pair. So in our example, any kerning pairs that end in "a" automatically have equivalent counterparts to those ending in "o", and vice-versa.

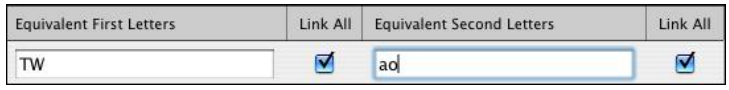
Note: Equivalent first letters are always connected to equivalent second letters.

The **Link all** checkboxes perform a function similar to what they do in metrics assistance. When these boxes are checked, any changes to kern pairs involving the members of that class will cause the other kerning pairs derived from the class to update automatically.

Here are some examples: You might want to follow along in Fontographer to get the most out of these explanations. We will begin with two kerning pairs that already exist –"To" and "Wa:"

Char	T	o	W	a
Width	618	575	1063	484
Left	1	47	15	39
Right	-2	43.454	-19	-28.02
Kern	0	To -150	oW 0	Wa -100

When we apply this set of kerning equivalence classes…



accepting the dialog box (by clicking **OK**), produces this:

Char	T	o	W	a
Width	618	575	1063	484
Left	1	47	15	39
Right	-2	43.454	-19	-28.02
Kern	0	To -150	oW 0	Wa -150

Fontographer has made the "To" and "Wa" kerning pair values equivalent. They are both now set at -150. In addition, Fontographer has also created the "Ta" and "Wo" kerning pairs below:

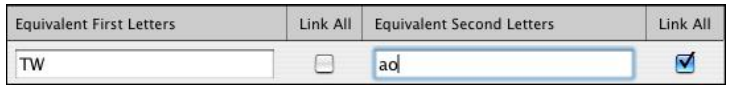
Char	T	a	W	o
Width	618	484	1063	575
Left	1	39	15	47
Right	-2	-28.02	-19	43.454
Kern	0	Ta -150	aW 0	Wo -150

In summary, we began with To = -150, Wa = -100. We ended with To = Wa = Ta = Wo = -150. Since the glyphs were all linked, they all became equal.

Next, let's explore what happens when we change the status of the **Link all** checkboxes. We'll start again from scratch, with just these two kerning pairs:

Char	T	o	W	a
Width	618	575	1063	484
Left	1	47	15	39
Right	-2	43.454	-19	-28.02
Kern	0	To -150	oW 0	Wa -100

Now, however, we will apply this slightly changed set of kerning equivalence classes:



The only difference is that the "T" and "W" are no longer linked together. This time, nothing has been changed for "To" or "Wa"; after all, "T" and "W" are not linked. The "a" and the "o" are, however, and so Fontographer has gone ahead and created the "Ta" and "Wo" pairs:

Char	T	a	W	o
Width	618	484	1063	575
Left	1	39	15	47
Right	-2	-28.02	-19	43.454
Kern	0	Ta -150	aW 0	Wo -100

Note that this time, the values of "Ta" and "Wo" are not the same.

In summary, we began with To = -150, Wa =-100. We ended with To = Ta = -150, and Wa = Wo = -100. This class sort of says anything-with-o will be equal to that-same-anything-with-a.

Now let's try one more permutation. We'll again start from scratch, with our trusty old "To" and "Wa" pairs:

Char	T	o	W	a
Width	618	575	1063	484
Left	1	47	15	39
Right	-2	43.454	-19	-28.02
Kern	0	To -150	oW 0	Wa -100

Now we'll unlink "a" and "o", but link "T" and "W". That gives us these two classes:

Equivalent First Letters	Link All	Equivalent Second Letters	Link All
TW	☑	ao	☐

When we apply this set of classes, it again does nothing to the "To" and "Wa" pairs. However, Fontographer did go ahead and create the "Ta" and "Wo" pairs:

Char	T	a	W	o
Width	618	484	1063	575
Left	1	39	15	47
Right	-2	-28.02	-19	43.454
Kern	0	Ta -100	all 0	Wo -150

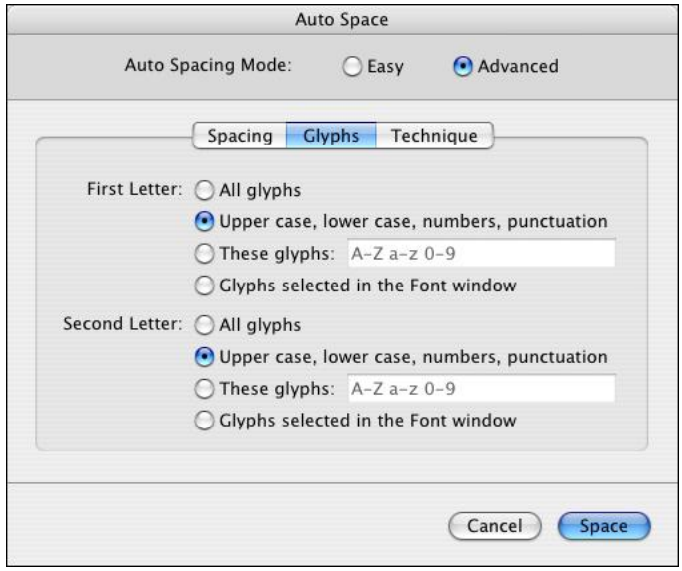
This time, however, the "Ta" got the value of -100 and "Wo" got -150. This is because the "T" and "W" are linked, but the "a" and "o" are not.

In summary, we began with To = -150, Wa = -100. We ended with To = Wo = -150, and Ta = Wa = -100. This class says essentially T-with-anything will always equal W-with-that-same-anything.

Advanced auto spacing

There is a lot more to auto spacing than merely what appears in the **Easy** mode. Switching to the **Advanced** mode causes a pop-up to appear at the top of the dialog box, which allows you to move through three screens of Auto Spacing controls.

The first dialog box is the Which glyphs dialog box.



The **Which glyphs** dialog box allows you to tell Fontographer which glyphs should get their widths set and which glyphs should be considered when choosing optimal widths. The **First letter** field defines which glyphs should get new widths. The **Second letter** field describes which companion glyphs Fontographer should consider.

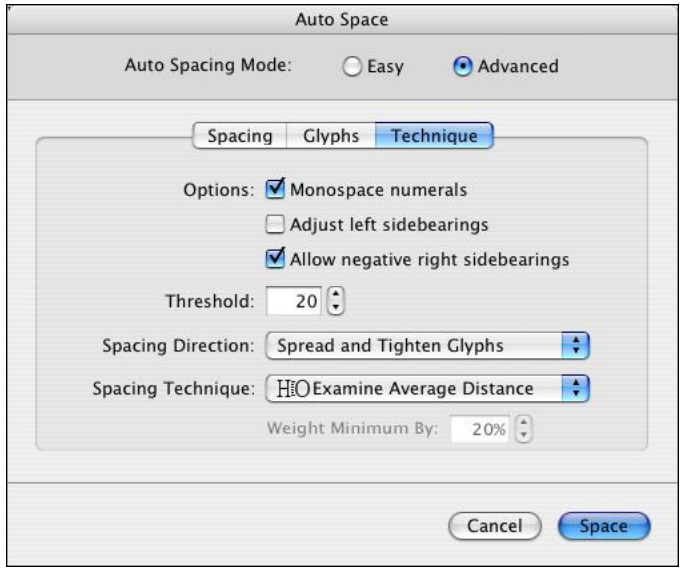
You may not want to have Fontographer set the widths of every glyph in the font. Suppose you have manually set the widths for the uppercase and lowercase glyphs. By selecting other glyphs in this field, you can have Fontographer space the rest of the font but not touch the glyphs you set by hand.

In addition, you might not want Fontographer to consider all the possible combinations of letters when determining optimal spacing. For instance, you might want to bias the spacing to favor the uppercase and lowercase letters.

By changing the selection in the **Second letter** field, you can optimize the spacing for the glyph combinations most likely to occur, and basically let Fontographer worry more about how “Th” is spaced instead of how “T‡” is spaced.

The second dialog box, the **How much** dialog box, determines the tightness or looseness of the spacing. This dialog box works just like it does in the Easy mode dialog box described earlier in this chapter.

The third dialog box is the **Technique** dialog box:



This dialog box allows you to customize some of the behavior of the **Auto Spacing** command.

Click **Monospace numerals** if you want all the numbers to have the same width.

Click **Adjust left sidebearings** if you want to let Fontographer (in its quest to achieve optimum spacing) move the character around relative to the origin. Some people are pretty adamant about where their glyphs sit relative to the origin, so they want Fontographer to leave them alone and do spacing solely by adjusting the widths, which is what happens if this box is left deselected.

Click **Allow negative right sidebearings** if you want to let Fontographer have portions of the glyph outlines extend to the right of the width. This will generally allow tighter spacing, but can in some cases cause certain glyphs to touch each other.

The **Threshold** field contains a value that influences the grouping of sets of glyph pairs Fontographer is considering. Threshold is a numerical value describing the distance necessary to define a kerning pair. This value is the kerning width used to group kerning pairs for spacing. By decreasing this value, fewer kerning pairs are included in the sample set used to determine optimum spacing. Having the value too small will cause only one or two values to be in each group. Having the value too large will cause most or all of the kerning pairs to be in each group. Neither of these will be helpful. Moderation is the key for determining the best value. By controlling the spacing of the sample set used to calculate kerning pairs, you influence the final result of your auto spacing. The useful range of this value is 10–80, with the best results being in the 20–30 range. Because this procedure is very complex, our advice is that if you feel the need to adjust this field, do so and see if you like it. If all of this sounds too complex then don't worry about this field – the default value will give you a good answer.

Note: The value for **Threshold** in auto spacing should probably be the same value used for **Don't kern a pair unless a kern of at least __ em units is needed** in auto kerning.

Spacing direction allows you to have Fontographer change spacing only by making glyphs narrower (tighten), only by making glyphs wider (spread), or by doing whatever is needed (spread and tighten).

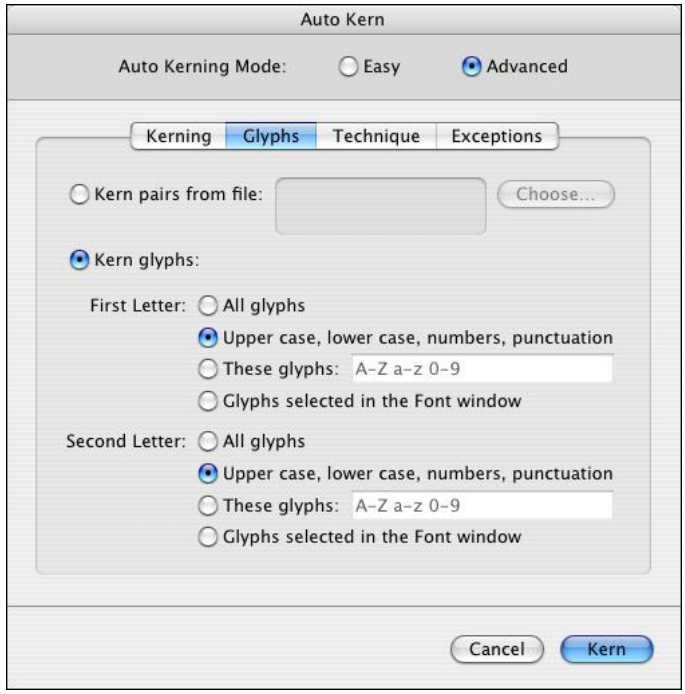
Spacing technique is another difficult-to-document feature. **Examine minimum distance** tells Fontographer to calculate spacing by looking only at the smallest distance two glyphs are from each other. For serif fonts, this can result in loose spacing because serifs frequently come close to touching each other. In this case, Fontographer will be essentially spacing the font by looking at the font's serifs and little else. **Examine average distance** and **Examine weighted distance** try to compensate for that problem by averaging the outlines a little bit – allowing the serifs to get closer if other parts of the glyph are further away. Of those two options, the weighted one is considered to be a more optical kind of comparison. Because fonts vary so widely, it is practically impossible to absolutely state the differences you will see with the various techniques. Our advice: try the different settings to see which one looks best to you.

Advanced auto kerning

Auto kerning also has a lot of hidden functionality. It is probably at its most powerful when applied to a font in several different ways. You can have auto kerning operate one way on some of the glyph set and another way for the rest, or you can use auto kerning in combination with manual kerning. Of course, you can always manually adjust the results of auto kerning as well. You can even use auto kerning as a diagnostic tool for your font. By auto kerning with different settings and then exporting the kerning and examining it, you can tell where the biggest spacing problems occur. Of course, anything with this many enhancements is going to take some practice and getting used to before you can learn to use it properly. But if you are really interested in kerning, it will be worthwhile.

When you choose the **Advanced** mode, a pop-up will appear that can navigate through four different auto kerning screens.

The first of these dialog boxes is called **Which glyphs**, and it looks something like this:



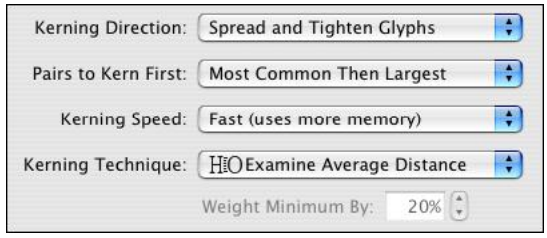
Use this dialog box to tell Fontographer which glyphs to auto kern. The first option, **Open file of pairs** allows you to select a text file of kerning pairs. Once you do that, Fontographer will do its normal auto kerning functions, but it will only create kerning pairs for the ones specified in that file. This is good for telling Fontographer exactly what you want done in the way of kerning.

The **Choose letters** option allows you to pick ranges of glyphs to kern. For instance, under First letter, you could simply enter "T" in the **These glyphs** field, and for the **Second letter** field, choose **All glyphs**. That would make Fontographer create only kerning pairs starting with "T". Or, it could be that you only want Fontographer to create kerning pairs among the most commonly used glyphs (in English, for example), in which case you would choose Upper case, lower case, numbers, and punctuation.

➲ ***Tip:*** Some people like to apply different kerning parameters to different parts of the font. You can fine-tune your results by invoking the **Auto Kerning** command many different times on the same font, but each time varying the kerning parameters and the glyphs to consider.

The next dialog box is the "How many and how much dialog box", and it is just like the **Easy** mode screen described earlier in this chapter.

The third dialog box is called **Technique**, and it looks like this:



The **Technique** dialog box allows you to choose different kerning techniques.

Kerning Direction is a way of having Fontographer create only negative kerning pairs (tighten), only positive kerning pairs (spread), or both (spread and tighten, which is the normal option).

Pairs to kern first is useful when you are controlling the total number of kerning pairs Fontographer is allowed to make (as specified in the How many and how much dialog box). If you have told Fontographer that it can only make 500 kerning pairs, for example, and Fontographer can find 2500 pairs that need kerning, Fontographer then needs a way to decide which 500 pairs to include.

✎ ***Note:*** If you tell Fontographer to create as many kerning pairs as it takes, this parameter doesn't matter; Fontographer will just include everything it finds. The four options are Doesn't matter, Most common pairs first, Largest pairs first, and Most common then largest.

However, if you choose Doesn't matter, Fontographer will simply choose the first 500 it finds.

Most common pairs first will cause Fontographer to give precedence to an internal list of common pairs, and output them first to make sure they are included. If you are really concerned about telling Fontographer which kerning pairs are important, choose the **Open file of pairs** option from the first dialog box.

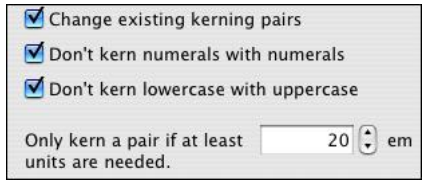
Largest pairs first will cause Fontographer to order the 2500 pairs it found from largest value to smallest, and output the 500 largest values. This is a "sure-fire" way to set the letter combinations that need kerning the most; however, you will find that many or most of these are often the goofy symbols glyphs, or punctuation glyphs (unless, of course, you told Fontographer not to consider those – which would have been smart – in the first screen).

The final option, **Most common then largest**, is what you will probably find yourself using the most. This option has Fontographer select the pairs it thinks are most common out of the 2500 found. When it has exhausted that set, it will generate the remaining pairs in order of magnitude. This way, you will get a lot of common pairs and also the ones that need kerning the most.

Kerning speed allows Fontographer to use less memory during auto kerning. Since auto kerning is already not blindingly fast, you are strongly encouraged to choose the Faster mode, and buy more memory if you need it.

Kerning technique tells Fontographer which internal algorithm it should use to compare glyphs. Examine minimum distance tells Fontographer to calculate kerning by looking only at the smallest distance two glyphs are from each other. For serif fonts, this can result in loose kerning because serifs frequently come very close to touching each other. In this case, Fontographer will be essentially kerning the font by looking at the font's serifs and little else. Examine average distance and Examine weighted distance try to compensate for that problem by averaging the outlines a little bit – it will allow the serifs to get closer if other parts of the glyph are further away. Of those two options, the weighted one is considered to be a more optical kind of comparison. Since fonts vary so drastically, and the sets of glyphs you can tell Fontographer to use for auto kerning are limitless, there is no meaningful way to document the differences in output that these various techniques will show. Our advice, as in auto spacing, is to try the different methods, and decide which ones give the best results for you and your fonts.

The final auto kerning dialog box is called **Exceptions**, and it looks like this:



Click **Change existing kerning pairs** if you want to let Fontographer, in the course of auto kerning, adjust kerning pairs that have already been created. Deselecting this checkbox locks the current pairs; if you go in the metrics window and manually set up a bunch of new pairs, you can now have Fontographer auto kern the rest, and be assured that it won't readjust any of the ones you created.

➲ ***Tip:*** You can also export kerning and experiment with it a bit.

Click **Don't kern numerals with numerals** if you don't want any kerning pairs created for the numerals. While number pairs often could benefit from kerning pairs, that will mess things up if the numbers ever have to appear in vertical columns: the columns of numbers won't line up perfectly if some of the numbers are involved in kerning pairs.

Check **Don't kern lowercase to uppercase** if you want Fontographer to skip all kerning pairs in which the first glyph is lowercase and the second glyph is uppercase. Having this checked is the recommended option, since those sorts of letter combinations almost never occur.

The bottom parameter is among those that exert the most influence on how many kerning pairs are created (unless you limit Fontographer to a specific number of pairs, which you can do in the **How many and how much** screen). This value tells Fontographer when to make kerning pairs and when not to. When Fontographer is considering a particular pair of letters, and the kerning amount Fontographer has decided that pair needs is greater than or equal to the reference value (20 in this case), then those two glyphs are made into a kerning pair. If the kerning value Fontographer came up with for those two letters is smaller than this value, then it is deemed a trivial kerning pair, and Fontographer will not create a kerning pair for those two letters.

You could sum up the preceding paragraph by saying that the size of every kerning pair Fontographer makes will be greater than or equal to the reference value you choose. Therefore, the larger the number you enter, the fewer kerning pairs will be created; the smaller the number, the more kerning pairs will be created.

Once you have done all the kerning you think your font needs, you can do auto kerning one last time, step up that value to 100 or 150 em units, and be assured that the most severely needed kern pairs will be created. For instance, you might want to have kerning pairs involving just the uppercase and lowercase glyphs, but you also might want to have 10 or 20 pairs involving the accent glyphs or the symbol set to ensure that the most severe cases are covered. In summary, by setting this value very high, you can have Fontographer find only the very largest pairs; conversely, you can set the value lower and have Fontographer find and create many more kerning pairs, involving smaller and smaller amounts.

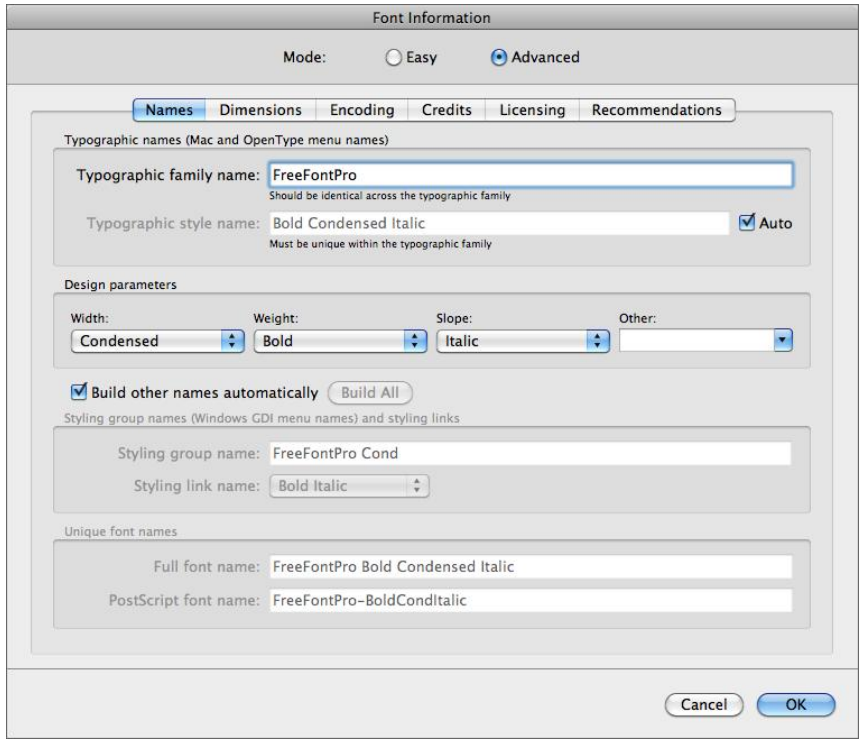
Font Info

The Font Info dialog box consists of five sections: names, dimensions, encoding, credits and licensing. Open the dialog by choosing **Font Info** from the **Element** menu or pressing **Command-Option-F**.

7

Names

If you need to see or edit all font names switch the Font Info dialog to the Advanced mode and select the **Names** tab if not yet selected.



The **Typographic family name** field is the main and the most important font name. Fill in this field at the first step. Do not use any special characters here.

If your font is condensed, bold, italic or have some other style parameters use pop-ups in the **Design parameters** section of the dialog. Choose appropriate width, weight and slope. For example, choose *Bold* as weight and *Italic* as slope if your font is bold-italic. If you want to add customization to your style name, then use the pop-up list **Other**.

The **Family name** and **Design parameters** are enough for Fontographer to build all other names automatically.

You will see the **Typographic style name** changes as you choose different design parameters. Switching the **Auto** option off allows you to customize the style name so you can call your bold italic font *Fat Slanted* or something else.

You may leave **Build other names automatically** turned on to allow Fontographer to build all other necessary names for you. If for some reason you need to change those names manually then you may switch this option off. Click on the **Build All** button at any time to let Fontographer build names automatically.

Name fields

You can alter the following font naming fields in Fontographer's Font Info dialog:

"Typographic family name" (TFN)

The **typographic family name (TFN)** is the name under which all fonts from one family are grouped in the font menu of typographically-aware applications (such as modern Adobe applications or most applications running on Mac OS X). It is sometimes called “Preferred family name”. The TFN is always determined by the font vendor, and is the first (out of three) key source of information, from which other font naming fields are built.

The TFN may consist of uppercase and lowercase English letters, spaces, numerals and has a length limit of 31 characters. The TFN may contain spaces but some vendors choose not to use any spaces here.

The TFN must be identical in all fonts that should appear as one family in the font menu of OpenType-aware and Mac applications.

"Typographic style name" (TSN)

The **typographic style name (TSN)** is the name under which a certain member of a font family appears in the font menu of typographically-aware applications (such as modern Adobe applications or most applications running on Mac OS X). It is sometimes called “Preferred style name”. The TSN is always determined by the font vendor, and is the second key source of information, from which other naming fields are built. Typically, this field describes the typographic properties of the font within the typographic family. The specific information contained here usually consists of style words that describe typographic parameters such as width, weight, slope, etc.

The TSN may consist of uppercase and lowercase English letters, spaces, numerals and has a length limit of 31 characters.

“Styling group name” (SGN)

The **styling group name (SGN)** is the name that appears in Windows GDI applications as the family name. It is sometimes called “Windows family name” or “Microsoft menu name”. All fonts that are associated with each other through styling links form a styling group.

The typographic family must be divided into styling groups, each having no more than four members which all must be connected by styling links with each other. Each styling group within a typographic family must have a unique SGN.

Within one typographic family, there must be exactly one default styling group, which must have the SGN identical to the TFN. The SGNs for the remaining styling groups are built automatically from the three key fields (TFN, TSN).

The SGN may consist of uppercase and lowercase English letters, spaces, numerals and has a length limit of 31 characters.

“Styling link name” (SLN)

The **styling link name (SLN)** is the name that appears in Windows GDI applications as the style name. It is sometimes called “Windows style name”. Only “Regular”, “Italic”, “Bold”, “Bold Italic” are permitted here (exactly as spelled). No other values are permitted.

“Full font name” (FFN)

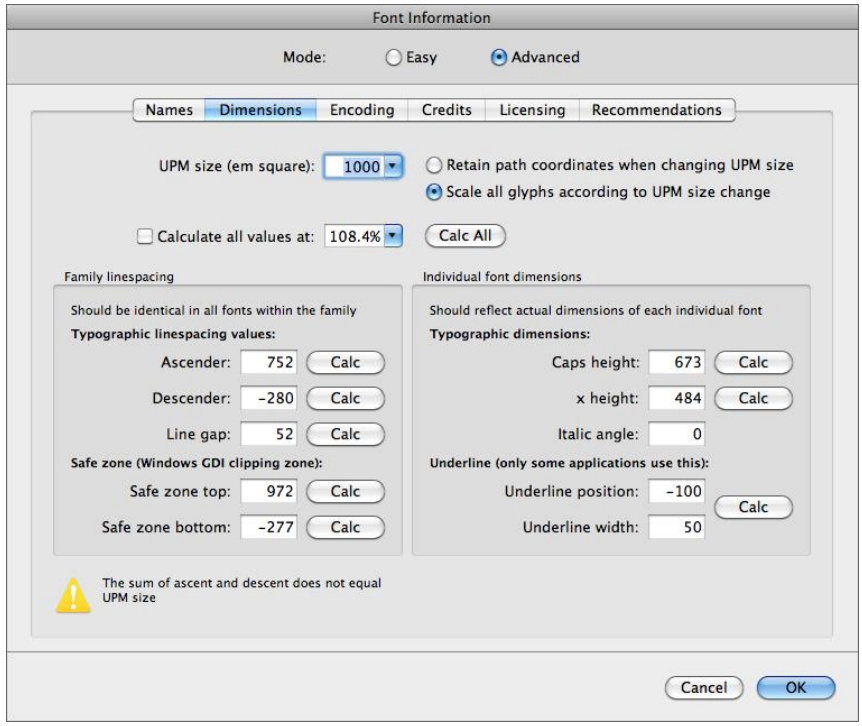
The **full font name (FFN)** is the field that is sometimes used by applications to refer to a font by its name (for example when storing information about fonts used in a document), and is sometimes displayed to the user (mostly by font viewing or managing applications).

The FFN may consist of uppercase and lowercase English letters, spaces, numerals and has a length limit of 63 characters. It is usually autogenerated by concatenating TFN and TSN separated by a space.

“PostScript font name” (PSN)

The **PostScript font name (PSN)** is the name usually used by PostScript print drivers to reference the font. The PSN may consist of uppercase and lowercase English letters and optionally one hyphen that separates the family name part from the style name part. The length limit is 29 characters. The PSN is usually autogenerated by concatenating TFN and TSN separated by a hyphen with all spaces removed.

Dimensions



All the numbers in this dialog box are in font units (UPM) – the fundamental parameter that influences point size, linespacing and the overall proportions of the font. The **Ascender** and **Descender** usually default to a sum of 1000 units for a PostScript font, or 2048 for a TrueType font. If they do not then the note is shown. This is not an error but just a warning. Other values are also allowed.

➲ ***Tip:*** Line gap is normally left at 0; but if you must set it, a good setting is 20% of the sum of the ascender and descender.

Safe zone top and **bottom** values correspond to OS/2 Win Ascender and OS/2 Win Descender accordingly.

Ascender, **Descender**, **Line gap**, **Safe zone top** and **Safe zone bottom** values should be kept consistent across an entire font family.

Caps Height, **x-height** and **Italic angle** should reflect actual dimensions of each individual font face in the family.

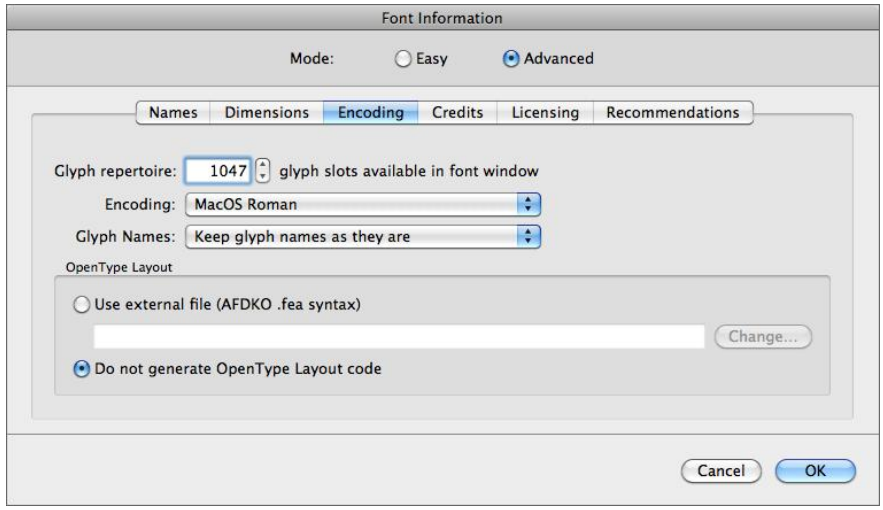
Since a PostScript underline is accomplished by defining a stroked line a certain distance from the bottom of the character, the **Underline position** entry allows you to change the distance between this stroked underline and the characters above it. The default indicates that this line will be drawn below the baseline. The **Underline width** indicates how wide the underline should be; the default is 20 units. Unfortunately there is no applications that use these settings for underline.

You would deselect the **Calculate all values at** if you want to manually change vertical metrics. Also you can click on individual **Calc** buttons to let Fontographer calculate the corresponding value for you.

With the **Retain path coordinates when changing UPM size** checked, the entire font will look smaller if you make the UPM size larger.

Encoding

The **Encoding** pop-up allows you reorganize the layout of the font window to display the key positions of the encoding you selected. It also defines the order of characters in the exported Type 1 fonts. In the **Encoding** pop-up, you have several choices. If you're creating OpenType fonts then one of the OpenType encodings should be used. If you're creating Type 1 fonts for Western languages (English, German, French, Spanish etc.), then in ninety-five percent of the cases, you'll want to use the MacOS Roman encoding option.



OpenType Layout options allow you to set your preferences in generating OpenType layout features for the particular font. You can override this later in the Format Options dialog box when generating the font (refer to Chapter 8, "Generating and Exporting Fonts").

Encoding options

The encoding of the font is the ordering of its characters. Version 5 of Fontographer has one dialog box that allows you to set the encoding. The **Encoding** pop-up in the Font Info>Encoding dialog box will reorganize the layout of the font window to display the key positions of the encoding you selected. On the other hand, it also defines the order of characters in the file when you select the **Generate Font Files** command in the **File** menu. Fontographer will output the font to match the encoding you select.

The most commonly used encodings for the Macintosh Type 1 fonts are MacOS Roman and Custom. The most commonly used encodings for the Windows Type 1 fonts are MS Windows 1252 Western (ANSI) and Custom.

➲ ***Tip:*** If you choose MS Windows 1252 Western (ANSI) encoding, be sure not to use slots 0–31. Slot 32 is reserved for the space character, so don't put a glyph there, or in 127, 128, 129, 141, 142, 143, 144, 158, 159, or 160.

Custom encoding allows for the custom naming of glyphs. The Glyph Information dialog box (which is opened by selecting a glyph and choosing **Selection Info** from the **Element** menu) contains a name field that allows you to change the glyph name. Custom encoding becomes the current encoding option any time you change a glyph name. Custom glyph naming is widely used by people who design non-Roman fonts.

Original encoding is like an elephant; it never forgets the encoding of the font at the time it was first opened in Fontographer. This is a very handy way to get back to where you began – wherever that was. This can be useful when opening fonts with strange encodings such as Sonata, Carta, or Zapf Dingbats.

Type 1 Adobe Standard encoding (ASE) is Adobe's default encoding for Type 1 fonts.

The set of OpenType encodings can be used when building OpenType TT and OpenType PS fonts. Select the one which better fits to your needs. And you can change it to another one at any time.

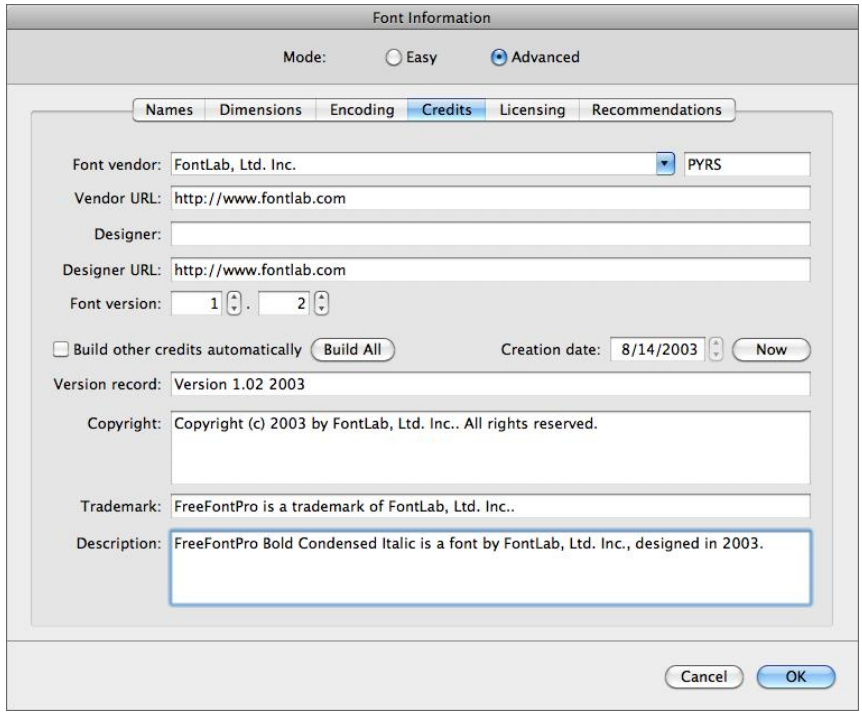
If you need to create your own encodings or change an existing one please see "Adding custom encodings" in Chapter 13, "Expert Advice".

If you need more information about encoding vectors, refer to:

http://partners.adobe.com/public/developer/opentype/index_char_sets.html

Credits

On the **Credits** page you can enter information about the creators of the font, font version, creation date. If you have created a new font you should enter your copyright data here. If you have edited an existing font that was not your creation you must not remove the information contained on this page, or you may violate copyright laws.



Font vendor	Human readable name of the company or person that created the font. If you created a new font enter your name or the name of your company here. All registered vendor names are placed in the drop-down list box from **vendors.dat** file
Font vendor ID	An up-to-four letter length ID that is assigned to most font producers to identify their fonts. An uppercase vendor ID must be registered with Microsoft or Apple. If you want to identify yourself without registering you may enter a lowercase four-letter vendor ID
Vendor URL	The www link to the site of the font vendor
Designer	Name of font designer
Designer URL	The www link to the site of font designer
Creation date	The date of font creation. It is set to today if you just created the font
Font version	Version and revision of the font
Version record	Font version records have a different format. You may enter the version record here or just press the **Build other credits** button to fill this record automatically
Copyright	Copyright message. Must include the © sign or the word "Copyright", the name of the company or person that owns the copyright and the copyright year
Trademark	Font trademark – used to save font's trademark notice
Description	Additional information that you want to include in Font Info

Vendor.dat File

Fontographer stores information about registered vendors in the *vendor.dat* file located in the [**Shared default data folder]/Data/** folder (typically, Macintosh HD/Library/Application Support/FontLab/Data/). This is a text file with a simple structure:

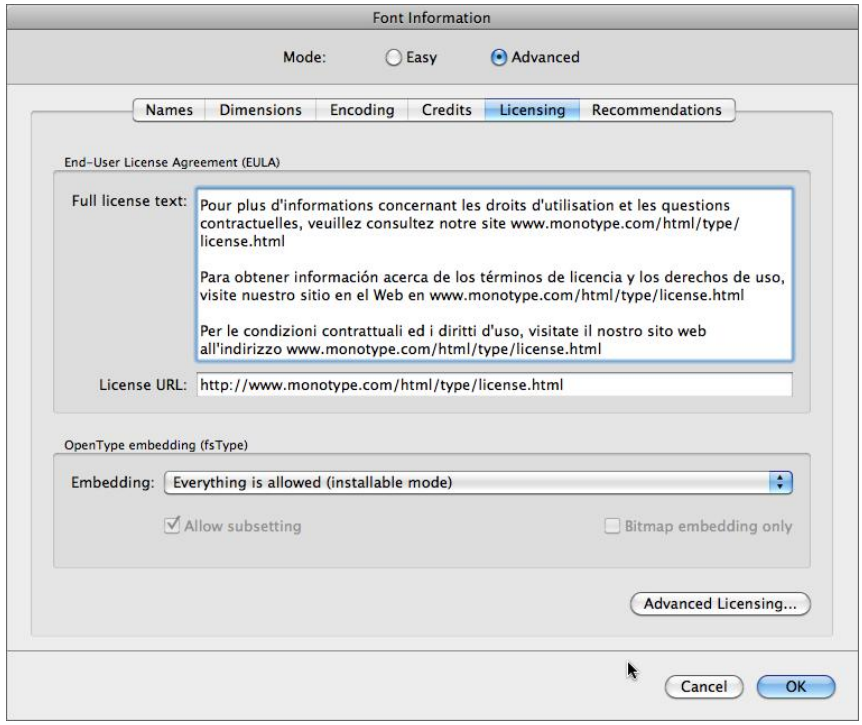
```
2REB 2Rebels
39BC Finley's Barcode Fonts
3ip Three Islands Press
918 RavenType
```

As you can see, it is just a vendor ID followed by vendor name. A single space is used as a separator.

If you want to change the file or add a new entry, just open it in any text editor (such as TextEdit) and make changes.

Licensing

The **Licensing** page contains the End-User License Agreement (EULA) information and the embedding information.



Full license text contains text information about how the font can be used. **License URL** is the link where additional license information can be found.

This **Embedding** popup menu controls how the font may be embedded into documents. Embedding is a feature of the operating system and some applications that allow programs to include fonts into documents (PDF, for example) to guarantee that they will be reproduced correctly. However, this feature may cause problems with font piracy. It is not very hard to extract embedded fonts from a document, so the TrueType font format includes a special setting that can control font embedding.

There are four types of font embedding:

Everything is allowed	After the document is opened the font works as if it was installed in the system
Embedding is not allowed	Embedding is not allowed for this font
Only printing and previewing is allowed	The font may be embedded, but editing of the document it contains is not allowed
Editing of the document is allowed	The font may be embedded and the document that contains the font may be viewed, printed and edited.

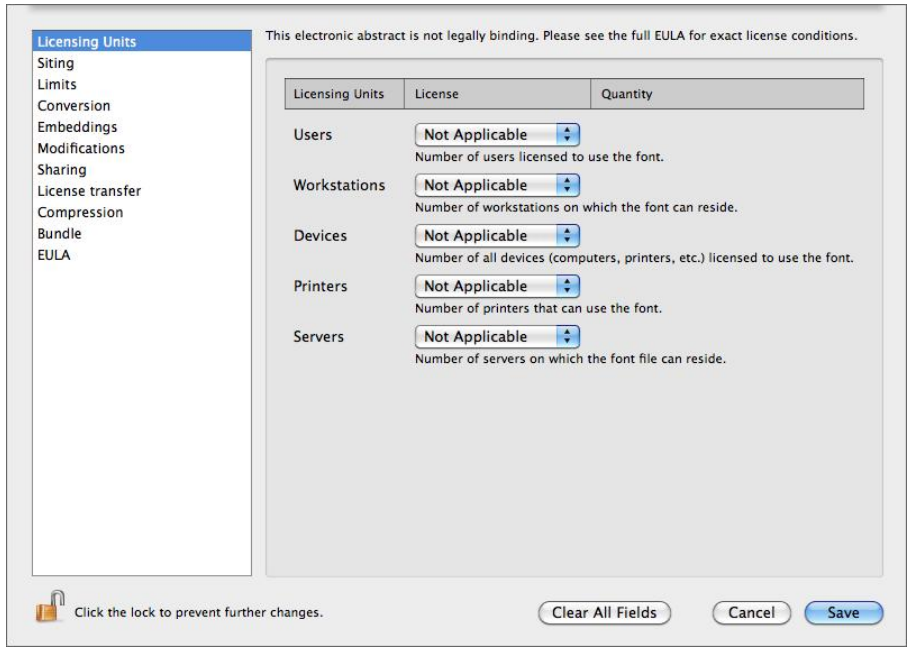
Additional options **Allow subsetting** and **Bitmap embedding only** are available.

Copyright Note

We decided to allow modification of the embedding setting only because we are sure that the users of Fontographer are professionals who respect others' rights to intellectual property. We assume that you will change the embedding setting only in your own fonts. You are not allowed to change this setting in fonts that were created by somebody else. Even if according to the font license you can modify the font for your own use, you must not “increase” the embedding rights for a font. So if embedding is not allowed leave it as it is.

Advanced Licensing

Clicking on the **Advanced Licensing** button will open an additional dialog where the electronic EULA of the font can be edited.



Electronic EULA Abstract (EEULAA) is an extra font table (EPAR table) containing a computerized form of the EULA. From this table users can know what they can do with a font without the pain of interpreting the legalese in its official EULA. One way of thinking about the EULA abstract is as a set of Frequently Asked Questions for font licenses, for example:

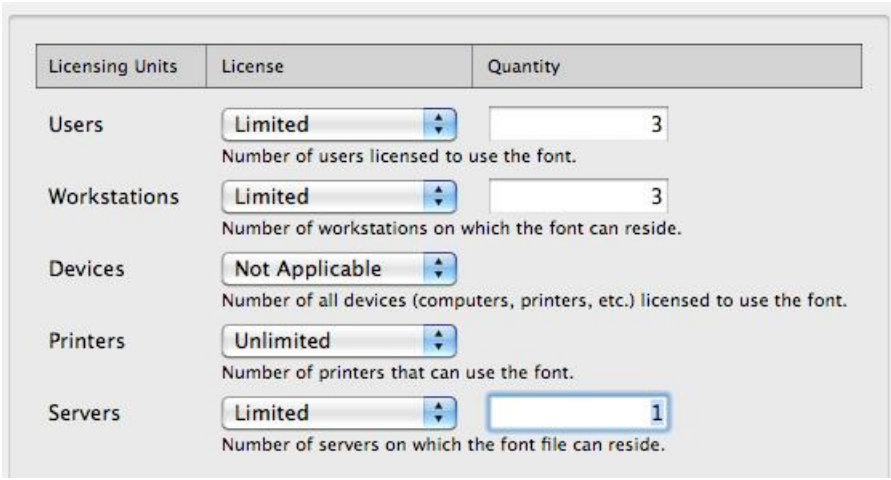
- Can I embed this font in a PDF?
- Can I send this font to a service bureau for printing my documents?
- How many computers can I install the font on?

Once a EEULAA table is created for a font you can protect it from unauthorized changes by adding password protection. To set or change the password, click on the **Change** button in the **EULA** section. Then click on the **Lock** in the lower left corner to prevent further editing:

Once the password is set and the dialog is locked Fontographer will display the contents of the EEULAA but will only save changes when the correct password is given.

You can find more information about EEULAA on http://www.eeulaa.org/.

Licensing Units



This section documents how a license is tied to something physical – people, places and/or devices. Licenses may be tied to individual person(s) and/or particular devices.

Users: A license tied to users means that only the specified users or number of users are licensed to use the font.

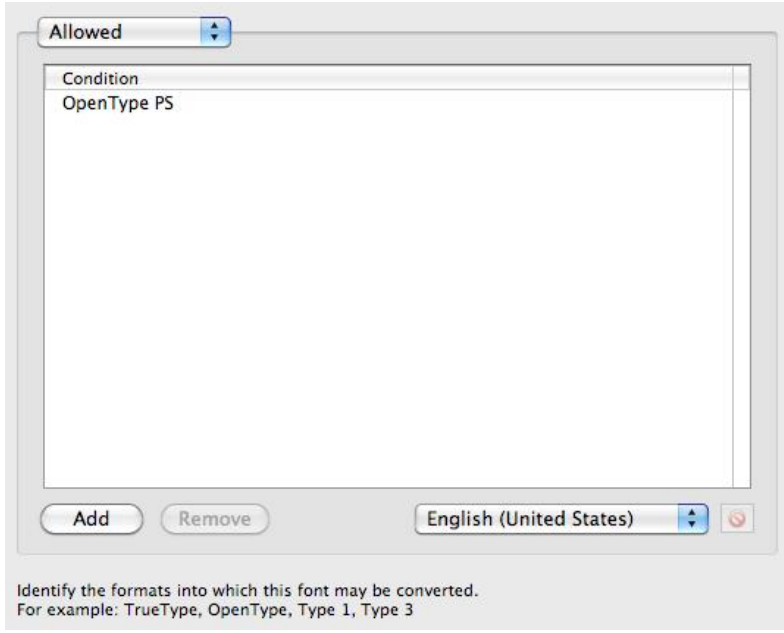
Workstations: A license tied to workstations means that the font may be used only on the specified workstations or number of workstations.

Devices: A license tied to devices means that the font may be used only on the specified devices or number of devices. "Devices" usually refers to computers, workstations, printers, servers, etc.

Printers: A license tied to printers means that the font may be used only on the specified printers or number of printers. This is rarely used anymore since modern fonts in general are not downloaded to printers.

Servers: A license tied to servers means that the font may be used only on the specified server computers or number of servers.

Conversion



Licenses may allow conversion of the font file. This can take several forms:

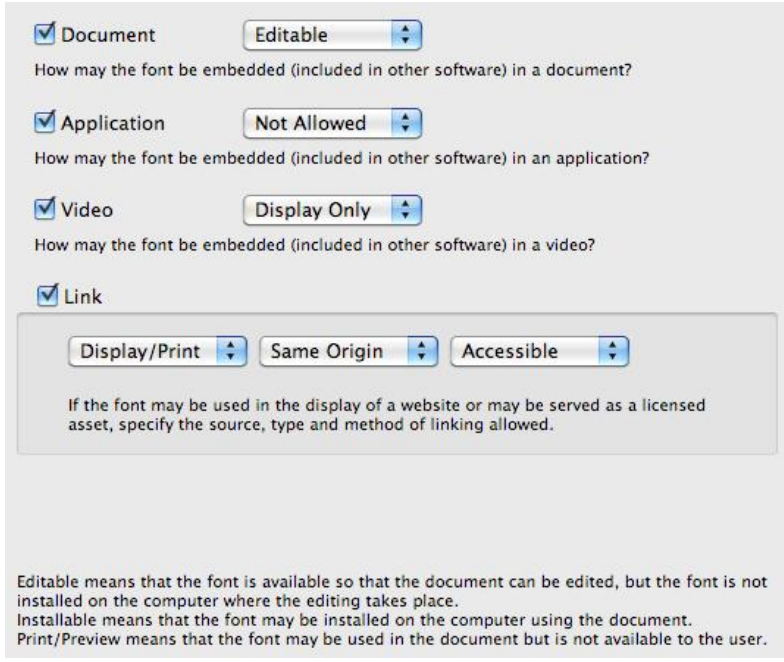
1. Format conversion – conversion between vector font formats such as Type 1, TrueType and OpenType.

2. Platform conversion – conversion to a platform-specific format.

3. Raster conversion – conversion between bitmap and vector font forms.

Since the possible types of conversion are virtually infinite this is a text field. The field may contain one of the following values: "All", "None", "Allowed-followed by a list of allowed conversions", "Disallowed-followed by a list of non-allowed conversion types".

Embeddings



Document Embedding means to incorporate the font into a document so that the font travels with the document. These are the same as the original document embedding bits. **Not Allowed** – the font may not travel with the document. **Installable** –the font may be installed and used on whatever device the document is on. **Editable** – means the font may be used in editing the document, but may not be installed on any device. **Print/Preview** – means the font may be used only to print a document or display on a device.

Application Embedding means to incorporate the font into a computer program/application so that the font travels with the program. **Not Allowed** – the font may not travel with the program/application. **Installable** – means the font may be installed and used on whatever device the application/program is on. **Editable** – means the font may be used in editing text in the course of running the application/program, but may not be installed on any device. **Print/Display** – means the font may be used only to print a document or display on a device in the course or running the application/program.

Video Embedding means to incorporate the font into a video so that the font travels with the video. **Not Allowed** – the font may not travel with the video. **Installable** – means the font may be installed and used on whatever device the video is on. **Editable** – means the font may be used in editing the video, but may not be installed on any device. **Display** – means the font may be used only to display text on a device.

Link: If the license allows the use of a font in the display of a website or to be served as a licensed asset, then this field specifies the source of linking allowed followed by the type and method.

1. The sources are: same-origin, root-string, and sub-licensee.

2. The types are display/print, edit/print.

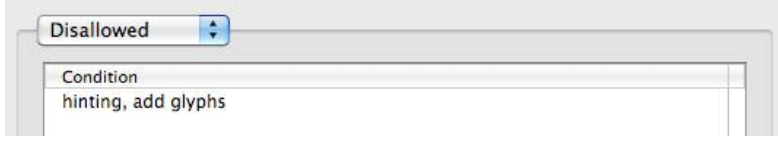
3. The methods are "Accessible" (straight @font-face call) or "Protected" (obfuscated call). E.g. root-string, display/print, encrypted, etc.)

Modifications

Font users may need additions or other modifications to a font for their uses. The license may or may not allow this or may allow it only under certain circumstances (e.g. that the modifications be done by the original designer).

1. This field excludes modifications such as conversion, compression, and subsetting, which are covered elsewhere.

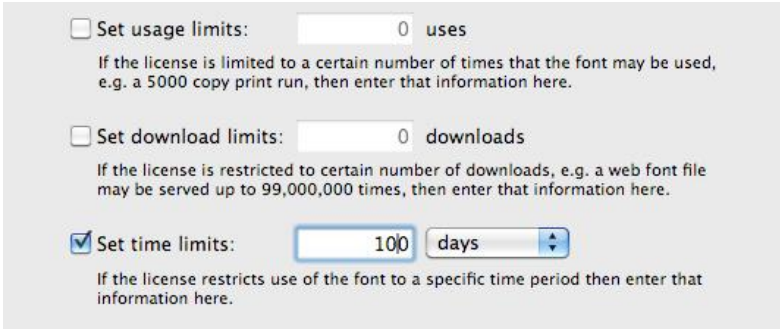
2. Possible kinds of modifications include glyph editing; adding new glyphs; changing metrics; hinting, etc. but are virtually unlimited. Therefore this is a text field with possible values of "All", "None", "Allowed-with a list of allowed modifications", "Disallowed-with a list of non-allowed modifications". Example:



Sharing

If a font may be temporarily shared with a service bureau then this field gives the conditions.

Meter



Licenses tied to time or events:

1. Pay-per-Use: A license for a specific or counted number of uses of the font.

2. Pay-per-Download: A license for a specific or counted number of downloads of the font.

3. Pay-per-Time: A license for a specific or counted number of time units.

License Transfer

A licensee may want to sell or give his font to someone else. The license may or may not allow this or may allow it under certain circumstances (for instance, that the change in license be registered with the vendor).

Site

Licenses may be tied to a specific physical location.

Location: A site license means that anyone on the site (department, building, campus - some contiguous area) is licensed to use the font.

Company: A company license means that anyone in the company (regardless of location) may use the font.

Third party: Where one party licenses the font for use by another party. The "third party" could be an individual or a group of people.

Network: means that anyone on a LAN, WAN, or the WWW may use the font.

If the license is a site license then this field contains the name/address/ description of the site.

Compression

Licenses may allow compression of the font file.

Subsetting: glyphs are removed from the font.

Algorithmic data compression: The font file data is rewritten in a different form.

Since the possible types of compression are virtually infinite this is a text field. The field may contain one of the following values: "All", "None", "Allowed-followed by a list of allowed compressions", "Disallowed-followed by a list of non-allowed compression types".

Bundle

Licenses may be tied to use or sale with specific software or hardware. A font may be bundled with other fonts, other programs or with computer hardware. If this is so then this field contains the name and/or details of the bundle.

EULA

EULA URL: The EEULAA can only give a summary of the most important parts of a EULA. In order to give the licensee access to the complete (and legally binding) EULA text the EULA should be accessible on the WWW and/or within the font. This field is for the URL where the full EULA is located.

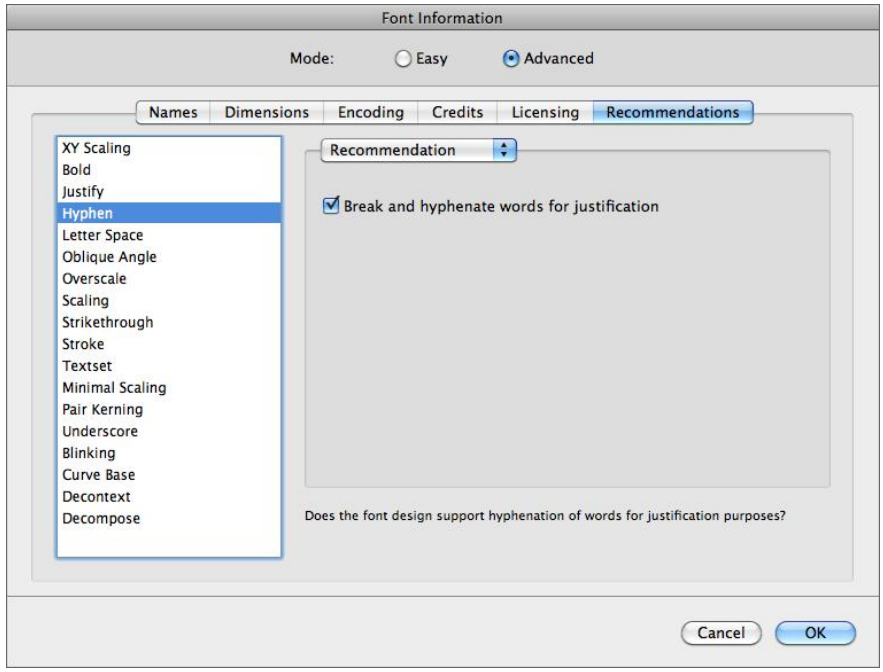
Upgrade URL: If a licensee finds that their license is inadequate for their purposes then they should have a way to contact the vendor and arrange for a license upgrade (e.g. more users, more locations, more embedding, etc.) that will allow them to use the font as needed. This field is for the URL of the license upgrade web page or equivalent.

Remote URL: The EPAR can also be accessed on a remote server. This field is for the URL/file where the remote EPAR is located. This is particularly useful for large licenses where many copies would need to be updated if there was a license change. Rather than read the EPAR data from the local table the application would read it from the remote EPAR file maintained by the vendor.

EEULAA Password: Click on the **Change** button to set or change the password. Then click on the **Lock** in the lower left corner to prevent further editing.

Recommendations

The **Recommendations** page allows you to edit the second part of the EPAR (Embedding Permissions and Recommendations) font table.



In order to use some fonts correctly some users need information about the conditions for which a font was designed. The EPAR table will contain this information covering a wide variety of ways in which a font might be used. These are optional, of course, so a font designer may choose not to make recommendations for some situations. Below is a list of definitions of the data fields for the recommendations section of the EPAR.

Parameter	Value type	Description
XY Scaling	**numeric range**	Recommendations for use of the font when scaling either x or y alone expressed as min/max % of em. Only one number (X)), is required because the value represents the range from 100% to X%.
Bold	**numeric percent**	Recommendations for bolding of the font expressed in % of em
Justify	**binary Y/N**	This is a binary recommendation, where the font either should or should not have inter letter space added to fit a line to a column width, for reasons in the design of the font

Hyphen	**binary Y/N**	This is a binary recommendation, where the font either should or should not have words broken at line ends for reasons of the design of the font.
Letter Space	**numeric percent range**	Recommendations for the positive and negative addition of space between letters in percent of em
Oblique Angle	**numeric range of angles**	Recommendations for the positive or negative angles for slanting the font
Overscale	**numeric %em, upm, pixel#**	Recommendations for minimal scaling of the font: smallest critical part (by % of em), largest upm per pixel, smallest number of x-ht pixels
Scaling	**numeric pts. pts, pts, pts; pxls, pxls, pxls, pxls**	Recommendations for optimal scaling of the font: in points: text optimal, text minimal, display optimal, display minimal; and in pixels: text optimal, text minimal, display optimal point, display minimal
Strike	**alphanumeric Y/N**	Recommendations for use of the font with an automatically generated stroke through the glyphs
Stroke	**numeric percent**	Recommendation for definition of the font's contours as other than zero stroke width: maximum stroke value in % of upm
Textset	**alphanumeric number**	Recommendation for Ideal length of a line of text in this font, in number of words
Minimal Scaling	**numeric pxls, pxls, pxls**	Recommendations for use of the glyph table's TrueType instructions. Minimum pixels per em with: no instructions, no x instructions, no y instructions
Pair Kerning	**binary Y/N**	Recommendation for use of the font with no pairs kerned
Underscore	**binary Y/N**	Recommendation for use of the font with an automatically generated stroke below the glyphs
Blinking	**binary Y/N**	Recommendation for use of the font for blinking
Curve Base	**alphanumeric text**	Recommendations for use of the font with curved baseline
Decontext	**alphanumeric text**	Recommendations for use of the font without feature tables regarding glyph substitution
Decompose	**alphanumeric text**	Recommendations for use of the font with decomposited glyphs

Exporting EPAR

The licensing and recommendation info is saved in your fog database file but you also can export it separately.

To export an EPAR file:

1. Select **EPAR** from the **Export** pop-up in the **File** menu.
2. Once you've named the file and chosen the folder to export the file to, press the **Save** button and the file will be exported without further comment.

The main reason of exporting EPAR information is applying it to other fonts later.

To import and apply exported EPAR information to your font:

1. Select **EPAR** from the **Import** pop-up in the **File** menu.
2. Find and select the file with epar extension, press the **Open** button and the file contents will be applied to your current open font.

Choose **Element > Font Info > Licensing > Advanced Licensing** and observe changes.

Printing

You've just created a font, or perhaps just a few glyphs, and now you would like to see those glyphs in various point sizes and kerning combinations. Maybe you created a large Do Not Disturb sign for your office door and want to print it now before you actually go through the font's installation procedures. You can print a sample of your choice any time your font is open, from any of its windows.

8

Fontographer gives you the choice of printing a variety of samples:

- All glyphs in the font
- Individual glyphs
- Specific typed text
- Text from a file
- A PostScript sample with several lines of text
- A key map, showing all the glyphs in the font and their respective widths, codes, and offset specifications
- A complete list of all the kerning pairs in the font
- A glyph sample in assorted sizes or just one giant glyph
- A sample showing all of the points in a selected glyph, and optionally, the x/y coordinates for each point.

Before you go to the Print Sample dialog box, it is important to decide which printer you want to use. Fontographer will print to both PostScript and non-PostScript printers, but the quality of output will vary. If you are printing to a PostScript printer, Fontographer hints the font before downloading it to the printer. This method will give you the highest quality prints. If you are printing to a non-PostScript printer, Fontographer must draw each glyph unhinted, so the quality may be slightly lower.

It should be noted that when Fontographer's print samples are output to a non-PostScript printer, they only show an approximation of what the font will look like when actually installed. Subtle variations will appear depending on the type of font generated (Type 1, Type 3, TrueType, and so on).

Fontographer's **Print** command in the **File** menu provides several options for printing font samples, most of which give you the opportunity to choose the point size of your printed sample.

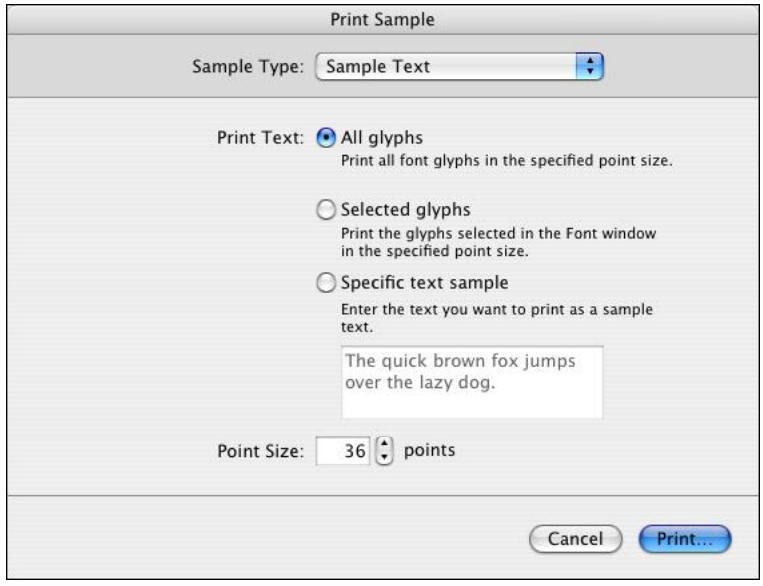
Sample text

The first choice in the **Sample type** pop-up is Sample text, which provides three printing options.

Choosing **All glyphs** prints all the glyphs of the font at the point size you designate.

To print a sample of all glyphs:

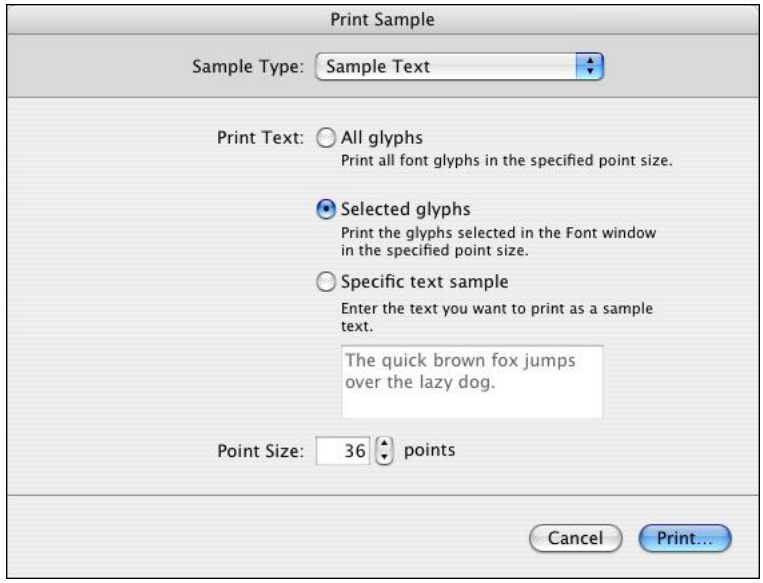
1. Choose **Print** from the **File** menu.
2. Choose Sample text from the **Sample type** pop-up.
3. Click the **All glyphs** radio button in the Print Sample dialog box.
4. Click **Print**.



Choosing **Selected glyphs** allows you to print glyphs selected in the Font window at a designated point size.

To print selected glyphs:

1. In the Font window, select the glyphs you wish to print.
2. Choose **Print** from the **File** menu.
3. Choose Sample text from the **Sample type** pop-up.

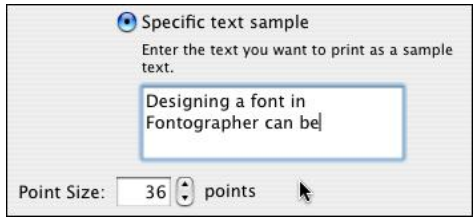


4. Click the **Selected glyphs** radio button in the Print Sample dialog box.
5. Click **Print**.

To print specific text samples:

1. Choose **Print** from the **File** menu.
2. Choose Sample text from the **Sample type** pop-up.
3. Click the lowest radio button in the Print Sample dialog box.

   (The lowest radio button is located above the text sample box.)



4. Type your text into the text box.

   Fontographer allows you to type up up to 256 glyphs in this box. If you press the RETURN or ENTER key while you're entering text, you will begin the printing process. However, text containing carriage returns can be pasted into this text box.

5. Click **Print**.

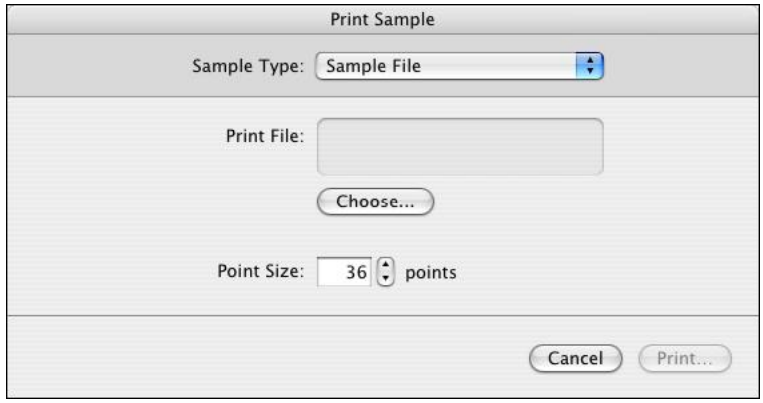
Note: The text box is a good place to test logos, kerning pairs, or glyph spacing in specific word and sentence combinations.

Sample file

Printing a sample file allows you to print the contents of a text file using the current font at any given point size.

To print a sample file:

1. Choose **Print** from the **File** menu.
2. Select Sample file from the **Sample type** menu.



3. Enter the desired point size or use the default setting of 36 points.
4. Click **Print**.

   At this time Fontographer asks you to select the text file that you would like to print.

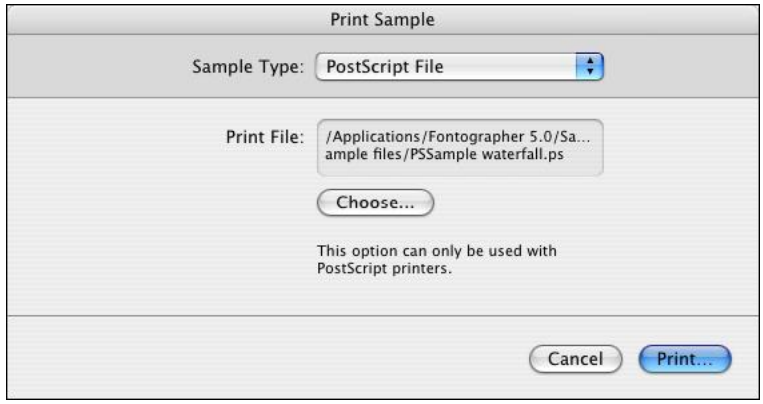
Note: Fontographer can print only plain text files as samples. This means Fontographer cannot print normal word processor files. You must first resave these files as Text files before Fontographer can print them.

PostScript file

The PostScript file option allows you to choose custom PostScript samples. Fontographer then sends this file to the printer along with the font. As in the Sample file print option, the file is chosen after the **Print** button is selected. This option is only available with PostScript printers.

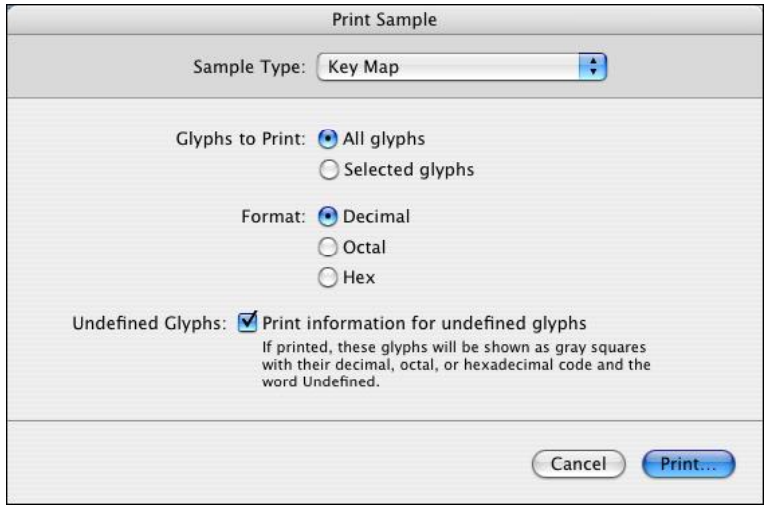
Note: For PostScript gurus the name of the font used in the file is *TestFont*. Fontographer will use any font you are currently working in as the TestFont, sending its information to the printer.

Several PostScript text files are provided with Fontographer 5. You may use them as is, or you may edit them with any text editor to define your own custom print sample.



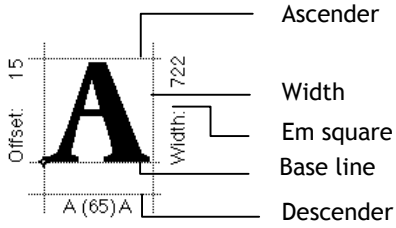
Key map

The Key map option allows you to print a sample for the entire font or only glyphs that you have selected.



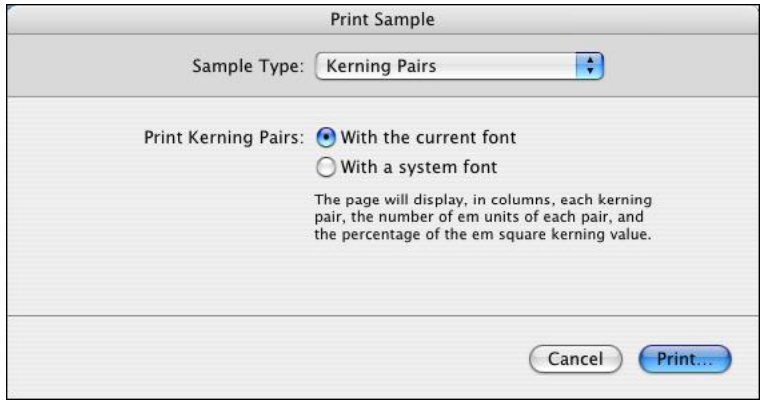
If you select the All glyphs option, Fontographer prints pages containing all the glyphs in the font. The printed pages consist of rows of glyphs, along with their offsets, widths, and corresponding keystroke codes. Fontographer gives you the option of showing decimal, octal, or hexadecimal locations. The Print undefined glyphs option allows you to print or omit undefined glyphs from your printout. If printed, these glyphs will be surrounded by a gray box, their decimal, octal, or hexadecimal code, and the word Undefined.

A glyph with a normal offset (not less than zero) will print a key map sample that looks something like this:

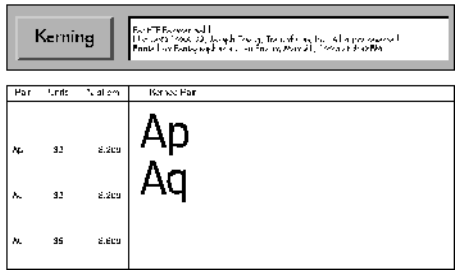


Kerning pairs

The Kerning pairs sample type option makes it possible to print a list of kerning pairs in the font. A full page, consisting of several columns will be printed showing each pair, the number of em units of each pair, and the percent of the em square each kerning value represents. The kerning pairs can either be printed in a monospaced font or in the current font open in Fontographer.



Choose Kerning Pairs...



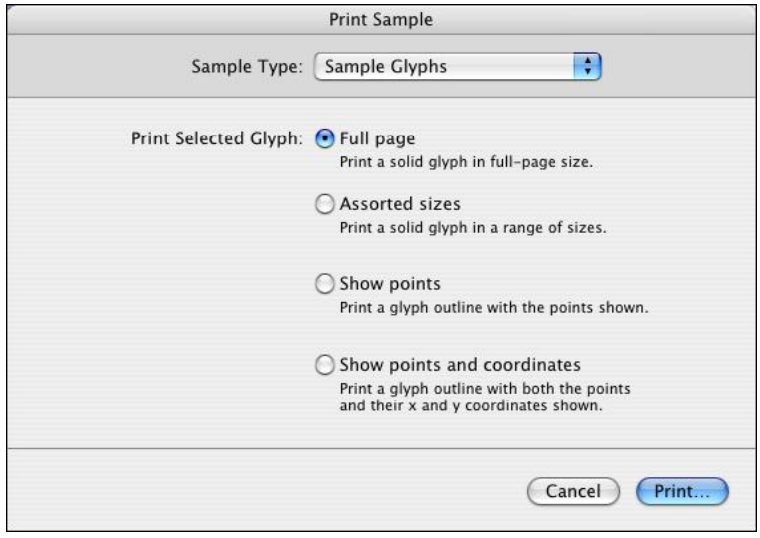
to print a kerning pairs list.

Sample Glyphs

The Glyphs sample type option provides you with four different choices: printing one full-page-sized filled glyph, several assorted sizes of the filled glyph, a sample that shows the points of the glyph, or a sample that shows both the points and x/y coordinates for each point.

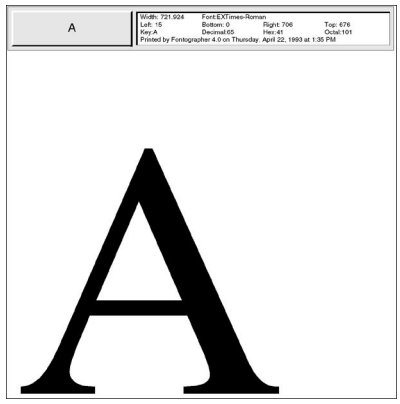
To print a full-page glyph sample:

1. Select the glyph(s) you want to print from the Font window, or open an Outline window for the glyph you want to print.
2. Choose **Print** from the **File** menu.
3. Choose Sample Glyphs from the **Sample type** pop-up.



4. Click the **Full page** radio button.
5. Click **Print**.

This is the quickest way to print a single glyph.

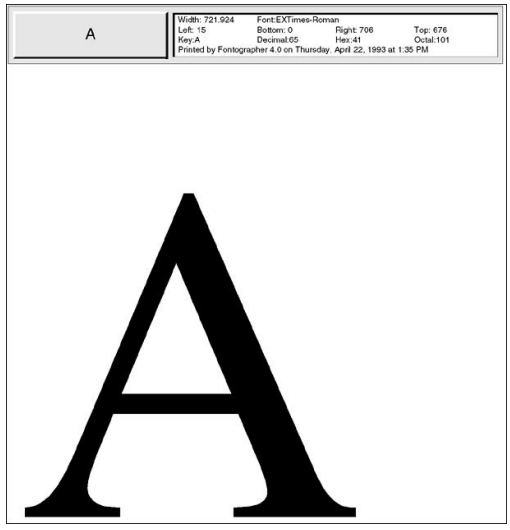


To print a text sample in assorted sizes:

1. Select the glyphs you want to print from the Font window. Use any letters you want to see printed or open an Outline window for the glyph you want to print.
2. Choose **Print** from the **File** menu.
3. Choose Glyphs from the **Sample type** pop-up.
4. Click **Assorted sizes**.
5. Click **Print**.

To print the other print samples, repeat the printing steps but instead click **Show points** or **Show points and coordinates**.

The print header



All printed samples have a similar print header. The gray box that occupies the left corner of the header indicates the print sample type. For instance, if you choose the Kerning pairs option, the header will read Kerning. If you choose Selected glyphs, the header will read Selected, and so on. An individual glyph such as the letter “S” will have only the glyph as a header.

The rest of the box will show the name of the font, the font size, any applicable textual information, and the date and time of printing. This is useful for archiving, filing, and other quality control procedures.

Generating and Exporting Fonts

You've been happily editing your new font, and it appears that everything's just the way you want it. Now what? Your new font isn't really a font yet: all you have is a bunch of characters in a database. (Maybe you've noticed the message displayed when you save your font: "Writing Fontographer database".) This database file won't work as a font because it hasn't been encoded into the proper structure. Fonts are resources that the system must have stored in a particular manner in order to be shared with applications that use fonts.

This chapter discusses all the font generation options available to you.

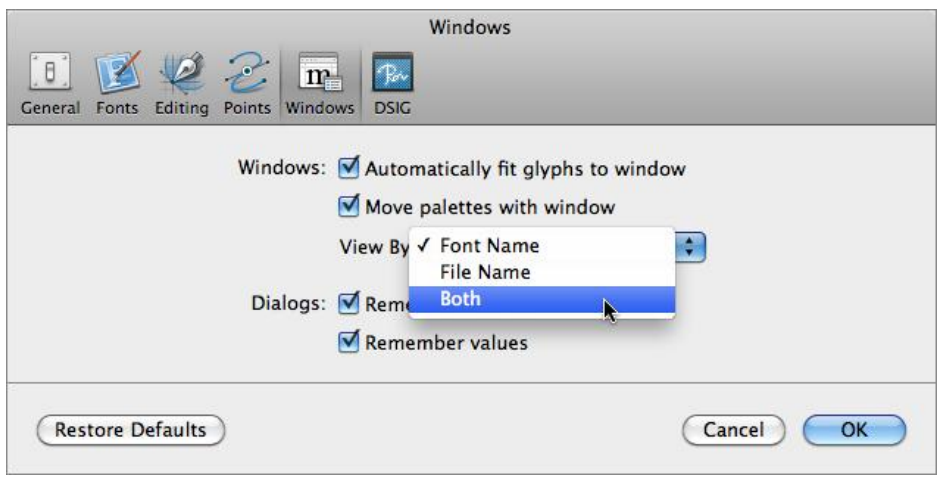
9

Before you do anything...

Fontographer 5 can generate font files of the following formats: Mac TrueType Suitcase (without extension or .dfont), Mac Type 1 and 3 (PostScript) and Mac Multiple Master + suitcase, TrueType/OpenType TT (.ttf), Windows Type 1 and 3 (PostScript) and Windows Multiple Master (.pfb), Unix/ASCII Type 1 (.pfa), OpenType PS (.otf).

Fonts on the computer all have names, and your fonts are no exception. If you have not entered a name for your font in the Font Info dialog box, be sure to do so before generating any font files.

Since Fontographer 5 allows you to save your font with a name other than the file name, confusion can arise as to what the file name of the font is. The default is for the font to be viewed by the font name specified in the Font Info dialog box. For those who keep font names and file names the same, this method works fine. If, however, you have many versions of the same font name, change the **View by** option to *File name* or *Both* in the **Windows** section of the Preferences dialog. This enables you to know at a glance which file you're working on:



Relevant Font Formats

The following lists the most relevant font formats and lists some of their advantages and disadvantages.

OpenType PS

Also known as: OpenType-CFF, PostScript-flavored OpenType, OTF

Filename extension: **.otf**

Pros: Works on Windows, Linux, Mac OS 8.6, 9, and OS X. Uses the Bezier curves that are preferred by designers and used in drawing apps such as Illustrator and Freehand so letterforms can be drawn precisely and outlines need not be converted. May contain up to 65,535 glyphs, supports Unicode and can contain OpenType Layout features.

Suitable for Western Roman fonts, non-Latin fonts, multilingual fonts and advanced typography. May include class kerning allowing for moderately-sized kerning tables. Uses Type 1 hinting that is relatively easy to create. Can include embedding rights information defining whether or not the font may be attached to electronic documents.

Cons: Type 1 hinting does not allow precise control in small screen sizes. Can theoretically contain bitmaps, but they are not displayed. Since this is a relatively new format, there are problems with some old applications (some styles are not displayed in menus, kerning for non-Western characters does not work.) The multilingual and advanced typography features only work with new OpenType-savvy applications, otherwise just the basic character set is available. Two alternative family namings within each font must be devised: one where a family contains an arbitrary number of styles, and second “brief family” where one family does not contain more than four styles. Does not work on Mac OS 8.

Recommendation: We recommend producing fonts in the OpenType PS format unless you have Mac customers running a pre-X Mac OS. For older systems (pre-X Mac OS) generate either a TrueType or a Type 1 font suitcase.

TrueType / OpenType TT

Also known as: Data-fork TrueType, Windows TrueType, TrueType-flavored OpenType, TTF

File extension: **.ttf**, also possible: **.otf**

Pros: Works on Windows, Linux and Mac OS X. May contain up to 65,535 glyphs, supports Unicode and can contain OpenType Layout features.

Suitable for Western Roman fonts, non-Latin fonts, multilingual fonts and advanced typography. May include class kerning allowing for moderately-sized kerning tables. TrueType hinting allows precise control in small screen sizes, can also contain bitmaps. Can include embedding rights information defining whether or not the font may be attached to electronic documents.

Cons: Does not work on Mac OS 8/9. May cause output problems on ten-year-old PostScript output and printing devices. The designer usually needs to convert the outlines from Bezier curves, which may introduce very slight changes in the shape. When converted back to Bezier curves (e.g. in Illustrator), the resulting curves have superfluous points. Manual TrueType hinting is laborious to create. The multilingual and advanced typography features only work with new OpenType-savvy applications, otherwise just the basic character set is available. For font families, requires two versions of the family name within each font: the first may contain any number of styles; the second “brief family” may contain only four styles.

Macintosh TrueType

Also known as: sfnt-based TrueType, TrueType suitcase

File extension: none

Pros: Works on all Macintosh systems, not cross-platform. May contain up to 65,535 glyphs, supports Unicode.

Suitable for Western Roman fonts, non-Latin fonts, multilingual fonts and advanced typography. TrueType hinting allows precise control in small screen sizes, can also contain bitmaps (in the same suitcase file). Can include embedding rights information defining whether or not the font may be attached to electronic documents.

Cons: Does not work on Windows. May cause output problems on ten-year-old PostScript output and printing devices. The designer usually needs to convert the outlines from Bezier curves, which may introduce very slight changes in the shape. When converted back to Bezier curves (e.g. in Illustrator), the resulting curves have superfluous points. Manual TrueType hinting is laborious to create. One family cannot contain more than four styles.

Macintosh Type 1

Also known as: Macintosh PostScript, LaserWriter font

File extension: none

Pros: Works on all Macintosh systems, not cross-platform. Works in all PostScript commercial output and printing devices. Uses the same curve system (Bezier) as drawing applications such as Illustrator and Freehand, so letterforms are easy to edit when converted to curves. Type 1 hinting is comparatively easy to create. Can contain bitmaps for small screen sizes. One family can contain more than four styles.

Cons: Does not work on Windows, not cross-platform. Contains two parts, the outline file and the bitmap font (suitcase), both of which must be in the same folder. Does not contain class kerning so kerning tables are large. Type 1 hinting does not allow precise control for very small screen sizes. Cannot include more than 256 encoded characters and lacks advanced layout features such as ligatures, making the format unsuitable for multilingual or non-Latin fonts.

Windows Type 1

Also known as: Windows PostScript, PC PostScript, PC Type 1

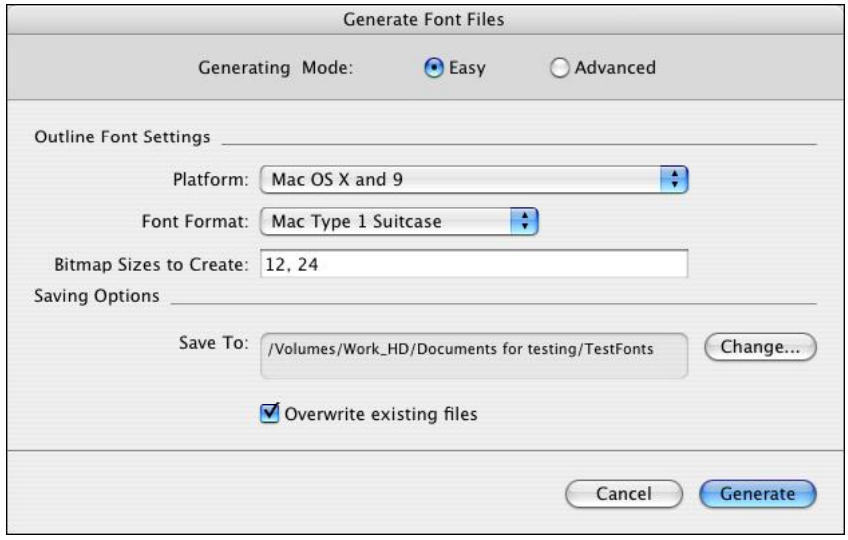
File extension: **.pfb**, with supplementary files **.afm, .inf, .pfm**

Pros: Works on Windows and Linux. Works in all PostScript commercial output and printing devices. Uses the same curve system (Bezier) as drawing applications such as Illustrator and Freehand, so letterforms are easy to edit when converted to curves. Type 1 hinting is comparatively easy to create.

Cons: Does not work on Mac OS 9 or X, not cross-platform. Contains two parts, the outline file (.pfb) and the metrics font (.pfm), both of which must be in the same folder. Does not contain class kerning so kerning tables are large.
Type 1 hinting does not allow precise control for very small screen sizes. Cannot include more than 256 encoded characters and lacks advanced layout features such as ligatures, making the format unsuitable for multilingual or non-Latin fonts. Cannot contain bitmaps for small screen sizes. One family cannot contain more than four styles.

Easy or advanced?

Fontographer is a do-all, end-all font editor. Since there is so much to see and do in it, we created many of our dialog boxes with an **Easy** option to simplify the process. Most novice users of Fontographer will be prudent enough to select the **Easy** option and let the program set up the desired font properly. But more advanced readers, beware! You know just enough to be dangerous with font generation.



Time for a pop quiz: What encoding do Sun fonts require? What does UPM size do? What happens in NeXTSTEP installation if you have no AFM file?

If you are unsure of the answers to these questions, then use the **Easy** option and let Fontographer make the best decisions for your fonts.

You experienced readers, dive right into the **Advanced** dialog boxes and the technical stuff; if you get stuck, then your experience will rescue you. If even that fails, then contact our Technical Support group. If your time is a terrible thing to waste, then you should also take advantage of the **Easy** option in the Generate Font Files dialog box.

In the Easy Generate Font Files dialog box, you have five simple decisions to make.

1. Select the platform for which you are preparing the font.
2. Determine the format (PostScript, TrueType or OpenType).
3. Choose the bitmap sizes – if any.

Note: Only Macintosh PostScript fonts require bitmaps. But you still may add them also to Macintosh TrueType Suitcase.

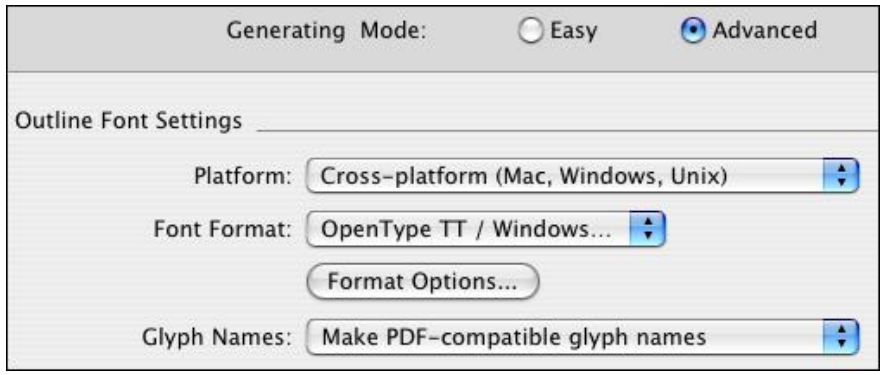
4. Set the folder into which you want the font files saved.
5. Decide whether you want to overwrite existing files with the same name.

Fontographer will then decide the sticky issues; for instance, if an AFM file should be generated for the NeXT fonts (yes), what ID should be selected for Mac FOND resource, or even how to set up the UPM size for Windows TrueType fonts (2,048).

This is all the information necessary to use the **Easy** mode. The rest of this chapter covers the more complicated stuff that appears in the **Advanced** mode.

Generating cross-platform fonts

Modern Mac OS X, Windows Vista, Windows 7 and Linux support fonts in TrueType/OpenType TT and OpenType PS format. So these two formats became cross-platform de-facto standard.



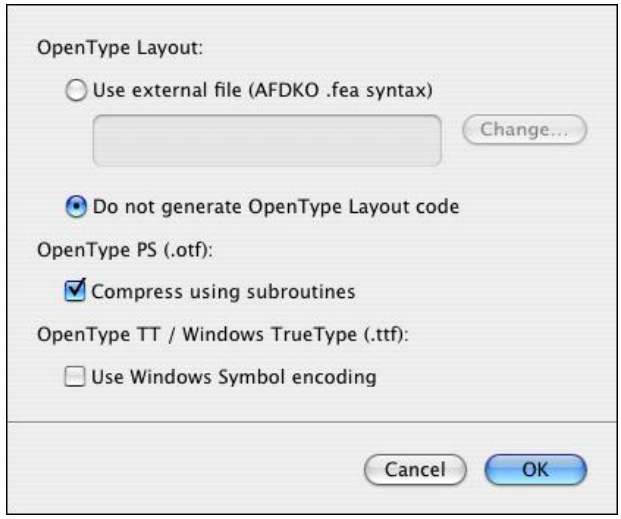
To generate cross-platform TrueType/OpenType TT or OpenType PS fonts:

1. Choose **Generate Font Files** from the **File** menu.
2. If you are not already in the Advanced mode, click the **Advanced** radio button in the Generate Font Files dialog box.
3. Select Cross-platform(...) from the **Platform** pop-up.
4. Select OpenType TT or OpenType PS from the **Font Format** pop-up. A Format Options dialog box appears when you click the **Format Options** button.
5. In the **Glyph Names** pop-up, you have two choices. Since both OpenType TT and OpenType PS formats are based on Unicode select Make PDF-compatible glyph names here.
6. Check the **Overwrite existing files** option. When this option is not checked, any file name created that conflicts with an existing file name in the same folder will have a number (2, 3 etc.) appended to its name.
7. Point Fontographer to the folder where you wish to save your fonts, via the **Change** button.
8. When all the options have been selected, press **Generate** to close the dialog box and generate the font.

OpenType Options

The OpenType Options dialog box appears when you click on the **Format Options** button and the selected font format is OpenType PS or OpenType TT/TrueType.

The default settings for OpenType PS export are shown below:



Fontographer cannot automatically generate OpenType layout features, but it can take the code from the external file. If you select the **Use external file** option Fontographer will take features code from the external file you point it to. You can edit this file with any text editor before generating the font and therefore have full control of the OpenType layout. Refer to Chapter 12, "OpenType Fonts" for details.

You may choose to not generate OpenType features at all. In this case click on the **Do not generate OpenType Layout code** option. With this option turned on, for OpenType PS (.otf) fonts, Fontographer will still generate an OpenType "kern" feature in the GPOS table, and an empty GSUB table (the latter is required because of a bug in InDesign CS3). For OpenType TT/TrueType (.ttf) fonts, no GPOS or GSUB will be generated. Kerning will be saved as a classic kerning table.

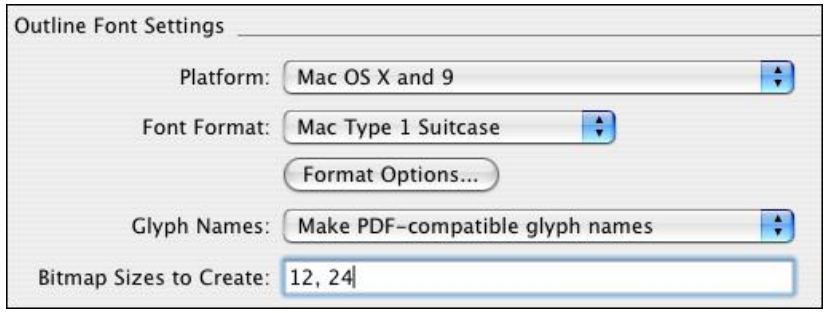
The **Compress using subroutines** option allows to automatically generate outline subroutines if font is generated as CFF-flavored. Outline subroutines store repetitive parts of outlines and allow reuse with references from outline definition code. The font can be more compact in this case.

The **Use Windows Symbol encoding** should only be selected if you wish to create a Wingdings style symbol font for Windows, or if you have some other special purpose (like getting some additional characters, and so forth). For more information, see "Symbol encoded Windows fonts" following.

Generating Macintosh fonts

Mac OS X has full support for OpenType PS (.otf) and OpenType TT/True Type (.ttf) fonts. But you may need to generate old-fashioned Macintosh fonts that will work on older Mac systems.

Macintosh fonts require two particular font resources: the outline font and the FOND. In the case of Type 1 and 3 (PostScript) fonts, these font resources are in separate files. For TrueType fonts, the two resources exist in the same file. This section will deal with the "how to" issue of generating the font resources. Much more information on how to make use of the fonts once they're generated will be covered in Chapter 11, "Installing and Removing Fonts".



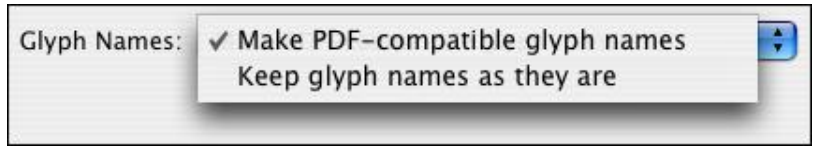
Mac Type 1 Suitcase

PostScript Type 1 fonts – also known as ATM fonts – are the fonts supported by Adobe Type Manager and all Mac systems. There are two principle advantages of Type 1 over the older Type 3 font format. When printing to low-resolution laser printers, Type 1 fonts are more legible at small point sizes. Also, ATM or OS will render the Type 1 PostScript fonts on-screen, thus removing the need for bitmaps for every point size.

Type 3 fonts aren't widely supported; most operating systems cannot work with them. You will probably never have the need to generate Type 3 fonts but these fonts will be discussed in detail in the next section.

To generate Macintosh PostScript Type 1 fonts with suitcases:

1. Choose **Generate Font Files** from the **File** menu.
2. If you are not already in the Advanced mode, click the **Advanced** radio button in the Generate Font Files dialog box.
3. Select Mac OS X and 9 from the **Platform** pop-up.
4. Select Mac Type 1 Suitcase from the **Font Format** pop-up.
5. The Type 1 Options dialog box appears when you click on the **Format Options** button. We recommend that you accept our default settings for the **Include hints** and **Use "Flex"** checkboxes. If you want more information on these fields, then refer to "When should you use hints?" and "A word about flex" on page 315 later in this chapter.
6. In the **Glyph Names** pop-up, you have two choices:



   Select Keep glyph names as they are if you have Custom glyph names and do not want Fontographer to change them.
7. See the section "Pack your suitcase: bitmap fonts" on page 319 to learn about bitmap options.

8. The **Overwrite existing files** checkbox is provided as a means to prevent the accidental removal of original files. If checked, Fontographer will replace older files with the same names as those currently being generated. If deselected, Fontographer will keep the older files alone, and change the name of the conflicting file by appending a number (2, 3, etc.) to the file name.

9. AFM files are not used on the Macintosh. So, if you're using a Macintosh, the **Create AFM file** option is turned off by default. If you want an AFM for your Macintosh font, just turn the checkbox on.

10. The default directory is indicated in the **Save to** field. If you press the **Change** button, you are presented with a standard file dialog box. Select the destination folder in the dialog box and press **Choose**.

Mac Type 1 Font

You also have the possibility to export Mac PostScript Type 1 font without an accompanying suitcase containing metrics and bitmaps. This is a single printer file (LWFN) containing POST resources. It cannot be installed and used in other applications without a suitcase but this option can be useful in some cases.

To generate Macintosh PostScript Type 1 font without a suitcase:

1. Choose **Generate Font Files** from the **File** menu.
2. If you are not already in the Advanced mode, click the **Advanced** radio button in the Generate Font Files dialog box.
3. Select Mac OS X and 9 from the **Platform** pop-up.
4. Select Mac Type 1 from the **Font Format** pop-up.
5. All the rest options are the same as for the Mac Type 1 Suitcase except that you cannot create suitcase file and therefore define bitmap sizes.

TrueType suitcase

Developed as a joint venture by Apple and Microsoft, TrueType fonts have become a popular outline font option for Macintosh and Windows systems.

You can create TrueType fonts using the default UPM size of 1000 built into Fontographer. In some specific cases, you may want to use a different UPM, for example 2000 or 2048. This may give you finer control of the character outline.

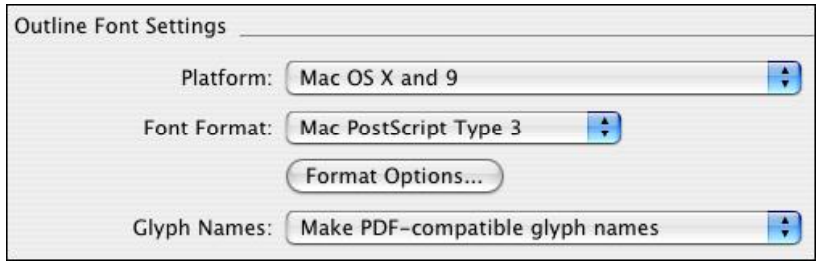
If you use a non-1000 UPM size, you can set your ascender and descender in **Font info** under the **Element** menu to adjust for the larger UPM, if you wish. You might try values of 1638 for ascender and 410 for descender to approximate the default UPM's 800/200 split.

To generate Macintosh TrueType fonts:

1. Choose **Generate Font Files** from the **File** menu.
2. If you are not already in the Advanced mode, click the **Advanced** radio button in the Generate Font Files dialog box.
3. Select Mac OS X and 9 from the **Platform** pop-up.
4. Select Mac TrueType Suitcase from the **Font Format** pop-up. A TrueType Options dialog box appears when you click on the **Format Options** button. The contents of the dialog are described above in the "OpenType Options" section.
5. In the **Glyph Names** pop-up, you have two choices. Select Make PDF-compatible glyph names here.
6. Bitmaps are not needed for Mac TrueType fonts.
7. Check the **Overwrite existing files** option. When this option is not checked, any file name created that conflicts with an existing file name in the same folder will have a number (2, 3 etc.) appended to its name.
8. Point Fontographer to the folder where you wish to save your fonts, via the **Change** button.
9. When all the options have been selected, press **Generate** to close the dialog box and generate the font.

PostScript Type 3

The steps for generating a Type 3 font are almost identical to those for generating a Type 1 font. But why would anyone want to generate a Type 3 font, since it won't work with ATM and is not as legible as Type 1 when printed to a 300-dpi printer? The advantage is the flexibility of the Type 3 format.



Let's assume you want an ornaments font that uses several tonal variations and filled-and-stroked objects in the same character. No problem with Type 3 fonts; absolutely impossible with Type 1 or with TrueType. The general rule is: for plain text fonts, go Type 1 or TrueType; for decorative or special-use fonts, experiment with Type 3. The rewards of this flexible format may surprise you.

To generate Macintosh PostScript Type 3 fonts:

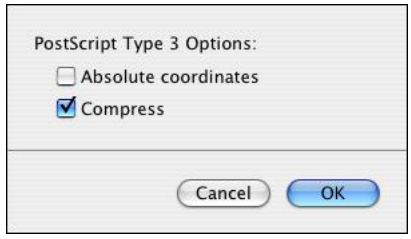
1. Choose **Generate Font Files** from the **File** menu.
2. Click the **Advanced** button, and then select the Mac OS X and 9 option from the **Platform** pop-up.
3. In the **Font Format** pop-up, select Mac PostScript Type 3.
4. The Type 3 Options dialog box appears when you click on the **Format Options** button. For more information on the **Absolute coordinates** and **Compress** checkboxes, see the section "Other Type 3 Formats" on page 314.
5. The **Overwrite existing files** checkbox is provided as a means to prevent the accidental removal of original files. If checked, Fontographer will replace older files with the same names as those currently being generated.

   If deselected, Fontographer will leave the older files alone, and change the name of the conflicting file by appending a number (2, 3 etc.) to its name.

6. AFM files are not used on the Macintosh. This option is turned off by default. If you think you've got a use for the AFM, then click the checkbox to turn it on.
7. The default directory is indicated in the **Save to** field. If you press the **Change** button, you are presented with a standard file dialog box. Select the destination folder in the dialog box and press **Choose**.
8. Press **Generate** to create your font files and close this dialog box.

Other Type 3 Formats

The **Absolute coordinates** checkbox creates a file extension of ".abs". This ASCII text file is often used by sign-cutters in their plastic cutting machines, which require absolute coordinates, rather than the usual relative coordinates. PostScript-Absolute requires a compressed file so when that option is checked, the **Compress** option should also be checked.



If you deselect the **Compress** checkbox, Fontographer will create a plain PostScript file. PostScript programmers use this to generate standard PostScript definitions of all the characters in the font. This will create a usable, if very large, Type 3 PostScript font.

Note: You cannot create uncompressed Type 1 fonts, because Type 1 fonts are compressed by definition.

Compressed is the standard Type 3 font format. This font file can be used for both automatically downloadable fonts or fonts that are to be downloaded to the printer's hard disk. Compressed PostScript is the preferred form for generating Type 3 fonts. PostScript is stored very inefficiently inside the printer, so compression is necessary to pack as much information into the memory as possible. Fontographer's compression scheme generates fonts that take up one-fifth the space, on the average, of uncompressed fonts. Nevertheless, a Type 3 compressed font is still larger than a Type 1 font.

When should you use hints?

Hints are information placed into a character's outline definition that adjusts it in a way that improves the character's perceived shape when it is drawn on the screen or on the printer. Hints almost always enhance the look of your font. Fontographer generates hints by default. You would do well to leave hinting on unless you have found some problem with the printing of your font that seems to indicate inaccurate hints.

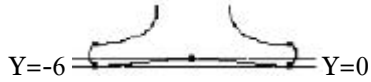
A word about flex

Flex is helpful only for Type 1 PostScript serif fonts that meet very specific design criteria. If your font doesn't match this exact model, then don't use flex. Here's what it does:

The purpose of flex is to eliminate slight indentations in the font's outline at small sizes if possible, while still keeping those same indentations at larger sizes. When the **Flex** box is checked, Fontographer applies subtle effects to cupped serifs and tapered stems. Garamond is a good example of a font that needs flex to properly render its serifs.

A segment will only respond to flex if it meets certain requirements:

- The segment must be composed of exactly two Bézier curve segments, typically created by placing a corner point, a curve point, and a corner point.
- The outer points must be perfectly vertical or horizontal (meaning that they have the same x or y coordinate).
- The difference between the end points' x/y coordinates and the middle point (known as flex height) must be 6 units or less in the flex direction.



Thus, for a serif flex, the middle point should be at Y = 0, and the end points should be at or above Y = -6.

Note: Applying flex to your font can add as much as 10K to the size of your font file.

Generating Windows fonts

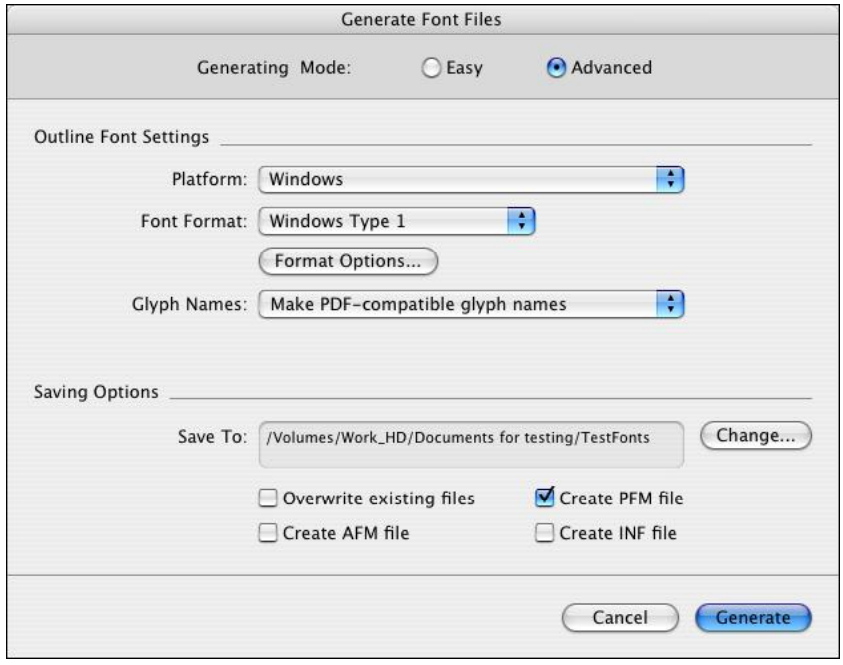
PostScript Type 1

Windows Type 1 PostScript fonts – also called ATM fonts – are intended for use in Microsoft Windows.

Microsoft Windows 2000 and later support Windows Type 1 fonts by default. Older Windows operating systems (3.1, NT, 95, 98, ME) require Adobe Type Manager (ATM) 2.x (or higher) to be installed. ATM doesn't work anymore on Windows Vista and Windows 7.

Windows PostScript fonts consist of two mandatory files: PFB and PFM. In addition, the font can include optional files: AFM and INF.

In the Advanced dialog box, you also get to choose which files should be output, whether or not the use of Flex is appropriate, and if hints should be included in the font.



To generate Windows PostScript Type 1 fonts:

1. Choose **Generate Font Files** from the **File** menu.
2. Select Windows from the **Platform** pop-up.
3. Select the Windows Type 1 option in the **Font Format** pop-up.
4. The Type 1 Options dialog box appears when you click on the **Format Options** button. We recommend that you accept our default settings for the **Include hints** and **Use "Flex"** checkboxes. If you want more information on these fields, then refer to "When should you use hints?" and "A word about flex" on page 315.
5. **Create PFM file** should be checked; the **AFM** and **INF** options should be deselected in most situations.
6. Point Fontographer to the destination folder, via the **Change** button.
7. When all the options have been selected, press **Generate** to close the dialog box and create your font files.

Other Options

Overwrite existing files

When this option is not checked, any file name created that conflicts with an existing file name in the same folder will have a number (2, 3, etc.) appended to its name.

Output AFM file

The AFM file is not used by Windows in normal installations. If you have a use for this metrics file, then check the box to generate the file.

Output PFM file

The PFM file is required by Windows. This option is on by default. This is a binary file containing metrics information (similar to information found in the AFM).

Output INF file

The INF is an information file used by some older DOS applications for name and style information. In rare situations, it can also be read by ATM and used with the AFM file in lieu of the PFM. Only generate this file if you know you will use it; otherwise, its presence may cause confusion when installing fonts in Windows.

Symbol encoded Windows fonts

When you generate a Windows OpenType TT/TrueType font, a checkbox allows for a special symbol encoding option. This is the same encoding format used in the dingbats font in Windows, Wingdings. Rather than converting your Windows font to the ANSI key layout (which results in some missing characters), Symbol encoding will output every character in your database from the Dec 32 (space) to Dec 255 (caron), except 127 and 160. All characters in the normal ASCII (Dec 32–126) sequence will remain accessible by striking the alphanumeric keys. But since extended ASCII characters on the Windows are accessed by typing the ALT+0XXX key combination, if you want the copyright symbol (©), hold down the ALT key and use the numeric keypad to type 0 followed by the decimal number for the copyright symbol, which is 169. Do likewise for every other extended ASCII (Dec 128–255) character. If you have any characters mapped to decimal 0–31, they won't be output to the Windows with Symbol encoding, nor will anything in Dec 127 (delete) or Dec 160 (nonbreaking space).

When Symbol encoding is selected, sequential character ordering should also be used. All other TrueType preferences do apply.

PostScript Type 3

This option has become a relic. We include it for those one or two folks who are using older applications that may require it. The instructions for using Type 3 fonts are very similar to the Type 1 discussion. The only additional note is that you will have to provide a bitmap screen font for the application. You can start with our BDF format (**File > Export > BDF**) and use some conversion tool in DOS to make it into a usable format.

Pack your suitcase: bitmap fonts

Bitmaps vs. outlines

In these days of OpenType fonts, the relationship between screen fonts and printer fonts is getting blurred. There was a time when they were both represented by individual files. But in the present day of OpenType and TrueType every platform we support can render the outline file to both the screen and the printer. So does that mean we never have to deal with bitmaps again? Not really.

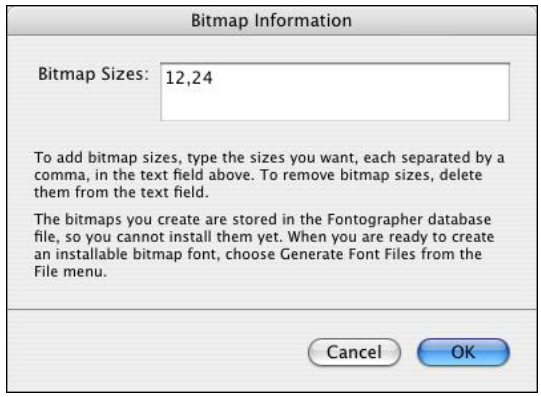
Think about your screen image. It is represented by a lot of pixels, each either turned on or turned off. When the font is rendered to the screen, the font display driver maps out the bits it wants turned on. That is essentially a bitmap created on the fly.

Now, consider the printed image. What is the resolution of your printer? Most laser printers print at least 300 dots (read bits) per inch. Thus, in one letter-size page, the particular page description driver determines which of the 8,145,000 dots it wants on; the ones turned on attract the black toner, the others don't. But before the page is printed, the intelligent printer has mapped out each bit and saved the page in RAM as a bitmap image. You can imagine how much memory that 300-dpi bitmap can consume.

While bitmap font files are rapidly becoming obsolete, bitmaps themselves – whether they are screen images or page descriptions – are always with us. Understanding their relationship will enable better use of fonts in general.

Fontographer 5 generates two varieties of bitmap fonts: one specifically for use as a screen image on the Macintosh, the .bmap (NFNT resources in the suitcase file), and one intended for distribution to other platforms (which may require further customization) – the BDF.

The bitmap sizes entered in the Bitmap Information dialog box are stored in your database.



These are not output to a file until you say so in the Generate Font files dialog box. If you won't be hand-editing your bitmaps, then there is no compelling reason to use this dialog box.

Note: It is not necessary to enter all the sizes you will ever want to output into the Bitmap Information dialog box prior to generating fonts. Any new sizes that you enter when generating fonts will be built on the fly and output to the bitmap file as is. (They will then be retained in the database file.)

Adding bitmap sizes

Type the desired point sizes in the Bitmap point sizes field and click **OK**. Adding a new size will automatically generate all the characters for the newly added size(s). They are limited to sizes between 1 and 255 points.

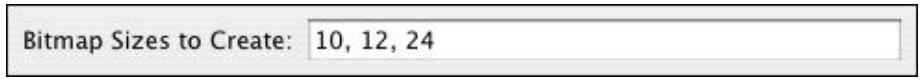
Deleting bitmap sizes

Unwanted bitmap sizes can be removed from the database file by removing the size from the Bitmap sizes field in the Bitmap Information dialog box. Fontographer will display a warning message before it deletes the bitmaps.

Bitmap format

NFNT

This information is specific to the Macintosh. If you have no desire to comprehend the mysteries of the NFNT and FOND, then you can proceed to the next section.



Fontographer generates NFNTs (New FoNTs) for the Macintosh Type 1 or TrueType suitcase.

The NFNT is the repository for the bitmaps. The spacing, kerning, naming, and metrics information is handled by the FOND. With the arrival of TrueType fonts on the Macintosh and System 7, the NFNT is being phased out and in many situations today, it is not even required. But the FOND remains a necessary part of the font handling equation on the Macintosh.

Exporting files

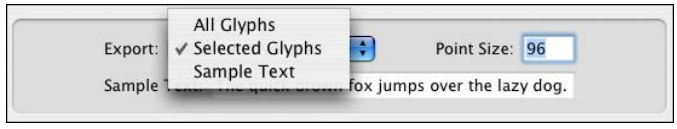
Fontographer is equipped with an **Export** menu item. From within its submenu you may export EPS, BDF, .vfb or metrics files. In this section we will discuss exporting your font characters as graphics for use in applications such as Adobe Illustrator or Adobe Photoshop. For more about exporting metrics, and the Fontographer Metrics file, refer to Chapter 6, "Metrics – Spacing and Kerning".

Exporting EPS files

The EPS option generates an Adobe Illustrator 1.1-style EPS (Encapsulated PostScript) format file. Programs such as Adobe Illustrator can open this file format directly for editing purposes. And once you open your EPS file, you can add or delete points, reshape, distort or fill characters with a specific color or pattern, or change the stroke weight.

To export an EPS file:

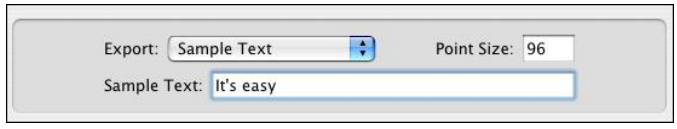
1. Select the **EPS format** from the **Export** pop-up in the **File** menu.
2. Select the point size necessary for your intended use.
3. Choose an **Export** option.



All glyphs will export all the glyphs in your font.

Selected glyphs will export only the glyphs currently selected in the font window or from the outline window (the active glyph).

Choosing **Sample text** will bring up a text entry box for you to type the text string you would like to export.



4. When you export an EPS file, Fontographer allows you to choose your own file name.
5. Once you've chosen the folder to export the file to, press the **Export** button and the file will be exported without further comment.

Exporting BDF files

This option is provided to generate Adobe Bitmap Distribution Format (BDF) files. BDFs can be used for creating screen fonts on computer systems such as the Sun, DEC, and DOS. Information on the format of the BDF file can be found on the web:

http://partners.adobe.com/public/developer/en/font/5005.BDF_Spec.pdf

To export a BDF font:

1. Select the **BDF** format from the **Export** pop-up in the **File** menu. If the item is not available this means you must create bitmaps with the **Element > Bitmap Info** command.

2. Select the bitmap size necessary for your intended use.

   Only existing bitmap sizes are listed here.

3. When you export a BDF font, Fontographer allows you to choose your own file name.

4. Once you've chosen the folder to export the file to, press the **Export** button and the file will be exported without further comment.

Creating a Font Family

Just as your own family has differences and similarities, so do your fonts.

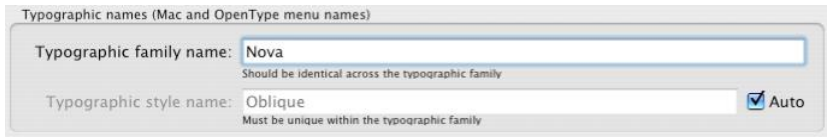
A family of fonts is defined as all the styles of one typeface. The group shares a common design but can differ in attributes such as character width, weight, and slope – that is, Roman and Italic.

10

A typical computer (font) family unit frequently contains four fonts – Plain, Italic, Bold, and BoldItalic. But this is not mandatory.

To appreciate the advantage of font families, imagine having twenty typefaces with four styles each installed in your system. In your font menu, there would be 80 fonts, since every typeface would appear four times, representing each of its styles. It would be inconvenient to search through the 80 entries every time you wanted to make your existing font bold. Using font families improves this scenario in two ways: it lets you use command keys to change the style of the font, and it shortens your font menu since there is just one listing for each family rather than one for each font.

In order to build families, you must use appropriately named fonts. In other words, their names must have the same base part. So you cannot choose Helvetica-Bold as the bold style in the Nova family; you must choose a font whose name starts with Nova.



When you name your font in Fontographer's Font Info > Names dialog box be sure to name each font properly. Make sure the Typographic Family Name text box has the same name in it for all the styles of the family you're building and every font in the family has it's own Typographic style name. This is important to remember for making families on all platforms.

Menu grouping and style linking

Traditionally, the Mac OS platform has always allowed **typographic family grouping** of fonts: an arbitrary number of fonts can appear in the font menu under one family name, with every font having a distinctive style name. This allows typographically authentic representation of font families. Many font families consist of a number of styles with different weights (light, regular, semibold, bold, black etc.), sometimes different widths (condensed, normal, extended) and often with accompanying italic styles.

Some of the fonts within one family are linked together through **styling links**, also known under the name "style-linking associations". A typical styling link is "this font acts as the bold style of another font" and "this font acts as the italic style of another font". Through styling links, the application knows which font should be used when the user applies italic or bold formatting — typically through clicking on an "I" or "B" icon in the application's toolbar. In the past, other styling links were also used such as "this font acts as the underline style of another font", but they are no longer relevant. These days, underlining is usually done by the application through drawing a line under the characters (the thickness of this line is specified in the font's parameters, but not all applications use that information).

In the Mac Type 1 format, multiple and complex styling links are possible. The "Black" member of the family font could act as the bold style of the "Bold" font, and at the same time, the "Bold" font could act as the bold style of the "Regular" font.

The original Windows graphic subsystem (GDI), still used by most Windows applications, only permits a more simplistic approach of forming font families: within one family, all fonts except one must have a styling link to another font, and the styling links cannot repeat or recurse. Since the two typical styling links are "is bold" and "is italic", a font family on Windows could have up to four members: the bold italic font acts as the italic style to the bold font and as the bold style to the italic font, the bold font acts as the bold style to the regular font, and finally, the italic font acts as the italic style to the regular font. All fonts that are linked to each other through a styling link for use in Windows GDI applications form a styling group.

Every font format (Mac Type 1, Windows Type 1 and OpenType/TrueType) has properties that represent both the typographic family grouping and the styling links, but they are stored or distributed in different ways, and sometimes there are more than one to choose from. Fontographer 5 suggests the optimal way to automatically create proper parameters for new fonts or translate those properties between formats.

In addition to the naming fields, Fontographer introduces a number of four design parameters that you need to define to build proper font family:

- width (also called FontStretch)
- weight (also called FontWeight)
- slope (also called inclination or slant, or FontStyle)

The design parameters of a font describe typographic properties of the underlying typeface. These parameters (width, weight and slope) are essential in the font selection model introduced by WPF (Windows Presentation Foundation). According to the WPF model, each font in a family must have a unique combination of those three parameters.

So your basic task is to choose the design parameters from dropdown lists.

Fontographer then can generate family naming and styling links based on your selection. When opening existing fonts, Fontographer tries to deduce the values for these parameters, and also preserve as much of the original family naming and styling links as possible.

In addition, you can specify a design parameter called **Other**. This parameter is just a text value where you can describe a design parameter of a typeface that is not width, weight or slope. For example, the optical size parameter can be set here. Also, if you do not wish to use OpenType Layout features for some reason, the **Other** parameter can be used to describe stylistic variations in the character set such as “Alt” or “Swash” or “OsF”.

Families: Windows, Sun, NeXTSTEP

Making the typographic family name the same is all you need to do to create a font family for Windows, NeXTSTEP and Sun environments. For more information about these systems, see Chapter 9, “Generating and Exporting Fonts”.

Font families on the Macintosh

If your font is a logo typeface or one that contains special symbols, you probably don't need to make a family of fonts. In fact, many people use Fontographer to make only one version of a typeface. In these cases, family information is not important. But if you are ambitious, and have made more than one member of a font family, you can use Style Merger® or TransType Pro to bring those styles together into a family of Macintosh typefaces. This concerns only Mac Type 1 and TrueType font families. If you generated your fonts in the OpenType TT or OpenType PS format you then already have everything set and you do not need further manipulations.

Note: Style Merger, an old Macromedia utility, works only in classic Macintosh systems. In Mac OS X you can merge your fonts with the help of TransType Pro – another product from Fontlab Ltd.

Style Merger and TransType Pro take styled Macintosh screen fonts and merge them into one Macintosh screen font family. Macintosh font family relationships are coordinated by special resources located in the screen font suitcases (FOND). Style Merger and TransType Pro will work with both PostScript and TrueType fonts. They never affect any fonts or font files already on your system; they simply read existing fonts and create a new file containing a family.

This allows you to quickly and easily build a family containing Plain, Bold, Italic, and BoldItalic fonts. If you use Fontographer to create four styled fonts of the same typeface, Style Merger and TransType Pro can merge the four separate fonts into one family, thus saving space in your font menu.

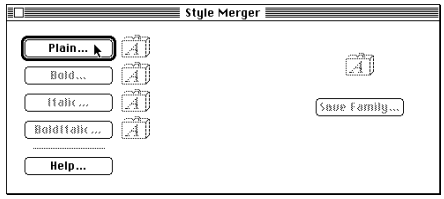
Before you run Style Merger or TransType Pro, generate any fonts you want to include in your family (see Chapter 9, "Generating and Exporting Fonts"). For this example, we started with these font files:

Now let's run Style Merger. It is so easy to use that the complete process takes only four steps.

To create a font family with Style Merger:

1. Double-click Style Merger icon.

Style Merger's main window appears.



2. Click the button marked **Plain** and then choose the bitmap or TrueType file that contains the Plain style of your font (*Nova-Normal.bmap* in this example).

3. Click the applicable buttons and then choose the appropriate bitmap font files for your Bold, Italic, and BoldItalic faces.

Note: If your typeface has fewer than the four base styles, then press only the relevant buttons, bypassing the button(s) that do not apply to your typeface.

4. Click the **Save Family** button and click **OK** to save your new family. You can even change the name of the suitcase file if you like. That's all there is to it.

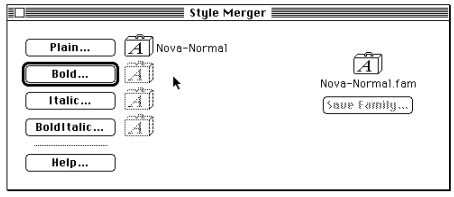
How Style Merger works

Style Merger's main window is mostly a collection of buttons. The button that you should click next always has a bold outline around it to guide you through the family building process (clicking the **Return** key always has the effect of clicking the currently bolded button). Buttons that you are not supposed to click yet are dimmed so that you can't click them by mistake.

On the left side of the window are the style buttons: **Plain**, **Bold**, **Italic**, and **BoldItalic**. When you start up Style Merger, **Plain** is the only button you can initially click (other than the **Help** button, which is always available). In addition, **Plain** has a bold outline around it, indicating that this is the button you should click first.

Clicking the **Plain** button begins the process of creating a family.

A file selection dialog box appears, allowing you to choose your fonts. Suppose you choose Nova-Normal, because that is the Plain style of Nova. Clicking **Plain** always wipes out any previously selected fonts and allows you to start over. So if you are making a family, and you choose the wrong fonts by accident, simply click **Plain** to automatically remove your mistakes and start over.

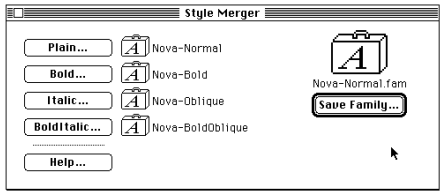


*Once you've chosen the Plain face and clicked **OK**, you will see the name of your Plain font appear to the right of the **Plain** button.*

After you choose the Plain bitmap, the little suitcase file image to the right of the button changes from gray to black, indicating that you've already selected a Plain font. In addition, a proposed new family name appears just above the **Save Family** button on the right. This name isn't set in stone: you can change it when you click the **Save Family** button. Notice that the **Save Family** button is disabled until you choose the second font to merge into the family.

Now that you've chosen a Plain face, Style Merger enables the **Bold**, **Italic**, and **BoldItalic** buttons, and it also puts a bold outline around the **Bold** button, directing you to click this button next. Of course, you can choose to add Bold, Italic, and BoldItalic styles in any order you want, so you can choose Italic next if you feel like being a rebel. To continue with the example, just click the **Bold** button and add fonts in order. Clicking the **Bold** button brings up the file selection dialog box again, just like it did for Plain. This time, choose the Bold style, which in this example is Nova-Bold. Once you have accepted the file selection dialog box, the name Nova-Bold appears next to the **Bold** button, and the **Save Family** button becomes active. This is because you could save the family at this point and be finished. This is useful when you don't have all four styles to merge into a family.

Style Merger bolds the **Italic** button next, so just choose Nova-Oblique as the italic face, and Nova-BoldOblique for the bold italic face. If you look at the Style Merger window, you can see that each style button has a name beside it, indicating that you can build a complete family now. And sure enough, Style Merger has bolded the **Save Family** button to tell you to do so. When you click that button, a dialog box appears asking where you want to save the family and what it should be called.



Typically, Style Merger suggests names like Nova-Normal.fam. This will be the name of the suitcase file Style Merger creates. The part of the name before the period is what the font will be called in your font menus (it becomes the name of the FOND resource). For instance, if you accept the name Nova-Normal.fam, Style Merger creates a file named "Nova-Normal.fam", and when you install that new family, you'll see Nova-Normal in your font menus. If you want to see just Nova, you should tell Style Merger to save the family as Nova.fam. After you press **OK** in the standard file dialog box, Style Merger builds the family and saves it. Then it cleans out all the font selections, and is ready to build another family. The window now looks exactly as it did when the program first started.

If you just want to build one family, you can quit now. To do so, type **COMMAND-Q**, choose **Quit** from the **File** menu, or simply click in the close box of Style Merger's window. If you want to create more families, you can leave Style Merger running to make some more.

Things you should know about Style Merger

Style Merger won't prevent you from choosing incorrect faces for particular styles. For instance, you could tell Style Merger to use Nova-Bold for the italic face. Your family will work just the way you set it up, but it would be a little confusing.

➲ ***Tip:*** If you make a mistake and choose the wrong font for a particular style, it's often easier to simply click **Plain** again and start from scratch. This is because fonts you've already selected will no longer appear in the file selection dialog box. This is really convenient for building a family because the names disappear once they've been used, but it is not so convenient when you make a mistake. Fortunately, choosing all the font names again takes only about 30 seconds.

Note: Please, do yourself a favor and always make sure the fonts you use to build families with are not installed. Always close any fonts installed with Suitcase, Master Juggler, Font Porter or whatever you installed them with; then build your family and reinstall your fonts. If you are using Style Merger and notice that the little suitcase icons change from capital "A's" to "B's", that means that the font you have just chosen was already installed. You may have to restart your Macintosh. If you have never heard of Suitcase, or Master Juggler, or installed fonts, then don't worry; you most likely will not encounter this situation.

Never try to open fonts that are in your System Folder.

Installing and Removing Fonts

You've devoted a lot of time to getting your font ready to use. You've tweaked, scaled, and skewed the characters; adjusted the kerning and spacing; printed out a sample to see what the characters look like; and finally generated the font. And since you're perfectly happy with your creation, you're ready to install the font and actually use it. Telling your system about your new font is the final step toward this goal. The manner in which the system is made aware of the font depends totally upon system configuration and computer platform. We'll examine the specific steps for each platform. You only need to install a font once to make it usable in all the programs on your computer.

11

Installing Macintosh Type 1 fonts

Macintosh Type 1 fonts consist of two parts: the **printer font file** which contains the PostScript outline data in the Type 1 format, and the **bitmap font file**, which is sometimes known as the font suitcase or the screen font.

In Mac OS X, the printer font files carry a font icon with the descriptor LWFN and the bitmap font files carry a font icon with the descriptor FFIL, and occasionally a ".bmap" file extension (but it is not mandatory).

Mac OS X doesn't use pfb, pfm, afm or inf files.

Installing Type 1 fonts in Mac OS X

If you double-click the suitcase font file in the Finder, a window opens in Font Book so you can preview the font. If you click **Install Font**, it's installed in the Library/Fonts/ folder in your home folder.

After you install a font, it appears in the All Fonts collection.

To make the font available to all users of the computer, drag it to the Computer folder in the Collection column of Font Book. You can change the default install location so that fonts are always available for all users of the computer. You can make this change in Font Book Preferences.

There are many third party utilities for installing and managing fonts in Mac OS X. Here are some names: Font Explorer X Pro, Suitcase Fusion, FontAgent Pro.

Installing Other fonts in Mac OS X

Mac TrueType, Windows OpenType TT (ttf) and OpenType PS (otf) fonts exist all in one file. There are no separate screen and printer files like there are for Type 1 fonts. Instead, everything comes in a single file (resource or data-fork based).

Note: When you install the font, both the outline and any bitmaps you have in the same file will be installed. The Macintosh Operating System will defer to the TrueType bitmaps over the TrueType outlines. Fontographer will not automatically generate TrueType bitmap sizes. You must specify the sizes for them to be included in the file if you need.

Installing other fonts in Mac OS X is the same as installing Type 1 fonts. You may refer to Mac OS X help for details.

Installing Windows fonts

Windows fonts generated by Fontographer can be used in any version of Windows starting from Windows 3.x but we will consider the most recent versions here: Windows XP, Windows Vista and Windows 7. Windows XP and later directly support Type 1, OpenType TT and OpenType PS fonts. There is a difference in the implementation of Type 1 and OpenType fonts in Windows, but they do have one similarity: neither depends upon a bitmap file for the screen image.

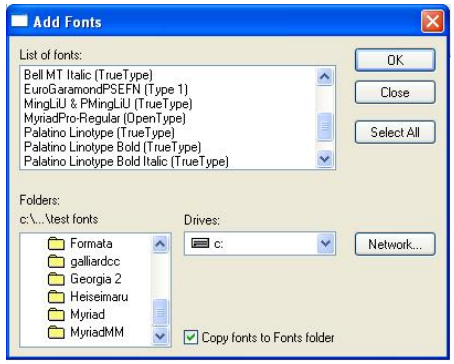
Multiple Master Type 1 fonts can be installed on Windows XP with the help of ATM 4.1 (Adobe Type Manager). Since ATM doesn't work properly on Windows Vista and was reported by Adobe as a discontinued project, Multiple Master fonts become more obsolete. Here is what Thomas Phinney said:

> "Well, the title pretty much says it all. ATM Light and Deluxe don't appear to work properly under Vista, and we don't currently have any plans to update them (we stopped selling and supporting ATM Deluxe quite some time back).
>
> However, multiple master (MM) fonts also don't really work at the system level under Windows without ATM (Light or Deluxe). With Adobe applications that use our shared font engine, you can still put MM fonts in a shared Adobe fonts folder, whether it's "C:\Program Files\Common Files\Adobe\Fonts" or the "Fonts" folder within an individual application folder. So it will still be possible to get MMs working under Vista for, say, InDesign, Illustrator and Photoshop. But not for Microsoft Office, QuarkXPress, Adobe Freehand, or many others.
>
> Sorry for the bad news. But I trust it was apparent that this sort of thing was coming sooner or later."

Installing fonts in Windows XP

To install both Type 1 and OpenType (ttf and otf) fonts follow these steps:

1. Open the Fonts control panel by choosing it from the **Start** menu. You'll see a fonts folder that lists all of the fonts currently installed on the system.
2. Select **Install New Font** from the **File** menu to bring up the Add Fonts dialog box.



   You'll see three windows labelled **List of fonts**, **Folders**, and **Drives** to help you in locating the fonts to be installed.

3. Find and click the folder where you told Fontographer to generate the fonts.

   The fonts will now display by name in the **List of fonts** window directly above the **Folders**.

4. Be sure that the **Copy fonts to Fonts folder** option is checked.
5. Select the fonts you want to install, or click **Select All**, click **OK** and the fonts will be copied into the system fonts folder.

   You can look in the fonts folder to verify that everything copied okay and you're done.

Installing fonts in Windows Vista/7

To install both Type 1 and OpenType (ttf and otf) fonts follow these steps:

1. Find and open the folder where you told Fontographer to generate the fonts.
2. Open the Fonts control panel by choosing it from the **Start** menu. You'll see a fonts folder that shows all of the fonts currently installed on the system.
3. Select and drag-drop your fonts to this folder. In the case of Type 1 fonts, select and drag-drop only the file with .pfm extension.

Another method:

1. Find and open the folder where you told Fontographer to generate the fonts.
2. Double-click on your font to open it for preview (or right-click and select Preview in the pop-up menu). In the case of Type 1 fonts, double-click on the file with .pfm extension.
3. In the font preview window click on the **Install** button to copy the font to the system Fonts folder.

   You can look in the fonts folder to verify that everything copied okay and you're done.

Removing installed fonts

It's a rare occasion when you want to remove a font from your computer. Removing a font is not necessarily the same as updating a font. On most platforms, replacing an old version of a font with a newer one is simply a matter of installing it again. If you do remove a font, any documents you've created with that font will revert back to some default font, a situation you should generally avoid. But if you don't have any documents that contain a certain font, you might want to remove it to save space on your computer's disk or to reduce clutter in your font menus. Disks containing fewer files generally work faster, and font menus with fewer fonts display more quickly. It's also true that programs start more slowly when there are lots of fonts installed. So there really are some valid reasons why you might want to remove unused fonts from your computer. Done carefully, font removal can speed your work and brighten your day.

Removing a Macintosh font

If the font was installed just by copying or moving it to the fonts folder:

1. Open one of the font folders (/Library/Fonts or /UserName/Library/Fonts).
2. Drag the font files (both bitmap and PostScript if it's a PostScript font) out of the Fonts folder and into some other folder on your disk or into the Trash.

If the font was installed with Font Book or some other font management utility then use this utility to disable or completely remove the font.

Removing a Windows font

If you were using Font Explorer or some other font utility for font installation then use it to disable or completely remove the font.

Otherwise:

1. Open the Fonts control panel by double-clicking its icon or choosing it in the **Start** menu.
2. Select the font you want removed.
3. Move it to some other place on your disk.
4. If you want the font permanently removed from your disk, move it to the Recycle Bin or choose **Delete** in the **File** menu.

OpenType Fonts

The OpenType font format, jointly developed by Microsoft and Adobe, allows us to combine the best features of the TrueType and Type 1 font formats.

OpenType fonts are stored in a single font file, use Unicode as their encoding and work in Windows and Mac OS X. They do not require bitmaps.

This all has been true for older TrueType fonts but the advantage of OpenType against older font formats is the support of layout features, which allow better typographic layout, and precise support of complex scripts.

OpenType fonts come in two sub-formats, so-called “flavors”:

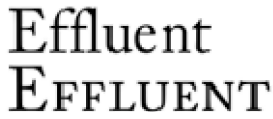
- **OpenType PS,** with the file extension “.otf”, also called OpenType-CFF or PostScript-flavored OpenType.
- **OpenType TT,** usually with the file extension “.ttf” (but the extension “.otf” is also permitted), also called TrueType-flavored OpenType. This format is backwards compatible with Windows TrueType (.ttf) fonts. In practical terms, any PC TrueType font is automatically an OpenType TT font.

12

Font Features

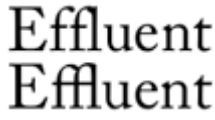
Both sorts of OpenType fonts may include so-called OpenType Layout features.

For example, the small caps layout feature (abbreviated *smcp*) may change all lowercase glyphs to their small caps counterparts.



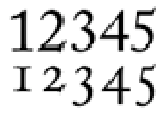
Small caps

The standard ligatures layout feature (abbreviated *liga*) can replace some letter combinations with ligatures.



Ligature

The old-style numerals layout feature (abbreviated *onum*) can replace lining figures with old-style figures.



Old Style Numerals

OpenType Layout features can serve typographic purposes like shown above. In this case, applications such as Adobe InDesign, Adobe Illustrator CS, Adobe Photoshop CS, Apple Pages or Apple Keynote on Mac OS X offer the user some interface to turn selected features on an off.

OpenType Layout features also play a crucial role in rendering complex scripts, i.e. writing systems such as Arabic, Devanagari or Thai. These writing systems have complex rules for displaying characters. For example Arabic uses different forms of letters if a letter is found at the beginning, in the middle or at the end of the word. Also, complex scripts often use vowel marks that are positioned dynamically over consonant letters. In all these cases, the layout features contain mapping rules that are automatically applied by the layout application.

Note that not all layout applications offer the same level of OpenType support. For example, Microsoft Word 2003 for Windows supports complex-script layout features for Arabic and Devanagari but does not support Western typographic layout features. Adobe InDesign CS2 U.S. English and Apple Keynote on Mac OS X support Western typographic layout features but do not support any complex-script layout features. Adobe InDesign CS Middle East edition supports Western and Arabic layout features, but does not support Devanagari.

Information about **using** OpenType fonts can be found at:

http://www.myfonts.com/info/opentype/
http://store.adobe.com/type/opentype/

Information about **developing** OpenType fonts can be found at:

http://www.microsoft.com/typography/SpecificationsOverview.mspx
http://www.microsoft.com/typography/developers/opentype/
http://partners.adobe.com/public/developer/opentype/

Probably the best thing about OT features is that they do not change the source string of characters.

Features and Lookups

Every feature consists of one or more *lookups*. A Lookup is an elementary procedure performed on a glyph sequence or positioning data. For instance, “replace the sequence of ‘f’ and ‘l’ with the ‘fl’ ligature glyph” is a lookup. A combination of similar lookups forms a feature.

The sequence of lookups is important. They are applied in the order they are defined. The sequence of features is also important, but the application or operating system may make changes in the feature preference.

By default all features and lookups are defined for the default language of the Latin script.

Scripts and Languages

The second great OpenType feature is support for multiple scripts and languages. With an OpenType font you can define different behaviors of the font when it is used to type text in different languages. For example, some ligatures that are necessary in English are not applicable to Turkish. Other features, like support for initial, medial and final forms of the characters are applicable only to Arabic script, and so on.

OpenType allows us to define script and language dependence at the lookup level, so the same feature may work differently when different languages are supported.

Feature Definition Language

Information about OpenType features is stored in a binary form inside the font file. This is not easy to modify and not easy to handle with visual tools (like the tools that Fontographer provides to edit outlines that are also stored in a binary form).

To define features in human-readable form Adobe has developed the feature definition language (FEA). It is very easy to read and it is the most compact way to represent OpenType font features.

Let's take a simple example: a ligature feature that covers the basic "fi" and "fl" ligatures that are present in almost every Western font. In feature-definition language this feature will be defined as follows:

```
feature liga{
   sub f i by fi;
   sub f l by fl;
} liga;
```

Other possible features are defined in a similar way, keeping the feature definition both compact and readable.

When Fontographer generates an OpenType font file it can try to compile the feature file (.fea) into the binary OpenType tables. With a few exceptions it works for most possible combinations of substitution and/or positioning features supported in Adobe OTFDK v2.5.

In the following sections we will describe the feature definition language in more detail. The next section covers the basic rules of the language.

Language Syntax

Information in this section is partially taken from the official Feature File Format specification by Adobe with their permission. Only those parts of the language that are supported by Fontographer are described.

Comments

The "#" character indicates the start of a comment; the comment extends until the end of the line.

Special characters

#	pound sign	Denotes start of comment
;	semicolon	Terminates a statement
,	comma	Separator in various lists
@	at sign	Identifies glyph class names
\	backslash	Distinguishes glyph names from an identical keyword
-	hyphen	Denotes glyph ranges in a glyph class
=	equal sign	Glyph class assignment operator
'	single quote	Marks a glyph or glyph class for contextual substitution or positioning
"	double quote	Marks a glyph or glyph class for contextual substitution or positioning
{ }	braces	Enclose a feature, lookup, table, or anonymous block
[ ]	square brackets	Enclose components of a glyph class
< >	angle brackets	Enclose a device, value record, contour point, anchor, or caret
( )	parentheses	Enclose the file name to be included

Number

A <number> is a signed decimal integer (without leading zeroes). For example:

```
-150
1000
```

Glyphs

These are represented by the glyph name. A glyph name may be up to 31 characters in length, must be entirely comprised of characters from the following set:

A-Z
a-z
0-9
. (period)
_ (underscore)

and must not start with a digit or period. The only exception is the special glyph ".notdef".

"twocents", "a1", and "_" are valid glyph names. "2cents" and ".twocents" are not.

An initial backslash serves to differentiate a glyph name from an identical keyword in the feature file language. For example, a glyph named "table" must be specified in the feature file as:

```
\table
```

Glyph classes

A feature file glyph class, <glyphclass>, represents a single glyph position in a sequence and is denoted by a list of glyphs enclosed in square brackets.

For example:

```
[endash emdash figuredash]
```

An example of a sequence that contains a glyph class is:

```
space [endash emdash figuredash] space
```

This would match any of the 3 sequences "space endash space", "space emdash space", or "space figuredash space" during OpenType layout.

A feature file glyph class that contains only one single glyph is known as a singleton glyph class.

A feature file glyph class is also used to represent the set of alternate glyphs in an alternate substitution lookup type rule.

Ranges

A range of glyphs is denoted by a hyphen:

```
[<firstGlyph> - <lastGlyph>]
```

Spaces around the hyphen are not required since hyphens are not permitted in feature file glyph names. For example:

```
[A-Z]
```

Named glyph classes

A glyph class can be named by assigning it to a glyph class name, which begins with the "@" character, and then referred to later on by the glyph class name. For example:

```
@dash = [endash emdash figuredash];          # Assignment
space @dash space # Usage
```

The part of the glyph class name after the "@" is subject to the same name restrictions that apply to a glyph name, except that its maximum length is 30.

Glyph class assignments can appear anywhere in the feature file. A glyph class name may be used in the feature file only after its definition.

When a glyph class name occurs within square brackets, its elements are simply added onto the other elements in the glyph class being defined. For example:

```
@Vowels.lc = [a e i o u];
@Vowels.uc = [A E I O U];
@Vowels = [@Vowels.lc @Vowels.uc y Y];
```

Here the last statement is equivalent to:

```
@Vowels = [a e i o u A E I O U y Y];
```

No square brackets are needed if a glyph class name is assigned to another single glyph class name. For example:

```
@Figures_lining_tabular = @FIGSDEFAULT;
```

Ranges, glyphs, and glyph class names can be combined in a glyph class. For example:

```
[zerooldstyle - nineoldstyle ampersandoldstyle          @smallCaps]
```

In Fontographer, you can define glyph classes in the external feature file.

Including files

Including files is accomplished by the directive:

```
include(<filename>)
```

In this Fontographer implementation an included file must be located in the same folder.

A maximum include depth of 5 ensures against infinite include loops (files that include each other).

Specifying features

Each feature is specified in a feature block:

```
feature <feature tag> {
   # specifications go here
} <feature tag>;
```

For example:

```
feature liga {
   # ...
} liga;
```

A feature file "rule" is a statement that specifies glyph substitution or glyph positioning. A feature block may contain glyph substitution rules, glyph positioning rules, or both.

Language system

In practice, most or all of the features in a font will be registered under the same set of language systems, and a particular feature's lookups will be identical across the language systems that the feature is registered under.

The "languagesystem" statement takes advantage of this fact. It is the simplest way to specify a language system in the feature file. One or more such statements may be present in the feature file at global scope (i.e. outside of the feature blocks or any other blocks) and before any of the feature blocks:

```
languagesystem <script tag> <language tag>;
```

When these statements are present, then each feature that does not contain an explicit "script" or "language" statement will be registered under every language system specified by the "languagesystem" statement(s).

If no "languagesystem" statement is present, then the implementation will behave exactly as though the following statement were present at the beginning of the feature file:

```
languagesystem latn DFLT;
```

Script and Language

Occasionally you may need to specify a feature whose lookups vary across the language systems of the feature, or whose language systems vary from the set of language systems of the rest of the features in the file (specified by the "languagesystem" statements).

In these cases, the "script" and "language" statements should be used within the feature block itself. (A "script" and/or "language" statement must be present before the first rule in the feature in order to indicate to the feature file parser that this feature is not to be registered under the language systems specified by the "languagesystem" statements).

The feature's lookups will be registered under the script and language attributes current at the definition of the lookup. The attributes may be changed as follows:

"script" statement:

```
script <script tag>;
```

For example:

```
script kana;
```

When a "script" statement is seen, the language attribute is implicitly set to 'DFLT', and the lookupflag attribute is implicitly set to 0. The script attribute stays the same until explicitly changed by another "script" statement or until the end of the feature.

"language" statement:

The language attribute stays the same until explicitly changed, until the script is changed, or until the end of the feature. To change the language attribute, use the "language" statement:

```
language <language tag> [excludeDFLT|includeDFLT] [required];
```

The script and lookupflag attributes stay the same as before. (If no "script" assignment statement has been seen thus far in the feature block, then the script attribute is set to 'latn', but it is recommended that an explicit "script" statement be used in such cases for clarity.)

Here is an example statement:

```
language DEU;
```

As a result of this statement, (a) the language attribute is changed to 'DEU ', and (b) the 'DFLT' lookups of the current script are automatically included into the language system specified by the current script and language attributes. If (b) is not desired, as may occasionally be the case, then the keyword "excludeDFLT" must follow the language tag. For example:

```
language DEU excludeDFLT;
```

The keyword "includeDFLT" may be used to explicitly indicate the default 'DFLT' lookup-inheriting behavior. For example:

```
language DEU includeDFLT; # Same as: language DEU;
```

lookupflag

The chapter "Common Table Formats" in the OpenType Font File Specification describes the LookupFlag field in the Lookup table.

The lookupflag attribute defaults to 0 at the start of a feature block.

The lookupflag attribute stays the same until explicitly changed, until a lookup reference statement is encountered that changes it, until the script is changed, or until the end of the feature.

To change the lookupflag attribute explicitly, use the lookupflag statement, which takes two formats:

lookupflag format A:

```
lookupflag <named lookupflag value> (, <named lookupflag value>)*;
```

Here, the individual lookup flag values to be set are expressed in a comma-separated list of one or more <named lookupflag value>s, in no particular order. A <named lookupflag value> is one of the following:

```
RightToLeft
IgnoreBaseGlyphs
IgnoreLigatures
IgnoreMarks
```

At most one of each of the above 5 kinds of <named lookupflag value> may be present in a lookupflag statement. For example, to skip over base glyphs and ligature glyphs:

lookupflag IgnoreBaseGlyphs, IgnoreLigatures;

lookupflag format B:

```
lookupflag <number>;
```

Here the entire lookup flag value is specified simply as a <number>. The format A example above could equivalently be expressed as:

```
lookupflag 6;
```

Format A is clearly a superior choice for human readability when the lookupflag value is not 0. However, a lookupflag value of 0 can be set only with format B, not with format A:

```
lookupflag 0;
```

lookup

You can label a set of rules and refer to it explicitly later on; in order to have different parts of the font tables refer to the same lookup. This decreases the size of the font in addition to freeing the editor from maintaining duplicate sets of rules.

To define and label a lookup, use a named lookup block:

```
lookup <label> {
   # rules to be grouped
} <label>;
```

To refer to the lookup later on, use a lookup reference statement:

```
lookup <label>;
```

For example:

```
lookup SHARED {    # lookup definition
   # ...
} SHARED;
# ...
lookup SHARED;     # lookup reference
```

Since the labelled block literally defines a single lookup in the font, the rules within the lookup block must be of the same lookup type and have the same lookupflag attribute. The lookup block must be specified within a feature block and may not contain any other kind of block.

Importing OpenType Fonts

Fontographer 5 **can** now import and generate OpenType PS (.otf) fonts. But it **cannot** read and decompile OpenType Layout features.

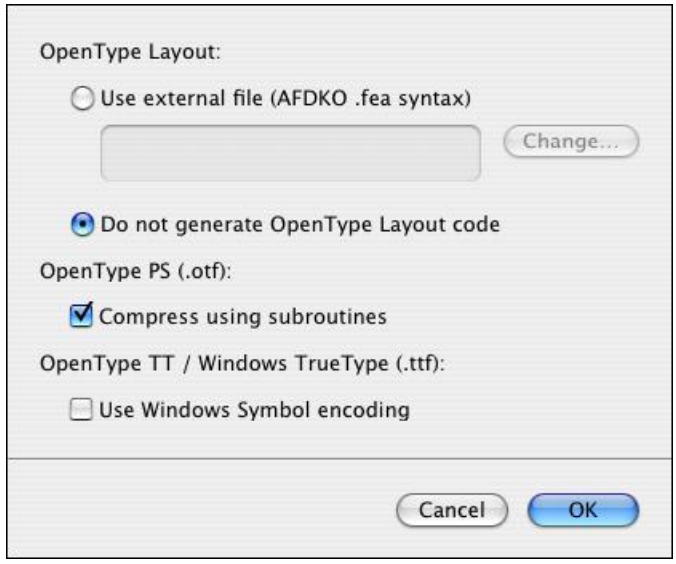
To open an OpenType PS font, choose **Open Font** or **Command-O** from the **File** menu. A file selection dialog box appears that works in the standard fashion. Once the selection dialog box is open, you can select a font file by clicking its name and then **Open**, or simply by double-clicking its name.

Fontographer shows the font contents in the font window. It also reads OpenType 'kern' feature code from the font and tries to decompile it to build a list of kerning pairs. You can then edit kerning in the Metrics window.

If you now export the font back into the OpenType PS format it will loose all its original OpenType Layout features found in the source font except kerning. Imported kerning will be automatically transformed into the new 'kern' feature.

Generating OpenType Fonts

When you generate your font in the OpenType TT/TrueType or OpenType PS format you can choose whether you want or not to include OpenType features in your font.



Choose the **Do not generate OpenType Layout code** option to generate font without OpenType features or choose the **Use external file (AFDKO .fea syntax)** option and select the external .fea file where you have the code for OpenType features prepared.

It is useful to name your file with features the same as your .fog database. For example, if your font file was named *MyFirstFont.fog* then name the file with features *MyFirstFont.fea*.

Fontographer will generate an OpenType "kern" feature in the GPOS table regardless of the option selected. Surely kerning pairs must be defined in the Metrics window in advance.

Preparing OpenType features

In the following sections we will discuss the different types of substitution and positioning lookups. Please note that you may combine substitution and positioning lookups in the same feature.

The algorithm that Fontographer 5 uses to generate OpenType layout code relies primarily on glyphnames found in the font. To generate proper OpenType features you must name your glyphs properly and write proper code.

Naming your glyphs

Each glyph in a font must have a unique glyphname, i.e. each glyphname may only occur once in a font.

Any glyphname must not be longer than 31 characters. The glyphname consists of a **basename**, optionally followed by a period (.), which is then followed by a **suffix**. Both the basename and the suffix may only include: uppercase English letters (A-Z), lowercase English letters (a-z), European digits (0-9), and underscore (_). Other characters such as spaces are not permitted. A glyphname must start with a letter or the underscore character — with the exception of the special glyphname ".notdef" that must start with the period. A glyphname must not start with a digit.

For example, "twocents", "a1", "a.1", and "_" are well-formed glyphnames (according to those minimal limitations), while "2cents", "two cents" and ".twocents" are not well-formed.

Substitution lookups

Substitution lookups (GSUB) deal with the replacement of the source glyph(s) with some other glyph(s).

The simplest example of a substitution lookup is the replacement of lowercase characters by small-caps versions.

Lookups may be *context-free* or *context-dependent*. Context-free lookups are applied every time the source sequence of glyphs is present, like when you want to replace the 'f' and 'l' sequence with the 'fl' ligature. In other cases you may need to apply a substitution only when a source sequence of glyphs is surrounded by some other glyphs. For instance, you may want to replace an uppercase character with a lowercase when it is followed by another lowercase character.

The OpenType specification declares the following types of basic substitutions:

Single	Replaces a single glyph with another single glyph: a->A
Ligature	Replaces multiple glyphs with a single glyph: f l -> fl
Multiple	Replaces a single glyph with multiple glyphs: $ -> d o l l a r
Alternate	Replaces a single glyph with one of the glyphs in a list: A -> A.version1 or A.version2

All these substitutions may be context-free or context-dependent.

Fontographer 5 supports all of the substitutions supported in AFDKO v2.5.

Single Substitution

This is the simplest substitution – it replaces a single glyph with another glyph or a class of glyphs with another class. The Class form of the substitution requires that the number of glyphs in the source and destination classes be the same.

A single substitution rule is specified in one of the following formats:

substitute <glyph> **by** <glyph>; # format A
substitute <glyphclass> **by** <glyph>; # format B
substitute <glyphclass> **by** <glyphclass>; # format C

You can use the codeword "sub" instead of the longer "substitute".

Format B specifies that all glyphs in the target glyph class will be replaced by the same replacement glyph.

Format C specifies that any of the glyphs in the target glyph class must be replaced by its corresponding glyph (in the order of glyphs in the glyph classes) in the replacement glyph class. If the replacement is a singleton glyph class, then the rule will be treated identically to a format B rule. If the replacement class has more than one glyph, then the number of elements in the target and replacement glyph classes must be the same.

For example:

```
sub a by a.smcp;    # format A
sub [one.fitted one.onum one.taboldstyle] by one;    # format B
sub [a - z] by [a.smcp - z.smcp];    # format C
sub @Capitals by @CapSwashes;    # format C
```

The third line in the above example produces the same effect in the font as:

```
sub a by a.smcp;
sub b by b.smcp;
sub c by c.smcp;
# ...
sub z by z.smcp;
```

Sample applications of this rule:

Replace different types of figures:

```
sub @figs_lnum by @figs_onum;
```

Replace lowercase glyphs by small-caps:

```
sub @lc by @sc;
```

Small SMALL

Ligature Substitution

The ligature substitution rule replaces several glyphs in sequence with a single glyph.

A Ligature substitution rule is specified as:

substitute <glyph sequence> **by** <glyph>;

<glyph sequence> must contain two or more <glyph|glyphclass>es. For example:

```
substitute [one one.onum] [slash fraction] [two two.onum] by onehalf;
```

Since the OpenType specification does not allow ligature substitutions to be specified on target sequences that contain glyph classes, the implementation software will enumerate all specific glyph sequences if glyph classes are detected in <glyph sequence>. Thus, the above example produces the same effect in the font as if the font editor manually enumerated all the sequences:

```
substitute one slash two by onehalf;
substitute one.onum slash two by onehalf;
substitute one fraction two by onehalf;
substitute one.onum fraction two by onehalf;
substitute one slash two.onum by onehalf;
substitute one.onum slash two.onum by onehalf;
substitute one fraction two.onum by onehalf;
substitute one.onum fraction two.onum by onehalf;
```

***Note*:** Variant glyphs should be named just like their default counterparts but with a suffix appended after a period. The suffix can typically be the feature tag for the layout feature the variant glyph will be most likely accessed with. So, a small cap a can be named a.smcp and an old-style digit 2 can be named two.onum. Non-standard names should be avoided.

Almost all fonts contain at least two ligatures: “fl” and “fi” which can be easily encoded as:

```
substitute f l by fl;
substitute f i by fi;
```

Some fonts add longer ligatures:

```
substitute f f i by f_f_i;
```

***Note*:** Ligature glyphs should be named using the underscore rule, e.g. f_f_odieresis for an ffö ligature. Only the fi and fl ligatures should be named without the underscores.

Multiple Substitution

The multiple substitution rule replaces a single glyph with a sequence of glyphs. It is specified as:

substitute <glyph> **by** <glyph sequence >;

<glyph sequence> contains two or more glyphs. It may not contain glyph classes. (If it did, the rule would be ambiguous as to which replacement sequence was required.) For example:

```
sub f_f_l by f f l;
sub $ by d o l l a r;
```

Note that "d o l l a r" is not the word "dollar" but the sequence of glyph names "d", "o", "l", "a", "r".

Alternate Substitution

Alternate substitution replaces a glyph with one of the glyphs in a pre-defined list of alternatives. The application that uses the font is expected to decide which glyph to choose. A good example of this lookup is to provide several versions of some glyph, like the ampersand. Another application is the selection of several different forms of ornaments.

An alternate substitution rule is specified as:

substitute <glyph> **from** <glyphclass>;

For example:

```
substitute ampersand from [ampersand.1 ampersand.2];
```

or ornament variations:

```
sub asterisk from [asterisk.ornm1 asterisk.ornm2 asterisk.ornm3 asterisk.ornm4];
```

Context Dependent Substitutions

A context-dependent rule can be any of the rules described above with one important difference: it defines a context that must include a target sequence of glyphs (or glyph classes).

In the simple form of, say, ligature substitution we simply write:

```
sub a b c by D;
```

In context-dependent substitution we can declare that “abc”, which is a target sequence of glyphs for a ligature substitution rule, must be a part of a larger context:

```
sub period a' b' c' period by D;
```

Only when “abc” is surrounded by two “period” glyphs will substitution take place. Note that we have marked the target glyphs with the single quote character positioned immediately after the glyph name.

The rule is specified as follows:

substitute <marked glyph sequence> # Target sequence with marked glyphs
by <glyph sequence>; # Sub-run replacement sequence

A <glyph sequence> comprises one or more glyphs or glyph classes.

<marked glyph sequence> is a <glyph sequence> in which a set of glyphs or glyph classes is identified, i.e. "marked". We will call this marked set of glyphs a sub-run. A sub-run is marked by inserting a single quote (') after each of its member elements.

This sub-run represents the target sequences of the lookups called by this rule. The lookup type of the lookup called by this rule is auto-detected from their target and replacement sequences in the same way as in their corresponding stand-alone (i.e. non-contextual) statements.

Example 1. This calls a lookup. The rule below means: in sequences "a d" or "e d" or "n d", substitute "d" by "d.alt".

```
substitute [a e n] d' by d.alt;
```

Example 2. This also calls a *single substitution* lookup. The rule below means: if a capital letter is followed by a small capital, then replace the small capital by its corresponding lowercase letter.

```
substitute [A-Z] [a.smcp-z.smcp]' by [a-z];
```

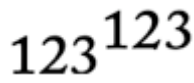
Example 3. This calls a *ligature substitution* lookup. The rule below means: in sequences "e t c" or "e.init t c", substitute the first two glyphs by the ampersand.

```
substitute [e e.init]' t' c by ampersand;
```

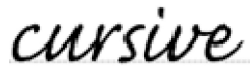
Positioning lookups

The OpenType specification allows you to define many positioning lookups (GPOS). The lookup types may be separated into three groups:

1. Basic lookups, single and pair positioning:



2. The Cursive attachment lookup that allows smooth connection of script and cursive glyphs:



3. Mark attachment lookups that define relative positions of glyphs and marks:

Fontographer 5 supports all of the positioning lookups supported in AFDKO v2.5 (http://partners.adobe.com/public/developer/opentype/). As with substitution lookups you must write positioning code in the external .fea file in order Fontographer could compile it at the font export stage.

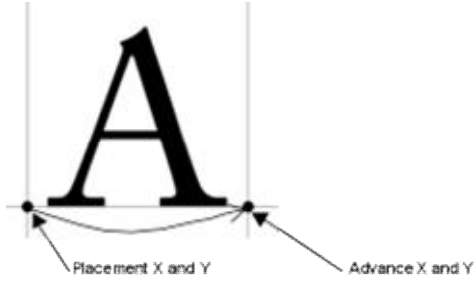
As in the case with substitution, positioning lookups may be *context-free* and *context-dependent*. Context-dependent lookups are the same as context-free but add a context that is verified against the source sequence of glyphs. Only when the context is matched is positioning performed.

Please note that glyph positioning is performed **after** substitution and that all positioning lookups must be defined for the glyph string that is a result of substitution.

Glyph positioning rules begin with the keyword "position"; this keyword may be abbreviated as "pos". (The "enumerate" or "ignore" keywords may precede the "position" keyword in some cases.) The GPOS lookup type is auto-detected from the format of the rest of the rule.

Glyph Geometry

Positioning lookups may change one of the glyph positioning metrics:



A single positioning lookup may tweak any of four values: placement_X, placement_Y, advance_Y and advance_Y.

Modification of the origin point of the glyph will shift it and all following glyphs. Modification of the advance vector will shift the next glyph in the glyph string.

Value Record

A <valuerecord> is used in positioning rules to define offsets to shift glyph origin or advance vector. It must be enclosed by angle brackets, except for format A, in which the angle brackets are optional. Note that the <metric> adjustments indicate values (in design units) to add to (positive values) or subtract from (negative values) the placement and advance values provided in the font (in the 'hmtx' and 'vmtx' tables).

Value record format A:

< <metric> > # Angle brackets are optional

Here the <metric> represents an X advance adjustment, except when used in the 'vkrn' feature, in which case it represents a Y advance adjustment. All other adjustments are implicitly set to 0. This is the simplest feature file <valuerecord> format, and is provided since it represents the most commonly used adjustment (i.e. for kerning). For example:

```
-3# without <>
<-3>        # with <>
```

Value record format B:

< <metric> <metric> <metric> <metric> >

Here, the <metric>s represent adjustments for X placement, Y placement, X advance, and Y advance, in that order. For example:

```
<-80 0 -160 0>      # X placement adj: -80; X advance adj: -160
```

Anchor

An <anchor> is used in some positioning rules. It takes 5 formats:

Anchor format A:

< anchor <metric> <metric> > #X coordinate, Y coordinate

For example:

```
<anchor 120 -20>
```

Anchor format B:

<anchor <metric> <metric> #X coordinate, Y coordinate

<contour point> >

For example:

```
<anchor 120 -20 contourpoint 5>
```

Anchor format C:

<anchor <metric> <metric> # X coordinate, Y coordinate

<device><device> > # X coord device, Y coord device

For example:

```
<anchor 120 -20 <device 11 1> <device NULL>>
```

Anchor format D, the null anchor:

<anchor NULL> # Anchor not defined

Anchor format E, the named anchor:

<anchor <name> >

For example:

```
<anchor TOP_ANCHOR_1>
```

Single Positioning

A Single Pos rule is specified as:

position <glyph|glyphclass> <valuerecord>;

Here, the <glyph|glyphclass> is adjusted by the <valuerecord>. For example, to reduce the left and right sidebearings of a glyph each by 80 design units:

```
position one <-80 0 -160 0>;
```

To shift the glyph up by 100 units:

```
position A <0 100 0 -100>;
```

Note that we changed the placement by 100 units but compensated for that with a -100 change applied to advance. This is needed to shift only the current glyph and not the following glyphs in the string.

Pair Positioning

Rules for this lookup type are usually used for kerning and must follow this format:

position <glyph|glyphclass> <glyph|glyphclass> <valuerecord format A>;

This format is provided since it closely parallels the way kerning is expressed in a plain pair kerning table. Here, the <valuerecord> must be of value record format A only, and corresponds to the first <glyph|glyphclass>.

Kerning can most easily be expressed with this format. This will result in adjusting the first glyph's X advance, except when in the 'vrkn' feature, in which case it will adjust the first glyph's Y advance. Some examples:

```
pos T a -100;          # specific pair (no glyph class present)
pos [T] a -100;        # class pair (singleton glyph class present)
pos T @a -100;         # class pair (glyph class present, even if singleton)
pos @T [a o u] -80;# class pair
```

Note that if at least one glyph class is present (even if it is a singleton glyph class), then the rule is interpreted as a class pair; otherwise, the rule is interpreted as a specific pair.

In the 'kern' feature, the specific glyph pairs will typically precede the glyph class pairs in the feature file, mirroring the way that they will be stored in the font.

```
feature kern {
  # specific pairs for all scripts
  # class pairs for all scripts
  } kern;
```

Enumerating Pairs

If some specific pairs are more conveniently represented as a class pair, but the editor does not want the pairs to be in a class kerning subtable, then the class pair must be preceded by the keyword "enumerate" (which can be abbreviated as "enum"). The implementation software will enumerate such pairs as specific pairs. Thus, these pairs can be thought of as "class exceptions" to class pairs. For example:

```
@Y_LC = [y yacute ydieresis];
@SMALL_PUNC = [comma semicolon period];
enum pos @Y_LC semicolon -80;       # specific pairs
pos f quoteright 30;            # specific pair
pos @Y_LC @SMALL_PUNC -100;       # class pair
```

The enum rule above can be replaced by:

```
pos y semicolon -80;
pos yacute semicolon -80;
pos ydieresis semicolon -80;
```

without changing the effect on the font.

Subtable Breaks

The implementation software will insert a subtable break within a run of class pair rules if a single subtable cannot be created due to class overlap. A warning will be given. For example:

```
pos [Ygrave] [colon semicolon] -55; # [line 99]          In first subtable
pos [Y Yacute] period -50;  # [line 100]        In first subtable
pos [Y Yacute Ygrave] period -60;    # [line 101]        In second subtable
```

will produce a warning that a new subtable has been started at line 101, and that some kern pairs within this subtable may never be accessed. The pair (Ygrave, period) will have a value of 0 if the above example comprised the entire lookup, since Ygrave is in the coverage (i.e. union of the first glyphs) of the first subtable.

Sometimes the class kerning subtable may get too large. The editor can force subtable breaks at appropriate points by inserting the statement:

```
subtable;
```

between two class kerning rules. The new subtable created will still be in the same lookup, so the editor must ensure that the coverages of the subtables thus created do not overlap. For example:

```
pos [Y Yacute] period -50;  # In first subtable
subtable;  # Force a subtable break here
pos [A Aacute Agrave] quoteright -30;          # In second subtable
```

If the subtable statement were not present, both rules would be represented within the same subtable.

Cursive attachment positioning

A Cursive Pos rule is specified as:
position cursive <glyph|glyphclass> <anchor> # Entry anchor <anchor>; # Exit anchor

The first <anchor> indicates the entry anchor point for <glyph|glyphclass>; the second, the exit anchor point.
For example, to define the entry point of glyph meem.medial to be at x=500, y=20, and the exit point to be at x=0, y=-20:

```
pos cursive meem.medial <anchor 500 20> <anchor 0 -20>;
```

A glyph may have a defined entry point, exit point, or both. <anchor> format D, the null anchor, must be used to indicate that an <anchor> is not defined.

```
pos cursive meem.end <anchor 500 20> <anchor NULL >;
```

Known Features

OpenType feature processing is an application-centric system. The application knows which features it needs to apply and then searches the font for these features. If a feature is present in the font, it is applied. To implement this system features must be standardized and registered so everybody will know what is done by a feature with a particular name.

The Microsoft Typography group performs feature registration:

http://www.microsoft.com/typography

The full list of registered features may be found in this document:

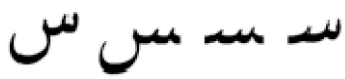
OpenType Layout tag registry
http://www.microsoft.com/typography/otspec/ttoreg.htm

This document contains descriptions of all name tags that can be used in an OpenType font for script, language and feature names.

We will provide brief definitions of the most commonly used features.

init, medi, fina and isol Features

These features are most commonly used in Arabic scripts and allow one to define four forms of Arabic characters: initial, final, medial and isolated:



Isolated Final Medial Initial

Note that Arabic text is written right-to-left.

Please note that it is the application or the operating system that detects word boundaries and applies one of the features. You are not expected to pay attention to that in the feature definition.

Usually these features are simple substitutions:

```
feature init{
   sub @isolated_forms by @initial_forms;
}init;
```

Latin Features

cpsp	Capital Spacing	Globally adjusts inter-glyph spacing for all-capital text.
		`feature cpsp {`
		`  pos @uppercase <7 0 14 0>;`
		`} cpsp;`

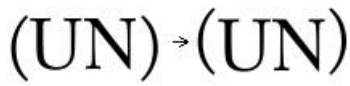
pnum	Proportional Figures	Replaces figure glyphs set on uniform (tabular) widths with corresponding glyphs set on glyph-specific (proportional) widths.
		`feature pnum {`
		`  sub @figures by @figures_prop;`
		`} pnum;`

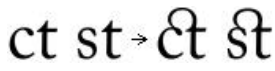
lnum	Lining Figures	This feature changes selected figures from oldstyle to the default lining form

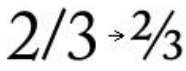
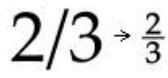
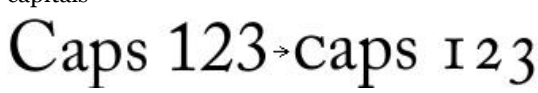
hist	Historical Forms	Replaces glyphs with their historical forms, like long form of s or Fraktur form of k.
		`feature hist {`
		`  sub s by longs;`
		`} hist;`

ordn	Ordinals	Replaces default alphabetic glyphs with the corresponding ordinal forms for use after figures
smcp	Small Capitals	This feature replaces lowercase characters with small capitals
sinf	Scientific Inferiors	Replaces lining or oldstyle figures with inferior figures (smaller glyphs which sit lower than the standard baseline, primarily for chemical or mathematical notation)
ornm	Ornaments	This feature lets the user access ornament glyphs in the font

liga	Standard Ligatures	Replaces a sequence of glyphs with a single glyph, which is preferred for typographic purposes. This feature covers the ligatures that the designer/manufacturer judges should be used in normal conditions
		`feature liga {`
		`  sub f f i by f_f_i;`
		`  sub f i by fi;`
		`  sub T h by T_h;`
		`  sub f f j by f_f_j;`
		`  sub f f l by f_f_l;`
		`  sub f f by f_f;`
		`  sub f j by f_j;`
		`  sub f l by fl;`
		`} liga;`

case	Case-Sensitive Forms	Shifts various punctuation marks up to a position that works better with all-capital sequences or sets of lining figures; also changes oldstyle figures to lining figures 

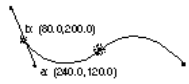
dlig	Discretionary Ligatures	Replaces a sequence of glyphs with a single glyph, which is preferred for typographic purposes. This feature covers those ligatures that may be used for special effect, at the user's preference
		`feature dlig {`
		`  sub c t by c_t;`
		`  sub s t by s_t;`
		`  sub longs h by longs_h;`
		`  sub longs i by longs_i;`
		`  sub longs l by longs_l;`
		`  sub longs t by logs_t;`
		`  sub longs longs by longs_longs;`
		`} dlig;`
		 

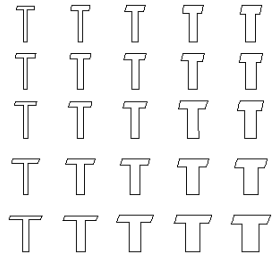
frac	Fractions	Replaces figures separated by a slash with 'common' (diagonal) fractions 
afrc	Alternative Fractions	Replaces figures separated by a slash with an alternative form 
dnom	Denominators	Replaces selected figures that follow a slash with denominator figures
c2sc	Small Capitals From Capitals	This feature replaces capital characters with small capitals 

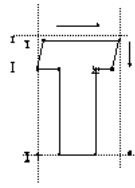
numr	Numerators	Replaces all with numerator figures
onum	Oldstyle Figures	This feature changes selected figures from the default lining style to oldstyle form
sups	Superscript	Replaces lining or oldstyle figures with superior figures and replaces lowercase letters with superior
sinf	Scientific Inferiors	Replaces lining or oldstyle figures with inferior figures.

Expert Advice

The Preferences dialog box is like the dials on your television: it's where you control the way Fontographer looks and behaves, customizing your work environment. Many of the commands and tools behave in several different ways, and you can easily choose between the various items in Preferences. Items chosen in Preferences are remembered, so the next time you run Fontographer, your customizations remain set.







13

Choose **Preferences** from the **Fontographer** menu.

A dialog box will appear and images at the top of the dialog box allow you to navigate through the various Preferences screens. Preferences are divided into six areas:

General preferences

International

Use this setting if you are going to work with Asian CJK fonts. It allows you to define vertical kerning and vertical **glyph metrics**: the vertical advance "width" (called *vertical advance vector*) for Asian glyphs used to type in vertical direction from top to bottom. If this option is on the additional "Vertical" button appears in the Metrics window:

Autosaving

Use the Autosave function that will periodically save the current font to protect your work from program or system crashes.

Use the check box to activate Autosave and enter the time interval (in minutes) at which you want to save the font. The font will be saved into the same folder where your font was last saved and will be named using the following structure:

FontName-Regular(Autosaved).fog

If Autosave was active and you have a system or program crash, you can open your last autosaved font.

Undo Settings

This is where you can set the number of undo levels and redo levels. You can have Fontographer remember up to 256 things to undo or redo. The penalty for this is, as usual, memory: the more undo levels you request, the more memory Fontographer will use. We recommend setting this preference between 10 and 100, depending upon how often you like to undo (and how far back you like to go).

When Fontographer runs out of memory (if this ever will happen in Mac OS X), it can throw away undo levels to free up some memory. If this happens Fontographer asks if you want Fontographer to throw away the undo levels.

Sounds

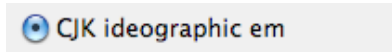
Fontographer 5 has the ability to play sounds during certain actions. It can play sounds when you snap to a point or snap to a guide in the outline window, and it can play a sound whenever the Please Wait dialog box closes. To turn sounds on, click on the **Application sounds**.

Fonts preferences

Linespacing Auto-Calculation

These two options define how Fontographer behaves when it needs to import, calculate and export different linespacing font values. Unless you do not need this for some special purpose leave the first option selected. This will keep linespacing in your font the same for all platforms and applications.

If **Asian CJK editing** option in General Preferences is on additional linespacing option becomes available:



Use this option when working with Asian CJK fonts.

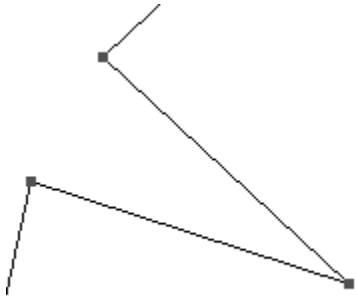
Opening Type 1 fonts

Many PostScript Type 1 fonts are saved using the Adobe Standard Encoding. This preference item allows you to either see the font in its original encoding (which is Adobe Standard), or to change native encoding to the default encoding, which is MacOS Roman on Macintosh.

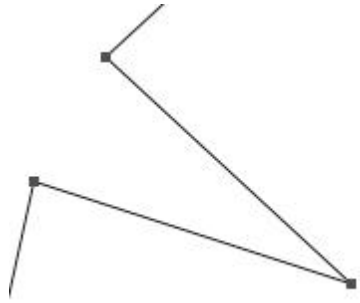
Editing preferences

Smooth outline

This setting lets you to select between non-anti-aliased and anti-aliased rendering of the outline. It doesn't influence the generated font but only the screen preview.

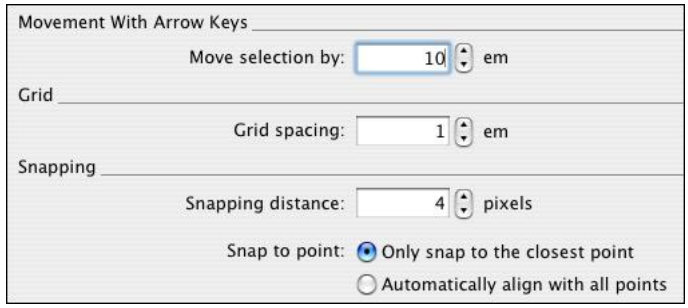


smoothing is off



smoothing is on

Distances



The **Move selection by** field lets Fontographer move points around by specified amounts when you press the ARROW keys. For instance, if you set this value to be 2.7, every time you press an ARROW key the selected points will move exactly 2.7 units in the direction of the arrow. In addition, if you hold down the SHIFT key and an ARROW key, the amount moved is ten times the normal value. Holding down the OPTION key moves by one-tenth the normal amount.

The **Grid spacing** field is for setting up an invisible grid that the Snap To Grid function snaps to. The **Align Points to Grid** command also uses this as the grid. One popular use for gridding is to have all your coordinates be on integral-number coordinates. Simply set a grid spacing of 1 em unit, make sure **Snap To Grid** is on, and then as you drag points around, they will always fall on em unit boundaries (that is, coordinates will always be 120, 66 rather than things like 120.223, 65.97).

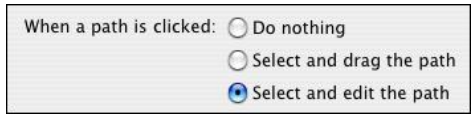
Snapping

Snapping distance specifies how close something has to be before a snap will occur. For instance, if **Snap To Point** is on, this preference states how close in pixels the object you're dragging has to be to another point before it automatically snaps to that point. If you are having trouble moving points around because they seem to be snapping to everything too often, try using a lower value in this field.

Snap to Point is a mode that makes aligning points much easier. Points you drag around with the mouse will snap to other points as you drag by them, as if they were magnets. This is how snap-to-point works when you have set the **Only snap to the closest point** mode. **Automatically align with all points**, on the other hand, makes objects snap to point extensions rather than just the points. In other words, let's say you have a point at x = 50, y = 100. In the **Only snap to the closest point** mode, you will snap to that point only when the point you are dragging comes close to 50,100. In the **Automatically align with all points** mode, however, the point you are dragging will snap anywhere along a vertical line at x = 50, or anywhere along a horizontal line at y = 100.

Paths

When a path is clicked on influences what happens when you click and drag the mouse on an actual path (that is, the line between the control points), as opposed to clicking one of the points. What you would expect to happen in this case depends upon your prior drawing program experience.



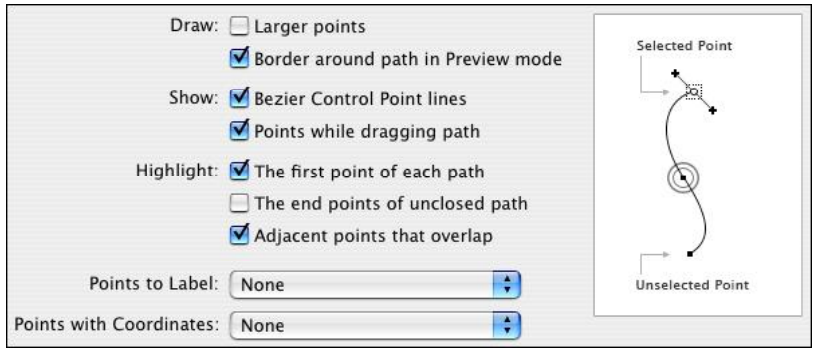
Do nothing is a good setting if you want to prevent yourself from modifying a path by clicking it accidentally.

Select and drag the path is a Adobe FreeHand-style operation. If you want to move an entire path, this preference allows you to do so merely by clicking it with the mouse and dragging it around. (This can also be accomplished, as in previous versions of Fontographer, by double-clicking one of the points and dragging the path around by a point.)

Select and edit the path is more of an Adobe Illustrator-style edit. This allows you to edit the path without having to click control points or BCPs; simply click directly on the path, and drag. This is especially fun to do in Preview mode, with the points turned off. This is a very exciting path editing feature, but it makes it easier to accidentally modify paths.

Note: This mode is meant to be used for minor path changes – large-scale movements may become unwieldy and difficult to predict.

Point display



The picture area of this dialog box is the most important part of this screen. It lets you see the effect of each of the items in the dialog box: simply change the state of one of the checkboxes, and look at the picture to see how that will affect the outline editing environment.

Rather than document all of the 128 different combinations of checkbox items and their effects, we'll just comment on a few of the items.

The first point in each path, when checked, draws a square around the first point in each path. This information is useful for doing font blending or for multiple master font creation. Some people objected to the visual clutter of showing that, and so we made it a preference so it can be turned off.

The end points of unclosed paths is a way to tell at a glance whether a path is opened or closed. Sometimes it is not clear in the outline window whether a path is opened or closed. (It is easy to lay down the last point of a path a little too far away from the first point, and so the path doesn't close like you think it did.) When this preference is on, big gray circles will appear around the two end points. The purpose of this preference is not to have it on all the time. What you should do is turn this preference on, flip rapidly through all the glyphs in the font, see if any have this problem of paths being left open, and then turn off the preference when you have finished.

Adjacent points that overlap, when checked, causes a double circle to appear on adjacent points (in the same path) that are on top of each other. This is a condition you almost never want, but that can occur by an inadvertent double-click of the mouse while laying down points. This preference should probably always remain on.

Points while dragging paths, when checked, will display all the control points as you drag something around. Turning this off causes all the control points to disappear when you're editing. Some think this results in a cleaner image to view while performing an edit.

Border around paths when in Preview mode, when checked, causes a dashed border to appear around lines during edits in Preview mode. This lets you always see your lines even when dragging them over black areas, which is not an uncommon occurrence in Preview mode. This preference makes editing somewhat slower, however, and can distort the image of the glyph while you are editing it.

Points to label is useful for font blending and multiple master font creation. This preference causes a small figure specifying point number to appear to the right of each control point, and BCP, if desired. This option can be used to determine path direction and path order. Having this on will slow editing performance a bit.

Points with coordinates is a way to have coordinates appear next to points, right in the outline window. It is probably most useful to show coordinates of only the selected points. That helps keep screen clutter down. Choosing to show coordinates will also slow editing.

Windows and dialog boxes

Windows

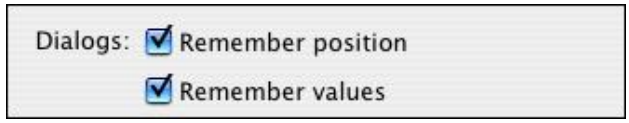
Automatically fit glyphs to windows, when checked, essentially performs an automatic **Fit in Window** command to be performed every time the outline window gets resized.

Move palettes with windows, when checked, locks the positions of the palettes with respect to the window frame. So if you always place your tool palette one inch in from the left of the outline window, and two inches down from the top, this preference will make sure that's where the palette stays, even when you move the window around on the screen or switch from one outline window to another.

View by

Use **View by** to choose whether you want to display your font windows by **Font name**, **File name**, or **Both**.

Dialogs



Remember position, when checked, tells Fontographer to open up each dialog box at the same place where that particular dialog box was last positioned. This is a handy way to have different dialog boxes appear in different places, rather than always centered in the screen.

Remember values, when checked, will cause Fontographer to remember all the values of all the controls in each dialog box (even if you quit Fontographer and start it up later). This is a time-saver for those dialog boxes where you frequently have to enter the same values, or choose the same settings.

DSIG preferences

This section controls settings for digital signatures that can be included in the OpenType (TT or PS) fonts that you generate. Please refer to

http://www.microsoft.com/typography/developers/dsig/default.aspx

for more information about digital signatures.

When **Generate digital signature** is enabled, a DSIG table will be generated that includes a digital signature. Note that you need to own a valid code signing digital certification to be able to digitally sign fonts. You should point Fontographer to your certificate file and private key file that you will receive from your certification authority. It is a good idea to store the private key file in a safe location, e.g. on a floppy disk or USB key, although you can use any location.

If the **Request private key password every time** option is selected, Fontographer will ask you for the password for your private key every time you generate a font in OpenType format.

Use the following password will remember the password for your private key.

The **Generate the time stamp** option tells Fontographer the time stamp must be generated and added to the font.

Please refer to the link above for more information.

Restore Defaults

This button is a fast way to throw away all of your Preference customizations and start again, with Fontographer's out of the box defaults for everything.

Note: Pressing the **Restore Defaults** button only sets the current Preference screen back to the default values.

Font blending – the technical details

Font blending, introduced earlier in Chapter 2, "Modifying Your Fonts", is a very easy, intuitive, and powerful feature of Fontographer. For casual use, you don't have to know much about it. For more industrial-strength use, however, there are some technical issues you might want to know about. Mastering a few key concepts will allow you to get the most out of font blending.

Fontographer's font blending, or interpolation, allows you to take two extreme variants of a typeface, and then automatically generate any number of intermediate versions within that range (or extrapolate to obtain versions outside of this range). You can edit those intermediate versions as desired, and quickly produce a family of weights, from just two master designs like Extra Light and Extra Bold.

The basic idea behind font blending is to create a few key designs, then let the computer do the work of producing a family of variants. Fontographer even gives you some powerful tools like **Change Weight** and **Remove Overlap** to help produce those key designs quickly, from a single base font.

Typically, you will not be able to successfully blend two entire fonts together; there will always be a few problem glyphs that can't be interpolated. You will know when this has happened, because Fontographer will display an alert saying that a problem occurred during blending. Usually almost every glyph will be interpolated (unless you have chosen two very disparate fonts). Glyphs that don't get blended are frequently some of the seldom-used symbol glyphs, because the upper 128 glyphs can vary greatly from font to font. If this is the case, you can decide you don't care because you never use those glyphs, and go on and use the font successfully. Or you can try to figure out why those several glyphs couldn't be interpolated like the rest, which is what this section is about.

The blending process

Font Blend is very much like the Adobe FreeHand blending of one object with another. The mathematical process for interpolation is very straightforward. Given two points, Fontographer will calculate another point that lies some specified fraction of the way between those points. Referring to *Figure 1*, assume we drew the inner and outer bold rectangles on the left, with the numbers indicating the ordering of the points.

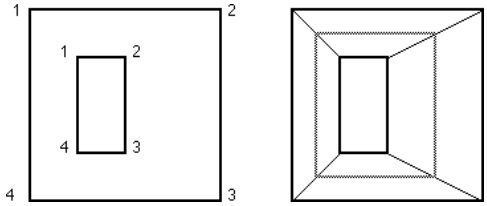


Figure 1

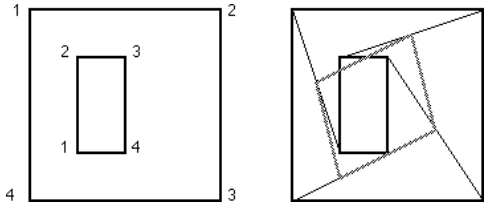
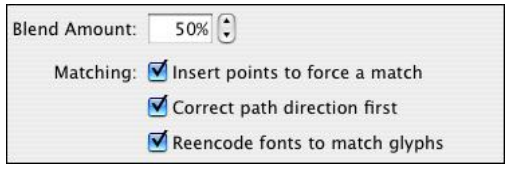


Figure 2

If you want to interpolate another rectangle that is 50% of the way between them (the gray rectangle on the right), you draw lines connecting the corresponding points of the two figures to be interpolated, then mark a point on each line that is 50% of the way from the outer point to the inner point. Connecting those points gives us the gray interpolated figure. You could interpolate 25% by marking the point 25% of the way from the outer to the inner point (or from the inner to the outer – it depends on how we set up the interpolate operation).

So far, the process is fairly simple. Now let's consider what can go wrong if the points of the two master figures don't correspond (like not having the path direction correct or the ordering of points be the same). The points in *Figure 2* again have been numbered so you can see what is going on.

The only change from the first example is that the inner figure has its points permuted (or reordered) slightly. You should remember this rule: Connect like-numbered points, then add a point halfway between the connected points, and connect those halfway points. Note in the example how, unlike the desired rectangle, the resulting diamond shape appears. It is likely that you will see something similar as you begin using Font Blend to make fonts. Clearly this is not an interpolate bug; Fontographer does exactly what it is supposed to do mathematically, and exactly what you asked it to do when you permuted the points.



Other aspects of blending make it a little difficult to use but follow clearly from the mathematics. Consider the case where the outer figure is a square, but the inner figure is a triangle. If you deselect **Insert points to force a match**, you run out of points to connect before you are done with the square, so it is pretty hard to know what to do with the extra point.

Similarly, if the outer figure is two squares and the inner figure is a single rectangle, you don't have enough figures to connect; Fontographer would prompt you that the number of paths didn't match. A more subtle error occurs when the two figures have the same number of points, but the points don't correspond well. Consider the case shown in *Figure 1*, but where the outer rectangle had four points inserted into the line from point 1 to point 2, and the inner rectangle had four points in the line from point 3 to point 4. Even though the point count is the same, the points don't correspond in a way that will give a pleasant result.

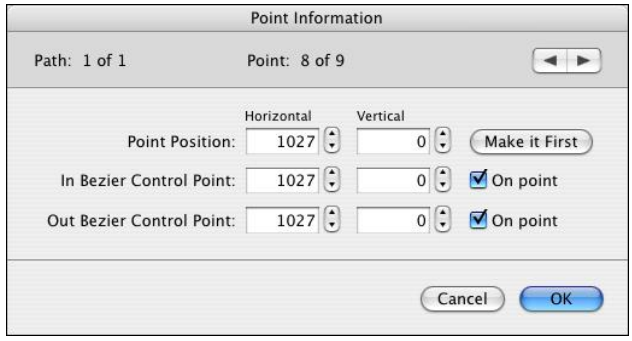
To make glyphs interpolate more easily, Fontographer reorders the points (and the paths) if you click the **Correct path directions** checkbox. If you correct path directions on all glyphs of each font you are trying to interpolate, Fontographer will almost always arrange the points so that they match up, and interpolating will give you the right result. On rare occasions, due to the way some glyphs might have been drawn, you can confuse Fontographer's point ordering technique. If this occurs, you will know it from the interpolate results (glyphs will appear imploded, or turned inside-out). Those cases will require manual point reordering with the Point Information dialog box. Select a point on the glyph, choose **Selection Info**, then move to the next or previous point until you are at the desired starting point. Click on the **Make It first** button, and Fontographer will make that point the first point of the path.

Note: The origin point of paths may be indicated by a box drawn around it. There is a preference to enable this feature.

➲ ***Tips:*** Choose one of the entries in the **Points to label** pop-up in the Point Display dialog box in Preferences, and numbers will appear in the outline window next to each point, showing point ordering information. Having this preference active is a good way to tell at a glance whether pairs of glyphs are likely to interpolate well or not.

➲ In some glyphs, Fontographer will be unable to automatically correct path ordering for blending. When this happens, you can fix the problem by selecting the offending paths, and choosing **Bring to Front** from the **Arrange** pop-up in the **Element** menu.

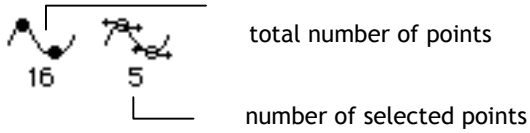
The Point Information dialog box also shows the total number of path contours and the total points in the current contour. See **Point** and **Path** in the illustration below.



This dialog appears when you select a single point in the Outline Window and choose **Selection Info** from the **Element** menu.

This is useful for finding mismatches between the same glyph in different fonts. You will have to figure out the best technique for getting the points to match on a glyph-by-glyph basis. Sometimes it is better to remove extra points from a glyph with too many points, and sometimes it is better to add points to a glyph with too few. Just remember how the points get matched up, and let that guide you as to where to insert or delete points.

➲ ***Tip:*** A quick way to see how many points are in a particular path without going into the Point Information dialog box is to double-click the path in the outline window, then read the value displayed under the number of selected points indicator in the info bar, as shown below.



Font hinting

Only about two percent of our Fontographer customers need to concern themselves with hinting. This statement is not meant to demean the other ninety-eight percent; Fontographer's autohinting abilities are quite good and should serve most cases well. Think of it like this: most people fly on commercial airplanes. They get around without any problems, and are usually happy with their flights. Very few of the people who take airplane rides need to know all the physics controlling airplane lift and related phenomena. Some people might want to know: they could be curious, or they might be interested in piloting their own airplane. The point is, however, that in some cases it isn't necessary to become an expert in all the details.

Hints are just like that. Fontographer's autohints will take you where you want to go; only those few who want to be pilots really need to manually edit them. So don't feel you need to read this section and master these concepts, or that you should go into the Hints layer and start changing things around. Feel free to check it all out, but be assured that you absolutely do not have to know a thing about hinting in order to success-fully use Fontographer.

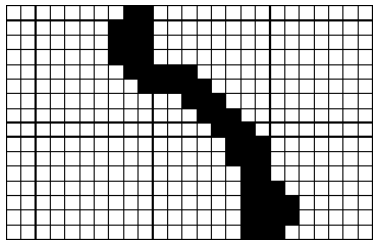
Are you still with us?

We highly recommend that you obtain and read *Adobe's Type One Font Specification* before getting too involved in hint editing. Although that publication deals entirely with PostScript Type 1, the concepts covered will be valuable no matter what font format you want to use.

What is hinting all about?

Hinting is a process by which the computer makes type look good at small sizes and low resolutions (72–600 dots-per-inch, or dpi). To understand hinting, you first have to understand a bit about how computers print.

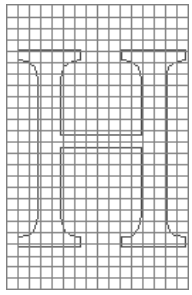
The reality of laser printing is that the smooth lines you get out of your printer aren't really smooth at all. They are composed of little dots (called pixels) which, like a mosaic, are all assembled on a grid to form a picture. At high enough resolution, your eye perceives smoothness even though there really is jaggedness.



Close-up of a printer's output.

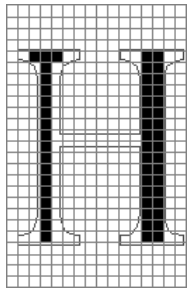
When the printer prints, it has to choose which pixels to turn on or off to best achieve the desired image. As you can imagine, it isn't quite as nice as just drawing smooth curves on paper with a pen.

To see how that relates to glyphs and hinting, let's do an example. We'll follow along as the computer constructs the letter "H" at a small point size, on a low-resolution printer. First, the computer must get the mathematical outline of the "H", scale it to the requested size, and position it in memory over the pixel grid at the requested location:



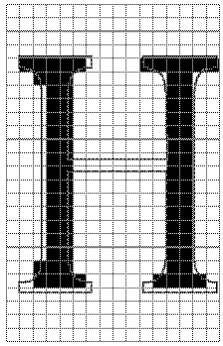
Remember that the printout you get is composed entirely of pixels, so if we were to print now, we would get a blank sheet because the computer hasn't yet turned on any pixels. That's the hard part; drawing the pure mathematical outline was easy.

The trick is to figure out, by examining the outline, which pixels to turn on and which to leave off. A reasonable start is to simply turn on all the pixels that are entirely contained by the glyph outline. That gives this result:



Or, shrunk down to size, this:

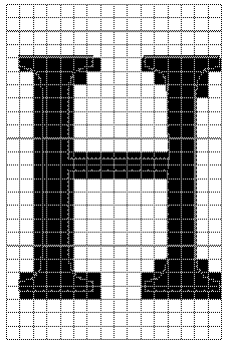
Since that was less than an ideal “H”, we obviously need to be cleverer. Let’s try turning on all the pixels that are mostly contained in the outline (that is, at least half of the pixel must be in the outline in order for it to be turned on). That yields this result:



Or, scaled down, this:

That didn’t turn out very well either. Note that, although we have at least a semblance of the four serifs, we still don’t even have a cross bar.

Let’s try one more way: we’ll turn on all the pixels that have any portion whatsoever of the outline touching them:



When scaled down, it looks like this:

To review, here are the three H's thus far:

And here is what the mathematical outline looks like:

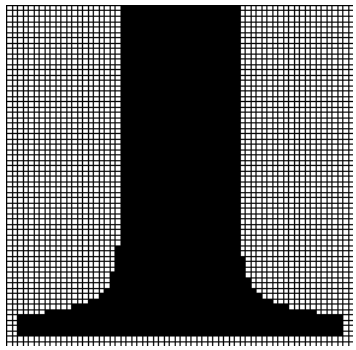
The last of our H's looks better than the other two, but unfortunately, it is still a terrible-looking letter. The left stem is 50% wider than the right stem; the serifs are chunky and irregular; the cross bar is too heavy; and most importantly, it doesn't match the outline.

What we would like is something like this:

That's where hinting comes in. As we've shown, you can't simply turn on pixels just by looking at the outline if you expect to wind up with a decent glyph. More intelligence has to be added in.

Hinting works by feature recognition. Hints basically define interesting features such as vertical stems, horizontal stems, serifs, cap heights, x-heights, and so forth. For example, in the case of the H, there is information in the font that says things like: where on the pixel grid the outlines fall; the left stem and the right stem must be the same width; and so forth. Hints can also control global font-wide attributes such as x-heights, so the tops of glyphs like x, n, c, m, and so on, are all aligned properly at all sizes.

Hinting is most important at low resolutions, because there are fewer pixels to define the glyph image. At high resolutions, it's easy to get a good image; there are so many pixels to deal with, the differences hinting makes are not normally noticeable or important. The figure below is how our "H" example might look on a Linotronic at very high resolutions.



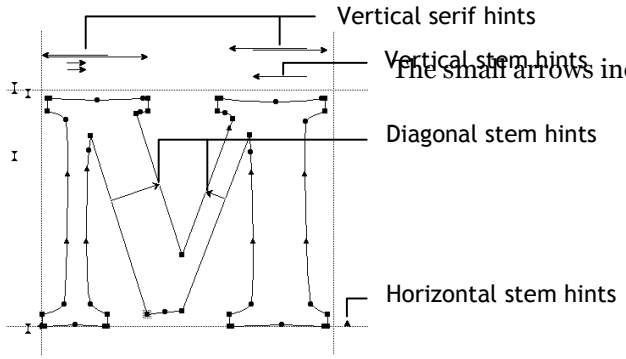
Since hinting can control at most one pixel's difference either way, the human eye can't usually perceive the difference hinting makes at high resolutions.

To sum up: A hint is special information placed into a glyph's outline definition that causes the glyph's outline to be adjusted in a way that improves the glyph's perceived shape when printed.

Hinting controls

Fontographer lets you control hints from two places: the Hints layer of the outline window, and the **Hints** menu.

To see actual hints, open an outline window, and make the Hints layer visible by clicking the box next to its name in the layers palette. Here is an example of a glyph with hints:



The small arrows indicate the presence of hints.

Autohint

Hinting in Fontographer is always in one of two modes: automatic or manual. Automatic means that Fontographer will recalculate hints automatically when you edit glyph outlines. Manual means that you have actually edited some hints. Manual mode tells Fontographer not to replace your manually edited hints with its own. Fontographer automatically switches to manual after you have edited hints.

Automatic or manual hinting varies from glyph to glyph. Whenever you open a PostScript font, Fontographer reads in the hints from that font and turns autohinting off. Note that Fontographer hasn't made up any new hints yet. It is waiting for the outline to be edited first. Once an outline has been edited, new hints may be calculated to match the new outline. When you open an existing TrueType font, Fontographer cannot read in the hints, which are compiled into the font. Fontographer will automatically turn autohinting on.

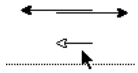
You can tell if Fontographer is on automatic by looking in the **Hints** menu. This menu contains an item called **Autohint**; a checkmark indicates that Fontographer will recalculate hints for the selected glyph every time the outline gets edited. If you don't want that, choose **Autohint** from the menu. The checkmark will go away, and autohinting will be turned off for that glyph. Be careful, because if you then turn **Autohint** back on by reselecting it from the menu, Fontographer will recalculate new hints for that glyph.

When autohinting is on, you can watch Fontographer come up with new hints as you edit a glyph. Just make the Hints layer visible, check to see that **Autohint** is on, and edit the glyph outline. You will see the hints adjust, disappear, move, and change, depending upon what you do with the outline. This is a good way to get a feeling for how Fontographer finds hints.

Editing hints in the outline window

You must make the Hints layer active before you can edit hints. You do this by clicking the Hints' layer name in the Layers palette.

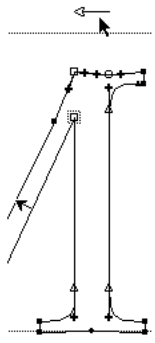
You can select hints with the mouse, just like you select points. Selected hints have a hollow looking arrowhead:



You can select more than one hint by dragging a marquee around a group of hints or by using the standard **Shift**-select method.

You can reorder hints by dragging them around (this is useful in the case of overlapping hints: PostScript fonts will get the hint that appears closest to the glyph outline in the case of overlap). Hints may also be reordered from the Common Stems dialog box, as described in "Common Stems" on page 405.

Changing the direction of a hint can be done by selecting the hint, and then clicking the arrowhead of the hint. To see what points were responsible for creating a particular hint, simply **Option**-click the hint, and those points will become selected:



Removing hints

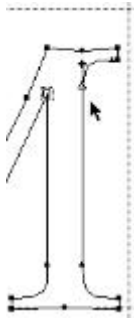
Removing a hint is simple. Just press the **Delete** key, or choose **Clear** from the **Edit** menu. Be careful – if you have just **Option** clicked on a hint to study the points that made it, you will remove both the hint and those points if you press the **Delete** key. Be sure to deselect the points, by pressing **Tab** or clicking somewhere else, then select just that hint before deleting anything.

Making new hints

Creating new hints is done in the **Hints** menu. To make this process faster, we have made sure each command in the **Hints** menu has its own keyboard equivalent.

To make a new vertical stem hint:

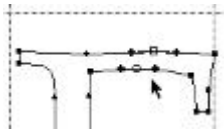
1. Select two points that define the stem.



2. Choose **Make Vertical Stem** from the **Hints** menu.

To make a new horizontal stem hint:

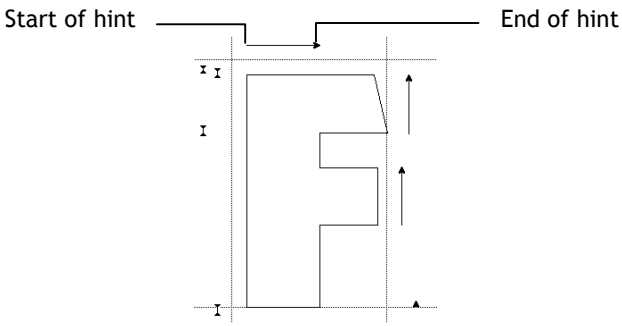
1. Select two points that define the stem:



2. Choose **Make Horizontal Stem** from the **Hints** menu.

Selection Info for hints

Selecting a hint and then choosing **Selection Info** from the **Element** menu brings up a dialog box that is somewhat analogous to the Point Location dialog box. It lets you traverse through all hints in the glyph and modify them.



Hint type

Hint type is a pop-up showing which kind of hint you are currently examining. The message to the right, 1 of 2, means that you are currently looking at the first of two hints of the indicated type (Vertical stem, in this case). To look at the other hints, simply choose the hint type you are interested in from the **Hint type** pop-up.

The area directly under the **Hint type** pop-up changes depending upon which kind of hints you are viewing. For Vertical stems, it lists the x coordinates of the stem (that is, the locations of the sides of the stem). These values can be edited by entering new ones.

The buttons

Flip Direction will change the direction of the current hint.

Add hint is a way to make a new hint anywhere you want; simply fill in the location fields in the left side of the dialog box to position the hint. Since most hints must involve actual points, this is really useful only for ghost hints (hints that exist only to pull part of a glyph into an alignment zone).

Remove hint deletes the current hint.

Forward and **Back** move the hint around, relative to the other hints of that type. This is the same thing as dragging a hint with the pointer tool, except that with the buttons, you only move in or out one hint level with each click. To move a hint all the way out or all the way in, keep clicking the appropriate button until it dims, which indicates that's the farthest the hint can go in that particular direction.

Hint Automatically performs the same task as the **Autohint** item in the **Hints** menu – it will turn on autohinting and rehint the glyph.

Next and **Previous** are the controls that change the currently selected hint. **Next** will choose the next-outer hint of the currently selected type, and **Previous** will choose the next-inner hint of that type. To move on to examining other groups of hints, choose a different hint type in the **Hint type** pop-up.

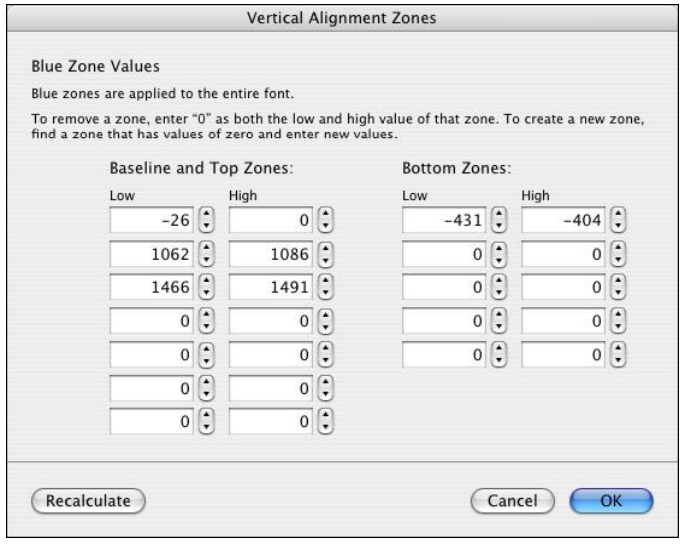
Vertical Alignment Zones

In addition to stem width hints, Type 1 font hints also control the vertical positions of glyphs. Due to the way the human vision works, curved letters like the "O" should be drawn taller than straight letters like an "E" if they are to appear to be the same height. The technical term for this is overshoot. However, at small sizes this overshoot should drop out and the letters should be the same height. The oversize is typically 4%, which means that if 4% of the em square is less than one pixel, there should be no overshoot. The way overshoot is controlled in a Type 1 font is through a blue zone. Only Adobe knows how they derived that term, but the concept is to specify important vertical alignment coordinates in a table.

Blue zones are not associated with particular glyphs; they apply to the entire font. However, it is helpful to think of them as corresponding to certain groups of glyphs. When Fontographer computes or recomputes the vertical alignment zones, the first pair of values is based on glyphs such as "A" and "C", which are normally drawn with the bottom of the letters on or near the baseline. The first value of the pair is the baseline overshoot (glyphs such as "C" and "O"). The second value is the normal baseline.

The remaining blue zones are computed from the tops of glyphs and the x-height (glyphs such as "c" and "x"), the cap height line ("H" and "O"), and the ascender line ("d" and "l"). Since the zones cannot overlap, the ascender zone may not be found. The Other blues are based on the bottoms of glyphs such as the descender line in glyphs "g" and "y".

Choosing **Alignment Zones** from the **Hints** menu brings up this dialog box.



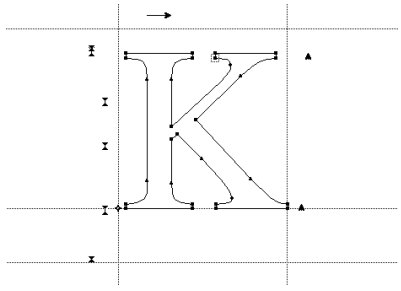
We don't really know why the PostScript hinting variables these fields were named after have anything to do with the color blue. But they do, and once again, you are encouraged to look at *Adobe's Type One Font Specification*, which covers PostScript hinting quite admirably. Be aware that, although the nomenclature in this dialog box is PostScript-centric, these values are used by Fontographer for vertical alignment in TrueType as well.

You can edit any of these values by entering new numbers into the various fields. If you ever have occasion to go back to how they used to be, you can press the **Recalculate** button. If that button is inactive, that means you haven't edited anything yet, so the recomputed values would be the same as those that are currently showing.

Fontographer will read in all the blue values from PostScript Type 1 fonts. It can also calculate blue values itself, by examining glyphs $21-$7E (decimal 33–126) to see where the topmost and bottommost parts of those outlines lie. Those areas are typically what should be included in the vertical alignment zones.

The fields in the Vertical Alignment Zones correspond exactly to the little I-beams that appear along the left side of the outline window, when the Hints layer is visible.

These I-beams are the vertical alignment zone indicators. You select them by clicking them with the mouse; selected zones appear to be hollow on each end. When a zone indicator has been selected, horizontal lines go through it to make it easier to tell which points fall inside the zone. You may change the size of any of the zones by dragging either the top or bottom part of the indicators. To describe the location of the zone numerically, you must use the Vertical Alignment Zones dialog box.

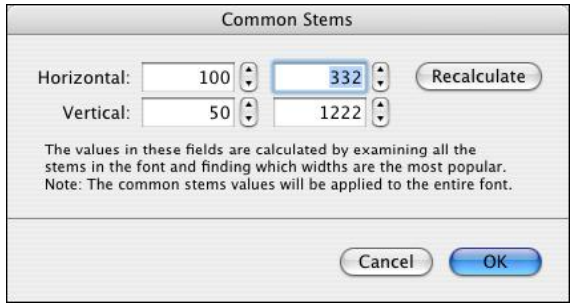


To remove a zone, you can select it and then press the **Delete** key. You may also go into the Vertical Alignment Zones dialog box and enter "0" for both the low and high value for a particular zone.

To create a new zone, go into the Vertical Alignment Zones dialog box, find a zone that has zeros in it, and enter new values. If there are no zones with zeros, that means all the available zones are used up, and you will have to remove an existing zone in order to add a new one.

Common Stems

Choosing **Common Stems** from the **Hints** menu brings up this dialog box.



Note: The Common stems values always apply to the entire font.

Changing these fields is useful for PostScript Type 1 and TrueType fonts. These fields are calculated by examining all the stems in the font and finding which widths are the most popular (that is, which widths occur the most). These values appear here, and get output in the PostScript font as StdHW, StdVW, StemSnapH, and StemSnapV. See the *Adobe Type One Font Format* book for more information on these fields. To throw away your edits to these values, or to get new values after substantially editing the font, press the **Recalculate** button.

What happens when Fontographer opens up PostScript Type 1 fonts

When a PostScript Type 1 font is read, Fontographer always keeps the hints that were defined as part of the font. Each glyph is marked as having user-edited hints. This lets you generate a Type 1 font and still leave the original hints unchanged. Before generating a TrueType font, however, you should select all glyphs and choose **Autohint** from the **Hints** menu. This will remove the original stem hints and find new ones.

Autohinting is on by default, so when you edit a glyph, its hints get recalculated. This is usually what you want to happen. If you change the outline, you don't want to still have the hints for the old outline – you want new hints to match the new outline, which is what autohinting gives you. However, sometimes you may want to make minor tweaks to the outline, and keep the exact original hints. You can accomplish this by turning autohinting off, in the **Hints** menu (the **Autohint** item is essentially a toggle; when it is on, as indicated by a checkmark, you turn it off simply by selecting it).

Note: If you turn autohinting back on, it will re-hint the selected glyphs. Usually, that's not a problem, because you either want autohinting or you want manual hinting – you rarely want to switch between them. To preserve hints for the entire font, go to the font window, **Select All**, and then turn **Autohinting** off. To rehint the entire font, go to the font window, **Select All**, and turn **Autohinting** off and back on again.

What happens when Fontographer opens TrueType fonts

Fontographer completely hints TrueType fonts when it opens them. This happens for a number of reasons, not the least of which is that Fontographer has to change the outline format from quadratic to cubic.

Using a text editor to tweak Fontographer 5 on your Macintosh

The tips and notes in this section are highly technical.

Adding custom encodings

Encoding tables are a useful mechanism to filter and view your glyphs in different arrangements. You can put a large number of glyphs into one font, assign a unique name to each character, and supply several encoding tables, allowing you to select different sets of characters in the font when you use different encodings.

For example, in symbol fonts the Greek characters take places that are usually occupied by Latin characters. With the encoding tables you can include both sets of characters. Just assign the correct names (like alpha for the 'A' character and A for the 'A' character) and later you can choose the symbol encoding to work with the Greek version of your font or choose Roman encoding to use the Latin characters.

In Fontographer you can include up to 32,000 glyphs in a font. If you need a font editor that supports more glyphs (up to 65,535), Fontlab Ltd. offers AsiaFont Studio, a multibyte font editor that is the "bigger brother" of Fontographer.

All encodings are stored in text files that can be edited in any text processor.

To create a custom encoding file:

1. Copy the .enc file located in the **[Shared default data folder]/Encoding/T1 non-Western** folder (typically, Macintosh HD/Library/Application Support/FontLab/Encoding/) to use as the basis for your new encoding file.
2. Open the copied file in any text editor (Macintosh TextEdit will do) and then edit it, following the same structure that you find in the original file.
3. Change the name of the encoding and the encoding index in the first line of the file. The first line should have the following structure:

   ```
   %%FONTLAB ENCODING: 7; Type 1 Adobe Symbol
   ```

   "%%FONTLAB ENCODING: " is the prefix of the file used to detect properly made encoding files and must not be changed. Note the space between ':' and the encoding index.

'**7**' is the index of the encoding vector. You must not change the encoding vector indexes of any of the encoding vectors or they will become unusable. If you make your own encodings the indexes of your files should not be used in any of the other files. The actual value of the index is not important, so you can assign indexes like 1001 or 10001.

The last part of the first line, "**Type 1 Adobe Symbol**", is the name of the encoding vector. It starts at the first non-space character after ';'. Pick a name on your own and type it there e.g. "A Glyph Definition Encoding" – use plain English letters, digits or simple punctuation such as [ ] ( ) but avoid too many special characters in the encoding name. Do not use ampersand (&).

4. Change the name of the group in the second line of the file:

```
%%GROUP:My Custom Encodings
```

The group name will become the submenu title in the **Encoding** popup menu. Note that there is no space after the ':' character. We recommend using your foundry name or your personal name in the encoding group name.

5. Edit the contents of the encoding table, e.g.:

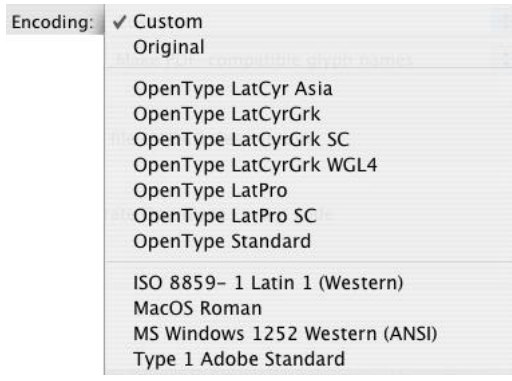
```
%%FONTLAB ENCODING: 1001; A Glyph Definition Encoding
%%GROUP:My Custom Encodings
%
A.smcp
A.titl
A.swsh
A.subs
```

If the encoding table will be used as source of Type 1 encoding on export, each glyphname should be followed by a space, followed by a decimal character code, e.g.:

```
%%FONTLAB ENCODING: 1001; A Glyph Definition Encoding
%%GROUP:My Custom Encodings
%
A 65
B 66
C 67
D 68
```

6. Save this encoding file with a different file name but be sure to use the .enc file extension. Put the .enc file in the **[Shared user data folder]/Encoding** folder (typically Macintosh HD/Users/Your Username/ Library/Application Support/FontLab/Encoding/) if you want to make the encoding available to all recent Fontlab Ltd. applications, or in the **[Application user data folder]/Encoding** folder (typically Macintosh HD/Users/Your Username/Library/ Application Support/FontLab/Fontographer 5/Encoding/) if you want to make the encoding available within Fontographer only. All custom .enc files should be located in these folders!

7. Open Fontographer and create a new font. Choose **Font Info** from the **Element** menu, click on the **Encoding** tab and select your own encoding from the **Encoding** pop-up. Now copy your glyphs into their proper locations in this font database.



Note: Mistakes in name entries will manifest themselves in the font, so be careful and accurate in your editing.

While in the font window, select Unicode from the **View by** pop-up. If a ** appears in the glyph label over the custom glyphs, then you know that your custom names did not map to known Unicode numbers. All of the Unicode data in Fontographer is stored in **[Shared default data folder]/Mapping/standard.nam** file. If you want to edit it, be extremely careful, or your other encodings will not work correctly. We strongly recommend against making any changes to **standard.nam**. It's better to make further changes to the specific glyph in the Glyph Info dialog box.

8. When your font is ready to generate, just choose **Generate Font Files** from the **File** menu.

A special note to designers of non-Roman Macintosh fonts

In Mac OS 7.1-9 with WorldScript, fonts can be assigned to a particular script system. WorldScript is also used for PostScript Type 1 in Mac OS X.

This assignment is achieved by using script ID number in the particular encoding file. For example, the MacOS Cyrillic encoding file contains the following string:

```
%!Macintosh Script:7
```

where **7** is the Cyrillic script ID. Fontographer takes this ID from the encoding file and assigns the exported Macintosh font the proper ID from the appropriate range (19456 – 19967 for Cyrillic). In other words, the proper FOND ID is assigned automatically according to the font encoding selected in the Font Info dialog box.

A table of the more popular script systems and their assigned ID range follows. For more information, refer to Inside Macintosh Volume VI, pages 13-6 through 13-9.

Script system	Script ID	Font ID range
Roman	0	2 – 16383
Japanese	1	16384 – 16895
Traditional Chinese	2	16896 – 17407
Korean	3	17408 – 17919
Arabic	4	17920 – 18431
Hebrew	5	18432 – 18943
Greek	6	18944 – 19455
Cyrillic	7	19456 – 19967
Simplified Chinese	25	28672 – 29183
Vietnamese	30	31232 – 31743

Setting developer IDs

If you are a font developer who has been issued a unique developer ID by Microsoft, then we have provided a means for you to add that ID into Fontographer by editing the *vendor.dat* file.

Find this file in the **[Shared default data folder]/Data/** folder (typically, Macintosh HD/Library/Application Support/FontLab/Data/) and add your vendor ID followed by vendor name.

Customizing Sample Text printout

Yet another TextEdit trick is to edit the *PageHeaderPS.lis*t resource to change the font used to label coordinates in the glyph points and coordinates printout. If you have noticed that the coordinate locations run over each other on paper, then you might try this quick fix.

1. Right-click on a copy of Fontographer in /Applications/Fontographer 5.0/ and choose **Show Package Contents**.
2. Open /Contents/Resources/PageHeaderPS.list in TextEdit
3. Change the following code:

```
(|_____Helvetica) RF
/descFont /       |_____Helvetica
          findfont def
```

Simply replace Helvetica with HelveticaNarrow or Helvetica-Narrow or whatever font you desire. Just be sure the font you specify is installed.

Character palette

You can use the Character Palette as a reference on special characters, such as mathematical symbols, letters with accent marks, or arrows and other "dingbats," which you may want to add to your font.

You can also use the Character Palette to view or enter Japanese, Traditional Chinese, Simplified Chinese, and Korean characters, as well as characters from other languages.

The Character Palette appears in the **Input** menu. The **Input** menu looks like a flag in the top-right corner of your menu bar. If you don't see the **Input** menu in the menu bar, open System Preferences and click **International**. Click **Input Menu**, then click the checkboxes next to **Character Palette** and **Show Input menu in menu bar**.

To use the Character Palette:

1. Open the application you want to type in, and place the insertion point where you want the special character or symbol to appear.
2. Click the **Input menu** icon in the menu bar and choose **Show Character Palette**.
3. Choose the type of characters you want to see from the **View** pop-up menu at the top of the Character Palette window.

   If you don't see the **View** pop-up menu, click the button in the upper-right corner of the Character Palette to show the top portion of the window. Click this button again to hide the top portion of the window.
4. Click an item in the list on the left to see the characters that are available in each category.
5. Double-click the character or symbol in the right column that you want to insert into your document. You can also select the character and click **Insert**. To see more options for each character, such as the variations in glyphs for some characters, click the **Character Info** triangle and then the **Font Variation** triangle at the bottom of the Character Palette window.

Multiple master fonts

You can use Fontographer to create multiple master fonts that can be used with Fontlab's TransType Pro. TransType Pro gives the type designer the ability to generate (or interpolate) different versions of master font designs. This means that you can customize your type to fit your design needs (almost literally on-the-fly). Suppose, for example, that you have a banner in your newsletter that needs its display type to be wider (and even heavier). You can use TransType Pro to make a wider version of your existing multiple master font, while still retaining the size and weight of the original typeface. TransType Pro has many potential benefits for graphic designers and publishers, such as copy and headline fitting, and the ability to adapt the design of a font to the point size at which it is imaged.

The multiple master technology can take interpolation to even higher dimensions. Instead of requiring merely two master fonts (as when using Fontographer's Font Blend feature), with multiple master typefaces you can utilize sixteen fonts (or more) for the most complex multiple master fonts.

Note: You must have TransType Pro or you may put multiple master fonts in the Library/Application Support/Adobe/Fonts/ folder. If you need more information about this contact Adobe: http://www.adobe.com/support/. There are also some fairly substantial technical notes available from Adobe about how to best design multiple master fonts.

Note: Multiple master PostScript fonts are supported in Mac OS X starting with version 10.2.

A quick overview

Actually, the term quick overview is somewhat of a misnomer since creating multiple master fonts is definitely not a simple procedure, so for your first font exercise we recommend that you create a one-dimensional font. In fact, if you have created an Extra Light and Extra Bold for interpolation, those would be excellent fonts to use as the basis for a one-dimensional multiple master font.

For one dimension, picture two fonts at opposite ends of a one-dimensional line segment. Typically, this single dimension could be weight, width, contrast, or optical size.

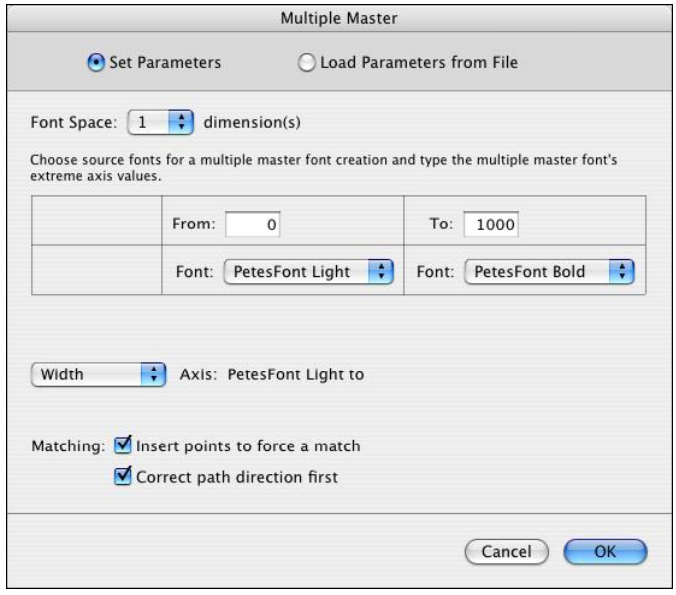
Light ⟵ Normal ⟶ **Bold**

R R **R**

In our example, we made a one-dimensional font that TransType Pro can use.

To create a one-dimensional multiple master font:

1. Open the thin version of your font (ours is PetesFontLight).
2. Now open the wider version of your font (ours is called PetesFontBold).
3. Choose **Multiple Master** from the **Element** menu.



The Multiple Master Setup dialog box appears.

The proper number of dimensions (1) is automatically selected in the pop-up at the top. The next two pop-ups show the names of the two open fonts. If there are other fonts open, you can select them by accessing these pop-ups. The final pop-up is the axis name. To change it to another name, select it and choose another. You can name your own axis by choosing **Other** from this pop-up. For this exercise choose **Width**.

Now your job is to decide how much wider you'd like your base font to be. We recommend that you examine existing multiple master fonts, including those from Adobe, for help in determining consistent numbering schemes.

4. Deselect the options **Insert points to force a match** and **Correct path directions first**.

5. Type in your values.

   We will discuss both these options later.

➲ ***Tip:*** Choose the Width option in the **View by** pop-up in each font window to view the different widths in your two fonts and get an idea of the minimum and maximum values you'd like to set here.

6. Press **OK** to close the Multiple Master dialog box.

7. Choose **Generate Font Files** to generate a multiple master font.

If there are any conflicts between your two fonts (like discrepancies in points, paths, and hinting values) Fontographer will notify you and generate an error report, known as the *Fontographer.log* file.

The *Fontographer.log* report will appear in your Fontographer folder.

Due to the precision required in multiple master fonts, this will probably happen during your first few attempts...but don't give up. Open the *Fontographer.log* file with any word processor or TextEdit. Scroll through your error report to see which characters have conflicts, make the changes indicated, and try again.

Next is a sample of an error report you will probably receive when generating a multiple master font.

```
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
>>>>>>MultiMaster setup complete<<<<<
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
Character 37 (%) failed compatibility check in Font 3
(Check your hinting)
hsbw 385 1000
hsbw 24 1000
Mismatched number of hints in character 41 (")")
Font 1 has 1 horizontal hints,
0 vertical hints
Font 2 has 2 horizontal hints,
0 vertical hints
Path 1 of character 97 in font 2 has a different number of points than the other fonts.
Character 101 in font 2 has a different number of paths than the other fonts.
```

This file has four basic error types:

- Character 37 (%) failed compatibility check in Font 3. This refers to the origin points. The "%" sign in the third font (remember the cube idea) has a different starting point than the other three fonts. Following this comment is a PostScript dump of the character provided as an aid for those who understand it.
- Mismatched number of hints in character 41 (")")

  - Font 1 has 1 horizontal hints, 0 vertical hints

  - Font 2 has 2 horizontal hints, 0 vertical hints
- Path 1 in character 97 in font 2 has a different number of points than the other fonts.
- Character 101 in font 2 has a different number of paths than the other fonts.

When you are finished generating your font, drag the bitmap and PostScript file into the multiple masters folder. If you are using Suitcase or Master Juggler, you'll need to open your font (to link it) before you'll be able to apply effects.

Now you can open your font in TransType Pro and create as many new versions of your font as there are combinations between your preset width values of 50 and 500.

Once you've got the knack of one-dimensional fonts, you can move on to two-, three-, and yes, even four-dimensional fonts. The next section gives you valuable information about how to start planning your multiple master fonts.

The genesis of a multiple master font

By Greg Thompson

When you draw a multiple master font, there is only one rule that you must obey: each corresponding character, "A" for example, must be drawn with the same kind and number of points in each of the 2, 4, 8, or 16 master fonts that together constitute the end points on the axes of your font. Oh yes, one other thing: if your multiple master font is to be hinted, each corresponding character must be hinted with the same kind and number of hints in all the fonts, and all the hints must also be in the same order and going in the same direction.

To avoid the process of making major "family" design changes to four different master fonts, I started NewtechMM as one medium weight normal width font, even though Newtech Medium would not be used as a master in the completed font. (I try out heavier or lighter drawings of certain characters as I go along just to see how they will translate later on, but for the most part I don't concern myself with other style variations until later.) When I am satisfied with Newtech Medium, I copy it two times and create Light Extended and Black Extended by modifying the original medium drawings. I use a combination of Fontographer's **Scale horizontally**, **Change Weight**, **Move**, and **Set Width** and manually move points. Some of the extended characters may require the addition of an extra point or two to remain true to the Newtech design. I chose to make the difference in weight dramatic, but it could be subtle.

These extremes are made as light and black as possible while still retaining the Newtech family resemblance. I then quickly check the extended styles for visual and technical compatibility by applying Fontographer's **Blend Fonts** feature. I use interpolation (font blending) extensively when working on a multiple master font to check progress since it allows me to find any problems while staying within Fontographer.

After I'm pretty sure everything's okay with the weight extremes (Light Extended and Black Extended), I copy and modify them to create the width extremes of Light Condensed and Black Condensed. If some characters require more or fewer points, I immediately make the same changes to the already-existing Extended styles. I might have to make slight design changes at this point to maintain familial resemblance or axis independ-ence.

Now I'm ready for the first attempt to generate an actual multiple master font from the Newtech masters. Turn hints off and generate Type 1 fonts from the four masters. Then follow Fontographer's documentation instructions for setting up a multiple master font, making sure to turn off **Correct Paths** and **Insert Points**.

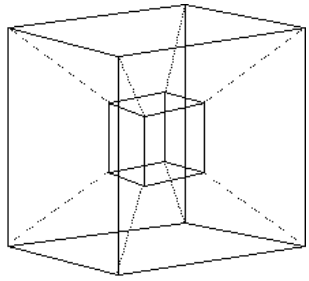
Multiple master font technology is really just interpolate and name technology. Why not try letters that bounce up and down, letters that flip around, or a little man that jumps around the screen?

Planning multiple master fonts

Start by planning very carefully just how many dimensions you want to deal with. For your first font, we recommend one or two dimensions. In fact, if you have created an Extra Light and Extra Bold for interpolation, those would be excellent fonts to use as the basis for a one-dimensional multiple master font. Adobe Systems, the developer of multiple master font technology, currently has only used one, two, or three dimensions; both three- and four-dimensional typefaces are very ambitious undertakings. Each dimension you add doubles the number of master fonts you will need.

Terminology: Master design: one of the PostScript Type 1 fonts a multiple master typeface is comprised of. Master design fonts do not show up in the font menu.

For illustrative purposes, we shall refer to the font design space as a hypercube, although it should be understood that a cube, square, and line segment are simple cases of the general hypercube.

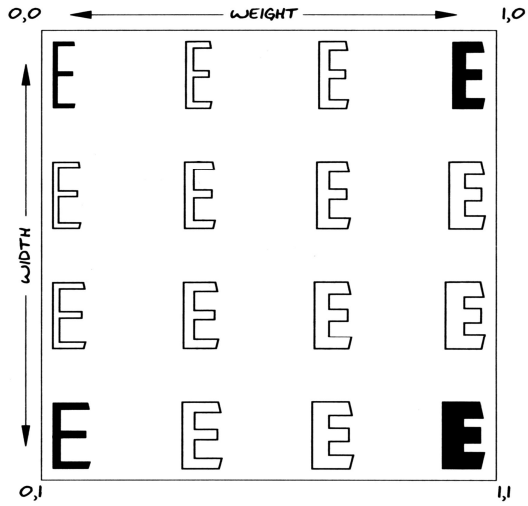


As we stated in the overview, for one-dimensional fonts, the two fonts are at opposite ends of a one-dimensional line segment. Typically, this single dimension could be weight, width, contrast, or optical size.

Two dimensions require four fonts at the corners of a square. Typically the two dimensions might be weight and width. Three dimensions require eight fonts at the corners of a cube (adding optical scaling), and four dimensions requires sixteen fonts at the corners of a hypercube. The generated multiple master font combines all the master design fonts into one gigantic PostScript–language font, so a four-dimensional multiple master font is fairly large (probably 350K or more).

Terminology: Normalized coordinates (also known as font space coordi-nates) the coordinates varying from 0 to 1 that label each corner of the multiple master's hypercube. Each master font has a coordinate in each dimension of either 0 or 1.

At this point, we encourage you to get a pad of paper and a pencil to sketch out a plan for creating your multiple master fonts. Draw out a picture of your font coordinate space, and sketch the font variations for each corner of the hypercube. Normalized font coordinate space always runs from 0 to 1 in each dimension. Label each corner with a coordinate of 0 or 1; this will be the font space coordinate of the corresponding master font. We recommend labelling the upper left corner 0,0 (or 0,0,0,0 for four-dimensional space), as shown here.



Font coordinate space

You can change the exact arrangement of fonts in this coordinate space, but you must be consistent in your labelling. Thus, you could make width vary from left to right, and weight vary from top to bottom; but you could not vertically interchange the two fonts on the right side and get meaningful results. Be advised that you will get neither predictable behavior from such a font, nor an error message from Fontographer. In addition, it is not particularly recommended; you should strive for consistency among different multiple master fonts, so applications can do intelligent things with them.

Notes about optical scaling

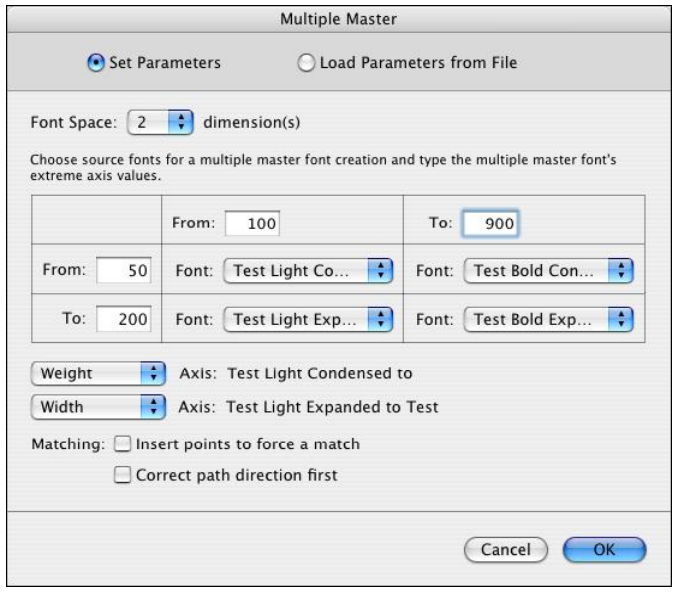
Including an optical size axis in your multiple master font provides a unique opportunity to optimize legibility for the entire range of sizes at which the font will be used. Traditionally, punch-cutters developing metal type would alter various aspects of a design depending on what looked correct for the size they were working on. With digital type, one master design gets scaled to all sizes, thus you lose some of the subtleties available in the best of the metal type era. Optical scaling gives this ability back to you.

Compared to a design intended for large point sizes, a small-size master design would have a larger x-height, wider counters and inter-character spacing, less contrast between thick and thin strokes, and heavier serifs. In designing an optical size axis, it is important to remember that the variations along an axis might be very subtle, and not always apparent when proofing on low-resolution laser printers.

You can now assign design coordinates to your font space coordinates.

Terminology: Design coordinate: a meaningful integer number used to relabel normalized coordinates – the only number a multiple master font user ever sees. Multiple master space cannot be divided any finer than integer design coordinates.

When running Adobe's Font Creator program, the numbers it displays are in their defined design coordinates. These are integers that map onto the font's normalized coordinate space. These numbers are basically arbitrary integers (related to the range of variation of any particular design, which you may assign as you wish, but should probably be related to Adobe's fonts for consistency). For example, if a design varies only slightly in width, the range should be smaller than if it varies greatly. For illustrative purposes, let us say the lightest weight font has a weight of 100 and the heaviest font has a weight of 900. The thinnest font has a width of 50 units, and the widest font is declared to be 200. (With these values, we have [(900 - 100) + 1] x [(200 - 50) + 1], or 120,951 possible derived fonts – the square is divided into 801 x 151 discrete positions by those numbers.)



Set ranges in Multiple Master Setup

Now that you have mapped out your multiple master design coordinate space, you should assign names to the master fonts. Let's call them:

- Test-LightCondensed (position [0,0])
- Test-BoldCondensed (position [1,0])
- Test-LightExpanded (position [0,1])
- Test-BoldExpanded (position [1,1])

Don't get too carried away with long names; remember that the Macintosh is limited to 29 characters in a PostScript font name.

Next, you should select an em square. When creating multiple master fonts, it is important for a number of font parameters to be the same, and the em square is the first place to start. The value 1000 is Fontographer's em square default, as well as the value Adobe uses for all Type 1 fonts. It's the number you should use for multiple master fonts. You can divide it up differently between ascent and descent if you like, but the font could have problems if you use something other than 1000 for your em square.

Now you can determine which fonts you would like to start with when you generate a multiple master typeface. Fontographer will generate any number of predefined variants (known as primary fonts, in multiple master lingo) when you create the multiple master face.

Terminology: Primary font: a generated font that, unlike a master font, need not lie on the corners of the design hypercube, but that can have any position desired in multiple master space. Primary fonts show up on the font menu; master design fonts do not.

The purpose of primary fonts is to have some fonts that correspond to the standard styles and weights of a typeface family, such as bold, black, light, and so on. Therefore, primary fonts are extremely unlikely to be at corners or edges of the cube; rather, they should be at the interior of the hypercube, which is where the more normal (read: less extreme) fonts will be found. Primary fonts are totally optional; their purpose is to have a selection of default, prebuilt fonts that will be compatible with existing applications, for users who haven't used the Font Creator yet.

Referring back to our illustration of the font coordinate space, you might want to create some intermediate fonts near each corner, but not at the extreme positions (which might be too extreme for general use). You might also want a font near the middle of the square, which could be a typical example of the regular or plain style of that typeface. Mark the positions where you would like a font, and write down the user coordinates of that position. You can choose [200 70], [200 150], [800 70], [800 150], and [400 150] as the primary fonts.

Ready to begin work? Great. Now just draw the four master designs. Be sure to make all the master fonts blendable. Corresponding characters in each master font must have the same number of contours and points, and each master design font must be able to be interpolated with the other masters.

➲ ***Tip:*** The easiest way to test this is to attempt to interpolate each master design with the design at the origin (0,0) of multiple master space. This process will interpolate along each diagonal of the hypercube and assures a high degree of compatibility between the master fonts (but not perfect compatibility).

If the results of the blend are what you expect, you are almost ready to generate a multiple master typeface.

Set up from file

Fontographer offers you the option of creating your one- and two-dimensional fonts via easy controls in the Multiple Master Setup dialog box. To set up more complicated multiple master fonts, you need to choose the **Load parameters from file** option, which allows for three- and four-dimensional fonts. Rather than build the world's most incredibly complex dialog box and interactive editing sequence, Fontographer uses a simple text file to control the creation of more complicated fonts. Simply create this file with any text editor, save it as a text-only file, and point Fontographer at that file via the **Choose** button.

Before you rush off to create your own control file, perhaps a bit of explanation will help you get it right. Please note that:

- Fields shown in double quotes should always be defined in straight double quotes (not typographical curly quotes).
- Fields in brackets should always be defined in brackets.
- All values should be defined in the order shown.
- You should never use commas between numbers in brackets.

A sample multiple master file follows, complete with explanation. Note that lines starting with a percent sign are comments and are ignored by the control file scanner. Do not use a tab or space at the beginning of your comment; you must use a % sign.

```
% Multiple master setup information for font 'Test'
% Created on Dec 30, 1992 by jve for multiple
% master tutorial.

% First, define the PostScript font name.
% It's recommended for ATM that all multiple
% master fonts end in MM.
% An optional second entry flags the
% existence of an Italic face for this font
"TestMM"   " TestMM-Italic"

% Next define the prefix name for this font. This is
% prepended to the primary name suffixes below to
% make a complete font name.
"TestMM_"
% This name must end in an            underscore.

% Now we must specify the number of dimensions
% of multiple master space. We call
% these 'axes'. This example is two dimensional.
2 axes
% type      long lbl  short lbl [[des_coord norm_value]...]
"Weight"    "Weight"" wt"      [[100 0] [900 1]]
"Width"     "Width"  " wd"     [[50 0] [200 1]]

% We have to declare how many master designs
% there are. This should be related to
% the number of axes as described in the text
% above.. The FOND name suffix is
```

```
% automatically built from the prefix above,
% prepended with %M, thus the names are
% "%MTestMM_100 50", etc. This %M is so the
% master fonts don't show up in the font menu.

4 masters
% Fonto. file name FOND suffix        normalized coord
"TestExtraLightExtraCond"  "%MTestMM_100 50"       [0 0]
"TestExtraBoldExtraCond"   "%MTestMM_900 50"       [1 0]
"TestExtraLightExtraExt"   "%MTestMM_100 200"      [0 1]
"TestExtraBoldExtraExt"    "%MTestMM_900 200"      [1 1]

% The style controls how much weight to add to
% get a bolder font. You can theoretically do other
% neat things with styles, but I don't believe in them
% yet... The style table below adds 200 to the weight
% of a 100-weight font, and 100 to the weight of a 900-
% weight font to get a bolder variant.

1 style

%Mac style          flag      axis      entries
1 0          1          [[100 200][900 100]]

% Now define our primary fonts

primaries

% name suffix       [user coord]      [FOND NFNT]     file name
"200 LT 70 CN"      [200 70]          [12345 23456]
"200 LT 150 EX"     [200 150]         [12346 23457]   "TestLTEX.bmap"
"800 BL 70 CN"      [800 70]          [12347 23458]
"800 BL 150 EX"     [800 150]         [12348 23459]
% Now, to turn ON the "Insert points to force match"
% and       "Correct path directions first" options, be
% sure to include the word "flags" in the last line of
% the setup file. If you want those options OFF,
% leave the flags out.
% That does it.!
```

Fontographer does very little checking of this input, assuming that you are a careful person. (And if you aren't now, that's fine; you will be by the time you get one of these to work.)

We break out sections of the setup file here. An explanation follows each section.

```
% First, define the PostScript font name.
% It's recommended for ATM that all multiple
% master fonts end in MM.
% An optional second entry flags the
% existence of an Italic face for this font
"TestMM"   " TestMM-Italic"
```

First, you need to declare the PostScript name of the font. This could be a name like MyFontMM-Roman, MyFontMM-Italic, or simply MyFontMM. We recommend that the name include "MM" to flag it as a multiple master font. A truncated version (truncated by the 5-3-3 rule used for all PostScript fonts) of this name will be used to name the PostScript-language font file. If the optional second entry on that line is present, it should contain the font name of the associated italic face. It informs Fontographer that there is an italic (or bold, bold italic, and so on) face defined for this plain face, and Fontographer will automatically build the correct style table to associate the italic face with this plain face. When you generate the associated italic face, the first name would be, in this example, TestMM-Italic, and the PostScript-language font file name would be *TestMMIta*.

```
% Next define the prefix name for this font. This is
% prepended to the primary name suffixes below to
% make a complete font name.
"TestMM_"
% This name must end in an          underscore.
```

The second declaration is that of the prefix name for the primary font names. It should be a 5-3-3 truncated version of the font name, ending in an underscore character. This is the name that will be added to the primary font names to make a complete font name. Adobe recommends the following naming convention:

```
FamilyNameMM_[Style/Char Set]     Number1
[Label1]   Numbern          [Labeln]
```

In the example above, the first primary font would thus be named TestMM_200 XL 70 XC (where "XL" and "XC" are the labels). Of course, this name would not be a legal PostScript font name; Fontographer replaces blanks with underscore characters when generating the actual font name, giving a PostScript font name of TestMM_200_XL_70_XC. It is important that the name, up to the first underscore, be the same as the PostScript font file name, so the font file may be found. The Style/Char Set – which is used when more than one MM font per family is required – and labels are optional, but recommended for clarity.

```
% Now we must specify the number of dimensions
% of multiple master space. We call
% these 'axes'. This example is two dimensional.
2 axes
% type      long lbl  short lbl[[des_coord norm_value]...]
"Weight"    "Weight"" wt"      [[100 0] [900 1]]
"Width"     "Width"  " wd"     [[50 0] [200 1]]
```

The axes are defined next, and you should label them properly. Adobe has set up a registry for axis types (the first keyword). Currently defined axis types are Weight, Width, and Optical Size. These should be in English. If you want to define more axis types, you should register them with the Adobe PostScript Developer Support Group. This first axis keyword is intended to allow applications to determine the axis type, so standardization is pretty important if you want to be a good citizen and create a multiple master font that works with most applications. It is highly recommended that you list the axes in this order (weight, width, then optical scale) if those are the axes you are using.

The second keyword is the long label presented to the multiple master user and may be translated into other languages as required. It is used in user-interface dialog boxes.

The third keyword is a short label and should be "wt" (Weight), "wd" (Width), or "op" (Optical Size). It is used in font name construction and may be language dependent. Like the long label, it is used in user-interface dialog boxes. Following the labels is the declaration of the user coordinate to design coordinate mapping. There may be several bracketed numbers in the outer brackets. In the example shown, the first line maps user coordinate 100 to design coordinate 0, and user coordinate 900 to design coordinate 1. The second line maps user coordinate 50 to 0, and user coordinate 200 to 1.

Notes about point mappings

Normally, the design coordinates of an optical size axis might vary from 6-point to 72-point – the normal range of use for most publishing purposes. Although linear interpolation over that range provides better quality than simple scaling of a single master design, the results can be significantly improved by using a nonlinear interpolation. This is necessary because design features must change more rapidly in the lower range of sizes than they do in the high range. Nonlinear mappings may be specified only in the Set up from file option.

For example, Adobe's Minion™ multiple master font maps the 6-point design coordinate to 0.0 of the normalized coordinates, 8-point to .35, 12-point to .5, 18-point to .75, and 72-point to 1.00. In the control file, that would look like this: ([6 0] [8 .35] [12 .5] [18 .75] [72 1]).

These values are a good starting point for text faces with similar design parameters; optimal values for any given typeface would depend both on the characteristics of the face and the dynamic range of the axes in the font.

In multiple master fonts, you can do this non-linear, piece-wise linear mapping for other axes than just optical size, but we're not sure if it has much of a design need for weight and width axes. Someone will probably find some creative use of it for Weight or Width, or they may dream up other axis types where it might be useful.

Note: Nonlinear mapping can only be specified in the setup from a file option.

The master declaration section declares the name of the font files containing the master design. These files may be in any format that Fontographer normally opens. Fontographer will look for the master fonts in the same folder or directory where the setup file is located and will open each, one at a time. You don't have to open them yourself.

% Fontographer file name	FOND name suffix	Normalized coord
"TestExtraLightExtraCond"	"%MTestMM_100 50"	[0 0]
"TestExtraBoldExtraCond"	"%MTestMM_900 50"	[1 0]
"TestExtraLightExtraExt"	"%MTestMM_100 200"	[0 1]
"TestExtraBoldExtraExt"	"%MTestMM_900 200"	[1 1]

Plan out your FOND ID and, optionally, NFNT ID numbers. Each primary font requires a FOND, but the NFNT for the 10-point bitmap of that primary font is optional. Fontographer automatically pulls the 10-point font from the master font file closest (in design space) to the primary font it is generating; so you may want to create a 10-point bitmap font for each master design. You can override this automatic font selection by specifying a bitmap file that will be opened during multiple master generation, if you want to make custom bitmaps for the primary fonts.

Note: This is optional since Adobe makes dummy bitmaps.

The FOND name for the master fonts is special. It should be preceded by a percent sign and an M. The percent sign creates a FOND with a hidden name, and the M signifies a multiple master font. The rest of the name, up through the underscore character, should be the same as the prefix name defined on the second noncomment line of the control file. The numbers following the name indicate the design space weights for each of the master fonts. Since this name is hidden, it isn't terribly clear why it is needed, but these FONDs seem to be necessary.

Finally, the normalized coordinate of each master design font should be declared. This is the label for the corner of the hypercube that the font is positioned on. Adobe recommends that you use the lightest, most condensed, smallest optically-scaled design as the origin font.

The style table controls how automatic bolding changes the design. Many applications have a technique for getting a bold face by typing **COMMAND-B** (or some similar sequence). If you would like your font to respond automatically to this, you can define a style table that will automatically generate a bold version of any weight. If you don't want this, specify zero for the number of styles.

```
% The style controls how much weight to add to
% get a bolder font. You can theoretically do other
% neat things with styles, but I don't believe in them
```

```
% yet... The style table below adds 200 to the weight
% of a 100-weight font, and 100 to the weight of a 900-
% weight font to get a bolder variant.

1 style

%Mac style          flag       axis       entries
1 0          1          [[100 200][900 100]]
```

You can easily make fonts bold with a simple menu selection. This ability has been carried over to multiple masters. You should decide, for each weight, how much to add to the weight to get a bold version. Thus in the above example, let us say that at the minimum weight of 100 we can produce a bold version of that weight by adding 200 to the weight. At a weight of 900, we can produce a bolder version by adding 100 to the weight. Multiple master fonts, when emboldened, will produce a bolder version by interpolating the weights. Thus at the intermediate weight of 500 (halfway between 100 and 900), a bolder version could be generated by adding 150 (halfway between 200 and 100), giving a total weight of 650. A bold version of the 900 weight face would be given a weight of 1000. You should think about the embolded weights, since you will have to specify them later.

You should specify the style code for bold (1), a flag (always 0 for now), the design axis that is to be varied (1 in this case, since the first axis is the weight axis that we wish to change), and the entries. The entries are more powerful (that is, complex) than necessary. You can have one or two pairs of entries. The first pair of numbers should be the starting design coordinate weight for boldness (100 in this case), followed by the weight to add at that design coordinate (200 in this case). The second (optional) pair of numbers is the ending design coordinate for boldness (900 in this case), followed by the weight to add at that ending coordinate (100 in this case). You can specify a single pair of numbers if you wish; then the second number is the boldness to add to all values, and the first number is meaningless.

Note: The first primary font is used as a default. It must not have the same design coordinates as a master design font or Font Creator might crash.

```
% Now define our primary fonts

primaries
```

% name suffix	[user coord]	[FOND NFNT]	file name (opt.)
"200 LT 70 CN"	[200 70]	[12345 23456]	
"200 LT 150 EX"	[200 150]	[12346 23457]	"TestLTEX.bmap"
"800 BL 70 CN"	[800 70]	[12347 23458]	
"800 BL 150 EX"	[800 150]	[12348 23459]	

The primary table is the last one in the control file. Primary font names consist of a prefix and a suffix. The prefix has already been defined, so these entries need only define the suffix. The suffix name uses two-letter abbreviations for human-readable indications of weight and width, interspersed with numeric values for those parameters. The numeric user coordinate position of this primary font is declared next, followed by the FOND and NFNT IDs. Optionally, you can specify a file name that Fontographer will open and pull a 10-point bitmap from when generating that primary font. If this file isn't specified, Fontographer will take the closest master design's 10-point bitmap, copying and renumbering as necessary. The source bitmap need not have the same FOND or NFNT ID as specified here; Fontographer renumbers properly as it copies.

Note: Putting a"0" in for the number tells Fontographer to pick a nonconflicting number. It will pick a different one each time you run. The *Fontographer.log* tells you which one it picked.

Suggested values for the two-letter abbreviation are shown here.

Axis	Long name	Abbreviation
Weight	Extra Light	XL
Weight	Light	LT
Weight	Regular	RG
Weight	Semibold	SB
Weight	Bold	BD
Weight	Black	BL
Weight	Extra Black	XB
Width	Extra Condensed	XC
Width	Condensed	CN
Width	Semi Condensed	SC
Width	Normal	NO
Width	Semi Extended	SE
Width	Extended	EX
Width	Extra Extended	XE

When you make a new font derived from your multiple master font by using Adobe's Font Creator, the names generated are always the two-letter abbreviation given with the axis specification. The only way to get fonts named with the two-letter abbreviations above is to create them as primary fonts when the multiple master font is generated.

We allow for the inclusion of one more entry in the setup file: determining the status of the checkboxes in the Font Blend and Multiple Master dialog boxes. By default, these boxes are checked when Fontographer is launched. However, since those using **Load parameters from file** will be the super-expert font designers creating three- and four-dimensional fonts, it is unlikely that any of them would want Fontographer to insert points to force a match or reset predetermined origin points. Therefore, these options are turned off when selecting **Load parameters from file**. You can turn them back on by typing the word "flags" anywhere in the last line.

Generating a multiple master font...

Now that your font is designed and the setup file is complete, you are ready for your first attempt at multiple master generation.

1. Open your first master font.
2. Go to the end of the **Element** menu and select **Multiple Master**.
3. When the dialog box appears, choose **Load parameters from file**.
4. Click the **Choose** button, and Fontographer will put up the standard file dialog box so you can locate your multiple master setup control file. (The one in our test sample is named *Test.MM.setup*.) Once you locate that file, Fontographer reads it, echoing what it read to a file named *Fontographer.log* so you can later consult that file for a list of the errors (most likely generated by your first attempts). Fontographer will automatically open the other master fonts designated in your setup file.
5. Then you select **Generate Font Files**, choosing Multiple Master, and Fontographer will produce a multiple master font in the PostScript-language font file (the one with the LaserWriter icon). It also produces a bitmap file named the same as the master font, but with a suffix of ".mmbm" (multiple master bitmap). This corresponds to the ".bmap" file of normal Macintosh font generation.
6. These fonts may be installed in the normal way.

Of course, there are still a few things that can go wrong. Not only must the points of corresponding master glyphs be compatible, but the same type of hints must be defined in each glyph. If you were unlucky enough to create a font that automatically hinted differently in the different glyphs, you will have to manually edit the glyphs that are different. See "Font hinting" on page 392 for details on how to do this.

The log file produced by Fontographer gives precise information about where the hints differ between glyphs. When it finds a mismatch, it tells you how many it found, the glyph(s) containing the error, along with an indication of the type of error. Instead of stopping, Fontographer continues on with the rest of the font, so it can find all the problems with one run. If the problem is a point mismatch, the error report lists the glyph in the font (1, 2, 36, 54, and so on) and tells you there is a different number of points. Similarly, if the curves and lines don't match properly, the error report tells you the point at which things went wrong, so you can go into the outline window and step through the points and contours until you find the one with the problem.

A more difficult-to-correct class of error occurs when the points match, but the origin points differ. In this case, Fontographer makes a complete error list of the Type 1 font instructions generated by each glyph. These are called compatibility check errors. You can look at this list to see how the glyphs differ. For help in understanding the instructions, we recommend getting a copy of *Adobe's Type 1 Font Specification*. A couple of the more frequently encountered instructions are vstem (vertical stem) and hstem (horizontal stem). If you have a hint mismatch, Fontographer informs you that you can typically open the offending glyphs in each master font, and correct the one that is wrong by manually adding or deleting hints.

Note: If you don't care whether or not your multiple master font has hints, you can ignore hinting by turning hints off in the Generate Font Files dialog box. This is a good way to avoid problems when generating Multiple Master fonts, but this approach involves the high penalty of making fonts without hints.

Reference

The Reference chapter contains detailed descriptions of every element of the program, menu commands, window options, and tools.

14

Windows in Fontographer
1. Font window

The font window shows you all the glyphs in your font at approx. 24 points. The **View by** menu at the top of the window allows you to change the label over each glyph slot. The default choice display mode is Character, which shows the character symbol above each slot.

If you choose Keystroke to display the keystrokes that correspond to each character, and scroll the window down so that you can see the international characters, you will see some cryptic things like SOe and Oee. SOe means that to produce this character you need to press **Shift**, **Option**, and the "e" key simultaneously, producing the character "‰". In the case of Oee the procedure is slightly different: press **Option** and the "e" key simultaneously, then release and press the "e" key by itself.

View by menu

Character

The Character item shows the symbol that corresponds to each slot in the font window.

Note: When you see a double-asterisk (**) above a glyph, it means that you cannot access the glyph from the keyboard using the currently selected encoding and keyboard layout. You can change encoding in the **Font Info** dialog box in the **Element** menu.

Keystroke

The Keystroke item corresponds to the keyboard sequence used to enter the glyph.

Note: When you see a double-asterisk (**) above a glyph, it means that you cannot access the glyph from the keyboard using the currently selected encoding and keyboard layout. You can change encoding in the **Font Info** dialog box in the **Element** menu or the keyboard layout in the **Input** menu.

Unicode

The Unicode item corresponds to the glyph's Unicode codepoint in hexadecimal notation. For example: "A" displays as "0041".

Decimal

The Decimal item shows the decimal value of a glyph's current slot. For example, "A" is in decimal location 65.

Hexadecimal

The Hexadecimal item shows hexadecimal (base 16) values. For example: "A" displays as "$41".

Octal

The Octal item shows octal (base 8) values. For example: "A" displays as "o101".

Width

The Width item shows the glyph's width in em units. Em units are not related to point size or any other physical measurement.

Note: In the Width mode, when you see a double-asterisk (**) above a glyph, it means that the slot is undefined.

Left sidebearing

The left sidebearing mode shows the measurement of the glyph's left sidebearing – the distance between the origin line and the left edge of the glyph's outline – in em units.

Note: A negative number indicates that some part of the glyph's outline extends beyond the left sidebearing. A positive number means that space exists between the sidebearing and the left edge of the outline. In the left sidebearing mode, when you see a double-asterisk (**) above a glyph, it means that the glyph is undefined.

Right sidebearing

The Right sidebearing mode shows the measurement of the glyph's right sidebearing (the distance between the width line and the right edge of the glyph's outline) in em units.

Note: A negative number indicates that some part of the glyph's outline extends beyond the right sidebearing. A positive number means that space exists between the sidebearing and the right edge of the outline. In the right sidebearing mode, when you see a double-asterisk (**) above a glyph, it means that the glyph is undefined.

Fill Tint

The Fill Tint item shows the percentage of black that will fill the glyph when you output a PostScript or Type 3 font. (We recommend using 100%.)

Note: In the Fill Tint mode, when you see a double-asterisk (**) above a character, it means that the glyph is unfilled or that it is undefined using currently selected encoding.

Stroke Tint

The Stroke Tint item shows the percentage of black that fills strokes when you output a PostScript or Type 3 font. (We recommend using 100%.)

Note: In the Stroke Tint mode, when you see a double-asterisk (**) above a glyph, it means that the glyph is not stroked or that it is undefined.

Stroke Weight

The Weight item shows the glyph's stroke weight in em square units when you output a PostScript Type 3 font.

Note: In the Weight mode, when you see a double-asterisk (**) above a glyph, it means that the glyph slot is empty, includes no weight in em square units, or is undefined.

Glyph properties

You can see different properties of the selected glyph slot at the right of the **View by** menu. Name, keystroke, Unicode codepoint, Unicode name, decimal or hex values can be combined in any order or combination. The "plus" button adds an item to the list. To remove an item, choose **Remove Item** in the item's pop-up menu.

Searching for glyphs

The font window has the Spotlight field at the right top corner, which allows you to quickly find particular glyphs in the font.

To search using the Spotlight:

1. Click on the Magnifying glass icon in the upper-right corner of your screen and select the criteria for the search: text ranges, glyph name, Unicode codepoint or Unicode name.
2. Type in the field. Glyph cells that meet the criteria will be marked green in the font window.

For example, typing "034" will find and mark glyphs "zero", "three" and "four" when *Search by text ranges* was selected or it will mark glyph "four" if *Search by Unicode codepoint* was selected.

2. Outline window

The outline window shows you the outlines (or strokes) of the glyph named in the title bar and allows you to edit them. At the top of the window (in the info bar) you will see position indicators. The numbers below them are horizontal and vertical distances measured in em units. These distances let you know the distance from the cursor or selected points to certain objects in the window.

Whenever you select a tool from the tool palette in the outline window, the information displayed in the information bar will update to accommodate the specific tool and its behavior. Some of the indicators that appear are horizontal and vertical displacement, angle indicator, and length. These indicators allow for precise control of your tools.

The lock at the bottom of the window indicates that you cannot change from the glyph in the outline window to some other glyph by merely typing the keystroke(s) of that glyph. If you click the lock, it will open and you will then be able to change glyphs simply by typing the keystroke(s) of the new glyph. You can also toggle the lock icon on and off by pressing the **Return** or **Enter** key. Fontographer will not save the state of this lock when you quit the program.

You can close all open outline windows by **Option**-clicking the close box of any open outline window. Any open bitmap or metrics windows will remain open.

Tool palette

Pointer tool

The pointer tool is for selecting and dragging objects. To change to the pointer tool when another tool is being used, while the lock icon is locked, type the ACCENT GRAVE key (`). To temporarily use the pointer tool when another tool is selected, press the COMMAND key and release to deselect the pointer tool.

Hand tool

The hand tool lets you scroll through the display area – useful for large characters. To temporarily change from the other tools to the hand tool, hold down the SPACEBAR.

Rectangle tool

The rectangle tool is for drawing rectangles, squares, and rectangles with rounded corners. To change the radius of curve at the corners of the square or rectangle, double-click the icon and make changes from the dialog box. To constrain the shape to a square, hold down the SHIFT key while dragging. Press the OPTION key and drag with the mouse to change the shape of the rectangle to a square, using the point clicked on as the center of the figure. To access the rectangle tool from another tool, while the lock icon is locked, type the number **1**.

Multigon tool

The multigon tool draws starbursts and regular polygons. Double-click the tool to bring up a dialog box that controls the shape. From the dialog box specify polygon or star shape, number of sides, and shape of points. To access the multigon tool from another tool, while the lock icon is locked, type the number **2**.

Oval tool

The oval tool creates ovals and circles. Hold down the OPTION key to draw the oval outward from the center. To constrain the oval to a circle, hold down the SHIFT key and drag. You can choose this tool, while the lock icon is locked, by typing the number **3**.

Straight line tool

The straight line tool draws straight lines without requiring you to place points manually. To constrain the line to the vertical, horizontal, and 45-degree angles, hold down the **Shift** key while dragging with the mouse. To access the straight line tool from another tool, while the lock icon is locked, type the number **4**. Press the **Option** key and click the mouse to draw a straight line that centers on the point where you click, and extends outward as you drag.

Freehand tool

The freehand tool can be used to draw paths freestyle or as either a variable-weight or calligraphic pen tool.

The calligraphic pen tool draws calligraphic lines. You can set the pen width and angle from the dialog box accessed by double-clicking the icon. To use the calligraphy pen when the selection pointer or other tools are selected, while the lock icon is locked, type the number **5**.

From the dialog box (accessed by double-clicking the icon) you can choose the freehand tool by selecting the pressure-sensitive option. Choose the maximum and minimum widths of strokes, as well as the shapes of caps and joins.

The **Tight curve fit** option lets you increase how much the outline conforms to the curved lines you draw. (The other option is normal fit, the default option, ordinarily used if you ignore this choice.) Choose **Draw dotted line** to display your lines as a series of hash marks; this option displays the strokes faster than does the normal option.

Note: Your strokes will not appear as a dotted line or as hash marks when printed.

Pen tool

The pen tool is for drawing outlines or strokes without having to change tools. You can choose this tool, while the lock icon is locked, by typing the number **6**.

Knife tool

Use the knife tool to cut paths or points. Access the knife when using another tool, while the lock icon is locked, by typing the number **7**.

Curve tool

The curve tool creates a point with curves on both sides. The angle of the curve can be adjusted from either of the incoming or outgoing BCPs, and the radius of the curve can be adjusted individually by the BCP on the appropriate side of the point. You can choose this tool, while the lock icon is locked, by typing the number **8**.

Corner tool

The corner tool connects two straight lines or two curves at a cusp. Both the incoming and outgoing paths can be adjusted with their respectively independent BCPs. You can choose this tool, while the lock icon is locked, by typing the number **9**.

Tangent tool

The tangent tool is used to connect straight lines to curves for smooth joins. You can choose this tool, while the lock icon is locked, by typing the number **0**.

Rotate tool

Use the rotate tool to rotate a character or its parts, centering the rotation around the place where the mouse clicks. Double-click the tool to access the Transform dialog box and specify degrees of rotation.

Click the mouse and drag to display a radius used to rotate the image on screen. (The rotation occurs around the point where the mouse clicked on the screen.) If you press the **Shift** key and click with the mouse and drag, you will constrain the movement of the radius to horizontal, vertical, and 45-degree angles.

Flip tool

The flip tool functions similarly to the rotate tool by allowing you to flip a figure horizontally or vertically, centering the movement around the click of the mouse. Double-click the tool to get the Transform dialog box where you can choose either direction, horizontal or vertical.

Click the mouse and press the **Shift** key to display a radius used to rotate the image on screen. (The rotation occurs around the point where the mouse clicked on the screen.) If you click with the mouse and drag, you will constrain the movement of the radius to horizontal, vertical, and 45-degree angles.

Note: The rotate and flip tools are similar in function. The difference between the two tools is that the flip tool constrains rotation to 45-degree increments unless the **Shift** key is held down, while the rotate tool requires the **Shift** key be depressed to constrain movement.

Scale tool

Use the scale tool to scale a figure horizontally and vertically, centering around the click of the mouse. Double-click the tool to access the Transform dialog box, where you can enter horizontal and/or vertical degrees for scaling. **Option**-double-clicking the Scale tool brings up the Transform dialog box with Scale Uniformly selected as the first transformation.

To scale horizontally and vertically, click the mouse and drag in either direction. To constrain scaling to horizontal, 45-degree, or vertical axes, just press the **Shift** key and drag to those locations.

Skew tool

The skew tool lets you skew figures, both horizontally and vertically, centering the movement around the click of the mouse. Double-click the tool to access the Transform dialog box where you can enter horizontal and/or vertical degrees to skew the selection.

To skew horizontally and vertically, click the mouse and drag in either direction. To constrain skewing to horizontal, 45-degree, or vertical axes, just press the **Shift** key and drag in any of those directions.

Measuring tool

The measuring tool measures distances in em units in the outline window.

Press the **Option** key and click the mouse to draw a measuring line that centers on the point where you click, and extends outward as you drag. If you press **Shift**, and click with the mouse and drag, you will constrain the movement of the measuring line to horizontal, vertical, and 45-degree angles.

Magnifying tool

Use the magnifying tool to enlarge or reduce the image in the window. With the tool selected, click the place you want to center in the window, to enlarge the image (or type **Command-Space**-click to temporarily invoke the tool). Press the **Option** key and click with this tool to reduce the image.

With the magnifying tool selected you can drag a box around the area you wish to magnify, and Fontographer will zoom into the area enclosed by the box. You can also use this method to zoom out by holding down the **Option** key.

Note: The same key combinations may be assigned to switching keyboard layouts in the **Input** menu. In this case pressing before **Command** helps.

Perspective tool

The Perspective tool works in conjunction with 3D rotate to apply three-dimensional rotations to two-dimensional objects, while still maintaining perspective. **Option**-double-clicking the tool brings up the Perspective Setup dialog box where you can set the Perspective Distance and select a Perspective Point of Basepoint, Center of selection, Character origin, or Mouse click. Double-clicking the tool brings up the Transform dialog box ready to apply a 3D move transformation.

Arc tool

The Arc tool allows you to create one-fourth of an oval and arc-like shapes. Double-click the tool to access the Arc Tool Setup dialog box, where you can choose to create Open, Flipped, or Concave arcs. You can also create different types of arcs without changing the Setup dialog box by using modifier keys. Hold down the **Command** key while creating a new arc to toggle between creating an open and a closed arc. Hold down the **Option** key to flip the arc both horizontally and vertically. The **Caps Lock** key toggles between creating a convex and a concave arc.

Note: Unlike **Command** and **Option**, you do not have to hold down the **Caps Lock** key while dragging the mouse.

To constrain the arc tool to create quarter circles, hold down the **Shift** key. The **Option** key is the demagnified move modifier key. By holding down **Option** when you create a new arc, you can increase/decrease the size of the arc in one em-unit increments.

Layers palette

The Layers palette displays in the outline window. It shows which layer is currently active and which ones are visible. The highlighted layer shows which one is currently in use. When an eye appears in the checkbox beside a layer, that layer is displayed in the current window. You can click the checkboxes of the layers on or off to show or hide them.

Outline layer

The Outline layer is where you will create and edit your outline glyph.

Template layer

The Template layer is where you will place images you want to use as a template for drawing. Any points you draw here will not be used for PostScript creation, but will show gray in the Outline and Guides layers. You can also paste artwork or scanned images into the Template layer for hand-tracing or autotracing. Each glyph in the font has its own Template layer.

Guides layer

The Guides layer is where you place guidelines and outlines to help you draw a glyph. Dragging from the origin and baselines creates horizontal and vertical guidelines that will appear behind every glyph in the font. Like guidelines, any outlines drawn here will not be used for PostScript creation. They will appear gray behind the glyphs.

Hints layer

The Hints layer displays hints that define vertical and horizontal stems. Here you can adjust the hinting of individual glyphs and edit glyph outlines.

Changing and hiding layers

You can turn layers on and off by clicking the checkbox next to their name in the Layers palette. When the lock icon is in the locked position, Fontographer also allows you to change layers simply by typing **O**, **T**, **G**, or **H**.

When the lock icon is in the locked position, you can also hide the Outline, Template, Guides, and Hints layers from view.

- Hold down the **Option** key, and type **O** to hide the Outline layer from view. Repeat to bring the Outline layer back into view.
- Repeat this procedure to hide the Template, Guides, and Hints layers, substituting **T**, **G**, and **H**, for the **O**.

Magnification

You can change the magnification of the outline window with the **Magnification** menu item in the **View** menu or you can use the magnifying tool or keyboard shortcuts.

To increase the magnification, hold down the **Command-Spacebar** keys and click the place you want to center in the window.

To reduce the magnification, hold down the **Option-Command-Spacebar** keys and click in the window.

Command-plus and **Command-minus** can be used for the same purpose.

Switching glyphs

If you need to change the glyph in the outline window to the next sequential glyph, you can use the **View** menu's **Next Glyph** item – **Command-]**.

To change to the previous sequential glyph, use the **View** menu's **Previous Glyph** item – **Command-[**. When the lock icon is off, you can change glyphs simply by typing them on the keyboard. You also can type glyph names when the lock icon is off. For example, typing "four" will show glyph "4" while typing "dollar" will show "$". In fact, typing only first letters of the glyph name will be enough.

3. Bitmap window

Use the bitmap window to edit the various sizes of bitmap fonts that Fontographer creates. It works very much like FatBits in a paint-type program. You can use a pencil or other tools to turn the dots in the bitmap on or off. The glyph's outline is visible in the background to help you place the dots properly. Each of the dots in this window is equivalent to one point (pixel) on the computer screen. At the top left area of the screen, you will see the current size preview of the bitmap.

Maximum ascender and descender lines will appear if a glyph's outline extends above the ascender line or descender line. If you have selected to preserve line spacing (in the Recalc Bitmaps dialog box), these guides will not appear for that glyph, as they will be the same as the normal ascender and descender lines.

At the bottom of the bitmap window is a lock icon. When the lock is closed, you will not be able to change glyphs by merely typing the keystroke(s) for the new glyph. If you click the lock, it will open and you will be able to change glyphs by simply typing the keystroke(s) of the new glyph.

Press the **Enter** or **Return** key to change the status of the lock.

You can close all open bitmap windows by **Option**-clicking the close box of any open bitmap window. Any open metrics or outline windows will remain open.

Tool palette

Straight line tool

The straight line tool draws straight lines without requiring you to place points manually. To constrain the line to the vertical, horizontal, and 45-degree angles, hold down the **Shift** key while dragging with the mouse. To access from another tool, while the lock icon is locked, type the number **1**.

Hand tool

Use the hand tool to move the display area – this is useful for large characters. To change from the other tools to the hand tool, hold down the **Spacebar**. You can use the tool as long as the **Spacebar** is held down. Or, while the lock icon is locked, type the number **2** to select the hand tool.

Pencil tool

Use the pencil tool to change the bits of a bitmap glyph in the bitmap window. To access the pencil tool from another tool, while the lock icon is locked, type the number **3**.

Eraser tool

The eraser tool will remove pixels when you drag it across them. Double-click the eraser tool icon to remove all pixels in the glyph bitmap. To use the eraser tool when another tool is selected, while the lock icon is locked, type the number **4**.

Marquee tool

Use the marquee tool to outline an area and display its pixel grid. The gridded section can be moved around the window by dragging it. To place the moved bits and deselect the marquee, click outside of the movable section. To use the marquee tool when another tool is selected, while the lock icon is locked, type the number **5**.

Move tool

Use the move tool to move the bitmap away from its outline in any direction. Click the glyph bitmap and drag to a new location. Access the move tool, while the lock icon is locked, by typing the number **6** when another tool is selected.

Measuring tool

The measuring tool measures distances in pixels. Press the **Shift** key and drag the mouse in any direction to use a deconstrained measuring line. Press the **Option** key and click to extend a measuring line that centers on the clicked point, and that rotates constraining to 45-degree, horizontal and vertical angles. Select the measuring tool when using another tool by typing the number **7**.

Note: You may have noticed that the **Shift** key has an opposite effect on the measuring tool in the bitmap window, when compared to the way it works in the outline window mentioned earlier. We found that users in the bitmap window most frequently want constrained measuring due to the organized way that the bits align, while the more freeform nature of outline editing requires that the measuring tool default to deconstrained measurement in the Outline layer.

Magnifying tool

Use the magnifying tool to enlarge or reduce the image in the window. With the tool selected, click the place you want to center in the window to enlarge the image or press **Option**-click with this tool to reduce the image. To temporarily invoke this tool, press **Command-Spacebar**-click to enlarge and **Command-Option-Spacebar**-click to reduce. Select the magnifying tool when another tool is selected by typing the number **8**.

Note: The same key combinations may be assigned to switching keyboard layouts in the **Input** menu. In this case pressing **Spacebar** before **Command** helps.

Command-plus and **Command-minus** can be used for magnification as well.

Ascent/Descent/Offset/Width

In the info bar at the top of the window, you find the glyph's ascent, descent, offset, and width numbers. The ascent is the maximum distance above the baseline measured in points. The descent is the maximum distance below the baseline measured in points. The offset is the number of points between the origin line and the leftmost point in the glyph. The width indicator shows us the width of the glyph in points.

Recalculate From outline

When you press this button, the bitmap will be recalculated in order to correspond with the outline of the glyph. This is useful when you change a glyph's outlines after having created bitmaps. To recalculate bitmaps for more than one glyph at a time use the **Recalc Bitmaps** command in the **Element** menu.

Scrolling

The **SPACEBAR** switches the current tool to a hand tool that will allow you to scroll the glyph back and forth in the window.

Switching characters

If you need to change the glyph in the bitmap window to the next sequential glyph, you can use the **View** menu's **Next Glyph** item – **COMMAND-]**.

To change to the previous sequential glyph, use the **View** menu's **Previous Glyph** item – **COMMAND-[**. When the lock icon is off, you can change glyphs simply by typing in the glyph keystroke.

Changing point sizes

To select the Next or Previous bitmap point size choose one of these menu items from the **View** menu. **COMMAND-OPTION+DOWN ARROW** will give you the next point size up. **COMMAND-OPTION+UP ARROW** will give you the previous point size.

Note: If only one size exists these items will appear dimmed. To generate more point sizes, go to the **Bitmap Info** item in the **Element** menu.

4. Metrics window

The metrics window displays the metrics for characters you type into the textbox at the top left. Here you can edit widths, sidebearings, and kerning pairs for those characters. Underneath the characters is a table that displays the numerical values in em units that apply to each character and its kerned pair. Changes to the kerning and width appear in the table below the screen area.

Width is the distance in em units between the origin point and the width line. Kern is the number of em units (negative) the character on the right overlaps or is pushed away from the character on the left (positive). You can change kerning distance, left and right sidebearings, and width by changing the numbers in the table, or by using the kerning and sidebearing lines for each character.

You can close all open metrics windows by **Option**-clicking the close box of any open metrics window. Any open bitmap or outline windows will remain open.

The keys to using the metrics window are:

- The **Tab** and **Right arrow** keys allow you to move between slots from left to right.
- The **Shift-Tab** and **Left arrow** keys allow you to move between slots from right to left.
- Typing a character when a slot is highlighted changes the character in that slot.

Kerning and sidebearing lines

Use the kerning or sidebearing lines to adjust the amount of space that a character has on its left and right edges. Click any character, and drag its sidebearings or kerning line to the desired location to affect the amount of empty space that will exist between that character and those on either side of it.

Key commands to change spacing and/or kerning

- **OPTION-H**

  Toggles sidebearings and kerning lines on or off when you click the screen area (not inside the textbox).

- **OPTION-G**

  Toggles Guides on or off when you click the screen area (not within the textbox).

- **OPTION-LEFT ARROW/RIGHT ARROW**

  When a character cell is selected, changes the selected character to the following character in the font sequence.

- **OPTION-LEFT ARROW/RIGHT ARROW**

  When a numeric cell is selected, adjusts the values up or down by one unit. Holding down the **SHIFT** key simultaneously adjusts by ten units at a time.

- **COMMAND-'**

  Views the Next kerning pair.

- **COMMAND-;**

  Shows the Previous kerning pair.

- **COMMAND-]**

  Shows the Next character.

- **COMMAND-[**

  Shows the Previous character.

Show kerning

To display the characters in the textbox with or without kerning information, click the **Show kerning** icon. When the option is switched on (K icon is filled), the kerning information is displayed on screen. Use this option to see how your font will appear in a program that does not support kerning.

Kerning is on

Kerning is off

Load Text

Use the **Text** link to load text from a file into the textbox and display area. Click the **Up** and **Down arrows** at the right of the textbox to scroll vertically within the file.

Menus

The Fontographer menu

About Fontographer

This dialog box shows you information about your current setup. You can see the version number in the right bottom corner. Click in the window to make it disappear.

Preferences

By selecting **Preferences** you can alter the default settings for Fontographer. The settings that you can change cover many aspects of the program, from the number of undo levels allowable (up to 256), to how points display (large or small, with or without BCP lines, and so forth), editing behavior of lines and points, and the way windows and dialog boxes operate. For a detailed description of what you can do from the Preferences dialog see Chapter 13, “Expert Advice”.

Quit

If you quit without first saving the changes you have made, Fontographer will ask if you want to save those changes. You can discard those changes by clicking the **Don’t Save** button. The **Cancel** button stops the process. If you quit with multiple, unsaved databases open, you will have the option to **Discard Changes**, **Cancel**, or **Review Changes**.

The File menu

New

In Fontographer when you select **New**, you automatically open a new font window. The default values for font attributes (such as the font name, ascent, descent, leading, and encoding) will be in use, unless you decide to change them from **Font Info** in the **Element** menu. (See also "Preferences" on page 457.)

Open

Open shows you a standard file dialog box that allows you to open a Fontographer (.fog) or Fontlab Studio (.vfb) database file, a PostScript font, a TrueType font, or an OpenType font file. The dialog box also allows you to display files according to their formats.

Open Recent

The **Open Recent** submenu contains the list of the most recently opened fonts. Next time you want one of them, just select the font file in the menu and Fontographer will open it.

Close

Use this command to close the active window. If you have made changes and close the font window, Fontographer will ask you if you want to save your changes.

Save

The **Save** command saves any changes you have made since opening the font.

Save As

Save As allows you to save your file with a different name (but not your font in use).

Revert to Saved

The **Revert to Saved** command restores your font to the last saved version on the disk, throwing away any changes you may have made since you last saved it. You can revert from any window.

Generate font files

To make generating fonts a little bit easier, Fontographer includes two approaches: *Easy* and *Advanced* mode. The Easy mode includes everything that is usually required. And ordinarily you won't want to bother with the Advanced mode. You just choose the platform and the type of font to output, the bitmap sizes, and where you want the generated files to be placed. Fontographer no longer creates AFMs unless you specifically ask for them in the Advanced mode. The Advanced mode includes several other choices, including bitmap sizes, glyph naming options, and so forth.

Import

Importing bitmap fonts, EPS, EPAR information or metrics into a font file is a simple matter using the **Import** item in the **File** menu. Just choose one of these three options from the submenu and import from another file. You can select whether or not you want to import bitmaps into the internal bitmap list or just into the Template layer, or you can do both. You can also import EPS outlines into a character. When you import metrics you can choose to import only kerning information or spacing information, both types, or ascent/descent. When importing ascent/descent, Fontographer looks for a .bmap or a TrueType font. When selecting the .bmap corresponding to the original PostScript font, we read the original ascender/descender information found in the FOND resource. You can also choose the kinds of files you want to appear in the dialog box display: suitcases, AFM files, PFM files, etc. This allows you to easily recognize your files, and select the right one for importing.

On the Macintosh, you can also import a specific character(s) from an installed TrueType font. This comes in handy when you wish to use specific characters from existing typefaces without disrupting changes you have already made to your open font.

Export

When you want to export metrics information, bitmap BDF font or an EPS graphic, just select **Export** from the **File** menu. You can choose **Metrics, BDF, EPAR** or **EPS** from the submenu. When you choose **EPS**, a dialog box appears allowing you to choose whether to export all characters or just a selected group, or the sample text, and the point size to use. You choose the file to export "to" from the dialog box. Exporting BDF and metrics works in pretty much the same way, but for metrics you choose to export either kerning or spacing, or both types of information, and you select the file format to use.

If you wish to convert your .fog files created in Fontographer 3.5-5 to FontLab Studio-compatible .vfb files, you can use the **Fontlab Font File** command.

Print

In the Print Sample dialog box, you can choose the sample type you want to print. Choose Sample text, Sample file, PostScript file, Key map, Kerning pairs, or Sample glyphs.

Printing Sample text offers three choices. All glyphs will print all the glyphs in the font you are modifying. Selected glyphs will print only those glyphs you select from the font window. Sample text will allow you to print the text you type in the print sample text box. Choose Sample file to print a text file with the new font style. PostScript file will print a custom PostScript file; some sample PostScript files are available in the PSFiles folder. You can use these files as they are or modify them with a text editor to define your own sample.

Key map prints all the glyphs in the font including offsets, widths and key codes, and offers you the choice of Selected or All glyphs, and format types.

Kerning pairs prints the kerning pairs in the current font or in a monospaced System font.

Sample glyphs prints four kinds of print samples: a single full-page sized glyph, assorted sizes of the same glyph, a sample showing the points of a glyph, or one that shows points and x/y coordinates.

The Edit menu

Undo

Fontographer has 256 possible levels of undo. The default setting is 100. This means that you can undo the last 100 things you did in Fontographer. The operation that can be undone will appear in the menu after the undo if you are using the outline or bitmap window. **Undo** is not available in the font window.

Redo

Redo allows you to undo an **Undo**. There are 256 possible levels of redo.

Cut

Cut deletes the selected object(s) and puts them in the Clipboard.

Copy

Copy makes a copy of the selected object(s) and puts them in the Clipboard.

Paste

Paste takes the object(s) in the Clipboard and puts them into the selected window or character slot.

Clear

Clear deletes the selected object(s) without doing anything to the Clipboard.

Copy Widths

Copy Widths will copy the width of a selected glyph to the Clipboard.

Copy Component

Copy Component copies a reference of a selected glyph into the Clipboard, which can then be pasted to allow layered objects. You must use **Copy Component** instead of **Copy** if you want to create a composite character.

Decompose Component

Decompose Component replaces a referenced glyph with explicit paths.

Select All

Select All selects everything in the currently active window and layer. From the font window you can choose **Select All** to select all the glyph slots that are filled; **Option-Select All** selects all slots, even if there are no paths in them.

Duplicate

Duplicate makes a copy of the selected object(s) and leaves it on the screen slightly displaced from the original(s).

Power duplicating occurs when you duplicate the same object several times (in the outline window).

Clone

Clone duplicates a point or path and places it directly over the original (in the outline window).

The View menu

Preview

In the outline window, the Preview mode displays the character or image filled, and stroked (if it's a stroked character), as it will appear when printed. You can edit the character in Preview mode.

Show points

If this item is checked, Fontographer will show the points on the path(s) in the outline window. When this item is deselected, Fontographer merely draws the character's outline(s) and does not show the points.

Magnification

You can magnify an image in Fontographer by selecting any one of the **Magnification** submenu options. **Fit in Window** will fit the character in the window. The other choices magnify at various levels: 6.25%, 12.5%, 25%, 50%, 100%, 200% etc.

Next Glyph

This item switches the currently selected glyph to the next glyph of the font.

Next Kerning Pair

If kerning pairs have been created for a font, you can choose **Next Kerning Pair** in the metrics window to change the currently selected kerning pair to the next one in the sequence of kerned pairs.

Next Point

Next Point changes the selected point to the next one in the path's sequence of points.

Next Point Size

If more than one size of bitmaps has been created for a font, you can choose this item in the bitmap window to switch the size of the bitmap character to the next larger point size.

Previous Glyph

This item switches the currently selected glyph to the previous glyph in the font.

Previous Kerning Pair

If kerning pairs have been created for a font, choose **Previous Kerning Pair** in the metrics window to change the currently selected pair to the previous one in the sequence of kerned pairs.

Previous Point

In the outline window, choose **Previous Point** to change the selected point to the previous point in the path's sequence of points.

Previous Point Size

This item switches the size of the bitmap character in the bitmap window to the next smaller point size.

Snap to Points

Choose **Snap to Points** to turn this item on or off. **Snap to Points** will make the selection snap to the nearest point within a certain number of pixels from the pointer. Or you can choose to align the selection with all the points in the Preferences dialog box.

Snap to Guides

Choose **Snap to Guides** when you want the selected point to snap to the nearest guideline within a certain distance from the pointer.

Snap to Grid

Choose **Snap to Grid** when you want a selected point to snap to the nearest intersection of invisible grid lines. You can change the grid size in Preferences.

The Element menu

Transform

The **Transform** menu item displays a dialog box that lets you choose various transformations in the outline window. You can choose to center transformations around the origin, the basepoint, the center of selection, or the last mouse click, and you can make up to four transformations simultaneously.

Flip

The **Flip** item allows you to flip any glyph(s) or selected parts of a glyph. The flip can be either horizontal or vertical and will occur relative to the basepoint, or you can flip around the center of selection or the glyph origin if you have selected objects in the outline window.

Move

The **Move** item allows you to move selected objects a specified distance horizontally, vertically, or both.

Rotate

The **Rotate** item allows you to rotate any glyph(s) or selected parts of a glyph. You can choose for the rotation to occur relative to the basepoint, the glyph origin, or around the center of selection or last mouse click if you have selected objects in the outline window.

Scale

The **Scale** item allows you to resize selected glyphs. If 100% scaling is selected, Fontographer does not change the size of the glyph(s). If you wish to make selected glyphs smaller, use numbers below 100. If you wish to make them larger, use numbers over 100.

You can scale the horizontal and vertical dimensions separately if you wish. A 200% horizontal scaling coupled with 100% vertical scaling will make selected glyphs twice as wide, while leaving them just as tall as they were before the scaling operation.

You can choose whether Fontographer will scale from the basepoint, the glyph origin, or around the center of selection or last mouse click if you have selected objects in the outline window.

Scale Uniformly

Uniform scaling will resize selected glyphs uniformly, with 100% representing the current size. Scale relative to the basepoint, the glyph origin, or the center of selection.

Skew

The **Skew** item allows you to skew any glyph(s) or selected parts of a glyph. The skewing will occur relative to the basepoint, the glyph origin, or you can skew around the center of selection or last mouse click if you have selected objects in the outline window. Positive degrees of horizontal skew slant the glyph to the left; negative degrees slant right.

Perspective

The **Perspective** item allows you to add perspective (slant) any glyph(s) or selected parts of a glyph. Perspective transformation does nothing by itself but works in conjunction with the **3D Rotate** to apply three-dimensional rotations to two-dimensional objects.

3D Rotate

The **3D Rotate** item allows you to rotate any glyph(s) or selected parts of a glyph around any of three axis in 3D space.

Arrange

Arrange allows you four choices when you are working with referenced or composite glyphs. It also can be used when you're reordering paths in font blending, or when you're creating a multiple master font.

- **Bring to Front**

  This item brings a selected object all the way to the front layer.

- **Send to Back**

  This item sends a selected object all the way to the back layer.

- **Bring Forward**

  This item brings selected objects forward by one layer.

- **Send Backward**

  This item sends a selected object back one layer.

Font Info

The Font Info dialog box has an Easy and Advanced modes. In the Advanced mode it consists of five sections: names, dimensions, encoding, credits and licensing.

Names

In the top of the dialog box you will see the family name of the current font and its style name. Next you will see the design parameters – the group of controls to build the style name. You should always fill the family name while the style name can be auto-generated from design parameters or filled manually.

The two lower sections (Styling group names and Unique font names) can be either auto-generated or filled manually (refer to Chapter 2, "Modifying Your Fonts" and to Chapter 7, "Font Info").

Dimensions

All the numbers in this dialog box are in font units (UPM). The ascender and descender default to a sum of 1000 units for a PostScript font, or 2048 for a TrueType font (refer to Chapter 8, "Generating and Exporting Fonts" and to Chapter 7, "Font Info").

Encoding

Here you set how glyphs are ordered in the font window and in the exported font. MacOS Roman encoding is the most relevant for Type 1 Roman fonts in Mac OS.

If you choose Adobe® standard character encoding, Fontographer displays and stores a font with Adobe encoding. Do not check this option if your fonts have all the upper 128 characters defined.

Notes: If you change the encoding and regenerate a Macintosh Type 1 font from an existing font file, you must regenerate, then remove and reinstall the bitmaps (suitcase file). If you do not do this, any characters of the font above 127 will be incorrect.

OpenType Layout options allow you to set your preferences in generating OpenType layout features for the particular font. You can override this later in the Format Options dialog box when generating the font (refer to Chapter 8, "Generating and Exporting Fonts" and to Chapter 7, "Font Info").

Credits

On the Credits page you can enter information about the creators of the font, font version, creation date. If you have created a new font you should enter your copyright data here. If you have edited an existing font that was not your creation you must not remove the information contained on this page, or you may violate copyright laws (refer to Chapter 7, "Font Info" for details).

Licensing

The Licensing page contains the End-User License Agreement (EULA) information and the embedding information (Chapter 7, "Font Info" for details).

Selection Info

When you choose **Selection Info**, a dialog box that relates to the type of selection you have made will display. In the font window, this choice will display glyph information, and in the outline window either glyph or point information will appear, depending on your selection. In the Hints layer, you get the Hint Info dialog box, and in the metrics and the bitmap windows the **Selection Info** menu item displays glyph information.

Glyph Information

Use the Glyph Information dialog box to change the fill and stroke characteristics of individual glyphs for Type 3 fonts. Normally a glyph will be either filled or stroked, but you can use this dialog box to create glyphs that are both filled and stroked. The **Tint**, **Weight**, **Cap**, and **Join** options allow you to set the characteristics of a stroked font.

The **Glyph Name** textbox allows you to change the name of the glyph – but be warned that doing so will also change the type of encoding your font uses if the current type does not include a slot with the name you select.

The glyph **Unicode Codepoint** textbox allows you to change the Unicode of the glyph. Changing Unicode codepoint usually means you should change the glyph name too.

The glyph **Unicode Name** shows the Unicode name of the glyph as it appears e.g. in the Character Palette. It is changed automatically when you change the Unicode codepoint.

- **Fill**

  If the **Fill outline** box is checked, you can change the percentage of black with the **Tint** option: 100% is black; 0% is white.

  The **Normal fill** will fill between paths that are alternately clockwise and counterclockwise and can be seen in Preview.

  The **Even/odd fill** will fill between every even/odd pair of paths, starting with the outermost path.

Note: The **Remove Overlap** option is sensitive to fill type.

- **Stroke**

  If the **Stroke outline** box is checked, you can change the percentage of black that fills the stroke with the **Tint** option: 100% is black; 0% is white.

- **Weight**

  The **Weight** option allows you to change the thickness of this individual glyph's stroke from the default set in the Font Info dialog box.

- **Cap and Join**

  In this dialog box, the **Cap** and **Join** options are for stroked fonts. The **Cap** options determine the shape of the endpoints of a stroked path. The **Join** options determine the shape of joins along a path.

Point Information

The Point Information dialog box appears when you select a point in the outline window and choose **Selection Info** from the **Element** menu. Point Information lets you change the location of a point horizontally and vertically, or designate it the first point in the path. You can also alter the position of its BCPs, return the BCPs to the point, or move to the next or previous point in the path.

Hint Information

The Hint Information dialog box appears when you select a hint in the outline window and choose **Selection Info** from the **Element** menu. Hint Information lets you navigate through your hints by hint type and set the starting and stopping point of a hint. You can also add, remove, and flip hints, and apply them to Type 1, TrueType, or bitmap fonts as applicable.

Bitmap Info

When you choose **Bitmap Info**, a Bitmap Information dialog box will display.

The Bitmap Information dialog box allows you to specify the point sizes you want to create. The bitmaps are stored in the Fontographer file. When you are ready to create an installable bitmap font, use the **Generate Font Files** command in the **File** menu.

Auto Trace

Auto Trace will auto trace any artwork or scanned image that you have pasted in the Template layer or have brought in as a reference using the **Import Bitmap** item in the **File** menu. You may use the Easy or Advanced mode to trace images.

In the Easy mode, the tight end of the slider makes the trace follow every possible contour on the image. The default normal or middle range of the slider is a good compromise between the tight and loose options. The loose end of the slider ignores little jagged edges and attempts to fit only the largest features of the image.

Advanced mode specifies the kind of fit curves will take: normal, loose, tight, or custom fit.

In the lower portion of the dialog box, you can click checkboxes that tell your computer to treat nearly flat paths as straight lines or to find extreme points.

Change Weight

Change Weight allows you to change the glyph's thickness. You can choose to correct path direction before changing the weight – to maintain proper filling. You can also limit the change in size to either vertical or horizontal size.

Clean Up Paths

Clean Up Paths improves the quality of your outlines. It removes unnecessary points and adds points where they are needed. Simply create the paths any way you'd like, and Fontographer will automatically clean them up for you.

Expand Stroke

The **Expand Stroke** item is used to expand stroked glyphs (such as old versions of Courier) into contoured (outline) or filled glyphs. You can choose cap and join types, and if you are using the calligraphy option, the width and angle of the pen.

Recalc Bitmaps

When you choose this item, selected bitmaps and point sizes will be recalculated. You can recalculate all the glyphs in your font, selected glyphs, or just the glyphs that are changed. You may also choose to preserve the line spacing, or the glyph shapes. Recalculating bitmaps is useful when you change a glyph's outlines after having created bitmaps.

Remove Overlap

The **Remove Overlap** item allows you to merge and remove overlapping areas. You'll get the best results if your path has a normal fill. If the path has an even/odd fill, you will be able to merge the paths, but you'll need to remove the overlapping segment(s) manually. Removing the overlap(s) may result in faster printing fonts and help the hinting process and eliminate some TrueType printing problems.

Correct Path Direction

Correct Path Direction examines all the selected glyphs or paths and, if necessary, automatically reorders their path directions. Outer paths are set to clockwise, the inner paths to counterclockwise.

Clockwise

This item will be selected if the selected path was drawn in a clockwise direction. You can change the direction of a path to clockwise by selecting this item.

Counterclockwise

This item will be selected if the selected path was drawn in a counterclockwise direction. You can change the direction of a path to counterclockwise by selecting this item.

Blend Fonts

Blend Fonts takes two fonts you have selected and develops a third font that is the offspring of this merger.

Multiple Master

This dialog box controls the creation of multiple master typefaces. You can choose the “master fonts”, as well as coordinate space and many other multiple master variables. To actually generate a multiple master typeface, use the Generate Font Files dialog box.

The Points Menu

Align Points

Align Points will align selected points along a horizontal or vertical axis. Fontographer decides which axis they are closer to forming.

Align Points to Grid

Align Points to Grid will align the selected points with the nearest intersection of grid lines. If no points are selected, then all points are aligned with the grid lines. Hold down the **Option** key during the selection to align BCPs as well.

Merge Points

Selecting this item will remove the selected point without breaking the path it is on. If you need to open a path, use the **Delete/Backspace** key or the knife tool.

Retract BCPs

This item will retract the extended BCP lines of selected points.

Split Points

Split Points will divide a selected point into two points. The path can be opened by dragging one point from on top of the other.

Auto Curvature

This item automatically adjusts the slope of the BCPs of a selected point to maintain a regular curve as the point is moved.

Curve Point

This item will be selected if the selected point is a curve point. Selecting this item while other kinds of points are selected will change those points to curve points.

Corner Point

This item will be selected if the selected point is a corner point. Selecting this item while points of other kinds are selected will change those points to corner points.

Tangent Point

This item will be selected if the selected point is a tangent point. Selecting this item while points of other kinds are selected will change those points to tangent points.

Set Basepoint

If you have not selected any points, choosing this item causes a dialog box to appear that will allow you to set the horizontal and vertical location of the basepoint. If you've selected a single point, Fontographer will move the basepoint to that point. If you have selected more than one point, Fontographer will set the basepoint to the center of the selection's bounding box.

Reset Basepoint

Click **Reset Basepoint** to return the basepoint to its original location at the intersection of the origin and the baseline.

The Metrics menu

Auto Space

Auto Space automatically spaces a font, setting widths for each glyph. You can select the Easy or the Advanced mode. With the Easy mode, you just select the spacing between glyphs that you prefer. Check the spacing in the metrics window by typing text in to the textbox. See "Advanced auto spacing" in Chapter 6, "Metrics – Spacing and Kerning", for a detailed explanation of the Advanced mode.

Auto Kern

Auto Kern automatically kerns a font. In the Easy mode you can choose how many kerning pairs you want, how close together you want the pairs kerned, and whether or not you want Fontographer to change existing pairs. (Select this item if you want to manually set some pairs but want Fontographer to set the others.) The Advanced mode allows you to choose which glyphs to kern, how many pairs to make, and a lot more. You can also specify the techniques to use, as well make choices about special cases. See "Advanced auto kerning" in Chapter 6, "Metrics – Spacing and Kerning", for more details.

Kerning Assistance

Kerning Assistance provides a table where you can name kerning pairs by typing glyphs in to columns. See "Kerning assistance" in Chapter 6, "Metrics – Spacing and Kerning", for a detailed explanation.

Metrics Assistance

Metrics Assistance allows you to set up a table for kerning that uses certain glyphs as bases or prototypes for the rest of the glyphs in any class you set up. See "Metrics assistance" in Chapter 6, "Metrics – Spacing and Kerning", for more information and detail.

Set Metrics

Use **Set Metrics** to set width, as well as left and right sidebearings. You can also apply values to specific sets of glyphs. Choose the glyphs you want to apply the metrics settings to, from the categories under Which glyph. These categories include: A–Z, a–z, 0–9, Punctuation, Accent glyphs, Symbols, or Selected glyphs. Add any other glyphs to apply the setting to in the provided textbox.

The next area in the dialog box tells you what to do. You can choose to set the sidebearings or width according to a particular glyph's measurement, or you can pick an em-unit value to type in to set the width, and/or sidebearings for the selected glyphs. You may also add an amount to that specification – either in em units, as a percentage of the distance represented by the glyph's metrical measurements, or the value you chose.

Set Width

This dialog box allows you to set the width of any selected glyph(s). If you are creating a monospaced font, select all the glyphs and set their width to a chosen number of units. Fontographer will set them all at once.

If the spacing of your glyphs is too tight or too loose, you can use the Change width by em units option to change the width of selected glyphs by a specified number of em square units.

If you need to increase or decrease the width of selected glyphs by a specified percentage, you can do so with the **Change width by % value** option. Enter a percentage change value in the box.

Equalize Sidebearings

If you want to make both left and right sidebearings for selected glyphs the same size, choose **Equalize sidebearings** from the **Metrics** menu. This improves the looks of typed columns of numbers. When points are selected, **Equalize Sidebearings** will center them between the origin and the width lines.

If you press the **Option** key and choose **Equalize Sidebearings**, the right sidebearing will become the same width as the left.

Clear Kerning Pairs

This option allows you to remove all kerning pairs from your font including vertical kerning in CJK fonts. You may wish to do this before choosing **Import Metrics** for a font. As a precautionary measure, a warning message allows you to cancel this option before Fontographer deletes all the kerning pairs.

The Hints menu

Add Vertical Hint

To make a new vertical stem hint, select two of the points that define the stem and choose this item from the menu.

Add Horizontal Hint

To make a new horizontal stem hint, select two points that define the stem and choose this item from the menu.

Common Stems

You can change the hint parameters for selected characters using the Common Stems dialog box. When you change hints for certain characters, you reset the default hints for just those characters. Other characters in the font retain the prior default hint settings.

See Chapter 13, “Expert Advice”, for more details on hints.

Alignment Zones

The vertical alignment zones correspond with the I-beam indicators appearing along the left-hand side of the outline window when the Hints layer is visible. When you click a zone indicator, the selected zone will appear with horizontal lines defining its parameters. You can alter a zone’s size by dragging on the upper or lower part of its indicator. Selected zone indicators appear hollow on each end.

Autohint

Use **Autohint** to recalculate hints automatically when you edit glyph outlines. Once an outline has been edited, new hints will be calculated to match the new outline. Autohinting is turned on if you see a checkmark beside it in the **Hints** menu. To turn it off, select it from the menu and the checkmark will disappear.

The Window menu

Open Outline Window

Select **Open Outline Window** to view or edit outlines.

Open Bitmap Window

Open Bitmap Window will open an existing bitmap. If none exist, the dialog box will ask if you want to create bitmaps.

Open Metrics Window

Open Metrics Window will display the metrics window for a selected glyph.

Show Layers Palette

This item displays or hides the movable Layers palette when you select it.

Show Tool Palette

Click this item to show or hide the movable tool palette.

The Window choices

In the lower section of the **Window** menu, there is a list of all open windows. Selecting a window in the list will bring that window to the front.

Special keys

Keyboard alternatives

To avoid having to constantly move the pointer back to the tool palette, you can use the number keys on your keyboard to switch between tools. When the lock icon is in the locked position, pressing the following keys will access the corresponding tool:

Outline tools	Bitmap tools
` Pointer	
1 Rectangle	1 Straight line
2 Multigon	2 Hand
3 Oval	3 Pencil
4 Straight line	4 Eraser
5 Calligraphy pen	5 Marquee
6 Pen	6 Move
7 Knife	7 Measuring tool
8 Curve point	8 Magnifier
9 Corner point	
0 Tangent point	

In the font window, the **Left** and **Right arrow** keys will move you to the previous glyph or to the next glyph. The **Up** and **Down arrows** select the glyph in the row above or below the present glyph.

In the outline window, the **arrow** keys will move selected points by ten em units in the direction of the arrows. If the default setting in Preferences is changed, then this number will reflect that change.

Option-arrow moves selected points by one em unit in the direction of the arrow. When the default setting in Preferences changes, this number will be 1/10th the distance of the new setting.

Shift-arrows move the selected points in the direction of the arrows by 10 times the default setting.

In the metrics window, use the **Left** and **Right arrow** keys to move to the next or previous glyph in the screen display when a glyph is selected.

The Full List of Keyboard Shortcuts

Mac shortcut	Description	Used In	Menu	Windows shortcut
⌘1	Change window zoom	Outline, Metrics, Bitmap windows	View > Magnification	^1
1	Selecting tool	Outline and Bitmap windows		1
⌘2	Change window zoom	Outline, Metrics, Bitmap windows	View > Magnification	^2
2	Selecting tool	Outline and Bitmap windows		2
⌘3	Change window zoom	Outline, Metrics, Bitmap windows	View > Magnification	^3
3	Selecting tool	Outline and Bitmap windows		3
⌘4	Change window zoom	Outline, Metrics, Bitmap windows	View > Magnification	^4
4	Selecting tool	Outline and Bitmap windows		4
⌘5	Change window zoom	Outline, Metrics, Bitmap windows	View > Magnification	^5
5	Selecting tool	Outline and Bitmap windows		5
⌘6	Change window zoom	Outline, Metrics, Bitmap windows	View > Magnification	^6
6	Selecting tool	Outline and Bitmap windows		6
⌘7	Change window zoom	Outline, Metrics, Bitmap windows	View > Magnification	^7
7	Selecting tool	Outline and Bitmap windows		7
⌘8	Curve Point	Outline window	Point	^8
8	Selecting tool	Outline and Bitmap windows		8
⌘9	Corner Point	Outline window	Point	^9
9	Selecting tool	Outline window		9
⌘0	Tangent Point	Outline window	Point	^0
0	Selecting tool	Outline window		0
⌘- (hyphen)	Zoom Out	Outline, Bitmap, Metrics windows		^- (hyphen)
⌘+ (grey plus sign)	Zoom In	Outline, Bitmap, Metrics windows		^+ (grey plus sign)
⌘⇧+ (plus sign)	Zoom In	Outline, Bitmap, Metrics windows		^⇧+ (plus sign)
⌘= (equal sign)	Equalize Sidebearings	Font, Outline, Metrics windows	Metrics	^= (equal sign)
⌘` (grave accent)	Activate the next open window in the application	Application		^Tab
` (grave accent)	Selection tool	Outline window		` (grave accent)
⌘⇧` (grave accent)	Activate the previous open window in the application	Application		^⇧Tab
⌘, (comma)	Preferences	Application	Application	^⇧P

⌘ [(left bracket)	Previous Glyph	Font, Outline, Bitmap, Metrics windows	View	^[(left bracket)
⌘] (right bracket)	Next Glyph	Font, Outline, Bitmap, Metrics windows	View	^] (right bracket)
⌘\ (back slash)	Transform	Font and Outline windows	Element	^\ (back slash)
⌘A	Select All	Application	Edit	^A
⌘^A	Select Search Results	Font window	Edit	^⌥S
⌘⇧A	Autohint	Font, Outline, Bitmap, Metrics windows	Hints	^⇧A
⌥Arrows	Move 10 times smaller distance	Moving point(s) and path(s)		⌥Arrows
⇧Arrows	Move 10 times larger distance	Moving point(s) and path(s)		⇧Arrows
⌘B	Set Basepoint	Outline window	Point	^B
⌘⇧B	Send to Back	Outline window	Element > Arrange	^⇧B
⌘⌥B	Reset Basepoint	Outline window	Point	^⌥B
⌘C	Copy	Application	Edit	^C
⌘⇧C	Clean Up Paths	Font, Outline windows	Element	^⇧C
⌘⌥ C	Copy	Application	Edit	n/a
⌘D	Duplicate	Outline window	Edit	^D
⌘Down arrow	Next Point	Outline window	View	^/(slash)
⌘⌥Down arrow	Next Point Size	Bitmap window	View	^Down arrow
⌘E	Align Points	Outline window	Point	⌘E
⌘⇧E	Expand Stroke	Font, Outline windows	Element	⌘⇧E
⌘⌥E	Align Points to Grid	Outline window	Point	⌘⌥E
↩ (Return)	Font window - open Outline window for the selected glyph; Outline and Bitmap windows - Lock/Unlock			↩ (Return)
⌘F	Move focus to Search field	Font window		^F
⌘⇧F	Bring to Front	Outline window	Element > Arrange	^⇧F
⌘F	Load Parameters from File	Multiple Master dialog		n/a
⌘⌥F	opens Font Info	Font window	Element	^⌥F
⌘G	Copy Component	Font, Outline windows	Edit	^G
⌘⌥G	Generate font	Font window	File	^⌥G
⌘⇧G	Snap to Guides	Outline window	View	^⇧G
G	Activate Guides layer	Outline window		G
⌥G	Change visibility of Guides layer	Outline, Bitmap, Metrics windows		n/a
⌘H	Hide application	Application	Application	n/a
⌘⌥H	Hide Others	Application	Application	n/a
⌘⇧H	Add Horizontal Hint	Outline window	Hints	^⇧H
H	Activate Hints layer	Outline window		H

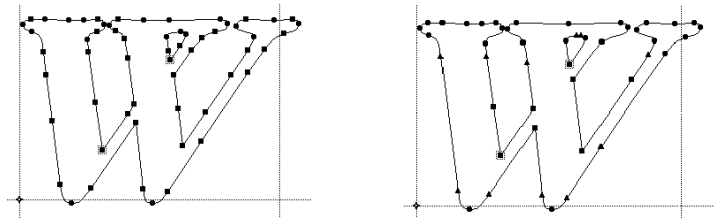
⌥H	Change visibility of Hints layer	Outline window		n/a
⌥H	Change visibility of sidebearings and kerning line	Metrics window		n/a
⌘I	Selection Info	Font, Outline windows	Element	^I
⌘⇧I	Snap to Grid	Outline window	View	^⇧I
⌘J	Open Bitmap window	Application	Window	^J
⌘⇧J	Common Stems	Application	Hints	^⇧J
⌘K	Open Metrics window	Application	Window	^K
⌘⇧K	Correct Path Direction	Font, Outline windows	Element	^⇧K
⌘L	Preview	Outline window	View	^L
⌘Left arrow	Previous Kerning Pair		View	^; (semicolon)
⌘M	Merge Points	Outline window	Point	^M
⌘⌥M	Set Width	Font, Outline, Bitmap, Metrics windows	Metrics	^⌥M
⌘⇧M	Set Metrics	Font, Outline, Bitmap, Metrics windows	Metrics	^⇧M
⇧Mouse drag	Draw square, circle, etc	Drawing objects		⇧Mouse drag
⌘N	New	Application	File	^N
⌘⇧N	Snap to Points	Outline window	View	^⇧N
⌘O	Open	Application	File	^O
⌘⇧O	Remove Overlap	Font, Outline windows	Element	^⇧O
O	Activate Outline layer	Outline window		O
⌥O	Change visibility of Outline layer	Outline, Bitmap window		n/a
⌘P	Print	Application	File	^P
⌘⇧P	Page Setup	Application	File	n/a
⌘Q	Quit application	Application	Application	⌥F4
⌘R	Retract BCPs	Outline window	Point	^R
⌘⇧R	Recalc Bitmaps	Application	Element	^⇧R
⌘Right arrow	Next Kerning Pair		View	^' (single quote)
⌘S	Save	Application	File	^S
⌘⇧S	Save As…	Application	File	^⇧S
Space	Select Hand tool temporarily	Outline, Bitmap, Metrics windows		Space
⌘Space	Zoom In	Outline, Bitmap, Metrics windows		^Space
⌘⌥Space	Zoom Out	Outline, Bitmap, Metrics windows		^⌥Space
⌘T	Fit in Window	Outline, Metrics, Bitmap windows	View > Magnification	^T
⌘⇧T	Auto Trace	Font, Outline windows	Element	^⇧T
T	Activate Template layer	Outline window		T

⌥T	Change visibility of Template layer	Outline window		n/a
TAB	Deselect everything in Outline window			TAB
⌘U	Show Points	Outline window	View	^U
⌘⇧U	Auto Curvature	Outline window	Point	^⇧U
⌘Up arrow	Previous Point	Outline window	View	^,(comma)
⌘⌥Up arrow	Previous Point Size	Bitmap window	View	^Up arrow
⌘V	Paste	Application	Edit	^V
⌘⇧V	Add Vertical Hint	Outline window	Hints	^⇧V
⌘W	Close	Application	File	^F4
⌘X	Cut	Application	Edit	^X
⌘Y	Open Outline window	Application	Window	^H
⌘⇧Y	Change Weight	Font, Outline windows	Element	^⇧Y
⌘Z	Undo operation	Application	Edit	^Z
⌘⇧Z	Redo operation	Application	Edit	^Y
⌥mouse wheel	Zoom	Outline, Metrics, Bitmap windows		⌥mouse wheel
⇧arrows, mouse button	Extent selection in the font window	Font window		⇧arrows, mouse button
⇧Menu: Edit->Paste	Shift-Option-Paste (scale pasted image in Outline window)	Outline window	Edit	n/a
⌘. (period)	Cancel operation	any dialog with Cancel and OK buttons		Esc
⌥Arrows	In spreadsheet area of Metrics window increment/decrement value by 1	Metrics window spreadsheet		^⌥Arrows
⇧⌥Arrows	In spreadsheet area of Metrics window increment/decrement value by 10	Metrics window spreadsheet		^⇧⌥Arrows
⌥mouse button	Option-click hint selects hint and its points	Outline window		⌥mouse button
⌥Edit->Select All	select all slots in Font window	Font window	Edit	n/a

Appendix A. Tips

We'd like to share some tricks we've learned while using Fontographer. Some of these are just reminders; others are shortcuts to solving typical problems. If you have additions, please send them to us.

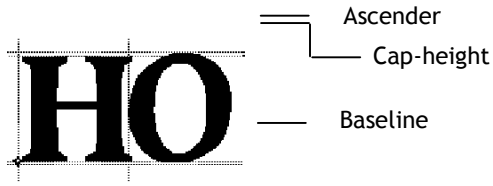
- Minimize the total number of points when drawing characters. Smaller characters draw faster, take up less space in the printer and on your disk, and usually generate smoother characters. Most standard Roman characters can be described in fewer than sixty points. **Clean Up Paths** makes sure your characters include the optimal number of points.



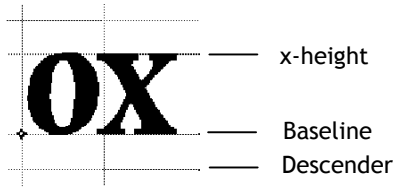
- Images that are imported or pasted into the Template layer are automatically sized to fit the em square.

  To maintain the original size of the image when pasting into the layer, hold down the **Option** key when you paste or import the image. The size of the image you see depends on the magnification level in the window.

- Open a line by using **Option**-knife to click a point and remove the section of line between the adjacent points on either side of it. You might do this to open a sans serif character so a serif foot can be glued onto it.
- To pull a BCP out of a point, select the point, press the **Option** key, and drag until the BCP handle appears.
- Move one or both BCP handles directly on top of their curve points to get sharp corners with slope control.
- Fontographer has vertical zones that are important for uniformity in the font. If letters fall into these zones, the font will have uniform height. If not, your characters could unexpectedly vary in height at small sizes. Some guidelines to keep in mind:



Capital letters' top coordinates should be between the top of the "H" and the "O."

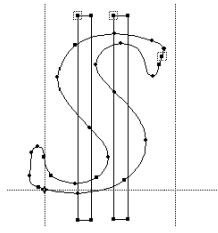


Lowercase letters' top coordinates should be between the top of the "x" and the "o." Lower coordinates should be between the baseline and the bottom of the "O."

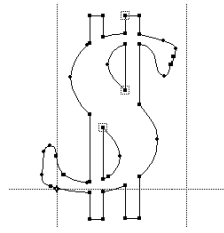
All characters' upper and lower coordinates will be automatically aligned if they fall into these zones.

Technically, the "O" and the "o" (and other curvy letters) are said to overshoot the "H" and the "x" by about 4%, which makes them appear to be the same size as straight letters. We take advantage of Type 1 fonts' ability to align upper and lower coordinates vertically within these overshoot zones, but only if the font is drawn according to the above rules (or if you adjust the vertical alignment zones from the Vertical Alignment Zones dialog box accessed through the **Hints** menu).

- Don't create outlines with overlapping paths.



For example, if you want to make a dollar sign out of an "S," you might add two rectangles in the center of the character and get this. That's fine for drawing programs, but not for fonts.



*Choose **Remove Overlap** in the **Elements** menu to remove the overlap in the paths.*

Answers to commonly asked questions

The Fontographer technical support group has compiled answers to our customer's most commonly asked questions. If you have any other questions, contact the Fontlab Ltd. technical support at http://www.fontlab.com/support/.

I've made a PostScript font using Fontographer and have printed it to my printer. I then made some changes to the font and tried printing again, but the changes didn't show up on the second printing. What is going on here and what can I do about it?

This problem should not occur with modern printers but usually happens with old PostScript Level 1 printers. Built into all Adobe PostScript printers is some software called a font cache. This font cache holds the imaged bitmaps of each of the characters that the printer has recently printed, so that it doesn't have to waste time re-interpreting the PostScript description of the character, but can instead pull the already imaged bitmap out of the font cache and place it on the page. Fontographer Type 1 fonts normally have an Outline ID that can be changed in the Generate fonts dialog box. This ID is the numeric handle by which the PostScript interpreter references the font cache images. If you temporarily change the number of the outline ID to zero (0), then the interpreter will not cache any of the characters of your font and you won't have this problem. Be sure to restore the original outline ID (or any random number between 4,000,000 and 4,999,999) when you're finished with the font and are satisfied that you won't be making any more changes. Another way to get around this is to flush the font cache after each iteration of your font by either turning the printer off for a few seconds, then turning it back on (for printers without hard disks) or by using Adobe's Font Downloader 4.x (or later), which can get rid of the font cache on a printer that has a hard disk. The LaserWriter IIf and IIg do not store the font cache on the hard disk, so it is merely necessary to restart these printers.

What is the difference between Type 1 and Type 3 PostScript fonts?

Type 1 fonts are smaller, faster to print, better looking, and work with the Adobe font driver (ATM or built into Windows or Mac OS X). However, Type 1 characters must be entirely black. Type 3, on the other hand, can have grayscale fills and strokes and other special effects. Type 3 fonts are bigger, slower, look worse in very small point sizes and at low resolutions (up to 600 dpi) and don't work on Windows or Mac OS X. About 99% of the time, you will want to create Type 1 fonts.

I need to exactly duplicate a font, with just a few changes to a few characters in that font. When I use Fontographer to open the font and get the outlines of the characters of the font I want to duplicate, I find that the resulting font doesn't have any of its characters kerned like the original. What should I do?

Choose Import from the File menu and then select its submenu Metrics to import the kerning pairs from the original font's AFM or PFM file or bitmap file.

Sometimes when I generate PostScript files with Fontographer, then look at those files in a Macintosh Finder window by Name, I see that the document type for those files contains the name of one of my other PostScript fonts. Why does this happen and what can I do about it?

Weird and wonderful are the ways in which the Macintosh Finder gets the information it shows you in the by Name view. We have found that if you wish to avoid this problem, always generate your PostScript files into a closed folder. If you have a lot of files that already have this problem and you'd like to fix them, you can use ResEdit or DiskTop to set the Bundle bit on each file, or you can use the public domain BundAid program to set them all at once. After having set the bundle bits, be sure to rebuild your desktop file by restarting your computer and then holding down the Option and Command keys until you see a dialog box that asks you if you want to rebuild your Desktop file. Answer Yes and your desktop file will be rebuilt and you won't see those pesky names any more.

My logo font has a fairly complex drawing in it that prints just fine to my LaserWriter at small point sizes, but won't print large sizes at all to my Linotronic (or any other image setter). Why won't it print and what can I do about it?

There is a limitation in Adobe PostScript Level 1 that limits the number of turn points in a PostScript character. Turn points are required when PostScript images any curved line because the PostScript imaging system really can't do curved lines at all; it just fakes them with lots of very short straight lines all lined up at angles to one another. Each time there's a new little straight line, you have a turn point. PostScript level 1 font characters can't have more than 1500 of these per character, and when you image a character on a 300-dpi printer like a LaserWriter, not as many turn points are generated because fewer are needed to define a curve at this low resolution. Thus, the LaserWriter will print the character because its low resolution avoids exceeding the 1500 turn point limit, while the image setter with its higher resolution exceeds the limit. Similarly, the number of turns required to image the curve at 12 points is far fewer than the number required at 120 points. The solution to this problem is either to make a Type 3 font with internal composites (as described in Chapter 1, "Basics") or to make a Type 1 font with the character broken out into parts in various characters so that the first few parts have zero width and the last part has the actual width of the entire logo. To get the whole logo, simply type the character strings, each of which will pile on top of the last one, until the final character in the series finally moves the cursor to the right to give the character its true width.

My .ttf font shows open rectangles in the character slots instead of the characters. What went wrong and how can I fix it?

If this is happening with signatures or symbols (logos, icons, and PICTs) your character is probably too complex for the TrueType rasterizer. You can get around this by either simplifying the object, or splitting it into multiple keystrokes (for example, AB instead of A) to access the image. For a signature, place John in the J slot and Smith in S. If the last name is too long, such as Supercalafragilisticexpialadocious, you'll need to split that too. This could take a bit of experimenting until you get it working. If ALL of the characters appear as rectangles in a non-pictorial font, it probably means the character mapping table got corrupted. To fix this, go back to the Fontographer database and make a note of the attributes (that is, family name, full name, ascent, descent, and so on) and open a new font and give it those same attributes. Then copy your characters into the new database, and save and regenerate the .ttf font. Uninstall the old font; install the new one, and you're ready to go. Check the Fontlab Ltd. website for a TechNote on this.

I've imported some characters into Fontographer from FreeHand (or Illustrator) by using Option-copy and the fonts seem to print OK, but I can't Convert to paths in Illustrator 3.x. Why?

Some further rules of Type 1 fonts state that no overlapping paths are allowed in a Type 1 character. If your imported characters were drawn in such a way as to overlap some of the paths, you will need to redraw those portions of the character that overlap. Another reason for this problem can be that a path in a character was accidentally left open (unclosed). Make sure that all outlines for a Type 1 character are closed paths that don't overlap one another. Another possibility is that somewhere in the font is a character where one point is exactly on top of another point. This will also cause problems for ATM and Illustrator.

I've been working on a font in Fontographer and suddenly, when I try to open my font, Fontographer tells me it can't open the file and displays an Error #-54. What's going on here and what can I do about it?

You may be using FontPorter from Adobe (it came free with ATM 2.0) and have dragged your bitmap file (the one with the .bmap extension on it) into the System Folder rather than installing it with Font/DA Mover. FontPorter still has a hold of the bitmap font and Fontographer is trying to get at it but can't, because FontPorter already has first dibs. Error #-54 is a permissions error telling you that Fontographer doesn't have permission to open the file because FontPorter already has it opened. To fix this problem, drag the .bmap file out of the System Folder and back into the folder where you were working on your font, then restart your computer. This problem could also potentially occur if you are using Suitcase or Master Juggler to attach the .bmap file to your system. Detach the file and Fontographer will work normally.

All I did was load an existing font, generate it as a Windows TrueType, and install it. It displays terribly, compared to the way it originally looked before I opened it in Fontographer. How can I make it look good again?

First – in case something got corrupted – select all from the original font, and copy and paste into a new font. If paste doesn't work, try copying one or two rows of the database at a time. Next you will need to rehint the font as follows: from the Hints menu select Vertical Alignment Zones, click Recalc, and click OK. Then select Hint Parameters from the Hints menu, click Recalc, and click OK. Go to the Hints menu and select Autohint twice to turn it off and back on.

Note: We are assuming you are creating a TrueType font. Also, if the above steps don't improve the character, it may be because you have manually hinted the font. The hinting may be causing the font corruption. Many commercial fonts (all of Microsoft's and Monotype's) use a delta hinting in their fonts. Delta hinting allows for precise control of the font's screen appearance. Fontographer doesn't work with delta hints but Fontlab Ltd.'s professional font editor Fontlab Studio does. You can use Fontlab Studio to delta-hint your font.

I want to use Fontographer to create my own version of a non-Roman font. Why can't I load an entire 2-Byte font into Fontographer?

2-Byte fonts contain tens of thousands of characters. Fontographer 5 has a limit of 32 000 characters. If you need to create a font with a larger number of characters, Fontlab Ltd. offers AsiaFont Studio, a high-end Asian font editor for that purpose.

Characters above the code 256 are accessed via Unicode. Unicode is supported in many applications on Mac OS X, many applications on Windows 2000/XP and some applications running on Windows 98/ME.

Fontographer 5 for the Macintosh has a built-in Unicode font encoding option that allows you to paste glyphs into 2,147 prenumbered Unicode slots. This makes life a lot easier for those who are using standard systems such as Cyrillic, Hebrew and most European characters.

For more information on Unicode, Code Pages, cmap tables, input systems, keyboard drivers, localization, and so forth, refer to the user's manual of Fontlab Studio that contains more technical information on that topic. You can download the Fontlab Studio manual in PDF format from http://www.fontlab.com/.

Appendix B. An Annotated Bibliography of Typography and the other Arts of the Book

by David S. Rose • Five Roses Press

The explosion of desktop-based, digital pre-press technology at the end of the twentieth century brought to a wider audience the previously specialized world of typography. Modern type design applications such as Fontographer give users the ability to create new digital typefaces from the imagination, to recreate classic faces that are otherwise unavailable in digital form, and to adapt existing faces for specific needs. For those artisans who still hand-set and print with traditional letterpress technology, a dwindling number of type foundries continue to provide classic metal faces. But for designers who combine the two worlds by printing letterpress from photopolymer plates, the options are unlimited.

As with any powerful tools, the more one knows of the history behind them, the better able one will be to utilize them. The books listed here are just a few of hundreds that have been written on the subject of typography over three centuries, but they will provide a solid start for reading in this area.

While many of the works listed are classics in the field, not all of them are currently in print. Those that are not available from the publisher (or from reprint houses such as Dover Publications) are generally available at most large libraries, and may often be found at antiquarian dealers who specialize in the field of Books about Books. The rapid adoption of the Internet by antiquarian book dealers now means that most of these books are a simple click away, through searches on web sites such as Bookfinder.com. Many of them may also become available as electronic books through Amazon, Apple or Google Books.

- Overviews of Printing Types
- History and Development of Lettering and Letterforms
- Type Designs from Various Periods
- Typography
- Book Design
- Type Designers
- Typeface Reference Works
- History of Printing
- Letterpress Printing
- Other Book Arts
- Bibliographies

Overviews of Printing Types

Printing Types: An Introduction by Alexander Lawson with Dwight Agner [Boston: Beacon Press, 1990] is a short (120 pages) easy-to-read overview that is exactly as advertised: an introduction. For over thirty years, Lawson taught the history of printing types at the Rochester Institute of Technology School of Printing, and this book grew out of his need for a simple handbook on the subject for his students. It is a well-designed and illustrated inexpensive paperback, and would probably be your best bet if you have a casual interest in the subject and only want to read one book. The latest edition, brought current through 1990, covers early digital typography as well. Lawson's *Anatomy of a Typeface* (see below) has become standard reading for many current courses on typography.

Printing Types: Their History, Forms, and Use by Daniel Berkeley Updike [New York: Dover Publications, 1980. Reprint of the second (1937) edition]. This is the classic work in the field of typographic history. Updike was a leader in the revival of traditional printing typefaces in the United States, and was the founder of the Merrymount Press (1893). A series of lectures he gave at Harvard from 1910-1917 served as the basis for Printing Types, which was first published in 1922. This Dover reprint is in two volumes, 618 pages of text plus 300 unnumbered illustrations. As Dover says in the jacket notes, "Printing Types presents the standards, the landmarks in typography that anyone connected with printing must know. In its mammoth, illustrated coverage, it is without a doubt the definitive guide to the subject."

Letters of Credit: A View of Type Design by Walter Tracy [Boston: David Godine, 1986]. A beautiful and profusely illustrated step-by-step demonstration of type-design aesthetics that traces the beginnings and the path of modern-day typesetting.

Fine Print on Type: the Best of Fine Print Magazine on Type and Typography by Charles A. Bigelow, Paul Hayden Duensing, Linnea Gentry [San Francisco: Fine Print, 1988] is an excellent selection of articles from *Fine Print* magazine, the late indispensable periodical with which anyone concerned with type should be familiar. Each issue was designed in a different style, printed by letterpress and included scholarly articles, typographic overviews, reviews, and notices of new books on typography. *Fine Print* was published quarterly through about 1990, after which the publication led a cliff-hanging existence as various groups and institutions tried to save it. While Fine Print is long gone, a final retrospective index is available from Oak Knoll Books (2003). It was succeeded briefly by *Bookways* (1991-1995), and more recently by *Parenthesis* (1998-), the journal of the Fine Press Book Association.

Type and Typefaces by J. Ben Lieberman [New Rochelle: The Myriade Press, 1978] is an alternative to the Lawson books, but less accurate, bigger (142 pages, 8 1/2 x 11, hardcover) and more enthusiastic. Ben Lieberman was a passionate amateur printer and the father of the American Chappel movement of hobby printers. An exuberant look at the history, classification, identification, and personalities of typography, it includes examples of over 1,000 typefaces. This book can be fun to read, despite its errors of both omission and commission.

History and Development of Lettering and Letterforms

Letters by James Hutchinson [New York: Van Nostrand Reinhold, 1983]. A stylishly designed, very readable history of alphabets, writing, and printing types.

The History and Technique of Lettering by Alexander Nesbitt [New York: Dover Publications, 1957]. A thorough history of type design from its origin through the mid-twentieth century, this book includes information on the development of letterforms since the invention of printing. It is written from an artist's perspective, and has a how-to section on lettering.

The Alphabet and Elements of Lettering by Frederic W. Goudy [New York: Dover Publications, 1963. Reprint of 1952 University of California edition]. This book brings Goudy's unique perspective as the most prolific type designer of the twentieth century to the letters of the alphabet and their development. More of a beautiful schematic than a well developed historical book.

Roman Lettering by L.C. Evetts [New York: Taplinger, 1979. Reprint of 1938 Pittman edition] includes a character-by-character analysis of the letters on Trajan's Column in Rome, which have served for centuries as one of the foundations of roman (serif) letter design. Evetts also includes charts showing the evolution of the roman alphabet through the centuries. Handsome lettering, with little text to clutter the presentation.

An ABC Book: ABC of Lettering and Printing Types by Erik Lindegren [New York: Greenwich House, 1982. Reprint of Vol. C of 1964 Museum Books edition]. A survey of type, calligraphy, and design, with examples of work from all periods, with an especially strong representation of lettering by Swedish, English, German, and American scribes and designers. A lively, well-designed introduction to letters.

Writing, Illuminating and Lettering by Edward Johnston [New York: Taplinger, 1980]. The comprehensive calligraphy manual by the man who led the twentieth century revival of calligraphy. Johnston's influence on English, American and German lettering and design was immense.

Type Designs from Various Periods

Anatomy of a Typeface by Alexander Lawson [Boston: David R. Godine, 1990] A great book from one of the leading typographic experts of the late twentieth century, this substantial work examines a wide variety of typefaces in great detail, and explains why they look the way they do. An excellent reference work for the designer and printer that will both improve your eye for the detail of font design and inform the choices you will make in specifying and setting type yourself.

From Gutenberg to OpenType: An Illustrated History of Type from the Earliest Letterforms to the Latest Digital Fonts by Robin Dodd [Point Roberts, WA: Hartley & Marks, 2006] A well-illustrated book of examples, Dodd provides a useful short history of type and typography from Gutenberg onwards.

Art of the Printed Book, 1455-1955: Masterpieces of Typography Through Five Centuries from the Collections of the Pierpont Morgan Library, New York by Joseph Blumenthal [New York: Pierpont Morgan Library, and Boston, MA: D.R. Godine, 1985]. Available both in hardcover and paperback, this collection by one of the great printer/scholars of the twentieth century is a must have for anyone interested in original source material. More than a hundred full pages of facsimiles from the Morgan Library provide an instant overview of the development of typographic design from Gutenberg to the mid-twentieth century.

Selected Essays on Books and Printing by A. F. Johnson [Amsterdam: Van Gendt, 1970]. Johnson was a scholar at the British Museum and, along with Daniel Berkeley Updike and Stanley Morison, was considered one of the experts in the field of typographic history. This lovely, massive (500 pages), and very expensive collection of some of his writings from 1927-1957 concentrates primarily on the typographic work of sixteenth century calligraphers and printers.

A View of Early Typography Up to About 1600 by Harry Carter (The Lyell Lectures, 1968) [London: Hyphen Press, 2002. Reprint of Oxford at the Clarendon Press, 1969]. Modestly titled but a classic book.

A History of the Old English Letter Foundries: with Notes, Historical And Bibliographical, on the Rise and Fall of English Typography by Talbot Baines Reed, rev. and enlarged by A. F. Johnston [Folkestone: Dawsons, 1974. Reprint of the 1952 Faber & Faber edition]. A classic history originally published in 1887.

Counterpunch: Making Type in the Sixteenth Century, Designing Typefaces Now by Fred Smeijers. [London: Hyphen Press, 1996]. Before digital fonts there was metal type, each letter of which had to be laboriously carved in metal "punches." Smeijers explains the techniques and how the technology affected type design. He argues that digital techniques should shape contemporary design. A fascinating advanced book, and a great read together with Theo Rehak's up-to-date work on *Practical Typecasting* (below).

Notes on a Century of Typography at the University Press, Oxford, 1693-1794 by Horace Hart [Oxford: Clarendon Press, 1970. Reprint of the 1900 edition] with an introduction and additional notes by Harry Carter. History of the types and typography of the Oxford University Press, generally regarded as the preeminent scholarly press in the western world.

Nineteenth Century Ornamented Type Faces by Nicolete Gray [Berkeley: University of California Press, 1976. Reprint of the 1938 Faber & Faber edition]. The definitive book on its subject.

American Wood Type, 1828-1900 by Rob Roy Kelly. [Liber Apertus Press, 2010. On-demand reprint of the 1977 Da Capo Press edition]. Notes on the evolution of decorated and large wood types, and comments on related trades. As with the Nicolete Gray book, this is the definitive work in its field. The book was issued in several editions, of which this (paperback) is the least expensive. (Kelly's wood type collection is owned by the University of Texas, and parts of it are online.)

The Typographic Book 1450-1935 by Stanley Morison and Kenneth Day [Chicago: University of Chicago Press, 1963]. A lush, expensive, visual treasury of almost 500 years of typography, including 357 plates.

Typography

Thinking with Type: A Critical Guide for Designers, Writers, Editors, & Students by Ellen Lupton. [New York: Princeton Architectural Press, 2004] A succinct and accessible guide about type, typography and design. Geared towards digital typography, but thought provoking for all.

The Elements of Typographic Style by Robert Bringhurst. [Point Roberts, WA: Hartley and Marks, 2004,3rd ed.] A highly acclaimed, although more advanced, standard work in the field. Bringhurst, a poet and typographer, has written a thoughtful book on fine design, which has gradually assumed a near-canonical status among serious contemporary designers.

A Typographic Workbook: A Primer to History, Techniques, and Artistry by Cynthia Busic-Snyder and Kate Clair [New York: Wiley, 1999. New edition expected 2010]. A good place to start for a basic grounding in typographic design.

Detail In Typography by Jost Hochuli [London: Hyphen Press, 2008] A succinct guide to issues of legibility and spacing in setting digital type.

The Complete Manual of Typography: A Guide to Setting Perfect Type by James Felici [Berkeley, CA: Peachpit Press, 2003] A practical manual for selecting and working with digital type.

The Crystal Goblet: Sixteen Essays on Typography by Beatrice Warde [Cleveland and New York: World Publishing Company, 1956]. From a major woman in the field of typography come some thought-provoking pieces, including the famous analogy that gave the collection its name. Mandatory reading for would-be typographers.

The Case for Legibility by John Ryder [London: The Bodley Head, 1979] "Not a typographer's manual nor a 'do-it-yourself' guide to book design, it is a personal statement of great sincerity and conviction by a distinguished practitioner of the art." Ryder also wrote *Printing for Pleasure*, one of the touchstones of the avocational letterpress printing movement.

First Principles of Typography by Stanley Morison [Cambridge: at the University Press, 1951]. An important book from the man credited with the creation of the Times Roman typeface for the London Times.

Asymmetric Typography by Jan Tschichold, translated by Ruari McLean [New York: Reinhold, 1967]. Jan Tschichold, a well-known typographer, inspired many people to rethink "conventional" theories of typography when this seminal work was published in the mid-1960s. Whether or not you agree with his approach, this book will widen your typographic horizons.

An Essay on Typography by Eric Gill [Boston: D.R. Godine, 1988, 1st U.S. ed.]. A classic typographic manifesto on the art and craft of letterforms from the designer of Gill Sans and the famous lettering for the London Underground.

Typography, A Manual of Design by Emil Ruder [Niederteufen, Switzerland: Arthur Niggli Ltd, 1977, 3rd ed.]. A fascinating, disciplined, and very Swiss analysis of typography and letterforms. Ruder's discussion and illustration of the importance of white space in letterforms and graphic designs is excellent background reading.

Report on the Typography of the Cambridge University Press by Bruce Rogers [London: Wynken de Worde Society, 1967. Reprint of the Cambridge 1950 edition]. Bruce Rogers is regarded by many as having been the greatest typographer and book designer of the twentieth century. After World War II he was commissioned by the Cambridge University Press to undertake a thorough review of all of the Press' publications and standards. The resulting Report had a major impact not only on the C.U.P., but also on the general typographic theory in both Britain and the U.S.

A Century for the Century: Fine Printed Books 1900-1999 by Martin Hutner and Jerry Kelly. [Boston: David R. Godine, 2004, revised and enlarged edition of 1999 Grolier Club exhibition]. Illustrated catalog from an important 1999 exhibition at the Grolier Club, which showed 100 beautiful and influential books from the Twentieth Century. A lovely work, and very useful as inspiration for typographers in seeing how type builds from characters, through paragraphs and pages, into books.

Designing with Type: the Essential Guide to Typography by James Craig and Susan E. Meyer [New York: Watson-Guptill Publications, 2006, 5th ed.]. A modern how-to book, often used as the primary textbook in college design courses. It is available at many large bookstores and from graphic arts dealers.

Finer Points in the Spacing & Arrangement of Type by Geoffrey Dowding. [Point Roberts, WA: Hartley & Marks, 1998, rev. ed.].

Book Design

Methods of Book Design: The Practice of an Industrial Craft by Hugh Williamson. [New Haven: Yale University Press, 1983]. An excellent book, not only for the author's typographical observations, but also as a comprehensive survey of printing at the height of letterpress. In some ways, a classic.

The Design of Books by Adrian Wilson. [New York: Reinhold Publishing Corporation, 1967]. A classic on the design, layout, and typography of traditional pages and books, written by a great letterpress printer and fine designer. Although the technology is dated, the principles remain constant.

Bookmaking: Editing, Design, Production by Marshall Lee. [New York: W.W. Norton, 2009] Originally written primarily about letterpress in 1965, this 500+ page work has recently been re-issued in a greatly updated third edition for the computer era.

Printing Poetry: A Workbook in Typographic Reification by Clifford Burke. [San Francisco: Scarab Press, 1980]. A very informative work on this subject that also applies to other letterpress printing. A classic work in its field, the book is a typographic example in its own right. Issued in an edition of only 1000, and priced accordingly.

Type Designers

Twentieth Century Type Designers by Sebastian Carter [New York: Taplinger Publishing Company, 1987]. An excellent look at the people behind the type faces, with in-depth profiles of designers such as Goudy, Rogers, Morison, Zapf, et al.

An A-Z of Type Designers by Neil MacMillan [New Haven, CT: Yale University Press, 2006]. Biographies of 260 type designers, from Gutenberg to the present, intended to succeed Rookledge's *International Handbook of Type Designers* (1991). Includes essays by Jean François Porchez, Erik Spiekermann, and John Downer.

Typologia: Studies in Type Design & Type Making, with Comments on the Invention of Typography, the First Types, Legibility, and Fine Printing by Frederic W. Goudy. [Berkeley: University of California Press, 1977. Reprint of 1940 edition]. Written by the most prolific type designer of the 20th century (creator of, among others, the eponymous Goudy Oldstyle, Copperplate Gothic and University of California Oldstyle), this reprint discusses the history, function, and meaning of type, and gives some very good insights into how a type designer works.

Jan Tschichold: Typographer by Ruari McLean [Boston: David R. Godine, 1975]. This biography puts Tschichold's career and writings in the context of developments in society around him. It is informative and thought provoking on its own, and serves as useful background to his writings on the subject.

Manuale Typographicum: 100 Typographical Arrangements with Considerations about Types, Typography and the Art of Printing Selected from Past and Present, Printed in Eighteen Languages by Hermann Zapf [Frankfurt, New York: Z-Presse, 1968]. Hermann Zapf is known primarily in digital circles for giving his name to the Zapf Dingbat font. But to the cognoscenti, he is also one of the most respected and creative typographers and type designers of the twentieth century, who created not only the Dingbat and Zapf Chancery fonts, but also Optima and Palatino and many other faces. *Manuale Typographicum* is a breathtaking "tour de force," consisting of 100 broadsides about type design in a wide variety of faces and styles. A superb source of inspiration and examples.

Hermann Zapf and His Design Philosophy by Hermann Zapf, Introduction by Carl Zahn [New Haven: Yale University Press, 90 color plates]. While the Manuale shows the master at work, this volume is a discourse on Zapf's insights into type design. An excellent book.

Edward Johnston by Priscilla Johnston [New York: Pentallic, 1976]. This biography of the twentieth century's most important calligrapher, written by his daughter, traces his career and influence. Unlike many printing books, this one is a delightful read.

Of the Just Shaping of Letters by Albrecht Dürer [New York: Dover Publications, 1965. Reprint of the Grolier Club translation of 1917]. Originally part of Dürer's theoretical treatise on applied geometry, here is the source for those famous capital letters set against a gridded background.

Champ Fleury by Geoffrey Tory, translated into English and annotated by George B. Ives [New York: Dover Publications, 1967. Reprint of the Grolier Club edition of 1927]. The other famous humanistic alphabet similar to the one discussed in the Dürer book, but this is the one with the letters shown against naked human bodies in addition to the grid system.

Typeface Reference Works

American Metal Typefaces of the Twentieth Century by Mac McGrew [New Castle, Delaware: Oak Knoll Books, 1994, 2nd rev, ed.]. The definitive work on the subject, and an essential reference for both graphic designers and current letterpress printers. Currently in print from the publisher.

The Encyclopedia of Type Faces 55th Anniversary Edition by W. Pincus Jaspert, W. Turner Berry, and A. F. Johnson [London: Cassell, 2009, 55th anniversary edition]. A standard comprehensive reference in the field, this work is a detailed listing of over 2,000 faces, arranged by name, with full information on their history, designers, etc. Although even after several editions it has numerous uncorrected errors (dates, foundries, names, even occasionally an incorrect specimen shown) it is still a required reference work on the subject.

Type and Typography: The Designer's Type Book by Ben Rosen [New York: Van Nostrand Reinhold Co., 1976 rev. ed.].

Fontbook: Digital Type Compendium, by FontShop International [Berlin: FontShop International, 2006]. An expensive compendium of (nearly) all digital fonts available at the time of publication, and perhaps the last time such a massive type specimen book will be printed on paper.

History of Printing

A Short History of the Printed Word by Warren Chappell and Robert Bringhurst [Point Roberts, WA: Hartley & Marks, 2000]. A once-over-very-lightly history, hitting the highlights in the development of type, printing and bookmaking. Updated by Bringhurst from 1970.

Five Hundred Years of Printing by S. H. Steinberg [New Castle: Oak Knoll Books, 1996. Enlarged edition revised from Penguin Books, 1974]. A 400-page small-print paperback, which is still in print, this covers Gutenberg through the early 20th century. Steinberg's style is a little dry. Since his death, the book (starting with the third edition) has been edited by James Moran.

Encyclopedia of the Book by Geoffrey Glaister [New Castle: Oak Knoll Books, 1996. Second edition of University of California Press 1979 edition issued previously as *Glossary of the Book*]. "Glaister" is an essential book for understanding terminology relating to the history of printing, the physical book (typography, paper, binding), book collecting and publishing. It may be supplemented by John Feather's *A Dictionary of Book History* (1986) and by John Carter's *ABC's of Book Collecting* (many editions).

The Book: The Story of Printing & Bookmaking by Douglas C. McMurtrie [New York: Oxford University Press, 1943]. Almost 700 pages of large type devoted to the history of the book, by one of the most prolific writers in the field. Easy to read, anecdotal, and illustrated. Although out of print, it is not particularly scarce and, if you can find it, probably the quickest way to get up to speed on printing history.

Letterpress Printing

While the craft of letterpress printing may at first glance appear completely archaic and anachronistic, the explosion of tools for digital typography, combined with the new technology of photopolymer plate production, has led to a resurgence of interest in this most fascinating, historic and rewarding craft. Consider the works below as a way to introduce yourself to a fifteenth century world that is still thriving in the twenty-first.

Introduction to Letterpress Printing in the 21st Century by David S. Rose [www.fiveroses.org/intro.htm]. The complete online Getting Started Guide to everything you need to know about acquiring a printing press, finding supplies, learning to print, and setting up your very own letterpress shop. An unabashedly enthusiastic primer by the author of this bibliography.

Letterpress Printing: A Manual for Modern Fine Press Printers by Paul Maravelas. [New Castle: Oak Knoll Press, 2005]. The most recent printer's manual for letterpress printing, oriented towards those who are repurposing old equipment for producing fine and book arts printing. The book is a useful introduction and compilation of information culled from sources online and in print. Follow it immediately with Cleeton and Pitkin, *General Printing*.

General Printing by Glen U. Cleeton and Charles W. Pitkin. [Liber Apertus Press, 2004. Reprint of the McKnight & McKnight Publishing Company, 1963 edition]. Probably the best all-around introductory book for traditional letterpress printing, this manual is profusely illustrated with detailed and useful photographs. The book appeared in three editions from 1941-1963 before the current reprint. Copies of the older editions of this book are readily available in both paperback and hardcover.

Printing Digital Type on the Hand-Operated Flatbed Cylinder Press by Gerald Lange [Marina del Rey: Bieler Press, 2009, 4th ed.]. This is one of the few letterpress manuals currently in print, and the only one specifically addressing both Vandercook proof presses (the gold standard for current fine letterpress printers) and photopolymer plates. This book is the authority on the technologies of "modern" limited edition letterpress printing. Subjects covered include digital type and computer practices; letterpress configuration; photopolymer plates, flat-bases, and processing equipment; photopolymer plate-making; plate registration and travel; impression; cylinder packing and makeready; presswork; ink and inking; and press operation and maintenance. It also includes an updated listing of manufacturers and distributors, as well as troubleshooting guides to problems encountered during the processing and printing of photopolymer plates.

Printing for Pleasure, A Practical Guide for Amateurs by John Ryder [Published in multiple editions from 1955-1977, in England and the US, by publishers including Chicago: Henry Regnery Co., (1977) and London: The Bodley Head (1976) This is still in print from The Bodley Head in the UK or Oak Knoll Books in North America]. A lovely, classy, little book, both pleasing to look at and inspirational for the novice amateur printer. This introductory work gives a light overview of the hobby of letterpress printing on both sides of the Atlantic, covering how to choose a press, type, paper and ink, as well as planning, design and production. A good place to start if you are just considering taking up this avocation, and a nice place to come back to every now and then to remind you why you are still printing.

Printing as a Hobby, by J. Ben Lieberman [New York: Sterling Publishing Co. & London: Oak Tree Press, 1963.]. This book is the brash, bigger, and less restrained American counterpart to the quintessentially British book by Ryder. Lieberman was an enthusiastic amateur printer, and this book is an exuberant, well-illustrated pitch for his hobby. The author was not a scholar (nor particularly an aesthete), but if you don't mind his unabashed "boosterism," you might find this book fun to read, despite its errors of both omission and commission (not unlike his later book, *Type and Typefaces*, described above).

Other Book Arts

Practical Typecasting by Theo Rehak. [New Castle: Oak Knoll Books, 1993]. The ultimate and definitive book on the subject, by the dean of American typefounders.

Hand Bookbinding: A Manual of Instruction by Aldren A. Watson. [New York: Dover Publications, 1996] A clear, thorough, inexpensive introduction to hand binding.

The Papermaker's Companion: The Ultimate Guide to Making And Using Handmade Paper by Helen Hiebert [North Adams, MA: Storey Publishing, 2000]. Extensive step by step instructions.

How to Marbleize Paper: Step-By-Step Instructions for 12 Traditional Patterns by Gabriele Grunebaum [New York: Dover Publications: 1984]. A slim, inexpensive, but useful paperback.

Bibliographies

A Typological Tally compiled by Tony Appleton [Brighton, Tony Appleton, 1973]. An enumerative list of thirteen hundred writings in English on printing history, typography, bookbinding, and papermaking, compiled by the late dean of book dealers in the field.

A Bibliography of Printing with Notes and Illustrations by F. C. Bigmore and C. W. H. Wyman [London: Oak Knoll Books, 1978]. Universally known as "Bigmore and Wyman," this is to printing bibliographies what Updike is to books about printing types. Published in 1880 (editions since then have been reprints), B&W provides excellent commentaries on just about every book that had been written on the subject as of the year it was published. Richard Gabriel-Rummonds' two volume book, *Nineteenth Century Printing Practices* includes a complete bibliography of letterpress printer's manuals, supplementing Bigmore and Wyman.

Type History & Design at Briar Press [www.briarpress.org] is a current, online directory to websites, organizations, museums, classified advertisements and mailing lists relating to the subject, part of the web's most useful repository of information about printing arts and history.

Many thanks to Howard Gralla, Alvin Eisenman, Robert Fleck, Kathy Schinhofen, Chuck Rowe, Earl Allen, Susan Lesch, Kathleen Tinkel, Michael J. Boyle, John Horn, Chris Simonds, Fritz Klinke, Roberta Lavadour, David Norton, Tom Parson, David Goodrich and the many members of the Letpress Internet mailing list for their suggestions before and during the compilation of this bibliography. A major debt of gratitude is particularly due to Paul Romaine, President of the American Printing History Association, and Jerry Kelly, typographer extraordinaire, who have been instrumental in helping update this bibliography for the 21st century.

An earlier version of this bibliography was originally published in conjunction with a prior release of Fontographer back in the 20th century. That version was, in turn, adapted and expanded from a still-earlier annotated checklist by the same author prepared for members of the erstwhile MAUG Forums on Compuserve, one of the first online services.

David S. Rose is a letterpress printer and graphic designer as well as a book collector, writer and teacher on the subject of printing history and typography. He is President of the Typophiles, the not-for-profit educational association of type lovers who for more than three quarters of a century have shared their abiding pleasure in fine printing through convivial meetings and handsomely produced publications. He is Vice Chairman of the Center for Book Arts in New York, Scribe of the Honorable Company of College Printers at Yale University, proprietor of the Five Roses Press, and author of the Internet's highest-ranked site on the subject of letterpress printing. He is a member of the Grolier Club, the Amalgamated Printers Association and the Westchester Chappel. More information is available at www.fiveroses.org.

Copyright © 1988-2010 by David S. Rose david@fiveroses.org. For re-publication inquiries, please contact the author. Revision: May 11, 2010

Appendix C. General Information

Fontographer background

Em square

Each character is described in terms of a rectangle called an em square, or the UPM (Units Per eM) size. The em square is the basis of all font dimensions. It defines the font height and the coordinate grid on which the characters are drawn.

Fontographer's em square is used as a normalization value when generating PostScript and also defines the precision possible in the font. The em square is divided into font units (typically 1000). The ascent and descent (found in the Font Info dialog) are measured in em units, as are all measurements in Fontographer.

When the font rasterizer produces an image of a letter at, say, 10 point, then the 10 point will correspond to the 1000 units. If your capital letter H is 700 units high, its physical size at the 10 point font size will be 7 point.

If you print out your characters at the point size that corresponds to the em square size (e.g. 1000 pt), then each font unit will have the size of exactly one point.

Unit

Fontographer units are values whose size is relative, varying with the size of the em-square and the point size of the output. Because outline fonts are scalable, units do not directly relate to points, pixels, or any physical distance. It is possible to relate units to physical distance if the em square is set up knowing the printing device resolution and scale at which the font is to be printed, but this defeats the whole concept of outline fonts. Think of them as design units whose dimension may vary.

Fontographer is capable of defining fonts with an em square of up to 8000 by 8000 units. This range is far greater than needed for any normal font. Most PostScript fonts have an em square of 1000 units, which is also Fontographer's default. Precise work for high-resolution printers above 1000 dpi might benefit from an em square of 2000 or more. The choice of em square size is left up to you.

Choosing a larger em square means that you have more precision when outputting the character. It does not mean that the printed characters are larger. Since a font is always normalized so that the em square is one point high, a font defined with an em square of 4000 prints out at exactly the same size as a font defined with an em square of 1000. It has four times the internal resolution, but that is insignificant on a 300 dot per inch printer except at very large point sizes. A 1000-point high character (almost 14 inches) would have at most 1/72 inch of inaccuracy if drawn on a 1000-unit em square. If drawn with a 4000-unit em square, the inaccuracy would be approximately one dot (1/288 of an inch). Proper positioning of the control points, whatever em square size is used, is more important than greater resolution.

Bitmap background

Bitmap fonts (Macintosh)

MacOS 9 and earlier use bitmap fonts for screen display of Type 1 fonts. Bitmap fonts are stored in the suitcase file, where they are accessible to all programs if installed in the system. Bitmap fonts cannot be smoothly scaled, so each Type 1 font face usually had several bitmap fonts in different sizes for display purposes.

Note: Mac Type 1 fonts require at least one bitmap size, but TrueType fonts or any Windows fonts don't require any.

Fontographer 5 for Mac can include bitmap glyphs in a Type 1 or TrueType suitcase. It can also export them into a BDF bitmap font, which you can further edit with a bitmap font editor such as Fontlab BitFonter. In Fontographer 5, there are three ways to generate bitmap glyphs from font outlines, which will produce different results. If you're interested in high-quality bitmap glyphs, you should explore all three possibilities to determine which give you the most satisfactory results.

Depending on the way you generate the bitmap glyphs, different rasterizers will be used:

1. If you generate the bitmaps inside Fontographer (using **Element>Bitmap Info** or **Window >Open Bitmap Window**), the bitmap glyphs will be generated using the classic Fontographer rasterizer. This rasterizer will produce similar results as older versions of Fontographer. Especially in small sizes, the results may not be very satisfactory.

2. If you don't have any bitmaps included in your font, and you generate a Mac Type 1 Suitcase font, the bitmap glyphs will be generated using a built-in version of the Adobe Type 1 rasterizer. This rasterizer will process the Type 1 hints specified in your font, and the result will be similar to the screen appearance of the Type 1 font if it were installed in Microsoft Windows. To do this, choose **File> Generate Font Files**, switch to Advanced mode, choose "Mac OS X and 9" as platform and "Mac Type 1 Suitcase" as format. Then, specify the **Bitmap Sizes to Create** and generate the suitcase. Typically, bitmaps generated this way will be of higher quality than those generated inside Fontographer.

3. If you don't have any bitmaps included in your font, and you generate a Mac TrueType Suitcase font, the bitmap glyphs will be generated using a built-in version of the Microsoft TrueType rasterizer. The Type 1 hints specified in your font will be converted into TrueType hinting using the high-quality Fontlab hint conversion algorithm, and bitmaps will be generated from the hinted TrueType outlines. The result will be similar to the screen appearance of the TrueType font if it were installed in Microsoft Windows. To do this, choose **File>Generate Font Files**, switch to Advanced mode, choose "Mac OS X and 9" as platform and "Mac TrueType Suitcase" as format. Then, specify the **Bitmap Sizes to Create** and generate the suitcase. Typically, bitmaps generated this way will also be of higher quality than those generated inside Fontographer, but will be different than those generated using the Adobe Type 1 rasterizer.

If you generated your bitmaps using the Adobe Type 1 rasterizer or the Microsoft TrueType rasterizer (i.e. method 2 or 3 described above), you can then choose **File>Import bitmaps**, and pick the Mac Suitcase you just generated (a Type 1 or a TrueType suitcase), and then **Select All** bitmap sizes to import. The bitmap glyphs will be imported into your Fontographer document. You can edit them using **Window>Open Bitmap Window**, or you can export them into the BDF format using **File>Export>BDF**.

FONDs (Macintosh)

A FOND is a table that creates the link from the bitmap font to the outline font. A FOND is automatically generated by Fontographer whenever a Mac Type 1 suitcase is generated. It contains several important types of information.

The Macintosh print manager examines the FOND, and if there is an outline font file available in the same font folder, it is downloaded to the printer before printing begins. If no outline font file is found, the bitmap font is used. The outline font file must be in the same folder as the suitcase file in order to be found for downloading.

The FOND also links all the bitmap fonts in a family so that custom-tuned italic or bold faces may be used in place of inferior derived styles. This component of the FOND is independent of outline fonts. FONDs also store additional information, such as character metrics and kerning.

PostScript

PostScript is a programming language developed by Adobe Systems, Inc. to drive high-resolution printers. Invented as a standard page description language with a well-documented behavior, it is used on printers produced by a number of different manufacturers.

PostScript's generality allows any picture to be described as a program; this is how fonts are constructed. Each letter is described by a small program that draws the letter outline using PostScript graphic commands. In order to print the character on a page, this character drawing program is run by the printer control software to draw and fill the outline, generating a high-resolution bitmap. The bitmap is generated at the current point size, and then copied to the page at the appropriate location.

Bézier curve

The PostScript graphic commands available are straight lines, arcs, and Bézier curves. A Bézier curve has control points that allow changing the shape of the curve. An arc has uniform curvature everywhere, but a Bézier curve allows non-uniform curvature. This property is useful in describing complex shapes with only a few well-chosen control points. Additionally, Bézier curves have desirable smoothness properties that lend themselves to efficient generating programs.

Another useful property of Bézier curves is that it is very easy to guarantee smooth joins at the endpoints of the curve. Smooth tangent joins are very important to high-quality letter shapes. Fontographer supports automatic tangent joins during character construction, so its characters are perfectly smooth where they should be smooth (but can be discontinuous if necessary).

Caching

Translating from the program into a bitmap is a complex process that takes an amount of time proportional to the complexity of the character. To minimize the amount of time spent generating bitmaps from the letter drawing programs, the bitmaps are saved on the printer's hard disk or in memory for later use. This saving process is called caching. The first time a particular letter is printed, its bitmap must be generated and cached before it can be drawn on the page. Once a letter has been converted, its bitmap is normally found in the cache, and is used directly.

Path

A path is composed of line or curve segments. These segments may be connected (the normal case) or disconnected (such as the dot on a lower-case “i”). Paths may be open or closed. A segment of a path is closed if the last point connects back to the first point, otherwise it is open.

Characters may consist of open paths or closed paths, but not both. Normally a character is constructed such that its outline is defined as a single closed path. The program that generates the bitmap can then just fill in the outline. Some characters, such as the letter “O,” have an inside and an outside. A simple-minded approach to filling such a letter would fill both outlines, resulting in a single solid circle. To handle this case, PostScript has two sophisticated approaches to filling.

Filling techniques

Winding number fill

The standard PostScript filling technique is called a winding number fill. This relies on one path being described in a clockwise direction, and the other path being described in a counterclockwise direction. A point is outside, and thus not filled, if a line away from that point in any direction crosses exactly as many counterclockwise paths as it crosses clockwise paths. In the case of the “O,” the outer path should be drawn clockwise and the inner path counterclockwise.

Technically, it doesn’t make any difference whether the outside path is clockwise or counterclockwise, but for the sake of consistency between Fontographer’s fonts and the proper operation of automatic hints, the outer paths should be clockwise and the inner paths counterclockwise.

Even-odd fill (Macintosh)

The other filling technique is called an even-odd fill. A point is outside (and thus not filled) if a line away from that point in any direction crosses an even number of paths, regardless of the path direction. In the case of the “O,” even-odd filling would have the desired result even if both paths were clockwise. Fontographer will fill the paths properly as long as the paths do not self-intersect.

Glossary

AAT (*Apple Advanced Typography) fonts*: The TrueType fonts especially designed for use with ATSUI. Like OpenType fonts these fonts have special features such as swashes, contextual forms, ligatures etc. These fonts are widely presented among system fonts in Mac OS X and are supported in Cocoa applications.

AFM (ASCII Font Metrics) file: a text file that contains the metrics information for a PC Type 1 font.

Alphabet/Script: The collection of characters used to write a particular language. "The" alphabet (as North Americans and English know it) is the script for the English language; Latin script is the script for most European, South-American and some Asian languages. Cyrillic script is used in all Slavonic languages (Russian, Ukrainian, Serbian, Bulgarian and many others). Note that a script usually includes many more characters than necessary for the one language. Latin script, for example, includes more than 200 characters.

Aperture: The degree of enclosure within a letter. Larger apertures may improve legibility if the increased space helps to differentiate similar letter forms (such as the o, e and c).

Apex: The upper point where two strokes meet, as in the tip of the A.

Application: A computer program, designed to perform a specific function such as word processing or illustrating.

Arm: A projecting stroke that extends from the vertical stem of a letter. The top of the upper case T and the horizontal strokes of the F, L and E are arms. The upward diagonals on the upper/lower case K/k may also be called arms.

Ascent / Ascender: A font's maximum distance above the baseline.

A*scent line / Ascender line*: The ascent line marks the top of the capital letters. An ascent guideline is automatically drawn at the vertical position specified when the font is created, and may be changed by using the Font Info command from the Element menu. Characters should not normally appear above the ascent line.

ASCII (American Standard Code for Information Interchange): A numbering scheme used for identifying printing characters.

ATM (Adobe Type Manager): The program that improves your screen font display by eliminating jagged edges on Type 1 fonts.

ATSUI (Apple Type Services for Unicode Imaging): Apple's technology and a set of routines that enable the rendering of Unicode-encoded text with advanced typographic features. It automatically handles many of the complexities inherent in text layout, including the correct rendering of text in bidirectional and vertical script systems.*AFM:* (Adobe Font Metrics) A specification for storing (in a text file) font metrics information such as character widths, kerning pairs, and character bounding boxes.

Baseline: The baseline is the imaginary horizontal line upon which all letters sit. When printing mixed fonts on a line, all baselines line up with one another. The baseline position does not need to be explicitly specified, since it is always at a vertical location of zero.

Basepoint: Fontographer's reference point from which distances are measured, and about which the special effect transformations may be performed.

BCP: Bézier control point. One of two points which guide a Bézier curve.

Bézier curve: Mathematical equations commonly used to describe the shapes of characters in electronic typography. The Bézier curve was named for Pierre Bézier, a French computer scientist who developed the mathematical representation used to describe that curve.

Bit: A contraction of BInary digiT, this word signifies the smallest unit of data a computer holds, and represents a two-way choice like on or off, or black or white.

Bitmap: A grid of individual dots or pixels that make up the graphic display. Each pixel (or picture element) corresponds to bits in the computer's memory.

Bitmap font: A character set created by turning on or off individual dots in a rectangular grid of dots.

Bitmap window: The window used to view and edit character bitmaps.

Blend: To merge two different font shapes to create a third.

Bowl: The curved strokes that either partially or fully enclose the counters of a letter (such as those in the lower case o, b, d or e, or the upper case B, C, D, etc.). The bowl usually refers to the main circular portion of a character, and not the subordinate curved loops or flourishes that may be part of ascenders or descenders.

Brackets: The curved or diagonal shapes that connect a serif to a letter stroke. These forms are also known as fillets. Brackets may be generous or reserved; they may taper to the midpoint or end of a serif. Not all serifs have brackets.

Byte: A unit of data consisting of a small number of bits; usually a byte equals a series of eight bits and signifies a character.

Caching: The process that saves bitmaps in memory or on the printer's hard disk in order to minimize the amount of time spent generating bitmaps. The first time a particular letter is imaged, its bitmap is generated and cached. Subsequent uses of that letter can use the cached version for faster printing.

Cap height: The height of uppercase letters in a typeface.

Cap line: The imaginary line which represents the uppermost part of capital letters and some characters' ascenders.

Character: A symbol in writing. A letter, punctuation mark, or figure.

Character label: The portion including a character's name located above the character slot in the Font Window.

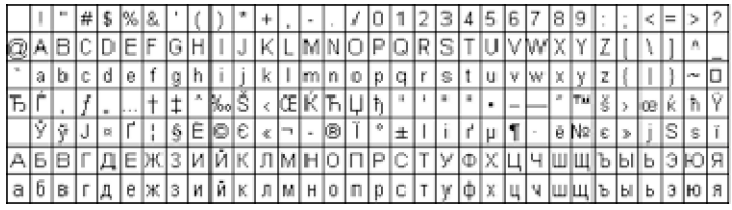
Character set: The characters, symbols, and numbers that make up one single font.

Character slot: The boxed area enclosing a font character in the Font Window.

Clipboard: The place where the most recent cut or copied image is stored.

CMap (character map): A table relating an encoding to a set of internal computer codes. For instance the computer may use the numbers between 1100 and 1356 to represent the characters in a font. When it needs character number 1234 it looks at the CMap table to find the corresponding code, which, in turn, directs it to the appropriate glyph.

Codepage: A 256-character portion of the Unicode encoding table (because that's how much we can address with one byte of data). The Russian codepage, for instance, contains the characters used in writing Russian:



Color: The visual tone or texture created by a block of type on a printed page. A typeface is considered to be most readable when the regular text setting produces an even shade of grey. Darker or lighter spots in text (caused by darker or lighter letters) are believed to slow reading and create fatigue for the reader.

Composite characters: The characters that have no outline, but link to other font characters. Good example of composite characters are accented characters, like 'Á', 'å' or 'ñ'.

Condensed: Characters which are narrowed to fit into a compact space. A properly condensed character should fit into a smaller space without making it too thin or reducing the character's height.

Contrast: The difference in width between the thick and the thin strokes of a letter. Historically, contrast was created by writing letters with a calligraphic pen (a pen with wide and narrow flat sides). Examples of high contrast typefaces include Bodoni and Didot. In general, sans serif faces (such as Helvetica) have low to medium contrast.

Counter: The negative space inside a letter. A counter can be fully enclosed (as in the lower case o) or partially enclosed (as in the lower case c, e and n). The shape and size of a counter can affect the legibility of a letter.

Crossbar: The horizontal stroke across the stem of the lower case letters t and f is called a crossbar. The angle, position and endings (terminals) of a crossbar may be unique, identifying features of a typeface.

Crosshair: The cross-like shape the pointer assumes when certain drawing tools from Fontographer's tool palette are selected.

Demagnified move: A special feature of Fontographer which allows precise point adjustment without requiring a zoom-in.

Descent / Descender: A font's maximum distance below the baseline.

Descent line / Descender line: The descent line marks the bottom of the lower case letters such as p, q, y, etc. An descent guideline is automatically drawn at the vertical position specified when the font is created, and may be changed by using the Font Info command from the Element menu. Characters should not normally appear below the descent line.

Dialog box: A window that displays when the computer needs more information from the user.

Discontinuous: Not adjacent to each other (as in discontinuous characters).

Display type: Display type is generally used for headlines or advertisements and is meant to attract attention. Display type is bold and heavier than text type because it is used in larger point sizes. More often, display type is highly stylized to the point where it is unreadable in small point sizes.

Downloadable font: A font that can be temporarily stored in the printer's memory.

Downloading: The process by which an outline font file is sent to a PostScript printer.

Dots Per Inch (dpi): The measure of resolution for a video monitor or printer. High-resolution printers contain usually at least 1000 dpi. Laser printers typically have a resolution of 300 dpi; monitors usually have 72, 75, or 90 dpi.

Ear: A small stroke that projects from the upper bowl of the lower case letter g, usually on the right-hand side. The ear is a feature that usually appears in serif typefaces. However, certain sans serif designs, especially grotesque or humanist sans serif, also include an ear or ear-like projection.

Encoding: The linear arrangement (also called the encoding vector) of a script. Alphabetical order, ABCDEFGHIJ…, is the encoding of the English alphabet. In earlier days (and on many older computers) ASCII (American Symbolic Code for Information Interchange) was the standard symbol encoding for computing.

Em: A unit of measure, which is the square of a face's point size. Traditionally, the width of a face's widest letter, the capital "M." For instance, if the "M" is 10 points wide, an em is equal to 10 points.

Em space: A space equal to the width of a typeface's point size. Often used for paragraph indentations. Traditionally, the em space was created by non-printing blocks of metal used to add space between words.

Em square: Each character is described in terms of a rectangle called an em square, or the UPM (Units Per eM) size. The em square is the basis of all font dimensions. It defines the font height and the coordinate grid on which the glyphs are drawn. The em square is divided into font units. Typically, the em square equals 1000 units. When the font rasterizer produces an image of a letter at, say, 10 point, then the 10 point will correspond to the 1000 units. If your capital letter H is 700 units high, its physical size at the 10 point font size will be 7 point.

This square is so named because historically, it used to be as wide as the letter "M" and the same height.

Em units: Measuring units in Fontographer whose size is relative. The em square can be visualized as being divided up by horizontal and vertical grid lines that result in box-like units of equal size. Em units are relative to the size of the em square and are not measured in points.

EPAR (Embedding Permissions and Recommendations): An extra OpenType font table that contains the EULA, Recommendations, and Personalization data.

EULA (End-User License Agreement): In the proprietary software industry, an end-user license agreement or software license agreement is the contract between the licensor and purchaser, establishing the purchaser's right to use the software (font). The license may define ways under which the copy can be used, in addition to the automatic rights of the buyer.

Expanded: A typeface whose letters have been made wider without visually adding weight.

Extended: A typeface whose letters are stretched (or expanded) horizontally while still retaining their original height.

Extenders: The parts of a letter that extend above the x-height or below the baseline. Both ascenders and descenders are letter extenders.

Eye: The enclosed space inside the upper half of the lower case letter e. The eye of the e must be large enough to avoid filling with ink when printed, even at small size. Otherwise, the e may be confused with a lower case c or o, or the number zero (0).

Face: A face (typeface) is a complete set of characters that share a similar appearance. Typical methods of categorization consist of measures such as thickness of stroke, angle of the stroke, roundness of letterforms, and many other dimensions that lie beyond the scope of this guide. Fontographer was the first consumer font editor that contributed to the popularization of type design outside of the highly-specialized large font foundries. Before the advent of Fontographer, the number of typefaces available for Windows or Mac OS was limited.

Family: All the type sizes and styles of one typeface. A complete character set of a font. The group shares a common design but can differ in attributes such as character width, weight, and posture (i.e., Roman vs. Italic). A typical computer family unit frequently contains four fonts – Roman, Italic, Bold, and BoldItalic –in all sizes.

Fill: In Fontographer, the degree of black within characters. (You can only specify the degree of fill for Type 3 fonts.)

Flex: A means of automatically suppressing small details, such as cupped serifs, that would print poorly at small sizes. At large sizes or high resolutions, the details are automatically reinstated. (Applies only to Type 1 fonts.)

FOG format: The internal format of Fontographer databases.

FON: Windows bitmap font format.

FOND (FONt family Descriptor): FONDs define the relationship between a plain Macintosh font and its styles (such as Bold, Italic, and BoldItalic). The FOND groups a family of fonts and contains the family name, the style, and size, as well as metrics information like fractional width tables and kerning tables.

FONT: Old Macintosh bitmap font size resource name. (See NFNT.)

Font: A font (derived from found, as in typefoundry) of type is a particular style of type in one body-size and one style of face. For Windows, the distinction between fonts and faces is not always straightforward. Since the distinction between face and font is not clear, the term font is used to describe both a particular size of a face, and the underlying face itself. When talking about the face in the abstract sense, the word face in its proper sense is used.

Font attributes: Characteristics which apply to the font as a whole (such as the ascent, descent, leading, etc.).

FontLab format: see *VFB format*

Fontographer format: see *FOG format*

Font Window: The graphic display of a character set in Fontographer from which individual character slots may be accessed.

Geometric Sans Serif: A subcategory of sans serif typefaces whose structure is based on regular geometric forms (the circle, square and triangle). Examples include Futura and Avante Garde.

Glyph: A graphic representation of a character. An “A” may appear in many different ways:

They are all the letter “A” but each is a different glyph.

Grotesque Sans Serif: The first sans serif typefaces that appeared in Europe during the nineteenth century were literally called 'grotesque' for their shocking, unadorned appearance. Grotesques, or 'grots', as they were also called, differ from later sans serif typefaces in that they are less unified and less systematic in their structure. In the United States, grotesque typefaces were also called gothic (for example, News Gothic, Franklin Gothic, and Trade Gothic).

Hints: Information embedded to enhance the appearance of characters printed or imaged at low resolutions (72-600 dpi). ATM and TrueType can take advantage of hints to render more uniformly shaped screen fonts across the character set.

Hints layer: The layer in the character’s outline window that displays hint information.

Hook: The descender on an upper/lower case J/j. If the lower case t has a curved base, the ending may also be referenced to as a hook.

Humanist Sans Serif: Sans serif typefaces influenced by humanist calligraphic writing. The early humanist sans serif (created in the first half of the twentieth century) are usually based on the proportions of ancient Roman inscriptions. Contemporary humanist sans serifs tend to have modern proportions (optically even widths), but calligraphic influence may still be seen in wider apertures and/or angled stroke endings.

INF (Information) file: a text file that contains information about a PC Type 1 font.

Italic: Best used to set off quotes, special phrases, and foreign words, italic letters have a redesigned structure that allows them to slant to the right. The first italic type was designed by Aldus Manutius in AD 1501 and was based on the handwriting style of that time.

Join: The intersection of two or more letter strokes, such as between a bowl and stem.

Justified text: Text that lines up at both the left and right margins. Also known as fully justified.

Kerning: Moving pairs of letters either closer together or farther apart to adjust and improve the space between them.

Kerning pairs: Combinations of character pairs where the space between them has been modified to improve readability.

Keystroke: A single pressing of a key on the keyboard.

Layers palette: The window of layers within the Outline Window.

Leading: The space, measured from baseline to baseline, added between successive rows of text in a document.

Left justified: Type that is aligned with its left margin. Also called flush left.

Leg: The descending lower right strokes of the upper case R and the upper/lower case K/k are called legs. The tail of a Q may also be called a leg if it extends to the lower right.

Link: The small, usually curved or diagonal stroke that connects the upper bowl and lower loop of a double story g. This feature may also be called the neck.

Linotype 100/300: High-resolution typesetting systems.

Lobe: The enclosed parts of the upper case B, R and P are called lobes.

Loop: In a double story g, the loop is the enclosed or partially enclosed lower curve that descends below the baseline. This term may also refer to enclosed or partially enclosed extenders on italic or script letters, or any other curved flourish.

Megabyte: A measuring unit; 1,048,576 bytes; denoted by the letters M or MB.

Mean line: The top (imaginary) point of all lowercase characters without ascenders. Also called x-height.

Metrics: Font information such as ascent, descent, leading, character width, and kerning.

MMPC2MAC: Macromedia PC to Macintosh font conversion utility. Converts files created for the Macintosh on a PC into Macintosh font files ready for installation.

Monospaced type: Like typewritten characters, these all have the same width and take up the same amount of space. Use of this type allows figures to be set in vertical rows without leaving a ragged appearance (as opposed to proportional type).

Multiple master font: A special type of font format that is an extension of the Type 1 font format. Multiple master fonts contain several font styles, called master fonts, in one font file. A program that uses multiple master font can not only select one of the master fonts, but it also can select an intermediate design created by the linear interpolation of the master fonts.

Neo-Grotesque Sans Serif: Neo-grotesques are grotesque sans serifs that have been adapted into new type designs with greater structural unity and visual refinement. The rise of the Bauhaus and Modernism in the 1920's emphasized simplicity and visual purity as design ideals; neo-grotesques evolved in sympathy with these aspects of the streamlined 'machine age'. Examples of neo-grotesques include Helvetica and Univers.

NFNT (New FoNT): Macintosh terminology for the part of a Type 1 or TrueType Macintosh font that contains the bitmap font.

Oblique: A right-slanted version of a Roman typeface without changes to the letter's design. Often confused with Italic.

Old Style: Characterized by variations in stroke width, bracketed serifs, high contrast, and a diagonal stroke. Some popular Old Styles include Garamond, Janson, and Caslon.

OpenType: OpenType font format, jointly developed by Microsoft and Adobe. OpenType fonts can be TrueType-flavored (or OpenType TT) and PostScript-flavored (or OpenType PS). Both are Unicode-encoded and support special features like swashes, contextual forms, ligatures etc.

Outline font: A font created by drawing the outlines of each character. A PostScript font is an outline font.

Outline layer: The layer in the Outline Window used in editing a character's outline.

Outline Window: The window that displays a characters points and paths, used for editing character outlines.

Overshoot: The degree to which a letter stroke extends over the cap-line or x-height.

Path: A sequence of points which may be connected, open, or closed.

Permanent font: A font which resides in a PostScript printer until the power is turned off.

PFB (Postscript Font Binary) file: A binary file that contains the glyph outline information for a PC Type 1 font.

PFM (Postscript Font Metrics) file: A binary file that contains the metrics information for a PC Type 1 font.

Pica: A unit of typographic measurement equal to 0.166 inches or 12 points.

PICT: A Macintosh graphics format that Fontographer and other graphic and page layout programs use.

Pixel (PICture ELement): Square dots that represent the smallest units displayed on a computer screen. Typical monitors display about 72 pixels per inch. Characters and graphics are created by turning pixels on or off.

Point: A unit of typographic measurement equal to approximately 1/72 inch (0.01383 inches).

Point size: A letter's type size is measured by its point size. The point system of type measurement was invented in 1737 by Pierre Fournier, a Parisian typefounder. Fournier's unit of measurement was 0.349 mm. In modern desktop publishing, a point is defined as 1/72 of an inch – this point is sometimes called the PostScript point. The point size always corresponds to the em square. When a font is set at 10 point, its em square (typically 1000 units) will be scaled to 10 point. If your capital letter H is 700 units high, it will be 7 point high.

Pop-up: A menu (also referred to as pull-down or drop-down) that appears in a dialog box or in a main menu when related information is selected.

PostScript: Adobe System's page description language. Programs like Macromedia FreeHand use PostScript to create complex pages, text, and graphics on-screen. This language is then sent to the printer to produce high-quality printed text and graphics.

POST resource: Macintosh terminology for the part of a Macintosh font that contains an Adobe Type 1 font.

Preview: A mode for editing or viewing a character which shows a filled outline.

Printer font: A font that permanently resides in the printer.

Proportionately spaced type: Type whose character widths vary according to the features of the letters (as opposed to monospaced type).

RAM: Random Access Memory. The computer printer's temporary place for storing data. When the computer or printer is turned off, the information in RAM is erased.

Rasterization: The process of converting outlines into bitmaps. The outlines are scaled to the desired size and filled by turning on pixels inside the outline. (See *pixel*.)

Reference: An image that refers to an original character; a composite.

Rendering: The actual placement of rasterized pixels on the monitor's display. Refers both to graphic objects and type, particularly for fonts using hints. Also called rasterization.

Resolution: The number of dots in an image's screen display or printed output. A monitor's resolution refers to the number of pixels per linear inch. Printed resolution refers to dots per linear inch. (See *Dots Per Inch*.)

Right justified: Type aligned with its right margin. Also known as flush right.

Sans serif: Typefaces without serifs are literally called 'sans serif'. Within this broad category there are four main sub-classifications: grotesque, neo-grotesque, geometric and humanist. Although sans serifs existed as early as 1816, the Modernist design movement of the 1920's popularized these designs. This manual's section headings are sans serif. Sans serif type is generally considered more modern, while serif type is considered more readable. In the United States, sans serif letters have been called gothic, which in Europe refers to black letters. Helvetica is an example of a sans serif face.

Scale: To change the size of a character or image by altering it proportionally.

Scan: To digitally capture an image and save it in a format that can be manipulated or altered from within a computer application; the image can be autotraced in Fontographer, thus creating a character with editable outlines.

Screen font: Bitmap fonts used for screen display.

Script: Letters are joined and should not be confused with cursive, which are not connected. Since script is difficult to read, its use should be limited to a few lines at a time. Early script typefaces were developed in the sixteenth century, and were based upon formal cursive handwriting.

Scroll bar: The window bars containing arrows that allow the document to be moved so that other parts of it become visible.

Serif: A small form that has been added to the ends of larger letter strokes. There are many different types of serifs, but in general, the main groups are pointed, hairline, square/slab or wedge/triangular. Serifs may be subtle or pronounced and bracketed or unbracketed. Serif ends may be blunt, rounded, tapered or pointed. A serif typeface is one that has any type or style of serifs. Typefaces without serifs are called sans serif.

Set-width: The width of a letter and its surrounding space; the space needed to set a line of text in a specific typeface. Some programs have tracking to adjust the typeface to make it set looser or tighter. Also known as advance width.

sfnt resource: Macintosh terminology for the part of a Macintosh font that contains a TrueType font.

Sidebearings: The distance between the origin and the left edge of a character (left sidebearing) and the distance between the width line and the right edge of a character (right sidebearing).

Skew: Creating an oblique image by transforming paths.

Spacing: The amount of unused area that exists between characters.

Spur: A small, serif-like projection at the end of certain letter strokes, such as at the base of the upper case G. Spurs are more common in serif typefaces, but spur-like projections exist in many sans serifs, especially grotesques and humanist sans serifs.

Stem: The main stroke of a letter (usually vertical or diagonal). For example, the center stroke of the T, or the left-hand stroke of the E, F or L. Not all letters have a stem (for example, neither C nor S). Certain upper case letters have two stems, as in the H, N and M. In diagonal letters, the stem is the downward stroke (i.e., the left side of the A, and the right side of the V and y).

Stroke: Any letter part (such as an arm, stem or bowl) may be referred to as a stroke.

Style: A visual variation of a basic typeface used to create emphasis. Type style is important since it can attract (or repel) the reader's eye. The four basic computer styles are Plain, Bold, Italic, and BoldItalic. Other styles could be defined: Many faces have condensed and extended versions, and some have light and heavy versions. Ultra-bold is frequently used for headline text.

Style Merger: Macromedia's utility that merges Macintosh styled fonts into one font family.

Suitcase: Macintosh terminology for a file which contains information about a font or family of fonts.

Swash capitals: Uppercase letters that have flourishes added to them. Originally designed to go with Italic typefaces.

Table: A set of data defining behaviors or relationships of a font. Digital fonts contain not only the drawings of their characters, but also information about how those characters should behave. Information about the spacing on each side of a character (metrics), how close particular characters should be to each other (kerning), CMaps and many other things can be kept in tables in a font.

Tail: The diagonal stroke of the upper case Q and the descenders on the lower case j and y. The legs of the K or R could be called tails if they are flowing/calligraphic in style.

Template layer: The layer of the Outline Window where background images and scans are inserted.

Terminal: When not finished with a serif, the shaped end of a letter stroke is called a terminal. Terminals may be angled, flared, rounded, or formed into circular/teardrop balls.

Textbox: Within a dialog box any rectangular outline that includes text.

Text type: Text type is used for larger masses of text and should be highly readable.

TTF: see *TrueType*

Tool palette: The collection of drawing tools in the outline and bitmap windows.

Tracking: The overall letterspacing in text. Tracking can also be used to tighten or loosen a block of type. Some programs have automatic tracking options which can add or remove small increments of space between the characters.

Transform: To alter an image by rotating, flipping, scaling, or skewing.

Transient font: A font which stays in the printer memory only until the current document is finished printing.

TransType: Universal font format converter for Mac and Windows from Fontlab Ltd.

TrueType: A font format using quadratic b-spline mathematics to describe glyph outlines. Developed and promulgated by Microsoft and Apple.

Type 1 (Adobe Type 1, PostScript Type 1): A font format using cubic b-spline mathematics to describe glyph outlines. Developed and promulgated by Adobe Systems.

Type 3 (Adobe Type 3, PostScript Type 3): Also referred to as user-defined fonts, these are non-Adobe encrypted fonts. They will not render on-screen in usual environments.

Typeface: A set of characters which share a similar appearance.

Undershoot: The degree to which a letter stroke extends under the baseline.

Unicode: A computing industry standard for the consistent representation and manipulation of text expressed in most of the world's writing systems. The latest version of Unicode consists of a repertoire of more than 107,000 characters covering 90 scripts.

Unicode range: The portion of Unicode dealing with a particular language or script. E.g. the Hebrew range, the Cyrillic range, the extended Latin range. Unicode range is not limited to 256 characters. It is usually a contiguous part of Unicode.

UPM size: see *em square*.

Vertex: A lower point where two strokes meet, as in the base of the V or N, or the central low point of the M.

VFB format: The internal database format of Fontlab products.

Weight: The measurement of a stroke's width; or, in general, the heaviness of a character or font. Common names for weights include demibold, light, and bold. Some typeface families have several weights, including ultra-bold and extra-light.

x-height: The x-height line marks the top of the lower case letters without ascenders or descenders, such as x and o. This line may be positioned anywhere you wish, since it is only a guideline. In general, typefaces with taller x-heights are perceived as larger and more readable than those with small x-heights.

Index

A

Add Fonts dialog, 339
adding guidelines, 127, 129
Adobe Type Manager, 308, 316, 338
Adobe Type One Font Format book, 405
Adobe Type One Font Specification, 392, 404
Advanced Auto Kerning, 256
Advanced Auto Spacing, 253
  spacing direction, 255
  spacing technique, 255
  which glyphs screen, 253
Advanced Tracing Options, 111
  allow curve fit errors, 112
AFM, 309
Arc Tool, 67, 69, 447
Arc Types, 67
arcs
  keyboard shortcuts for, 68
arrow keys, 19, 39, 180, 234, 381, 454, 480
ascender, 22, 128, 265
ascent, 24, 35, 160, 198, 201, 453, 459, 513
Ascent Values, 201
ascent, 128
ASCII character, 318
ASCII glyph
  representation of, 28
Asian CJK, 18, 377
assisted metrics, 245
auto curvature
  example of, 192
Auto Kerning, 226
  advanced, 256
  easy mode, 226
  technique screen, 257
Auto Spacing
  advanced mode, 253
Auto Spacing, 216
  easy mode, 218
autohinting, 398, 402, 405
Autosave
  preferences, 19, 377
Autotracing, 107

B

baseline, 25, 102
basepoint, 25, 102
basic shape tools, 55
BCPs
  for control points, 189
  for curve points, 189
  for tangent points, 191
  placing, 188
  retracting, 183, 191
Bézier control points, 45, 188
Bitmap
  enlarging, 210
  importing, 213, 459
  recalculating, 212
  reducing, 210
Bitmap Files
  Exporting EPS, 322
Bitmap Fonts, 214, 319
Bitmap Format, 321
Bitmap Information dialog box, 197
Bitmap point sizes, 320
Bitmap Window, 30, 34, 197, 450
  changing point size, 453
  eraser tool, 205
  hand tool, 204
  lock icon, 198
  magnifying tool, 207
  marquee selection tool, 205
  measuring tool, 207
  move tool, 206

opening, 199
pencil tool, 201, 205
Recalculate from outline button, 35, 198
straight line tool, 204
switching glyphs, 210, 453
Blending fonts, 150, 388
Blue Zones, 403

C

Calligraphic Pen tool, 57
Cap and Join, 470
Change button, 88
Change Path Direction, 163
changing width, 99
Character Information dialog, 161
character mode, 26
Character Palette, 412
character(s)
viewing, 437
Choose letters option
in advanced autokerning, 257
clearing kerning, 237
click-selecting objects, 47
Clockwise, 163
Cloning, 173
Closing windows, 72
Composite Glyphs
creating, 93
constraining tools, 69
control point tools, 59
Control Point(s)
hiding, 187
Copy Component command, 93
corner point tool, 60
Corner Point(s), 167
Corner Tool, 444
Counterclockwise, 163
Creating
hints, 400
ligatures, 97
new fonts, 105
stroked fonts, 132
variable weight glyphs, 148
Cursor Key Distance, 19
curve point tool, 59, 166
Curve Point(s), 166
Curve Tool, 444
Custom Encoding, 268, 407

D

Database font, 86
Decimal viewing, 437
Decompose Component command, 96
Default Preferences, 387
descender, 22, 23, 128, 265
descent, 24, 35, 128, 201, 453
deselecting objects, 45
Design parameters, 262
Dialog boxes
placement, 21
setting preferences for, 21
Dialogs
preferences, 385
Don't kern lowercase to uppercase
in advanced autokerning, 259
dragging, 42
drag-selecting objects, 46
drawing layers, 50
DSIG preferences, 386
Duplicating, 173

E

editing behavior preferences, 19
Editing hints in outline window, 399
editing preferences, 380
EEULAA, 274
em square, 22
Embedding, 272, 273
Encoding
custom, 268
original, 268
preferences, 379
Type 1 Adobe standard, 269
End Caps
butt end caps, 135
round end caps, 135

square end caps, 135
enlarging an image, 40
enlarging glyphs, 40
EPAR, 274, 281, 459
exporting, 283
importing, 283
EPS
exporting, 322
importing, 159, 459
Equivalence Classes, 245, 247
Eraser Tool, 205, 451
EULA, 274
Examine Average Distance
in advanced autokerning, 258
in advanced autospacing, 255
Examine Minimum Distance
in advanced autokerning, 258
in advanced autospacing, 255
Examine Weighted Distance
in advanced autokerning, 258
in advanced autospacing, 255
Exceptions screen
in advanced autokerning, 258
Excess Points
removing, 179
Expand Stroke, 136, 471
Exporting EPAR, 283
Exporting metrics, 238

F

feature definition language, 347
Fill Tint
viewing, 438
viewing by, 28
Fill(s)
even/odd, 162
normal, 163
winding number, 163
Fit in Window option, 41
flex, 315
Flip Tool, 62, 445
Flip Transformation, 114
font blending, 150, 388, 417
Font Families, 326
Macintosh, 330
Font Formats, 79, 299
Font Info, 467
advanced mode, 262
Credits, 270
Dimensions, 265
Encoding, 267
Licensing, 272
Names, 262
Font Info dialog box, 298
font label(s), 27
font preview window, 340
Font vendor, 271
font vendors, 271, 411
font window
selecting glyphs in font window, 38
view by, 437
Font(s)
installing in Windows, 340
Font(s)
adding bitmap sizes, 320
database, 86
families, 325
generating, 303
generating cross-platform, 305
generating families with Style Merger, 331
generating for Macintosh, 307
generating for Windows, 316
interpolating, 389
new, 79, 81
non-Roman ID range, 410
OpenType, 305
script systems, 410
spacing, 219
suitcases, 319
window, 436
Fontlab Studio 5, 490
Fontographer Metrics File, 239
Fonts control panel, 340
Fonts preferences, 379
freehand drawing tool, 56, 140, 148
Freehand Tool, 443
full font name, 264

G

Generate digital signature
  preferences, 386
Generating
  multiple master fonts, 432
  OpenType fonts, 305
  other PostScript Type 3 formats, 314
  PostScript Type 1 fonts for Macintosh, 308, 310
  PostScript Type 1 fonts for Windows, 316
  PostScript Type 3 fonts for Macintosh, 312
  PostScript Type 3 fonts for Windows, 318
  TrueType fonts, 311
generating font(s), 88
Ghost Hint, 402
glyph class, 349
Glyph Display
  choosing new glyph, 232
  in metrics window, 231
Glyph Information, 469
Glyph Information dialog, 268
glyph name, 349
glyph properties, 440
glyph slots, 28
glyph(s)
  changing in outline window, 33
  information, 469
  select discontinuous, 39
  selecting, 38
  selecting a single glyph, 38
  selecting range of, 38
  selecting with arrow keys, 39
  spacing, 219
  undefined, 29
  variable weight, 148
  vertical spacing, 236
  viewing in outline window, 40
Grid Spacing, 20
  preferences, 381
Guidelines
  adding, 127
  changing, 52
  display, 203
  hide, 203
Guides layer, 52

H

Hand Tool, 42, 55, 442, 451
  in bitmap window, 204
Hexadecimal viewing, 437
Hinting
  automatic, 398
  changing direction, 399
  definition, 396
  flipping direction, 399
  fonts, 392
  horizontal stem, 400
  illustration, 397
  manual, 398
  vertical stems, 401
Hints
  ghost, 402
  removing, 400
Hints layer, 53
Horizontal Size
  don't change option, 83
Horizontal Stem
  hinting, 400
Hypercube, 419

I

image
  enlarging, 41
  reducing, 41
Importing
  bitmaps, 459
  EPAR, 283
  EPS, 459
  EPS images, 159
  metrics, 459
  outline path data, 159
Importing metrics, 237
info bar, 26, 31, 32
Input menu, 412

installing
  fonts in Windows, 340
installing font(s), 89
interpolating
  fonts, 389

K

Kern Distance, 37
Kerning
  definition, 454
  tables, 237
Kerning Assistance, 249
  dialog, 249
  link all, 250
Kerning Direction
  in advanced autokerning, 257
Kerning Pairs
  printing, 293
Kerning Speed
  in advanced autokerning, 258
Kerning Tables
  storing, 237
Kerning technique
  in advanced autokerning, 258
Key Map
  printing, 292
keyboard shortcuts
  arcs, 68
keystroke mode, 26
Knife Tool, 58, 443

L

language system, 352
Largest Pairs First
  direction to kern, 258
  in advanced autokerning, 258
Layer(s)
  drawing, 50
  guides, 52, 448
  hints, 53, 448
  outline, 50, 448
  palette, 54, 448
  template, 51, 448
  visible, 203
leading, 23
Left Sidebearing, 37
  viewing, 438
  viewing by, 27
license, 272
ligatures, 97
Line gap, 23
Line Joins
  bevel joins, 135
  miter joins, 135
  round joins, 135
Linespacing
  preferences, 19
Lock Icon, 33, 198
locking glyphs, 33
Logo(s)
  changing width, 159
  scaling, 159
lookup, 346, 355

M

Mac OS X
  installing OpenType fonts, 337
  installing PostScript fonts, 336
  installing Windows fonts, 337
Macintosh
  font families, 330
  installing PostScript fonts, 336
  removing fonts in Mac OS X, 341
Magnification
  command keys, 210
  decreasing, 449
  from the View menu, 210
  increasing, 449
  options, 41
Magnifying Tool, 66, 446, 452
  in bitmap window, 207
  using, 40
Marquee Selection Tool
  in bitmap window, 205
Marquee Tool, 451
Measuring Tool, 66, 446
  constraining, 70
  in bitmap window, 452

menus, 71
Metrics
  importing, 459
Metrics Assistance
  link all, 247
  link to base, 246
Metrics Display
  activating a cell, 234
  moving from cell to cell, 234
Metrics Window, 30, 36
  glyph display area, 228
  opening, 36
  show kerning icon, 229, 456
  spreadsheet area, 228
  text link, 229, 456
Most Common Pairs First
  in advanced autokerning, 258
Most Common Then Largest
  in advanced autokerning, 258
Move selection distance, 381
Move Tools, 35, 451
  in bitmap window, 206
Move transformation, 115
Moving
  by dragging, 42
  template images, 108
Multigon Tool, 55, 442
Multiple Master Font(s), 413
  creating one-dimensional, 414
  FOND name, 428
  Greg Thompson article, 417
  NFNT, 428
  suggested name values, 430
Multiple Points
  selecting, 169

N

naming font(s), 85
naming glyphs, 358
New Font, 79, 81
New Hints
  creating, 400
Next Bitmap Point Size, 211, 453
Next Glyph Item, 449, 453
Next Point In Path
  selecting, 182
NFNT Format, 321

O

objects
  selecting, 45
Octal viewing, 437
Offset, 453
Offset Value, 202
ont version, 270
Open Font, 77
opening
  bitmap window, 34
  glyph's outline window, 31
  metrics window, 479
  windows, 30
OpenType features
  preparing, 358
OpenType fonts, 305, 343
  generating, 357
  importing, 356
  Latin features, 371
OpenType layout features, 306, 344
OpenType Options, 306
OpenType PS, 299
OpenType TT, 300
OpenType-CFF, 299
origin line, 25, 52
Original Encoding, 268
OTF, 299
Outline layer, 50
Outline vs. Stroked Glyphs, 132
Outline Window, 31
  changing glyphs, 33
  editing hints, 399
  next glyph item, 449
  previous glyph item, 449
  viewing glyphs in, 40
Oval Tool, 56
Oval Tool, 69, 442
Overwrite existing files option, 88

P

Pairs To Kern First
  in advanced autokerning, 257
palettes
  using, 54
Pasting Images
  into template layer, 107
Path Direction indicator, 33, 165
Path(s)
  closed, 162
  correcting direction, 164
  filling, 162
  open, 162
  reversing direction, 165
  selecting parts of, 48
Pen Tool, 57, 443
Pencil Tool, 205, 451
Perspective Tool, 65, 446
Point Display, 21
  preferences, 383
Point Information dialog, 182, 391
Point Mapping
  multiple master fonts, 427
Point Size
  bitmap, 197, 211, 308, 310, 320
Point(s)
  corner, 167
  excess, 179
  moving with mouse, 180
  tangent, 168
Pointer Tool, 55, 442
positioning lookups, 364
PostScript, 301
PostScript files
  printing, 291
PostScript font name, 264
PostScript Font(s)
  installing on Macintosh, 336
PostScript Type 1 Fonts
  opening, 405
PostScript Type 3
  generating for Macintosh, 312
  generating for Windows, 318
  generating other formats, 314
Preferences
  automatically fit glyphs to windows, 385
  autosave, 19, 377
  cursor editing behavior, 181
  dialog boxes, 21
  editing behavior, 19, 184
  grid spacing, 381
  linespacing, 19
  move palettes with windows, 385
  move selection distance, 381
  point display, 21, 186
  remember dialog box positions, 385
  remember dialog box values, 385
  setting, 18
  snapping distance, 381
  snap-to-point, 381
  undo, 18
  windows, 21
pressure sensitive tool, 57
Preview, 43, 463
Previous Bitmap Point Size, 211, 453
Previous Glyph Item, 453
Previous Point In Path
  selecting, 182
Print Sample dialog, 286, 460
Printing
  all glyphs, 287
  glyphs, 460
  kerning pairs, 293, 460
  key map, 292, 460
  PostScript file, 291, 460
  sample file, 290, 460
  sample glyphs, 294
  sample text, 460
  selected glyphs, 288
  specific text samples, 289

Q

quitting Fontographer, 72

R

Recalculate from outline, 35, 198

recently used font(s), 78
recommendations, 281
Rectangle Tool, 55, 69, 442
Redo, 71, 209
reducing an image, 40
reducing glyphs, 40
Reference Glyph(s)
  changing, 95
Remove Overlap command, 164
Removing
  Macintosh font, 341
  Windows font, 341
Removing Fonts, 341
Removing Hints, 400
Resizing
  template image, 108
Restore Defaults, 387
reverting font, 87
Right Sidebearing, 37
  viewing, 438
  viewing by, 28
Rotate Tool, 61, 444
Rotate transformation, 117

S

safe zone bottom, 24, 265
safe zone top, 24, 265
Sample Glyphs
  printing, 294
Sample Text
  printing, 287
Saving font(s), 86
Scale, 118
Scale Tool, 63, 122, 445
script, 352
scroll bar, 27
scroll bars, 31, 36
scrolling
  with hand tool, 42
Searching glyphs, 440
Selecting
  all the points, 169
  entire path, 169
  glyph(s), 38
  objects, 45
  parts of a path, 48
  path or points, 49
Set Basepoint dialog, 102
Set Width
  change width by % value, 477
  change width by em units, 477
Setting
  basepoint, 102
  guidelines, 127
  preferences, 18
  stroke attributes, 133
shift-selecting objects, 47
Show kerning icon, 36
Show Points option, 44
Sidebearings
  viewing by, 27
Size
  horizontal, 83
  vertical, 83
sizing glyphs, 40
Skew glyphs, 91
Skew Tool, 64, 120, 445
Skew transformation, 120
Smooth outline, 19, 380
Snapping Distance
  preferences, 381
Snap-to-Point
  preferences, 381
sounds, 378
Spacing
  key commands, 455
Spacing Direction
  in advanced autospacing, 255
Spacing Technique
  in advanced autospacing, 255
Special Keys, 480
Spreadsheet Area
  in metrics window, 234
Straight Line Tool, 56, 204, 443, 451
  constraining, 204
Stroke Tint
  viewing, 438
  viewing by, 28

Stroke Weight
viewing, 439
viewing by, 28
Stroke(s), 469
expanding, 471
style linking, 327
Style Merger, 330
creating families, 331
substitution lookups, 359
Sutcliffe
Judy, 171
Switching Glyphs
in bitmap window, 210
Symbol Encoded Windows Fonts, 318
System 7.1
WorldScript, 410

T

Tangent point tool, 60, 168
Tangent Tool, 444
Technique screen
in advanced autospacing, 254
Template Image
moving, 108
resizing, 108
Template Layer, 51
pasting images, 107
Terminology
design coordinate, 421
master design, 419
normalized coordinates, 419
primary font, 423
Text String Notation, 230
title bar, 26
tool palette, 54
Tool(s)
arc, 69
arc tool, 67, 447
calligraphic pen, 57
constraining, 69
control point, 59
corner, 444
corner point, 60
curve, 444
curve point, 59
eraser, 451
flip, 62, 445
freehand, 443
freehand drawing, 56
hand, 55, 442, 451
knife, 58, 443
magnifying, 66, 446, 452
marquee, 451
measuring, 66, 446, 452
move, 451
multigon, 55, 442
oval, 69, 442
palette for, 54
pen, 57, 443
pencil, 451
perspective, 65, 446
pointer, 55
pressure sensitive, 57
rectangle, 69, 442
rotate, 61, 444
scale, 63, 445
skew, 64, 445
straight line, 443, 451
tangent, 444
tangent point, 60
transformation, 61
Tracing
images, 109
Transform command, 100
transformation tools, 61
TransType Pro, 330
TrueType, 300
TrueType Collection, 81
TrueType Font(s), 196
generating, 311
importing characters, 459
TrueType Fonts
opening, 406
TTC, 81
Type 1, 301
Typographic family name, 262
Typographic style name, 262

U

Underline position, 266
Underline width, 266
Undo, 71, 209
  preferences, 18
Unicode Codepoint, 469
Unicode Name, 469
UPM size, 22, 265

V

Value(s)
  ascent, 201
  descent, 201
  offset, 202
  width, 202
Variable Weight Glyphs
  creating, 148
Vertical Alignment Zones, 403
Vertical Alignment Zones dialog, 403
vertical kerning, 236, 237
Vertical Size
  don't change option, 83
Vertical spacing, 236
Vertical Stem
  hinting, 400
View By Character
  in font window, 437
View By Keystroke
  in font window, 437
View by menu, 27
  in font window, 437
View By Unicode
  in font window, 437
View menu
  changing point size, 453
  next glyph item, 453
  previous glyph item, 453
  using, 33
Viewing by
  character, 27
  code, 27
  fill tint, 28
  keystroke, 27
  sidebearings, 27
  stroke tint, 28
  stroke weight, 28
  Unicode codepoint, 27
viewing glyphs, 40
viewing modes, 43

W

Weight, 470
  changing, 82
When a path is clicked on
  preferences, 382
Which Glyphs screen
  in advanced autokerning, 256
  in advanced autospacing, 253
Width Value, 202
Width(s), 25, 453
  changing, 99
  definition, 454
  in metrics display, 37
  viewing, 438
Win Ascender, 265
Win Descender, 265
Winding Number Fill, 162
window
  preferences, 385
windows
  closing, 72
  opening, 30
  setting preferences for, 21
windows placement, 21
Windows Type 1 Font(s)
  generating options, 317
Wingdings, 318
WorldScript, 410

X

x-height, 24, 265

Z

zooming in on glyphs, 40
zooming out on glyphs, 40

Acknowledgments

Originally written and edited at Altsys and Macromedia by Katharine Green, Pete Mason, Jim Von Ehr, and Taeza Pittman.

Revised at Fontlab Ltd. by Alex Petrov and Adam Twardoch.