FontLAB

Fontographer 4.7

FONT EDITOR FOR DESKTOP PUBLISHING –
EASY EDITING OF POSTSCRIPT & TRUETYPE
USER'S MANUAL FOR MACINTOSH

Fontographer 4.7

for Mac OS® X

User Manual

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 4.7 [11/2005]

Contents

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 rival those produced by professional typographers.

Fontographer 4.7 generates ATM-compatible Type 1 fonts, as well as Type 3 PostScript fonts, TrueType fonts, PICTs 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 Macromedia 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 *Using Fontographer* manual is designed for Macintosh users only. If you have Fontographer 4.1 for Windows 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 *Using Fontographer* 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 *Using Fontographer* 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 *Using Fontographer* manual for information that will help you plan and create your fonts.

System requirements

If you are using the Macintosh version of Fontographer 4.7, you must have a Macintosh computer running Mac OS X 10.3 or later operating system with 5.1 MB of hard drive space, and at least 10 MB of available RAM. It's also a good idea to make sure you have the latest version of Apple's System update for your machine.

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/

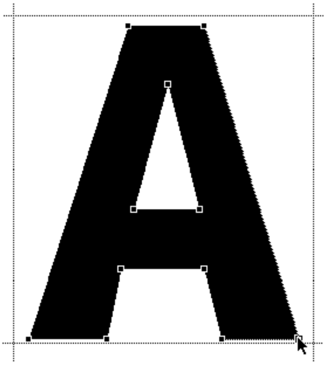
The Fontographer technical notes collection is available for download (in PDF format). Technical notes address various issues and problems concerning Fontographer 4.x:

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.

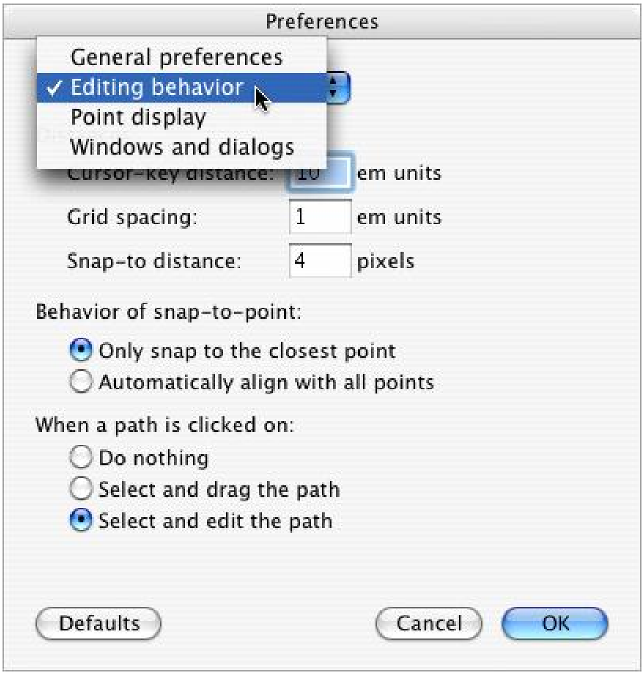


Create characters from scratch
or by autotracing a scanned image.

1

Setting preferences

Fontographer 4.7 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 11, “Expert Advice”, for more information.



Cursor-key distance: 10 em units

Grid spacing: 1 em units

Snap-to distance: 4 pixels

Behavior of snap-to-point:

- ● Only snap to the closest point
- ○ Automatically align with all points

When a path is clicked on:

- ○ Do nothing
- ○ Select and drag the path
- ● Select and edit the path

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 eight 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 101. The more undo levels you allow, the more memory you use, and the less you have for the font you’re editing.

Choose **Preferences** from the **Fontographer** menu to bring up the Preferences dialog box. The General preferences dialog box lets you select the number of undo levels that you want, and whether or not you want Fontographer to throw away undo levels when there isn't enough memory. Or you can tell Fontographer to consult you first.

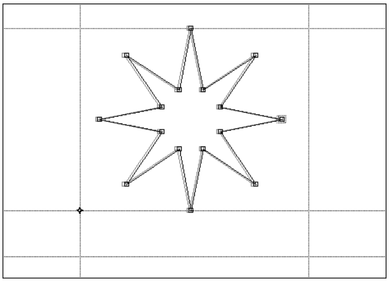
Editing behavior

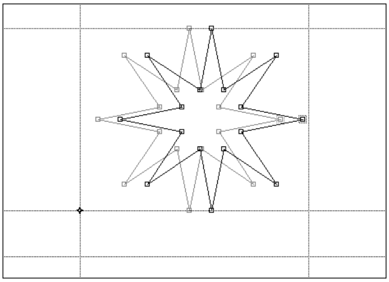
To set Preferences related to editing behavior, choose Editing Behavior from the pop-up in the Preferences dialog box.

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.

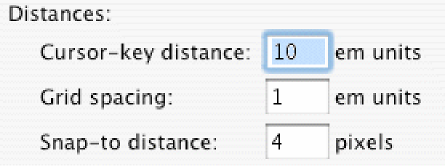




Grid spacing

There is an invisible key grid in the outline window that can be used to position characters. 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.



Distances:

Cursor-key distance: 10 em units

Grid spacing: 1 em units

Snap-to distance: 4 pixels

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

➲ ***Tip:*** Listen to your snaps by placing the *Fontographer Sounds* file (provided with Fontographer) into the same folder that holds the Fontographer application.

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

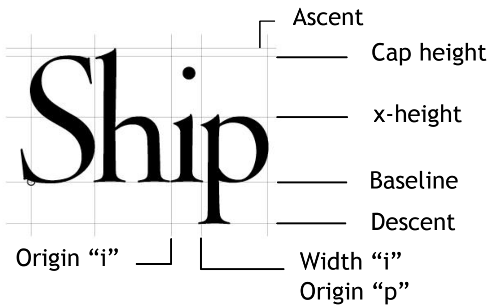
Preferences lets you set the placement of windows, remember values and positions of dialog boxes, control how characters fit into windows, and have palettes move with the windows.

For more information about setting any of the preferences, see “General preferences” in Chapter 11, “Expert Advice”.

The em square

Each character 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. 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 ascent and descent. 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 em 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.

Ascent

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

➲ ***Tip:*** Characters normally should not extend above the ascent line. Any character that extends above the ascent line or below the descent line may have its bitmap representation vertically scaled to fit between the ascent and descent. See “When should you recalculate bitmaps?” in Chapter 5, “Editing Bitmaps”, for more information.

Descent

A font’s maximum distance below the baseline is called its descent. Fontographer automatically places a descent guideline at the bottom of the em square in the outline window. The descent 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:*** Characters should not drop below the descent line or they may interfere with characters 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).

You can use ascent and descent to control the size of the em square in which your characters are drawn. Fontographer’s default values are: ascent 800 and descent 200. You may change these proportions as needed. The default values provide a 1000-unit-high em square.

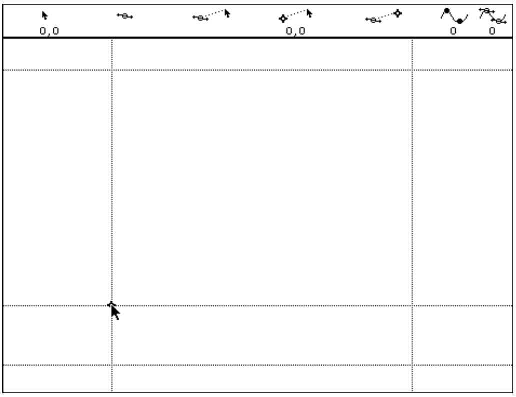
x-height

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:

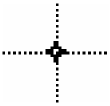


Baseline

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 star-shape at the vertical intersection of the origin line and baseline is the basepoint:



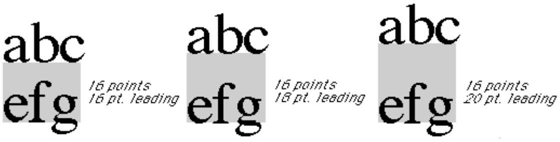
This is a movable reference point for position measurements or as an aid to line up points. When you open a new character's outline window (or display a new character in an existing window), the basepoint is positioned at location 0,0 – at the origin point of the em square. Fontographer allows you to 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 character slot. When the printer draws characters on a page, the origin line of the next character is placed on top of the width line of the current character. Widths may be zero but cannot be negative.

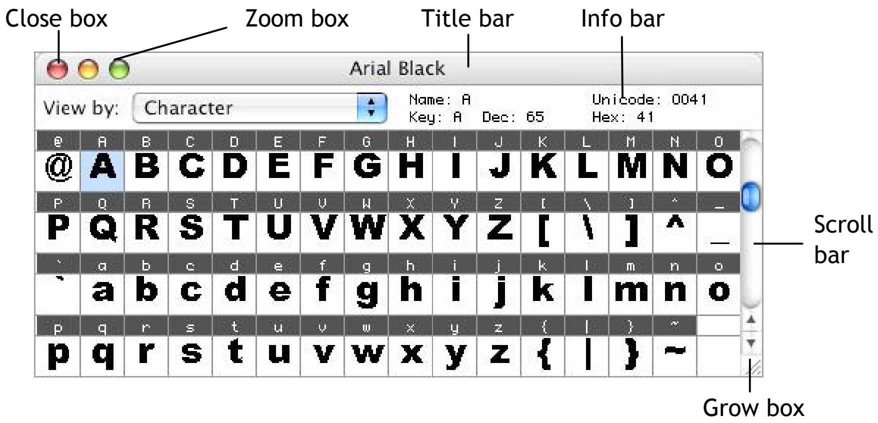
Leading

Leading is the space – measured in points – between lines of text, from baseline to baseline. It specifies how much space there is between each line. 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, leading is specified only in the screen (bitmap) font; this controls the spacing on the printer. Some applications ignore this leading value. Leading size is not used directly by a PostScript font but is used by Fontographer when calculating the default leading for bitmap generation.



Leading is the space between the baseline of one line and the baseline of the following line.

The font window



Arial Black

View by: Character

Name: A
Key: A Dec: 65
Unicode: 0041
Hex: 41

@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	

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

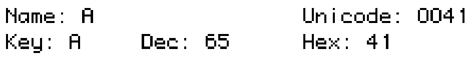
The font window also has other standard features like close, grow, and zoom boxes.

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 twelve different types of font labels. The character mode shows the symbols for each character in the Latin alphabet, while the keystroke mode shows you the keys you type to access the characters. In some cases the character symbol and the keystroke may be different.

You'll find more detailed information about font labeling information in "Character slots" on page 29.

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



Name: A
Key: A Dec: 65
Unicode: 0041
Hex: 41

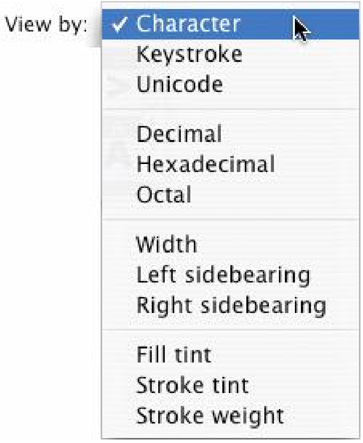
Title bar

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 characters.

View by pop-up



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

- Character displays the system character corresponding to each slot in the font window.
- Keystroke displays the keyboard sequence used to access a character.

- Unicode number is a value assigned to a character based on an international numbering system with 16-bit numeric designations for each character in every language used (or planned for use) in electronic information systems.
- Decimal code number is a number value assigned to a character 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 character based on a numbering system with a base of 8.
- Width view of a character displays the character's width in em units.
- Left sidebearing shows the distance from the left edge of a character to its origin, in em units.
- Right sidebearing displays the distance from the right edge of a character to its width line.

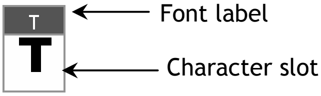
And when viewing Type 3 fonts:

- Fill tint shows you what percentage of black fills a character. 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 character.
- Stroke weight displays the weight of the character's strokes in em units.

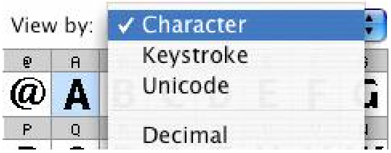
Character slots

Character slots are visible in the font window. Character slots can show the character 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 character.

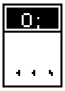
When the label is outlined, the outline window contains paths, a defined width, or a scan. When the label is black, it means the character has been changed.



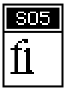
Font labels display in one of several ways. Viewing by character shows the Macintosh standard representation of that ASCII character code in the font label, while viewing by keystroke shows the keystrokes you'll need to press to type that character. Change view modes with the font window's **View by**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



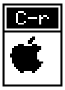
Below, the Macintosh keystroke for "ellipsis" is shown. The font label displays the **Option-;** keyboard command that is needed to type it.



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

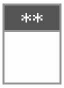


For this character the keyboard command is **Control-R**.

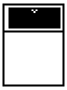


Note: Characters accessed by typing the keyboard's control keys cannot be typed from non-Apple Desktop Bus keyboards.

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



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



With the keystroke view, an empty character slot displays as shown above.

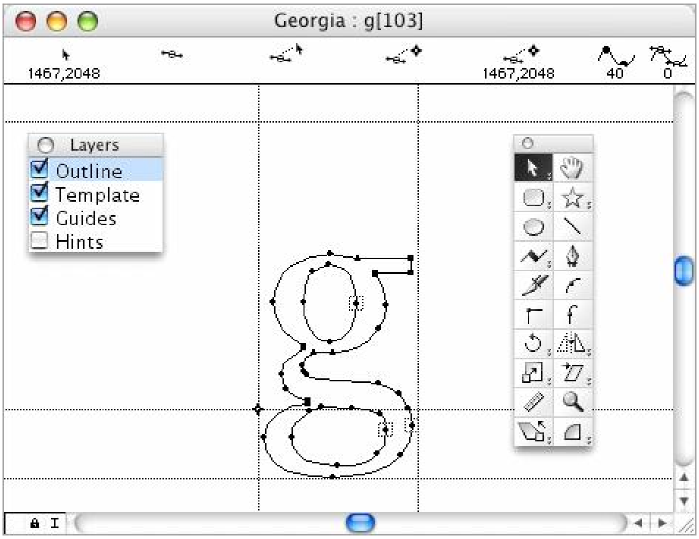
Opening windows

When you are familiar with the font window, look more closely at the characters you see in this window. There are three possible windows from which to view each character. Each window gives a different perspective of the character. 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 character editing. A character's outline reveals its filled or unfilled shape bounded by paths and points.
- The bitmap window contains a character image displayed in pixels. Changing this bitmapped image alters the screen font's appearance.
- The metrics window displays a filled character image and provides tools to modify kerning and spacing.

Any window can be accessed either from the font window or from Fontographer's **Windows** 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 character's outline window:

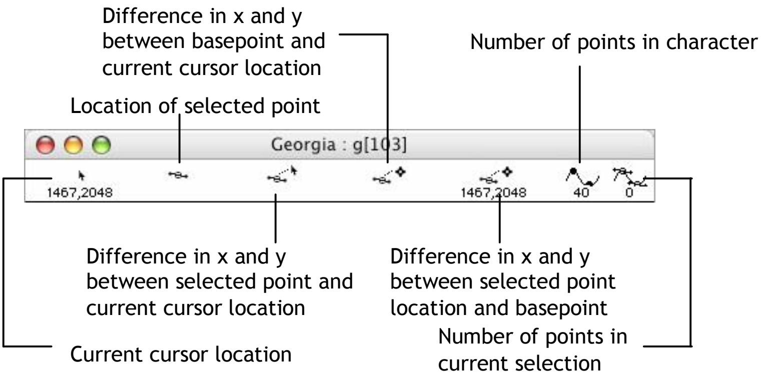
You can open the outline window in one of four ways:

- Double-click the selected character.
- Select the character and type **Command-H**.
- Select the character and press the **Return** key.
- Select the character and choose **Open Outline Window** from the **Windows** 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 character 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.

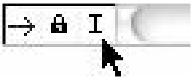
The info bar


Difference in x and y between basepoint and current cursor location
Number of points in character
Location of selected point
Georgia : g[103]
1467,2048
1467,2048
40
0
Difference in x and y between selected point and current cursor location
Difference in x and y between selected point location and basepoint
Number of points in current selection
Current cursor location

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 character'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 character window (next to the lock icon) to display or hide the info bar.



Changing characters in the outline window

Using the keyboard

You can move to another character’s window by typing the desired character 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

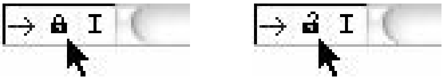
The lock icon 🔒 is operated by clicking it or by pressing **RETURN**. This toggles the icon on or off.

The lock appears black when the character is locked into position. This prevents the character 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 character’s outline window by typing the new character’s key.

Using the View menu

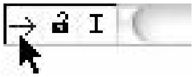
When any character is selected in an active font window, or when any outline window is open, you can choose **Next Character**, **COMMAND-]**, or **Previous Character**, **COMMAND-[**, from the **View** menu to move forward or backward in the sequence.



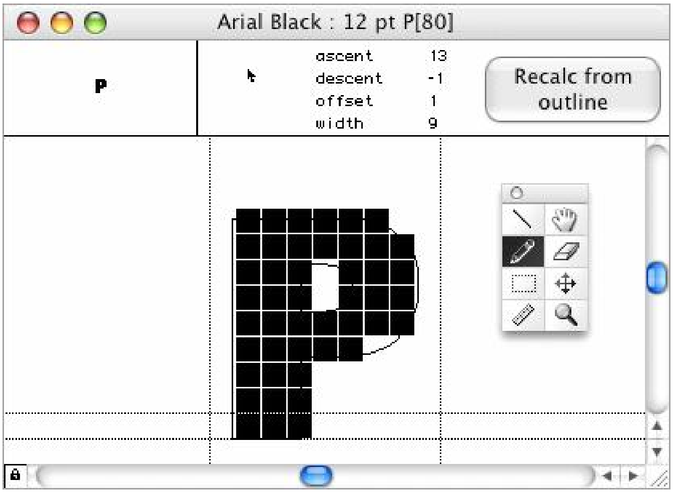
Click the lock icon to lock or unlock the window.

Path direction indicator

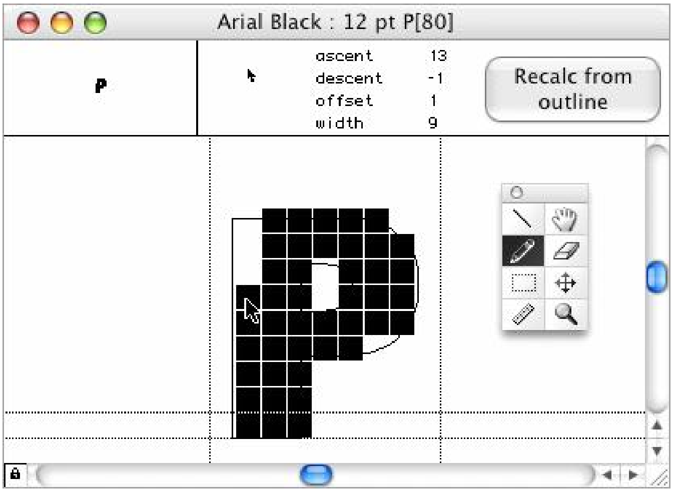
The path direction indicator tells you the direction of any selected path. You can use this feature to make sure that the paths of any character you create or edit are drawn in the necessary alternating pattern, from the outside to the inside of a character, so that the character will print with an accurate 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 character:

- Press and hold down the **OPTION** key while you double-click the desired character.
- Select the character and type **COMMAND-J**.
- Select the character and choose **Open Bitmap Window** from the **Windows** menu.

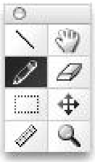
The bitmap window shows how characters look as they appear on the screen. However, bitmaps aren't normally used in printing your type or graphics, so time spent editing them will have no impact on the printed results. Therefore, tweaking and adjusting bitmaps may not be for you.

Actually, advances in programs related to type production have almost made editing bitmaps unnecessary. If you use ATM (Adobe Type Manager – a program that improves the look of screen fonts) or TrueType fonts (Apple and Microsoft's outline font format that uses 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 character 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 character, the cursor location, and the ascent, descent, offset, and width values for the character. To the right is the **Recalc 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).

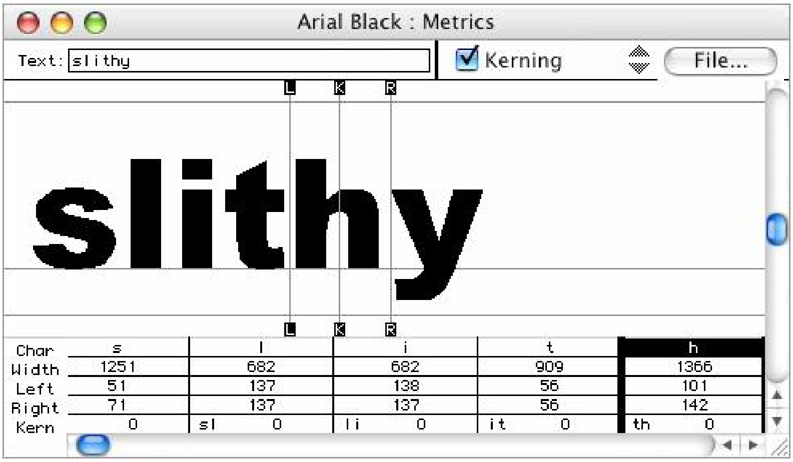
In the lower left corner you will find the lock icon for allowing or preventing access to other characters with the type of a keystroke. When locked, the current character's bitmap window is locked in place; when unlocked, the icon appears hollow and you can access other characters by typing their keystrokes.

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



Text: slithy

☑ Kerning

Char	s	l	i	t	h
Width	1251	682	682	909	1366
Left	51	137	138	56	101
Right	71	137	137	56	142
Kern	0	sl 0	li 0	it 0	th 0

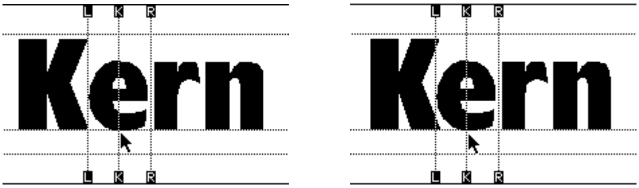
To open the metrics window:

There are three ways to open the metrics window:

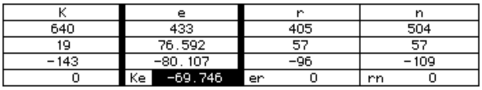
- Select a character in the font window and type **Command-K**.
- Select a character in the font window and then choose **Open Metrics Window** from the **Windows** menu.
- Type **Command-K** to access the metrics window directly from the outline window.

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 **Kerning** checkbox will show the text sample kerned; when it is deselected the sample appears unkerned. The **File** button 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.



K	e	r	n
640	433	405	504
19	76.592	57	57
-143	-80. 107	-96	-109
0	Ke -69.746	er 0	rn 0

Or type the values in the spreadsheet area.

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

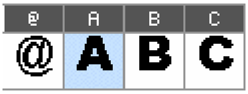
Selecting characters in the font window

Fontographer lets you select characters individually or in ranges.

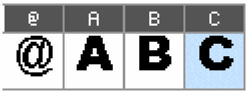
Selecting a single character:

Click a character slot with the mouse or type the character's keystrokes. The selected character 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.



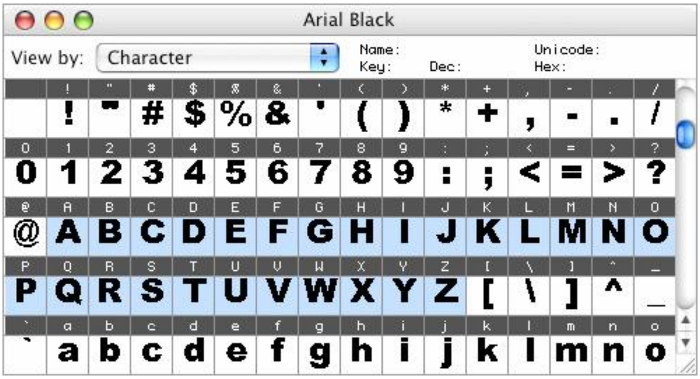
Selecting a range of characters:

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

Position the pointer on the first character in the range and drag across the others you want to include. Release the mouse button when all the characters 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 characters "A–Z" will be selected.


Arial Black

View by: Character

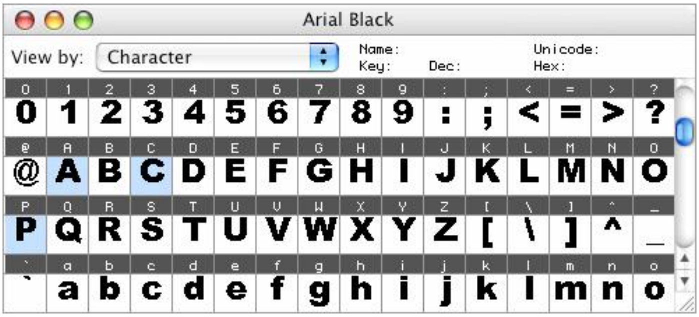
Name:
Key:
Dec:
Unicode:
Hex:
! " # $ % & ' ( ) * + , - . /
0 1 2 3 4 5 6 7 8 9 : ; < = > ?
@ A B C D E F G H I J K L M N O
P Q R S T U V W X Y Z [\ ] ^ _
` a b c d e f g h i j k l m n o

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 characters or to add additional ones. But what if the characters you want to select are not in a continuous range?

To select discontinuous characters:

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

   All three characters will be selected.


Arial Black

View by: Character

Name:
Key:
Dec:
Unicode:
Hex:
0 1 2 3 4 5 6 7 8 9 : ; < = > ?
@ A B C D E F G H I J K L M N O
P Q R S T U V W X Y Z [\ ] ^ _
` a b c d e f g h i j k l m n o

Selecting characters with arrow keys:

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

Viewing characters in the outline windows

Viewing and sizing the character

When you want to enlarge or reduce the character 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 character

Press **COMMAND-SPACEBAR** and click in the character edit area. The location you click will center in the window.

Reduce or zoom out on the character

Press **COMMAND-OPTION-SPACEBAR** and click the area that you want to center in the window.

The reduce command decreases the view of the character by zooming out from it.

Using the magnifying tool

Use the magnifying tool 🔍 to enlarge or reduce an entire character, 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:

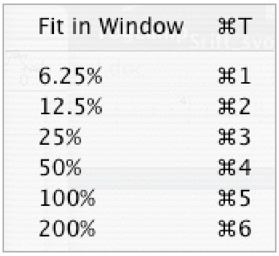
- Select the magnifying tool or press **Command-Option-Spacebar** and click the part of the window that you want to reduce.
- Select the points or path for reduction with the pointer and choose a magnification menu item.
- Select the magnifying tool and drag a box around the area to reduce while pressing **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 character so the em square just fits into the window. It also centers the character within the window.

Magnification

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



Or you can fit the character'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

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

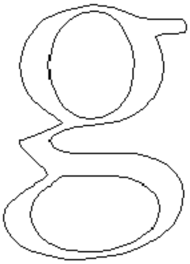
In Fontographer, you can scroll by dragging an object or a path outside the window area. As long as the cursor is outside the image area and the mouse is pressed down, the screen view will scroll in the direction of the movement. For example:

Viewing modes

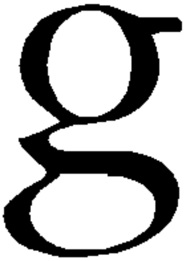
Fontographer’s **View** menu provides different ways of looking at the image in the character window. Each mode offers a different perspective of the character image when you are constructing or editing a font.

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 character while you’re in the preview mode.



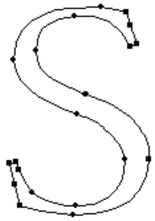
Outline



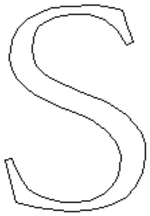
Filled character

Show Points

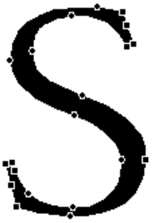
From the **View** menu you can select **Show Points**, **Command-U** to display all the points in the character. Corner points are square-shaped, tangent points are triangular, and curve points are circular.



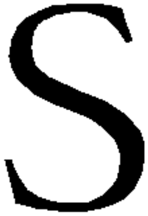
Points showing in Outline mode



Points not showing



Points showing in Preview mode



Points not showing

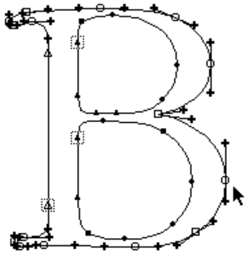
➲ ***Tip:*** To temporarily display points when they are turned off, click anywhere outside the path.

Selecting and deselecting objects

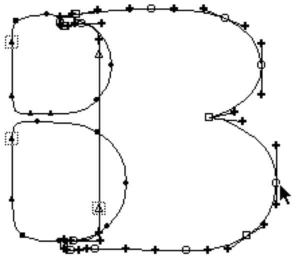
You work with objects in the outline window by selecting and altering them. To select an object, choose the pointer from the tool palette and use one of the following methods.

- 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 character slot.

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



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



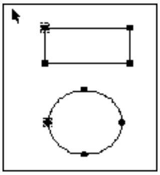
Only that outline is selected.

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.

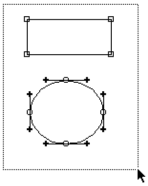
You will also notice a dotted box around one point in each path. This is the first point or origin of the path. Use the Preferences dialog box to turn this option on or off. In some transformations (like font blending and multiple master fonts), the first point is aligned with initial points in other 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 11, "Expert Advice". Multiple master fonts are discussed in "Multiple master fonts" in Chapter 11, "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.



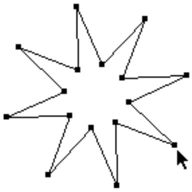
Position the pointer outside the area you want to select.



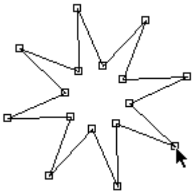
Drag the mouse around it.

Click-selecting objects

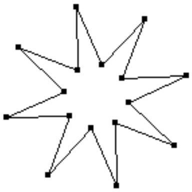
Double-clicking an object (composed of a single path) selects it. To deselect the object, click outside the shape or press the **Tab** key.



Double-click an object…



to select it.

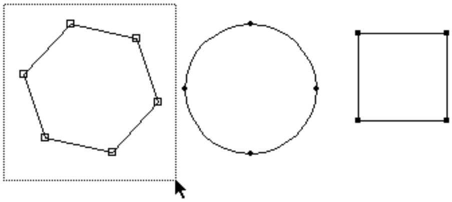


*Press the **Tab** key to deselect it.*

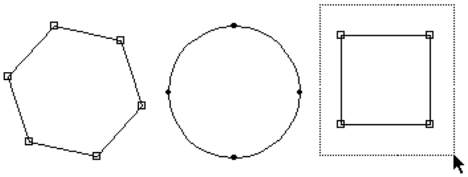
Shift-selecting objects

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.



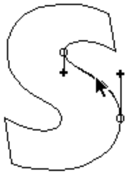
*Press the **SHIFT** key and drag around another object.*

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 characters as well.

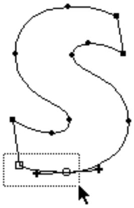
You can select a segment and its two connecting points in three different ways:

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



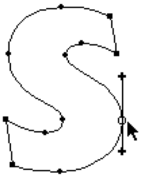
Click directly on the path.

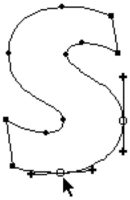
- Drag the pointer around the segment (and its two adjoining points) and release the mouse.



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.



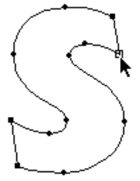


*Select a point and hold down the **SHIFT** key…*

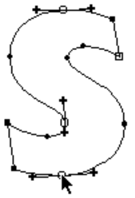
while you select another point.

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.

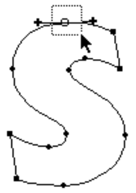


Click a point.

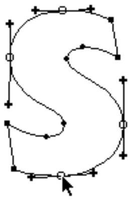


*Hold down the **SHIFT** key to continue selecting points.*

- 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.



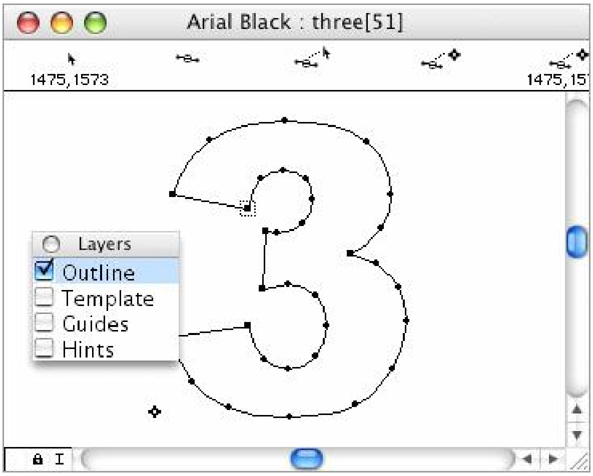
*Press the **SHIFT** key to continue selecting other points.*

Drawing layers

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

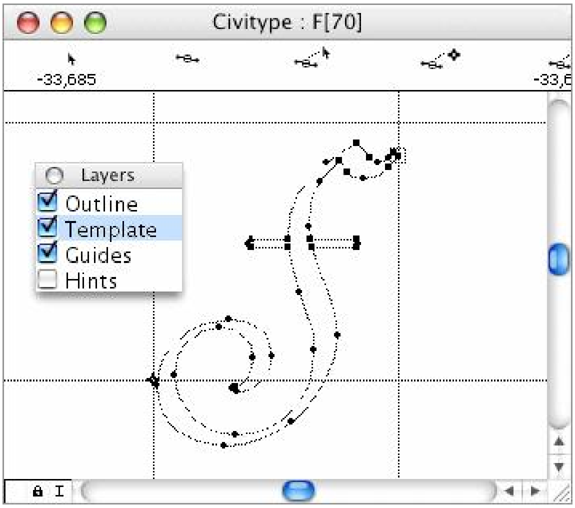
Outline layer

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



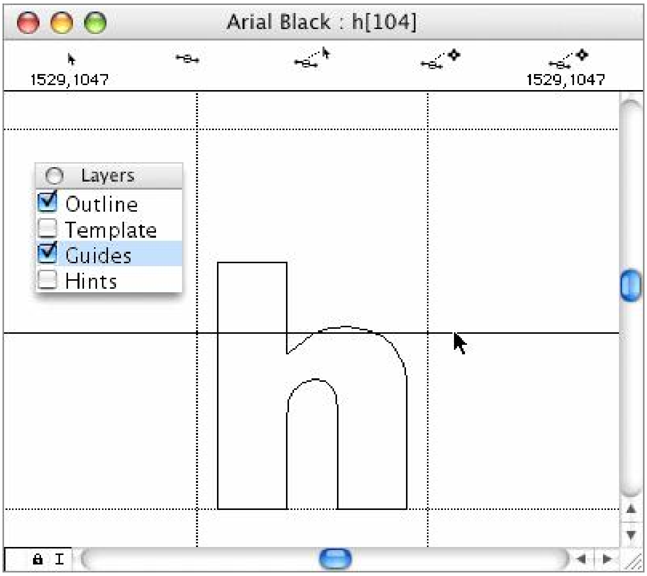
Template layer

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 printed 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

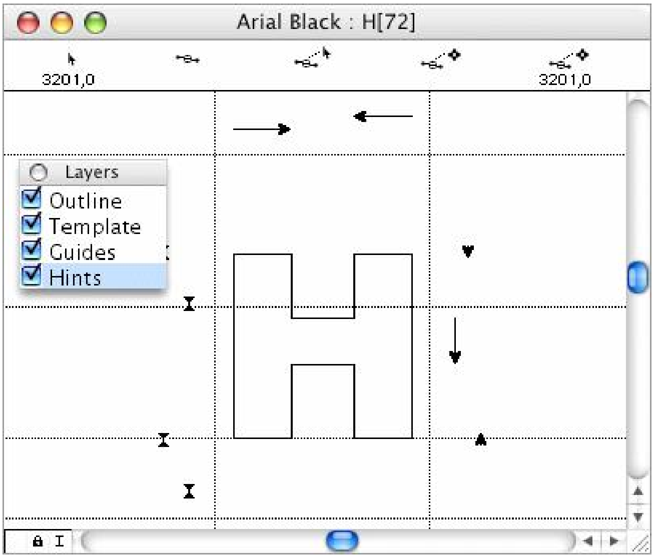
The guides layer can be used as a drawing aid. Additional guidelines can be pulled out of the origin line and the baseline. These lines serve as guidelines in character construction. For example, a horizontal line can be used to set the x-height. You can create multiple horizontal and vertical guidelines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



Changing the guidelines for one character will change the guidelines in all the other character windows in the font; likewise, any edits to the guidelines in this layer will immediately appear in every other character in the font. However, any images created in the guides layer do not print since this layer is just a design aid for all the characters. 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 character will be printed when output to low-resolution printers. The object of Fontographer's hinting process is to preserve the shape of the characters when they are printed at smaller sizes, including serifs, x-height, stem widths, and other features that define the characters in a font.

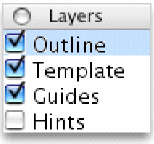


For more information about the hinting process and the hints layer, refer to "Font hinting" in Chapter 11, "Expert Advice".

Using the palettes

Layers palette

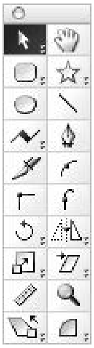
In Fontographer, the Layers palette displays when you open the outline window. It can be moved or closed like any window, and selected or deselected for display from the **Windows** menu.



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

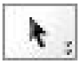
To change layers, click the mouse on the name, not the checkbox, of the desired layer. The layer appears highlighted within the palette and display on the screen. To display another layer in the outline window, click the appropriate checkbox. When a layer's checkbox is selected, its contents display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

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 **Windows** menu.

Pointer tool



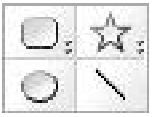
Use the pointer tool to select elements and drag points and objects. You can select the pointer by clicking its indicator with the mouse. When the pointer is active it is shaped like an arrow.

To change any other tool in the tool palette to a pointer, press the TILDE (~) key (to the left of the number 1). 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 character 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.

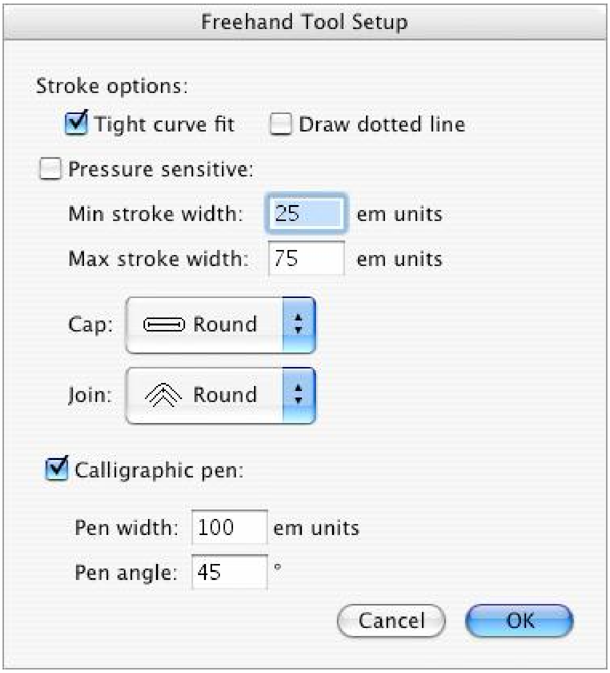
When the lock icon is in the locked position, you can use numeric keystroke equivalents to select the tools. The keystroke commands are: rectangle tool (1), multigon tool (2), oval tool (3), and straight line tool (4).

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.



Freehand Tool Setup

Stroke options:

☑ Tight curve fit ☐ Draw dotted line

☐ Pressure sensitive:

Min stroke width: 25 em units

Max stroke width: 75 em units

Cap: Round

Join: Round

☑ Calligraphic pen:

Pen width: 100 em units

Pen angle: 45 °

Calligraphic pen tool

The calligraphic pen tool creates calligraphic or freehand characters.

You can create calligraphic characters 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.

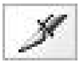
When the lock icon is in the locked position, the numeric keystroke equivalent for choosing the calligraphic/freehand tool is 5.

Pen tool

The pen tool is a multipurpose tool similar to those in Macromedia 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.

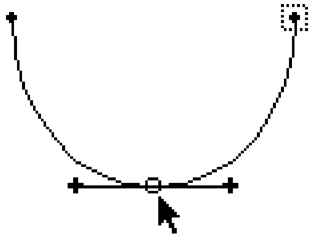
For more information on splitting paths using the knife tool, refer to "Splitting a path" in Chapter 4, "Altering Outlines".

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.

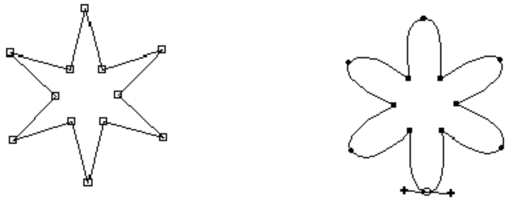


A curve point determines that any adjoining line segments will be curves, regardless of the type of points they are attached to. This is different than the way a corner point operates – the shape of its attached segments depends on the connecting points beside it in the path. Select the curve point tool 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 character stems; or you can join corner points to curve or tangent points to create gentle curves or cusps.

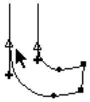


The shape of line segments extending from corner points is determined by points connected to the corner point.

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

Tangent point tool

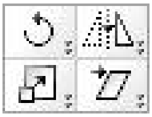
The tangent point tool is used to connect straight lines to curves with a smooth tangent join.



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.

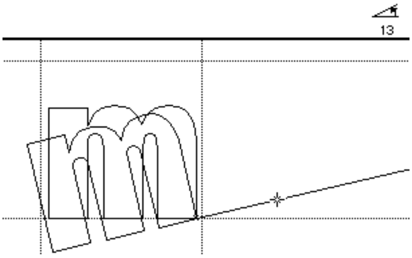
Transformation tools



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

Rotate tool

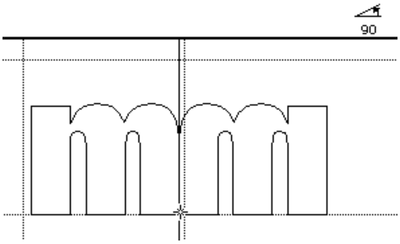
The rotate tool is used to rotate objects.



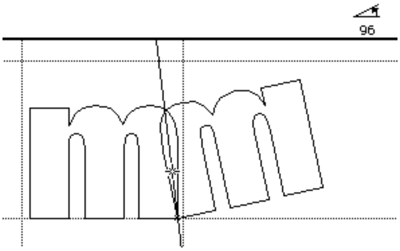
Holding down the **Shift** key while you rotate confines the rotation to 45-degree increments.

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.

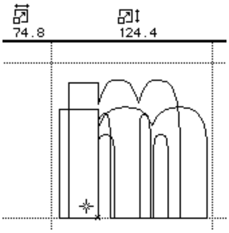


You can drag and flip freely by holding down the **SHIFT** key as you use the flip tool.

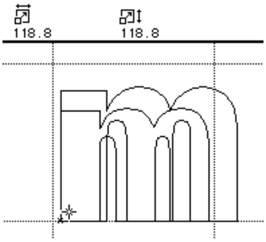


Scale tool

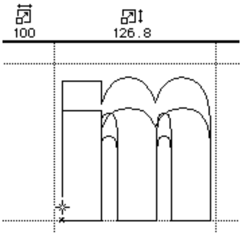
The scale tool increases or decreases the size of the object. Depending on the horizontal or vertical direction you drag, you can condense and/or stretch the object. The scale values are displayed in the info bar.



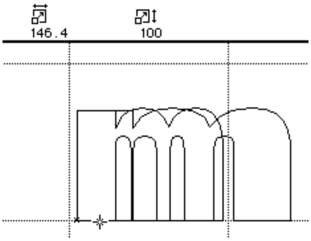
If you hold down the Shift key and drag at a 45-degree angle (up or down), you can scale proportionately.



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



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

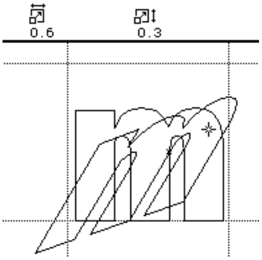


Double-clicking it brings up the transform dialog box with Scale selected as the first transform.

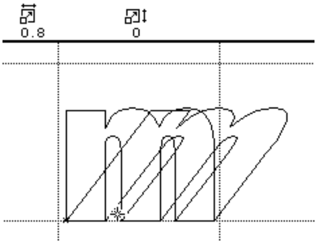
When you **Option**-double-click the scale tool, the Transform dialog box appears with Scale Uniformly selected as the first transformation.

Skew tool

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



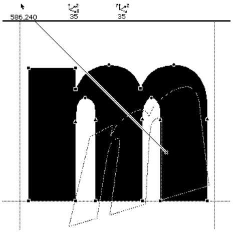
To skew in 45-degree incremental angles, hold down the **Shift** key while you skew.



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 characters in the font precisely.

Perspective tool

With the perspective tool, you can create space-age 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, character origin, or mouse click.

Changing the Perspective Distance/Point will not change the appearance of the character 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 character in your outline window.

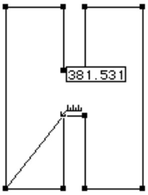
Note: When a 3D transformation has been applied to a two-dimensional object, the object becomes two-dimensional again. The object itself does not maintain three-dimensional coordinates. To do multiple 3D transformations while maintaining 3D coordinates in between each transformation you must use the 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 characters 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 increases or decreases the view of 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 reduce 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.

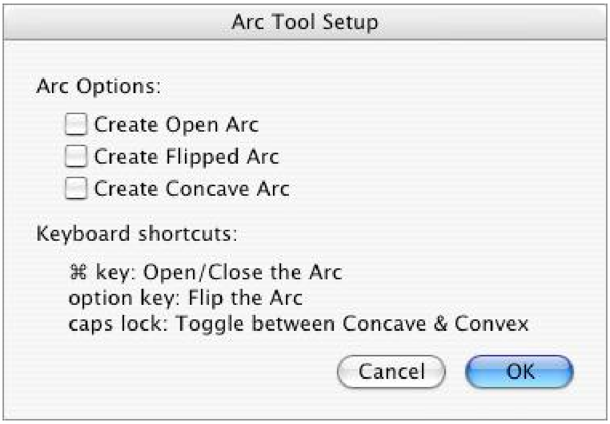
The magnifying tool can be temporarily accessed from any other tool by holding down **COMMAND-SPACEBAR**. To demagnify the view, hold down **OPTION-COMMAND-SPACEBAR**.

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 options 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 Tool Setup

Arc Options:

□ Create Open Arc
□ Create Flipped Arc
□ Create Concave Arc

Keyboard shortcuts:

⌘ key: Open/Close the Arc
option key: Flip the Arc
caps lock: Toggle between Concave & Convex

Arc Options

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

When the Create Open Arc option is checked in the Arc Tool Setup dialog box, the arc tool creates open paths. Two examples of open path arcs are and .

By clicking the checkbox or the words next to the checkbox, you can toggle the checkbox on and off.

By default the arc tool creates convex arcs. These arcs will resemble quarter ovals like ◸, ◿, ╭ and ╯. With the Create Concave Arc option checked in the Arc Tool Setup dialog box, the arc tool creates concave arcs like ┏ and ┛. Notice that the Create Open Arc option has no effect when the Create Concave Arc checkbox is on.

By default the arc tool creates unflipped arcs like ◸, ╭ , and ┏. With the Created Flipped Arc option checked, the same sequence of steps creates arcs like these: ◿, ╯and ┛.

Keyboard shortcuts

You can create different types of arcs without changing the arc options. The arc options determine the default type of arc that is created. However, 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 Create Open Arc turned on, 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.

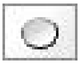
The **CONTROL** key is the demagnified move modifier key. By holding down the **CONTROL** key when you create a new arc, you can increase/decrease the size of the arc in one em-unit increments.

Constraining tools

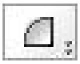
When you want to constrain the action of a tool to 45-degree increments, use the tool normally while holding down the **SHIFT** key. Each tool's constraint mode may operate with slightly different results.

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.

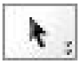


To use the oval tool to draw a circle, hold down the **SHIFT** key and drag the mouse.

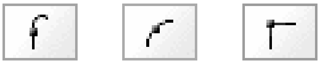


To use the arc tool to draw a quarter circle, hold the **SHIFT** key down while moving the mouse at approximately a 45-degree angle (up and right, up and left, down and left, or down and right.)

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.



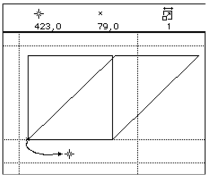
Use the **SHIFT** key with corner, curve, or tangent point tools while dragging points to constrain their movement. In addition, if the **SHIFT** key is held down when placing new points with these tools, the new point is automatically aligned (or constrained to) the previous point.

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

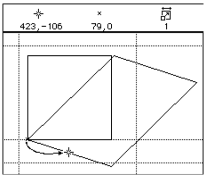
Note: The flip tool is automatically constrained, because you will most often want to flip paths and objects in exact 90-degree increments. Hold down the **Shift** key to deconstrain the flip tool.

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

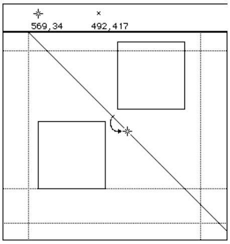
Note: In the bitmap window, the measuring tool is automatically constrained, because you will usually want to measure pixels straight across, or up and down. So, when you're in the bitmap window, hold down the **Shift** key to deconstrain the measuring tool.



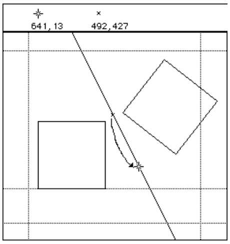
*Constrain skewing by using the **Shift** key.*



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

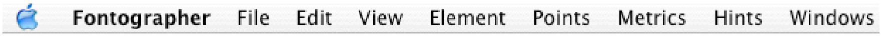


*This rotation is constrained by holding down the **Shift** key.*

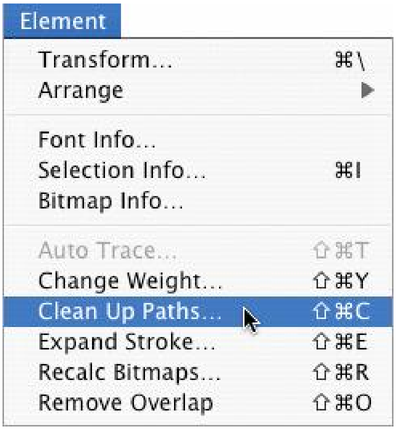


*It is rotated unconstrained without the **Shift** key.*

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

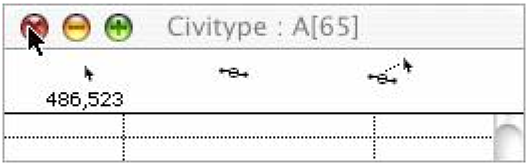
- Whenever you are in a character’s outline window and you want to undo a command or a series of commands, you can choose **Undo** from the **Edit** menu or type **Command-Z**.
- To redo what you’ve undone, choose **Redo** from the **Edit** menu or type **Command-Y**.

The default setting for the number of commands you can undo or redo is eight, but this number can be changed (up to 101) by resetting the preferences. See “Setting preferences” on page 18, or in “General preferences” in Chapter 11, “Expert Advice”.

Closing and Quitting

Closing

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

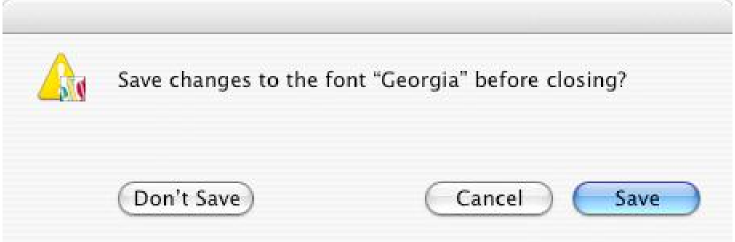


- Select **Close** from the **File** menu or type the keyboard alternative, **Command-W**.

Quitting

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

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


Save changes to the font "Georgia" before closing?

- 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.

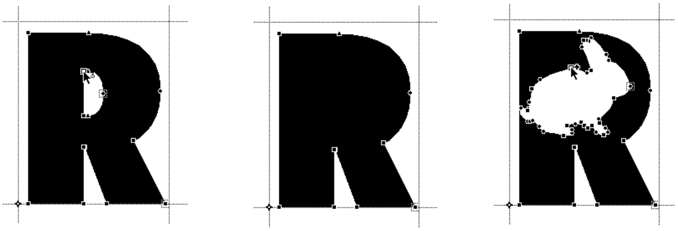
Fontographer offers new Quit dialog boxes for greater efficiency when quitting with multiple, unsaved documents open.

- **Close All** quits without further interruption, leaving all documents unsaved.
- **Save All** saves all open documents and quits Fontographer. If any database has not been saved, the Save As dialog box appears.
- **Review** 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, canceling, or not saving changes to each one.

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

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.

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 12, "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

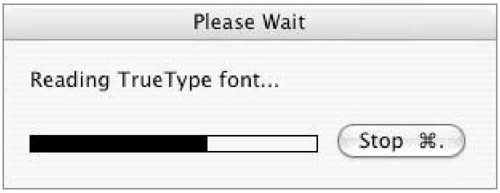
Opening a Font

Existing fonts

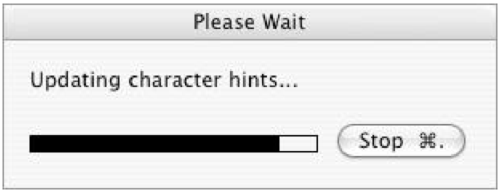
To open an existing font, choose **Open Font** or **Command-O** from the **File** menu. A file selection dialog box appears that works in the standard fashion. Once the selection dialog box is open, you can select a font file by clicking its name and then **Open**, or simply by double-clicking its name.

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

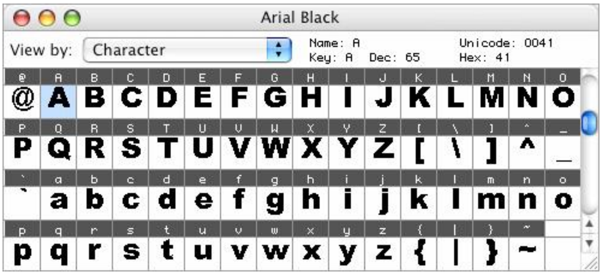


Reading TrueType font...



Updating character hints...

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



Arial Black

View by: Character

Name: A
Key: A Dec: 65

Unicode: 0041
Hex: 41

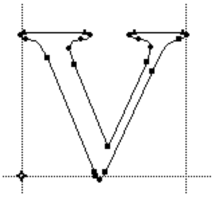
@ A B C D E F G H I J K L M N O
P Q R S T U V W X Y Z [\ ] ^ _
` a b c d e f g h i j k l m n o
p q r s t u v w x y z { | } ~

until Fontographer displays the Font Window.

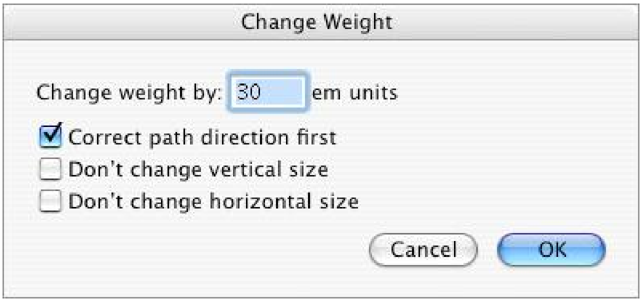
✐ ***Note:*** You can open as many fonts as you like depending on the amount of memory you have available. The average font takes between 200K and 300K of RAM.

Changing the character's weight

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



Open any character's outline window.



Change Weight

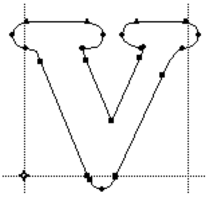
Change weight by: 30 em units

☑ Correct path direction first

□ Don't change vertical size

□ Don't change horizontal size

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

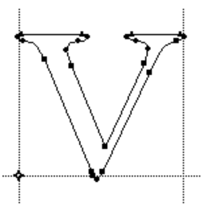


Fontographer changes the weight of the character.

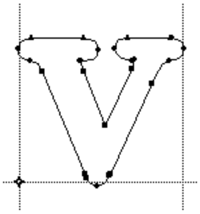
To change weight:

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

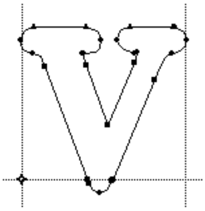
   The Change Weight dialog box appears.
3. Enter 30 in the Change weight by text edit box and click **OK**. Fontographer increases the weight of the “v” by 30 em units.



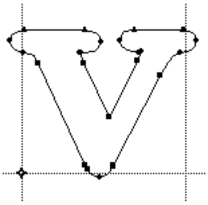
Original character.



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 character or entire font without affecting the vertical or horizontal size of the character.

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 **OK**.

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.

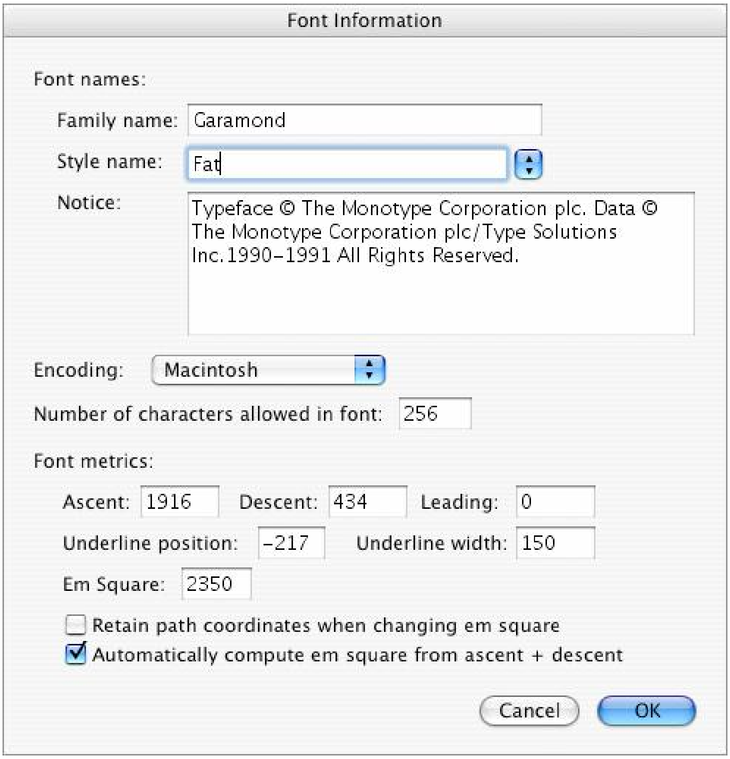
You can now apply the desired weight to the entire font or just to selected characters. From the font window, use the pointer tool to click and SHIFT-click the desired characters. (You can choose them all by choosing **Select All** from the **Edit** menu.) Then repeat the procedure above to apply the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

➲ ***Tip:*** You can’t undo global changes in the font window, so it’s a good idea to try your changes out on one character first. We recommend testing out your weight changes on one character 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 characters 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. For this exercise, name your font something simple like Garamond-Fat. (Don't forget the hyphen before the Style Name.) For more information about naming your font, refer to Chapter 8, "Generating and Exporting Fonts".

If you want to use a custom style name, be sure to let Fontographer assign a standard style name first and then reassign your custom name. So, if you want to call your bold-italic version of a font fat slanted, use the pop-up and select bold italic first, then change the style name to fat slanted.


Font Information

Font names:

Family name: Garamond

Style name: Fat

Notice: Typeface © The Monotype Corporation plc. Data © The Monotype Corporation plc/Type Solutions Inc.1990-1991 All Rights Reserved.

Encoding: Macintosh

Number of characters allowed in font: 256

Font metrics:

Ascent: 1916 Descent: 434 Leading: 0

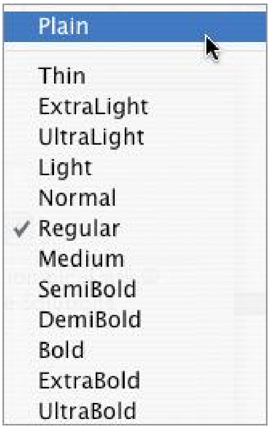
Underline position: -217 Underline width: 150

Em Square: 2350

□ Retain path coordinates when changing em square

☑ Automatically compute em square from ascent + descent

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



Fontographer also provides you a list of predefined style names from which to choose.

This will allow Fontographer to recognize and assign the proper style to your custom name.

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.ttf (for a TrueType font), and you'll have to start over.

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 Macromedia FreeHand, 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 92 and Chapter 8, "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".

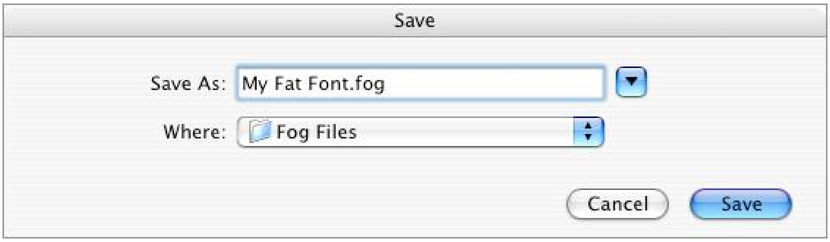
The standard file saving dialog box appears. You can name your databases anything you like, because there's no relationship between the name of the actual font you'll use in your programs and the name of the database itself.

➲ ***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 92.

Save As

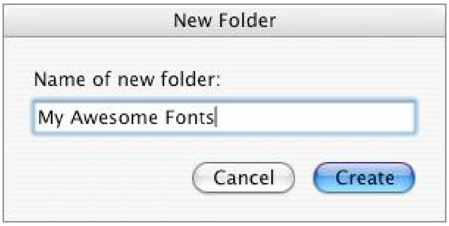
Choose **Save As** when you want to make another copy of a currently open font file. This will save the changes you made, without overwriting the original file. You can save the file in its current directory or you can save it in another location listed in the dialog box.


Save

Save As: My Fat Font.fog

Where: Fog Files

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


New Folder

Name of new folder: My Awesome Fonts

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** 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.

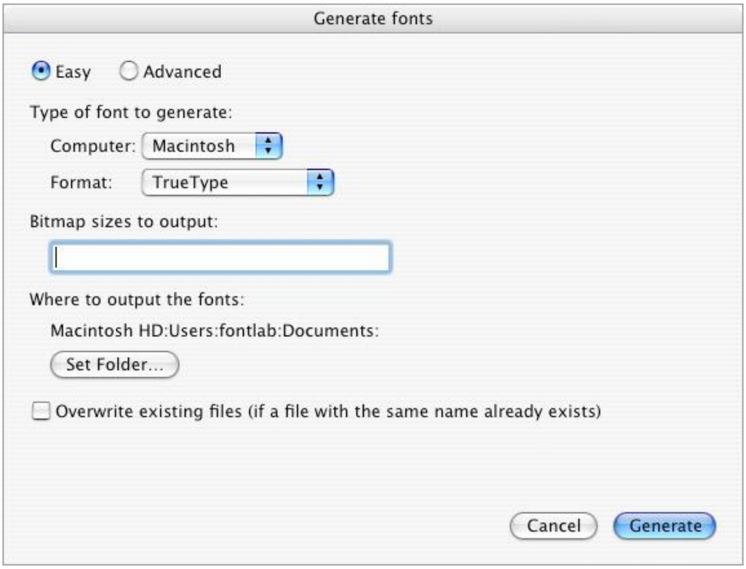
Generating your font

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 10, “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 computer you’re generating fonts for and select 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 PostScript font on the Macintosh.



Generate fonts

● Easy ○ Advanced

Type of font to generate:

Computer: Macintosh

Format: TrueType

Bitmap sizes to output:

Where to output the fonts:

Macintosh HD:Users:fontlab:Documents:

□ Overwrite existing files (if a file with the same name already exists)

The **Set Folder** 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 bullet (•). For more information about font generation options, refer to Chapter 8, "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 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.

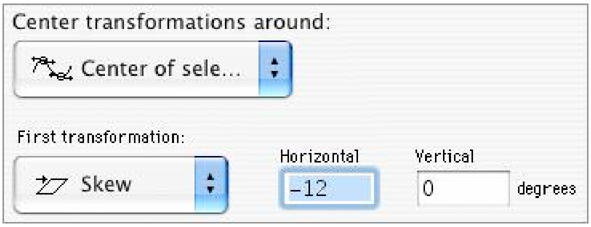
About font piracy

"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 characters that compliment the face. But, skewing is a really easy way to create a new typeface that can add emphasis to your text.



Center transformations around:

Center of sele…

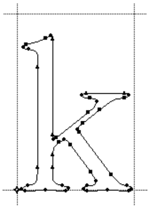
First transformation:

Skew

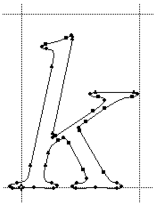
Horizontal: -12

Vertical: 0 degrees

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



Original character



Skewed character

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

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

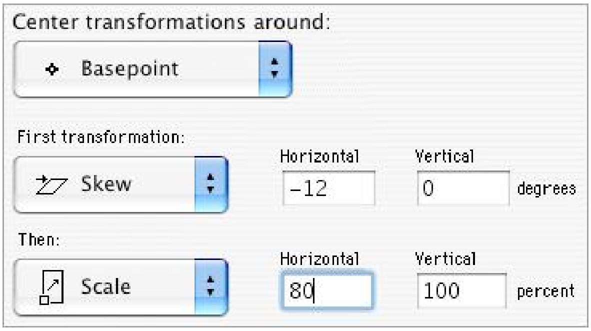
To skew a character:

1. Go to the Font Window and double-click the character “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 character.

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).



Center transformations around:

Basepoint

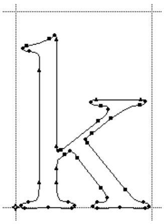
First transformation:

Skew — Horizontal: -12 — Vertical: 0 degrees

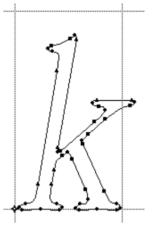
Then:

Scale — Horizontal: 80 — Vertical: 100 percent

Choose more than one transformation at once.



Original character



Transformed character

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 character.

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

Skewing multiple characters

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

Creating a fraction using composite characters

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 characters made up of more than one character combined in a single character 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 character slots.

2. Click the "8" character slot in the Font Window.

   It's not necessary to open the outline window to copy the character and then select **Copy Reference** 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 characters to 60% of their original size.

6. Position your pointer away from the characters, 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 character's bounding box). Composite characters do not show the points you normally see. (You cannot edit points in a composite character unless you first choose **Unlink Reference**.)

7. Drag the box containing the 8 toward the lower right corner.

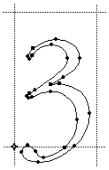
8. Double-click the path or any point on the number 3 to select all of it, and then move the 3 toward the top left corner.

   You can create the divisor line by copying the forward slash into your character. You can also draw the divisor line if you prefer. However, it is often much easier to use existing characters to create parts.

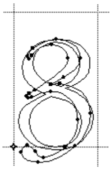
9. Select the forward-slash character in the font window.

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

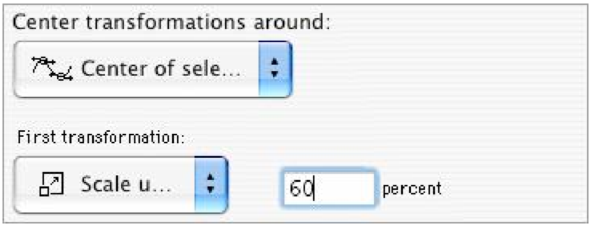
11. Paste the forward-slash character into the 3 character slot.



Open a character.



Paste in another character using the ***Copy Reference*** *command.*



Center transformations around:

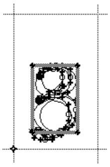
Center of sele...

First transformation:

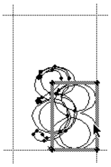
Scale u...

60 percent

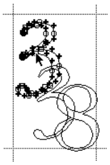
Choose ***Scale*** *from the Transform dialog box.*



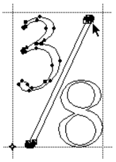
Fontographer scales both characters.



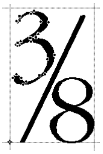
Move the eight toward the bottom right corner and the three toward the top left corner.



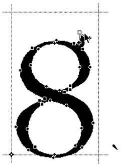
Move the three toward the top left corner.



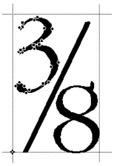
Copy and paste the forward slash character into the outline window.



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



Make any change to the original character.



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

To change a reference character:

1. Open the outline window for the actual character 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 character.

➲ ***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 characters. If the entire set uses reference characters, all the fractions in the set will be updated automatically when you modify the original elements.

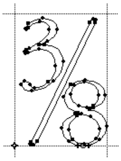
✐ ***Note:*** Because a composite character references a drawing rather than duplicating it, all the instructions required for creating that shape in the printer need only appear once, thus saving memory and processing time.

Unlinking a reference character

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

To unlink a reference character:

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



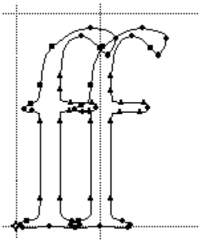
*Choose **Unlink Reference** to remove any reference to the original character.*

Creating a ligature

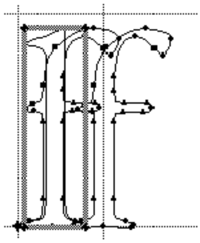
A ligature is a character made up of one or more characters. Most commercial fonts have some commonly used ligatures like æ, and œ. However, Fontographer makes it easy to create ligatures of your own without drawing a thing.

To create a ligature:

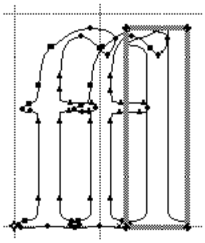
1. Open the outline window for the “f”.
2. Choose Select All and then choose Duplicate to create another “f”.
3. Move the new “f” to the right.
4. Go back to the font window.
5. Copy the “l” into the same window using the **Copy Reference** command from the **Edit** menu.
6. Move the referenced “l” to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



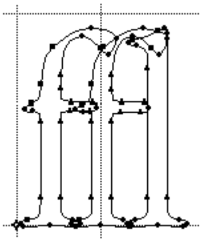
Duplicate the "f".



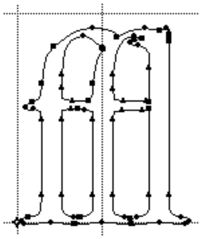
Copy a reference character into the window.



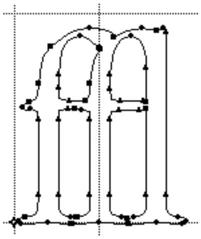
Move it into place.



*Choose **Unlink Reference**.*



*Then choose **Remove Overlap**.*



Remove any extra points.

Changing the character width

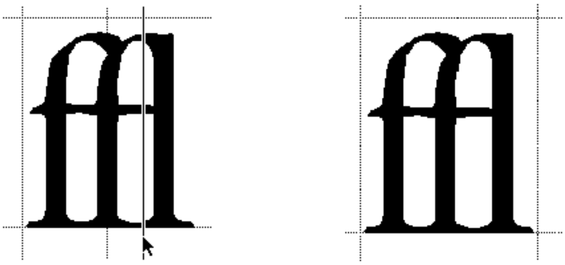
You probably noticed the vertical line that runs through the second “f”. This is the character width line. Width is a moveable vertical line that specifies the width of each character. When you print a line of text, the origin line of the next character is placed on top of the width line of the current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

To change the character’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 character 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 character 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 character, since you don’t have to worry about selecting all the points. You can unlink the reference once you’ve moved the character into the position you want.

Creating a condensed character or font

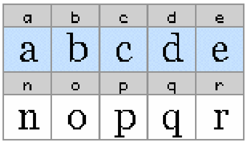
Fontographer has the ability to modify character images to produce interesting special effects. You can create these effects in either the outline window (on one particular character) or the font window (on the whole font). For example, you can create an oblique font by selecting all the characters in 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 character 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 character at time.

To create a condensed character:

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

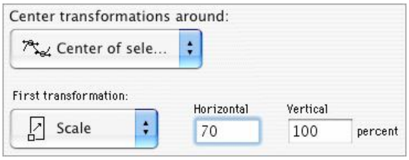


a	b	c	d	e
a	b	c	d	e
n	o	p	q	r
n	o	p	q	r

Click the characters you want to scale.

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

The Transform dialog box appears.



Center transformations around: Center of sele...

First transformation: Scale

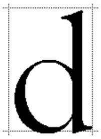
Horizontal: 70

Vertical: 100 percent

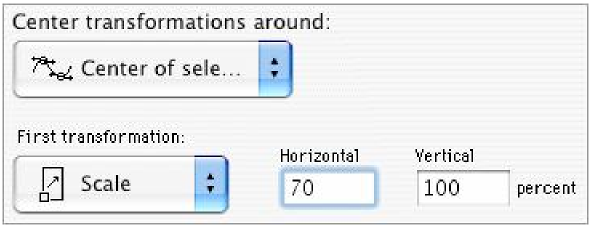
Select Transform from the Element menu and then enter the amount to scale.

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

You can apply the transformations in one of four ways:



Original character

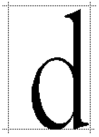


Center transformations around: Center of sele…

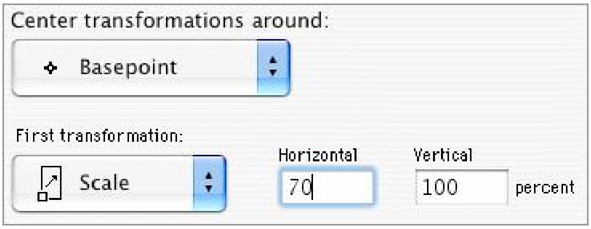
First transformation: Scale

Horizontal: 70

Vertical: 100 percent



Scaled from the center of the selection

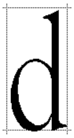


Center transformations around: Basepoint

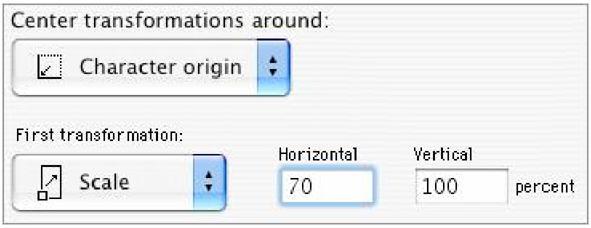
First transformation: Scale

Horizontal: 70

Vertical: 100 percent



Scaled from the basepoint

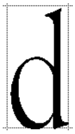


Center transformations around: Character origin

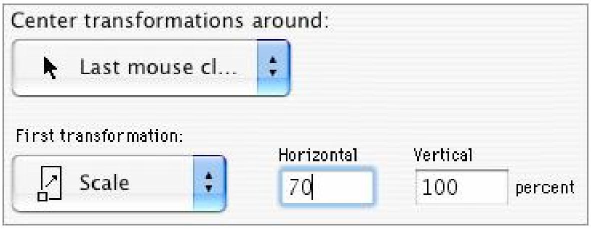
First transformation: Scale

Horizontal: 70

Vertical: 100 percent



Scaled from the character's origin

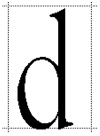


Center transformations around: Last mouse cl…

First transformation: Scale

Horizontal: 70

Vertical: 100 percent

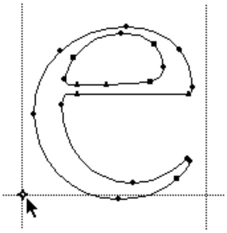


Scaled from the last mouse click

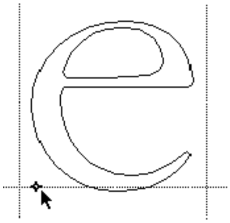
Setting the basepoint

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.

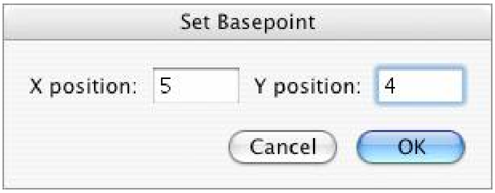
The basepoint is used to accurately and quickly align points and characters. The basepoint is generally at the character's origin (where the origin line and the baseline intersect at 0,0); however, you can position the basepoint anywhere. Fontographer allows you to set each character's basepoint differently. 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.



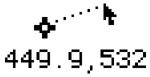
Set Basepoint

X position: 5 Y position: 4

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 character parts. Just place the basepoint on a reference point of the character 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

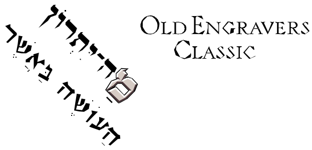
Imagine being able to digitize an old fashioned typeface and transfer it to your documents, or that logo you created with a pen or pencil before you'd ever heard of a graphics program. What about the beautiful effects that are created with calligraphy fountain pens? Wouldn't it be great if you could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

Arabia Felix Face
Mesopotamia Face

Calligraphic and graphic fonts by Judith Sutcliffe



Hebrew fonts and classic fonts by Dennis Ortiz-

Bad face
Hip face
Dog face
Dig face

Fun and wacky fonts by Paul Sych

Autotracing

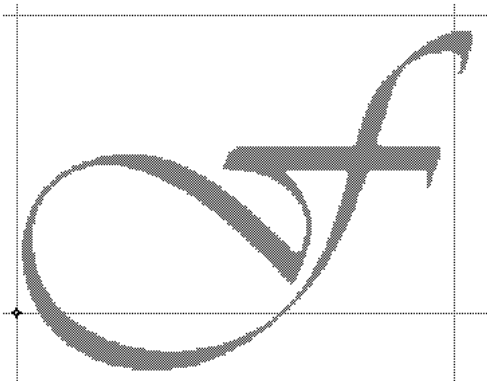
The bitmap option is one of Fontographer's most advanced features. Autotracing is probably most useful for tracing scanned images. Suppose you have an existing character, logo, or image that you want to assign to a keystroke. On the Macintosh, you can scan your image, save it in PICT format, and place 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.

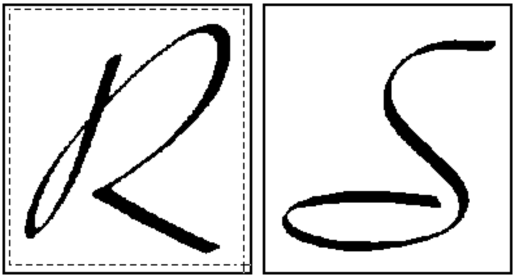


Fontographer will automatically paste the image into the Template layer where it will be shown as a dimmed image.

✐ ***Note:*** Images pasted into the Template layer are automatically sized to fit the em square. If you hold down **SHIFT-OPTION**, the image will automatically scale to fit between the ascent 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.

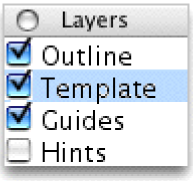
To make sure you copy the image in at the same size, open your scan in a program that can preserve the full resolution of copied bitmaps in its Clipboard. Draw a box around the largest of your characters (the “M” or “W” would be a good choice). Use this box to define the area you’ll be marqueeing and 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.

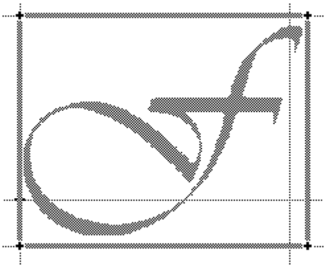
To move a Template image:

1. Click the Template layer to select it.



2. Click the template image with the selection pointer.

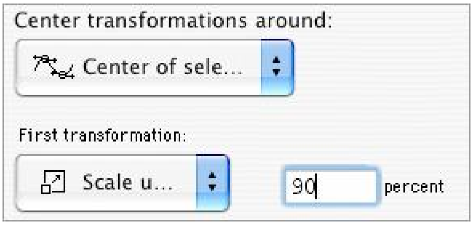
   A gray bounding box appears.



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:

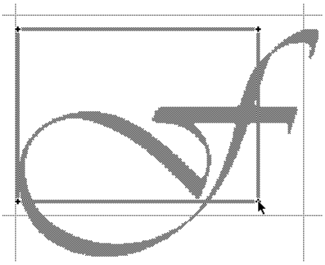
1. Click the template image to select it.
2. Choose **Transform** from the **Element** menu and Scale uniformly 90 percent.



Center transformations around: Center of sele...

First transformation: Scale u... 90 percent

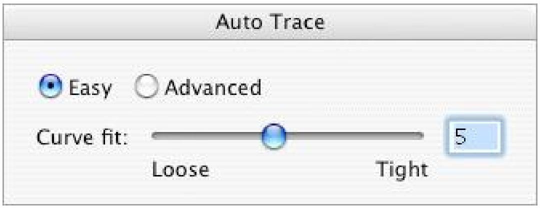
Note: Dragging the handles interactively scales the background image as well.



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.



Auto Trace

● Easy ○ Advanced

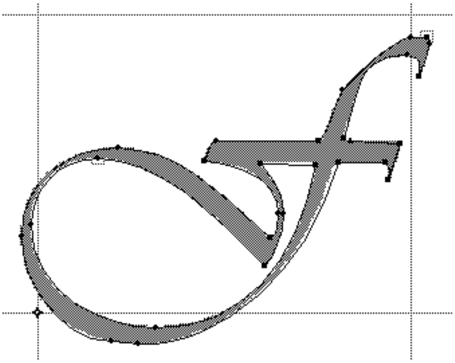
Curve fit: 5

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.

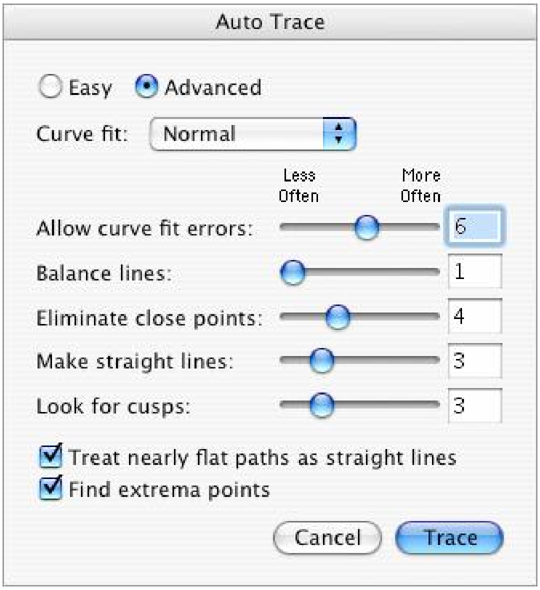


You can cancel the tracing operation at any time by clicking the **Cancel** button or by typing **COMMAND-PERIOD**.

➲ ***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.



○ Easy ● Advanced

Curve fit: Normal

Allow curve fit errors: 6

Balance lines: 1

Eliminate close points: 4

Make straight lines: 3

Look for cusps: 3

☑ Treat nearly flat paths as straight lines

☑ Find extrema points

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 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 characters. On the other hand, in a character'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 character.

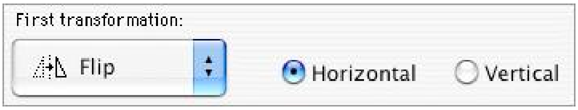
Flip

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 character.
2. Choose the **Flip** pop-up as the first transformation.
3. Click the **Horizontal** radio button.

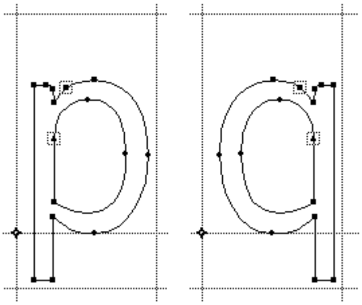


First transformation:

Flip

● Horizontal ○ Vertical

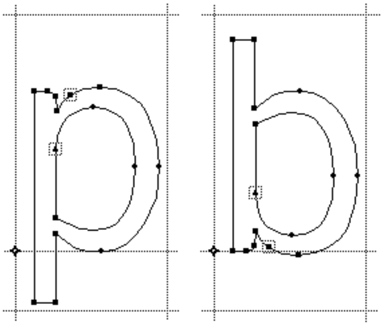
4. Click the **Transform** button.



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.



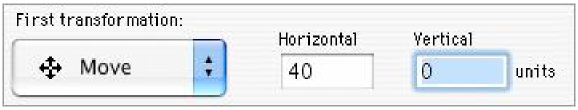
Change the "p" to a "b" using the Vertical Flip transformation.

Move

The Move transformation option can move whole characters, 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.

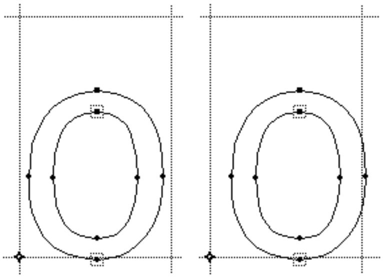


First transformation: Move

Horizontal: 40

Vertical: 0 units

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

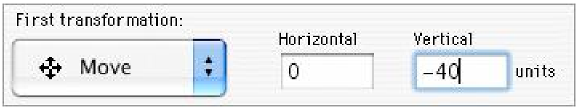


Move the "O" to the right using the Move transformation.

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.

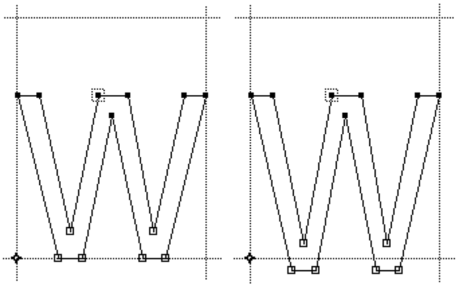


First transformation: Move

Horizontal: 0

Vertical: -40 units

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.

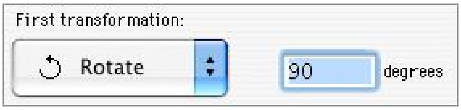
Rotate

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.

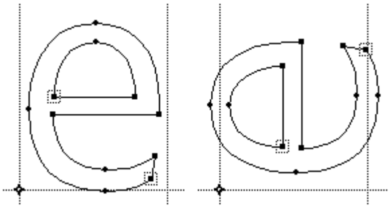


First transformation:

Rotate

90 degrees

3. Click **Transform**.



Rotate the "e" using the Rotate transformation.

Scale

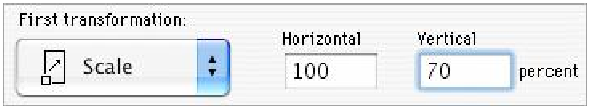
There are two Scale transformation options: Scale and Scale uniformly. Both of these options are used to increase or decrease the size of an image by a specific scale factor.

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 character vertically:

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



First transformation: Scale

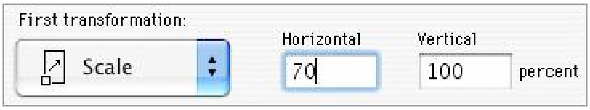
Horizontal: 100

Vertical: 70 percent

3. Click **Transform**.

To scale a character horizontally:

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

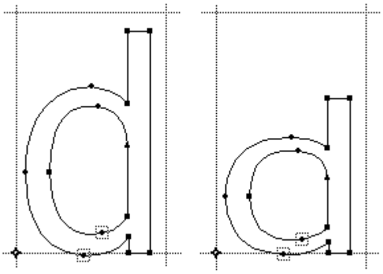


First transformation: Scale

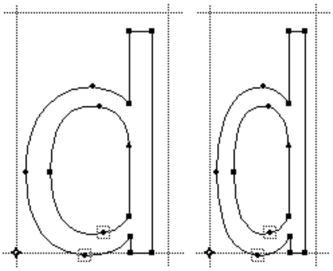
Horizontal: 70

Vertical: 100 percent

3. Click **Transform**.



Use scaling to create vertically scaled...



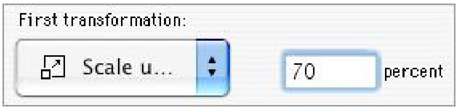
or condensed versions of your character or font.

Scale uniformly

This transformation option scales the entire image uniformly. If you select a character 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%.

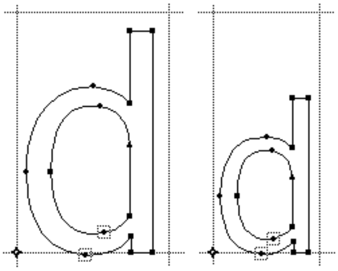
To scale uniformly:

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



First transformation: Scale u... 70 percent

3. Click **Transform**.



Scale the "d" uniformly.

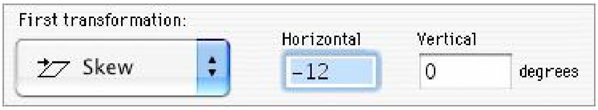
Skew

The Skew transformation option applies a slant to the image. Negative values slant the character to the right, positive values to the left. Vertical skewing can be used to create oblique characters. 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 characters:

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

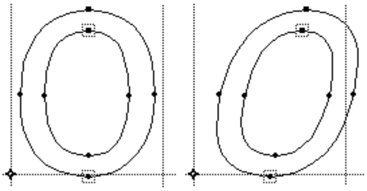


First transformation: Skew

Horizontal: -12

Vertical: 0 degrees

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

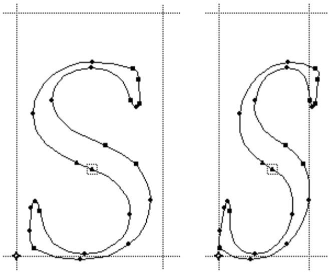


Skew the "o" with the Skew transformation.

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 character, or to the entire font.

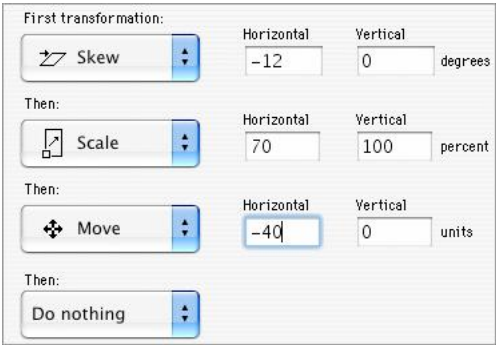
Suppose you want to create a condensed oblique font and move it closer to the baseline to compensate for the skew angle. It's easier than you might think.



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

To apply multiple transformations:

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



First transformation: Skew — Horizontal -12, Vertical 0 degrees

Then: Scale — Horizontal 70, Vertical 100 percent

Then: Move — Horizontal -40, Vertical 0 units

Then: Do nothing

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

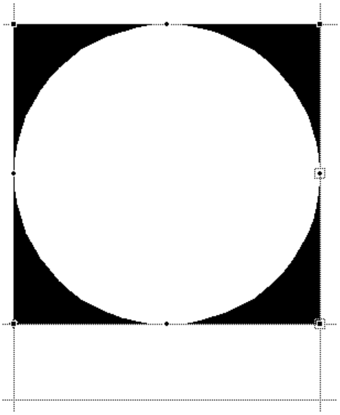
3D transformations using the Transform dialog box

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.

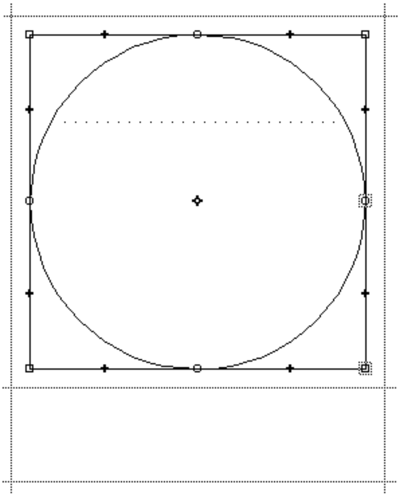
   The character should look like this with **Preview** selected from the **View** menu or by pressing **Command**:



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.

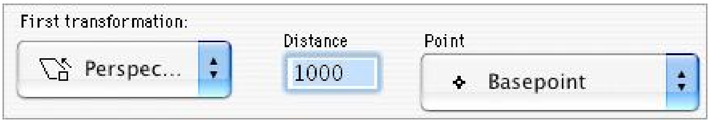
7. Type 90 into the text edit field and press RETURN or click **Transform**.

   The character 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.



First transformation: Perspec...

Distance 1000

Point Basepoint

2. Set the Distance to 1000 and the Point to Basepoint and press the RETURN key or click **OK**.

   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.

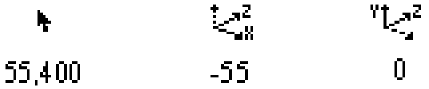
5. Choose **Copy** from the **Edit** menu.

   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 character).

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:



```
55,400          -55          0
```

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

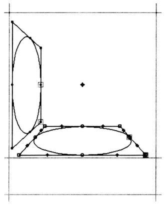
   You have just rotated the selected points by 90 degrees in the XZ plane. You can also think of it as rotating around the Y axis.

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 character 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.

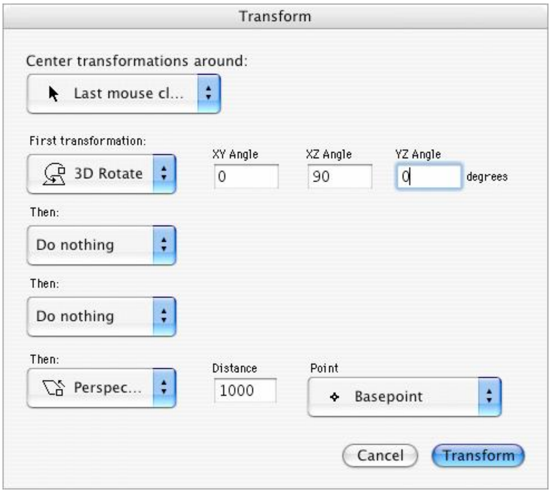
4. Double-click the Perspective tool.

   The Transform dialog box will be brought up ready to apply a 3D rotation.
5. Choose Last Mouse Click from the **Center transformations around** 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:



Center transformations around: Last mouse cl...

First transformation: 3D Rotate — XY Angle 0, XZ Angle 90, YZ Angle 0 degrees

Then: Do nothing

Then: Do nothing

Then: Perspec... — Distance 1000, Point Basepoint

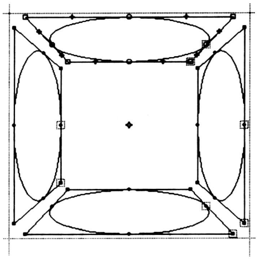
8. Press the **Return** key.

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 character 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.

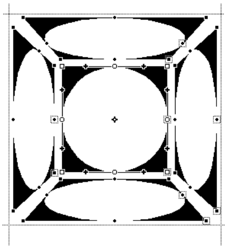
You will now create the back side of the cube by moving the selected points backward (by 800 em units) into the third dimension.

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 **OK**.
6. Select **Preview** to view your completed characters:

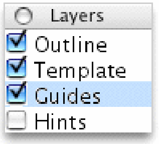


Guidelines

Setting guidelines

The Guides layer is used to construct drawing guidelines that are common to every character 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 character 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 character'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.



☑ Outline
☑ Template
☑ Guides
□ Hints

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

🔒

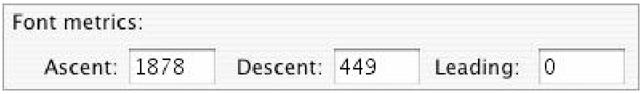
when the lock icon is in the locked state.

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 character 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 Ascent and Descent text boxes.



Font metrics:

Ascent: 1878 Descent: 449 Leading: 0

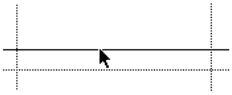
Note: Most word processing and page layout programs now support leading, but a few may still ignore the information you enter in the leading field.

Adding new guidelines

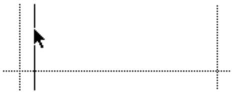
Additional guidelines can be set two ways in the Guides layer.

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...

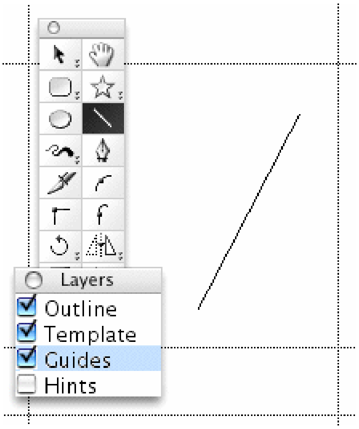


or origin line to create a guideline.

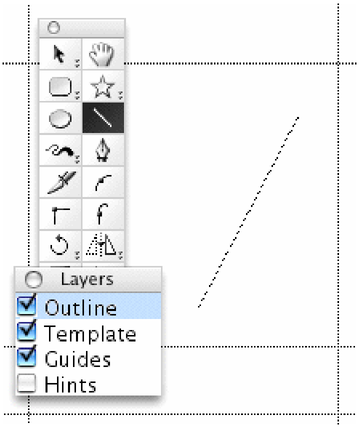
Tip: Remove these guidelines by **Option**-clicking them, or use the pointer tool to drag them back into the origin or baseline that they originated from.

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.



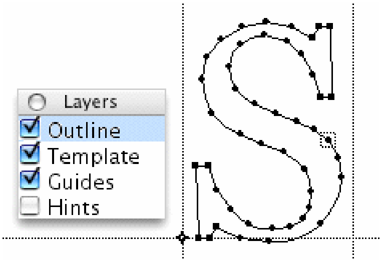
Drag your own guide in the Guides layer…



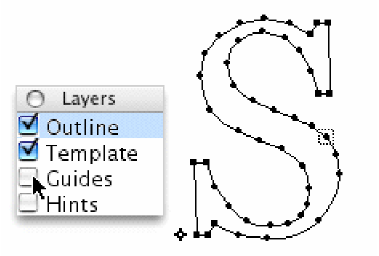
and it will appear in the Outline layer of every character in the font.

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**.



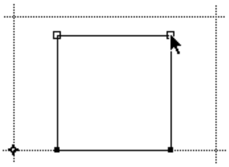
Turn guides on...



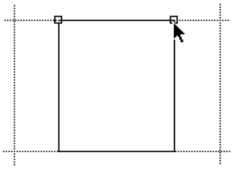
or off.

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 Behavior) 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

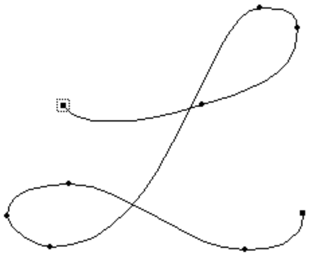
Outline versus stroked characters

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

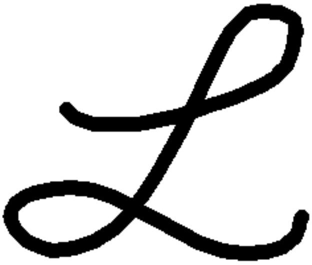
Any character that is constructed entirely of equal weight strokes can be drawn as a stroked character. For example, we drew the letters in this Fontographer logo as stroked characters. You must generate these as Type 3 fonts, since neither TrueType nor Type 1 fonts allow stroked characters.

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 character into an outline character (or font).



Draw characters like this with the freehand tool. This character’s stroke weigh is 40 em units.



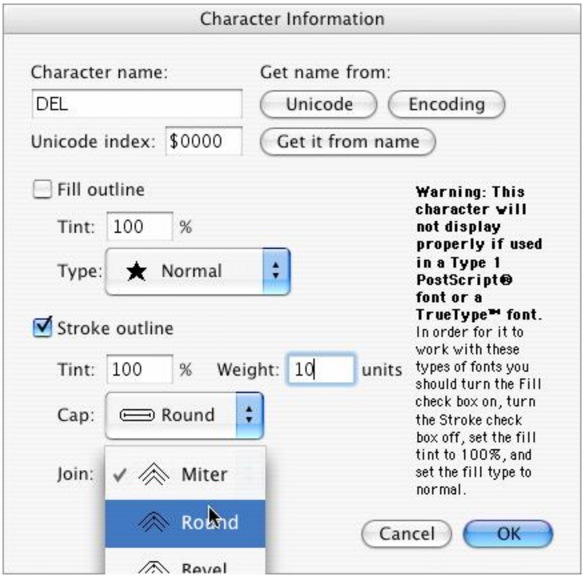
Preview the character with or without the points.

Setting stroke attributes

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

To change a filled character 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.



Character Information

Character name: DEL

Unicode index: $0000

□ Fill outline

Tint: 100 %

Type: ★ Normal

☑ Stroke outline

Tint: 100 % Weight: 10 units

Cap: Round

Join: ✓ Miter

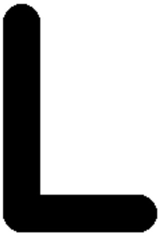
Warning: This character will not display properly if used in a Type 1 PostScript® font or a TrueType™ font. In order for it to work with these types of fonts you should turn the Fill check box on, turn the Stroke check box off, set the fill tint to 100%, and set the fill type to normal.

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 characters to be rounded. Each of these options is discussed after this example.

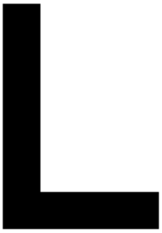
5. Click **OK** and get ready to draw a character.
6. Choose a drawing tool or one of the control point tools.

7. Draw an “L”.
8. Turn on Preview to see what your character actually looks like.

Draw with a corner point tool…



to create round capped characters…



or square capped characters.

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.

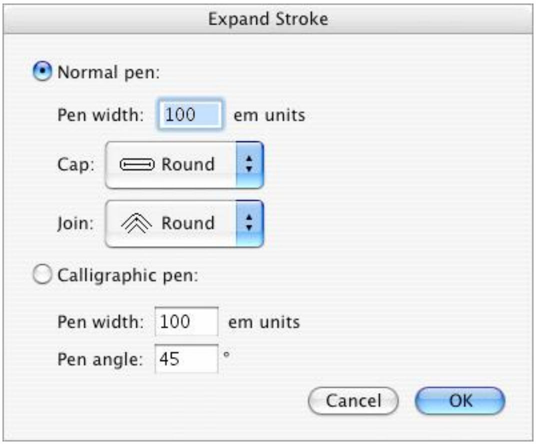
- Miter joins extend the outer edges of the lines until they meet at an angle, like the corners of a picture frame.
- Round joins draw a circle with a diameter equal to the stroke width at each bend.
- Bevel joins are drawn as if the joining segments were stroked with butt cap ends and the resulting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

Expand stroke

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

To expand the stroke width:

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



Expand Stroke

● Normal pen:

Pen width: 100 em units

Cap: Round

Join: Round

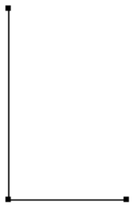
○ Calligraphic pen:

Pen width: 100 em units

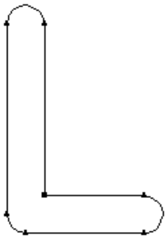
Pen angle: 45 °

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

   Fontographer automatically changes the stroked character into an outline character.



Change a stroked character...

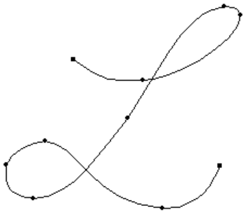


into an outline character.

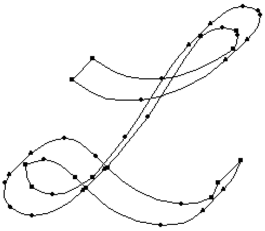
The **Expand Stroke** command can also be used to automatically change your stroked character into a calligraphic character.

To change a stroked character into a calligraphic character:

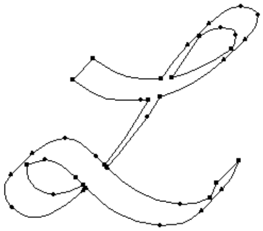
1. Choose **Expand Stroke**.
2. Click **Calligraphic Pen**, enter a value, and click **OK**.
3. Choose **Selection Info** and change the character to a Filled character.
4. Choose **Remove Overlap** from the **Element** menu.



Draw a stroked character with the freehand tool.



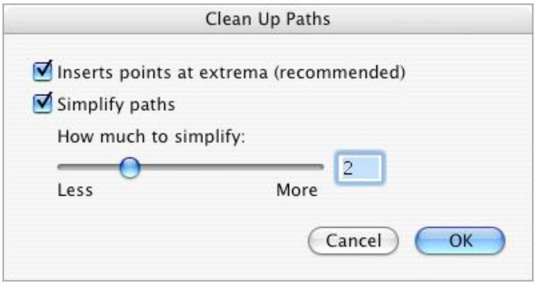
Change it into a filled character.



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 less points, your printing time will be faster as well.



Clean Up Paths

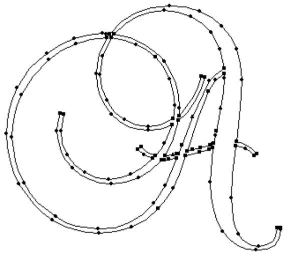
☑ Inserts points at extrema (recommended)

☑ Simplify paths

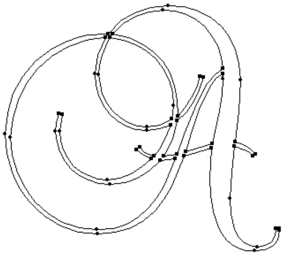
How much to simplify: 2

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 character 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 characters 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 characters

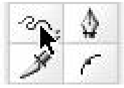
One of the most notable features of Fontographer is the freehand drawing tool that you can use directly with the mouse or with a pressure-sensitive pen and digitizing tablet. Additionally, you can use the freehand drawing tool as either a calligraphic pen or a variable-weight pen.

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

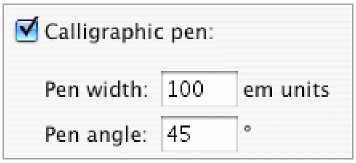
To use the calligraphic pen:

1. Double-click the freehand drawing tool.

   The Freehand Tool Setup dialog box appears.



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

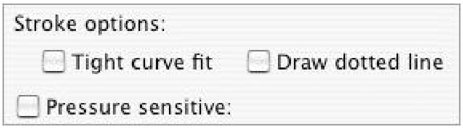


☑ Calligraphic pen:

Pen width: 100 em units

Pen angle: 45 °

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



Stroke options:

□ Tight curve fit □ Draw dotted line

□ Pressure sensitive:

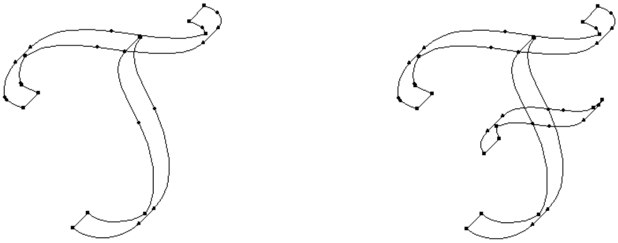
4. Enter a Pen width of 100 em units.

   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 characters.

5. Click **OK**.

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

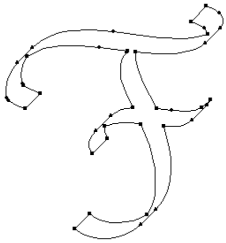
6. Press and hold down the mouse button while moving it around on your desk or mouse pad, or use a digitizing tablet to draw a calligraphic character.



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 character 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 character without points and filled.

Turn off points and then preview your new character.

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.

Calligraphy 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 characteristic 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 is 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 characters. 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 characteristic 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 111.

The second way is without a drawing tablet: Choose a couple of letters that are most characteristic 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 characters 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 characters 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 characters 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 characters 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 character 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 character 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 character that's looking good, you can get a little fancier for your uppercase characters. 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 characters 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" characters, 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 character 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 characters 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 characters. 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 (bearing in mind that there are differences between the System 6 and System 7 key positions). Then make a separate font and put the upper ASCII characters into it, on uppercase/lowercase keys and supply both fonts for Windows users.

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 character. The margins of characters need to be set so that the outgoing stroke of one letter overlaps the incoming stroke of the following character. This takes some careful experimentation with and slight manipulation of each character'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 characters 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 characters. The **Remove Overlap** command will be very handy. You can always adjust each new character 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 characters. 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 characters

Variable weight characters 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.

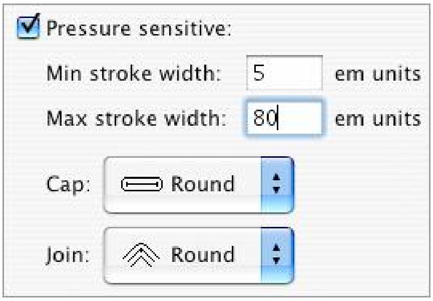
Note: If you use a pressure-sensitive pen and tablet, make the appropriate tablet settings before you draw your characters.

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.



☑ Pressure sensitive:

Min stroke width: 5 em units

Max stroke width: 80 em units

Cap: Round

Join: Round

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).

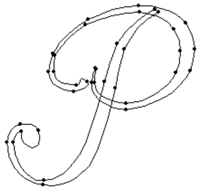
   Try experimenting with different values, and line caps and joins to get different line effects.

5. Click **OK**.

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

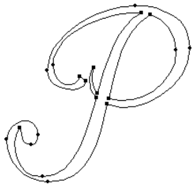
   If you are using a pressure-sensitive pen, go to step 6; if you are using a mouse, skip to step 8.

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



Draw a character.

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

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 character.

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 our own Macromedia FreeHand's blending of one object to another. It helps if your fonts are similar in characteristics. 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.

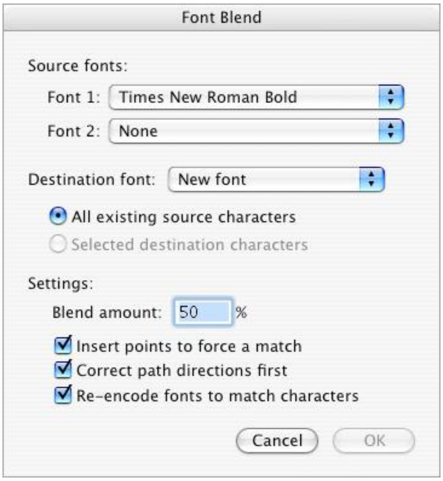
To blend fonts:

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.



Font Blend

Source fonts:

Font 1: Times New Roman Bold

Font 2: None

Destination font: New font

● All existing source characters

○ Selected destination characters

Settings:

Blend amount: 50 %

☑ Insert points to force a match

☑ Correct path directions first

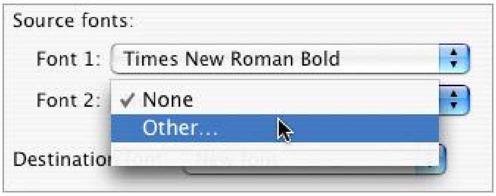
☑ Re-encode fonts to match characters

The font you opened in step 1 automatically appears as the Font 1 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 **Font 1** and **Font 2** Source fonts pop-ups.

3. Choose a Source font for the **Font 2** 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.



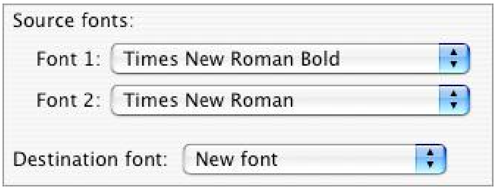
Source fonts:

Font 1: Times New Roman Bold

Font 2: ✓ None
Other...

Destinatio

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.



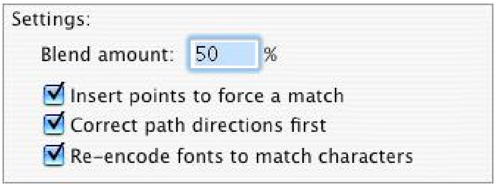
Source fonts:

Font 1: Times New Roman Bold

Font 2: Times New Roman

Destination font: New font

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



Settings:

Blend amount: 50 %

☑ Insert points to force a match

☑ Correct path directions first

☑ Re-encode fonts to match characters

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

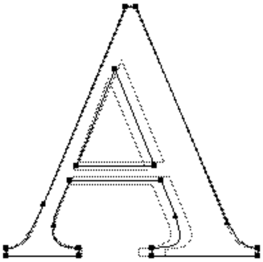
Note: If you leave the option set at **Correct Path Directions First**, Fontographer sets the rightmost point as the origin point. If you want your origin points to remain in the same place (in your new Blended font), you should leave this option deselected.

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.

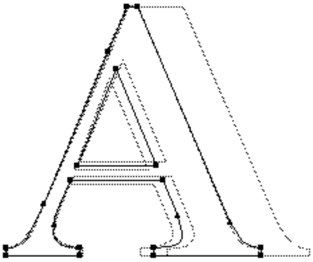
   The two source fonts are now dynamically linked to the new font. If you open any outline window in the new font, you will see both source outlines in the Template layer (see illustration below).

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



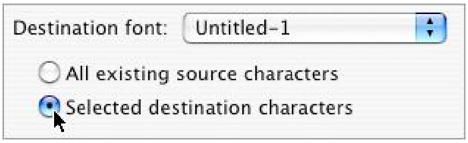
To change selected destination characters:

1. Make some changes to a character in one of your Source fonts.



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

2. Click the **Selected destination characters** radio button.



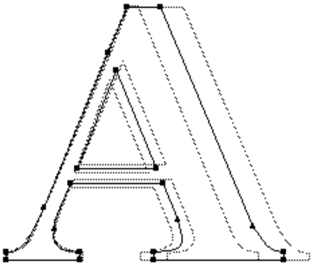
Destination font: Untitled-1

○ All existing source characters

● Selected destination characters

3. Click **OK**.

   Fontographer will automatically reblend the characters.



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

You can continue making minor tweaks to the blended character (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 characters didn’t match during blending. Consult your log file for details about the unblended characters”. When you open the text file *Fontog.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			character missing in font 2

The message “different # of paths” refers to the character decimal location in the fonts. So character 290 in Font 1 and Character 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 character 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 character such as ç is a referenced character in one font, but is an actual outline character in the other. You can correct this by using the **Unlink Reference** command on the referenced character. 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, “character missing in font 2” means that although one font has this character, the other does not, and Fontographer can’t blend what isn’t there. If the two fonts do not have the same number of character slots, your blended font will have the same number of characters as the font with the fewest character slots.

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

Just remember, the key to success using **Blend Fonts** is that your characters 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 character slots.

For information about the blending process, refer to “Font blending – the technical details” in Chapter 11, “Expert Advice”.

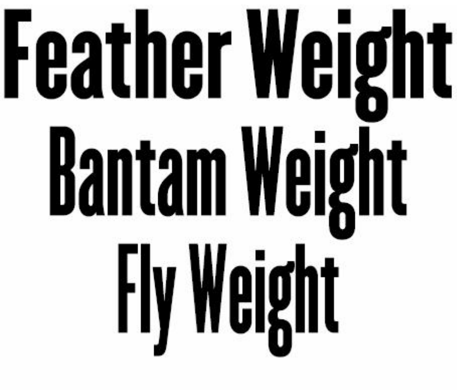
Jonathan Hoefler says:

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.

CAN'T BEAT
THE REAL THING
Enjoy
Coca-Cola
CLASSIC

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 character 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 Macromedia 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 character‘s Outline window.

   You can delete the character 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 Macromedia FreeHand or Adobe Illustrator EPS file.

   Fontographer automatically imports the image into the character’s outline window. The image will be scaled to fit between the character’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 character logo to display on screen and/or print, you may need to break the character into parts that can be placed into multiple keystrokes.

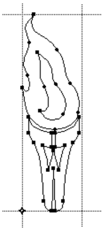
Pasting EPS outlines from the Clipboard

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

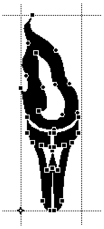
- While you are in FreeHand or Illustrator, select the graphics you wish to copy, then press the **Option** or **Control** key and select **Copy** from the **Edit** menu.
- In Fontographer, select a character slot and paste in the image.

The path data will be pasted into your character 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...

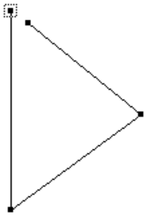
or with points turned off.

Paths and points

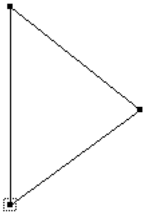
Paths are made of straight or curved line segments that are either connected or unconnected. Characters like the lowercase “i” and “j” are composed of two separate non-overlapping paths: the dots and the stems. When you join the end points of a path, you’ve closed that path.

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 176.



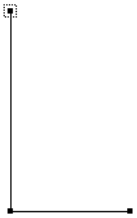
Open path



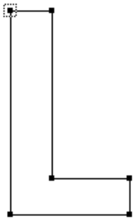
Closed path

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 character fill off, choose **Selection Info** from the **Element** menu and deselect **Fill** in the Character Information dialog box. If you want to have outline and path connections without a fill (this is a character 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 characters, refer to Chapter 3, “Creating New Fonts”.



Open path



Closed path

Path direction and fills

A character with an open path remains unfilled by PostScript, but characters 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 Character 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 Character 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 174 of this chapter).

When you choose **Selection Info**, you will only get the Character 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.

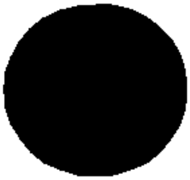
This results in normally filled characters, like in the “o” below.

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



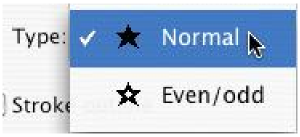
If the inside path of the character is changed to a clockwise direction (the same direction of the outside path), everything in the character becomes filled.

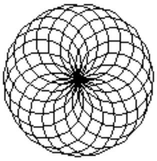
Even/odd fill

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.

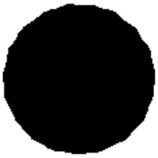
In the previous illustration of the "o", even/odd filling would give the desired result even if both paths were clockwise.

➲ ***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.

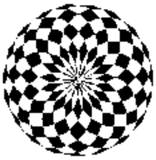




Original illustration



With a normal fill



With an even/odd fill

Correct path direction

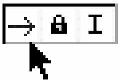
Path directions can be automatically corrected – you can tell Fontographer to examine all the path directions and reorder them if necessary. Path directions must be correctly set for proper filling of characters. Choosing **Correct Path Direction** from the **Element** menu tells Fontographer to examine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

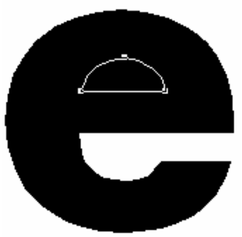
✐ ***Note:*** You can stop this operation at any time by typing **COMMAND-PERIOD**.

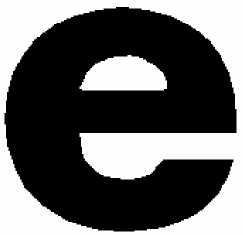
Reverse path direction

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 character.



Click the path direction indicator.





Change this character... *to this character.*

Types of points

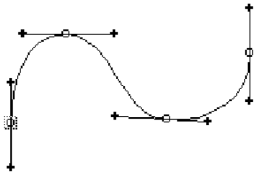
Fontographer uses three different types of points: corner points, curve points, and tangent points. Don't be confused by the different point types – like control handles, they're nothing more than different ways of working with the same basic element. The shape of characters is determined by the kinds 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 characters 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.

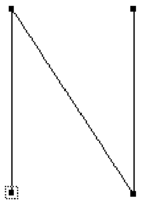


- ***Tips:*** Click the existing path with the curve point tool to add curve points.
- When you select a curve point, Fontographer displays the point as a hollow circle. You can change an existing point to a curve point by selecting **Curve Point** from the **Points** menu.
- You can use either the **Option**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

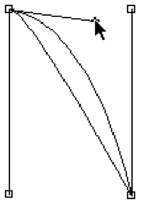
The corner point tool is used to join straight segments to curved segments or to other straight segments at an angle, or to connect two curve segments at a cusp. The corner point tool can be selected by clicking its indicator. When the lock icon is in the locked position, the numeric equivalent for choosing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



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

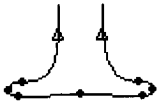
When you select a corner point, Fontographer displays the point as a hollow square. You can change an existing point to a corner point by selecting **Corner Point** from the **Points** menu.

➲ ***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.



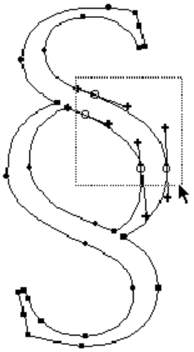
When you select a tangent point, Fontographer displays the point as a hollow triangle. You can change an existing point to a tangent point by selecting **Tangent Point** from the **Points** menu.

➲ ***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.

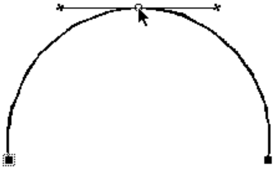
Changing a point type

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:

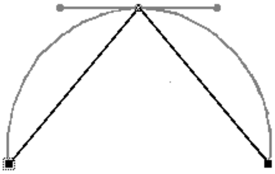
There are two ways to change point types:

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

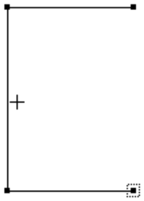
In general, with font characters and other graphic images, the fewer points you include the more graceful the image. On the other hand, there are situations when you need to add points to get more control. One professional typographer – Judy Sutcliffe – recommends roughing out characters by drawing them 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 characters, 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:

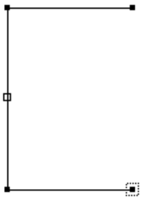
1. Select the appropriate control point tool (or the pen tool).
2. Position the pointer on the figure where the new point should be inserted (on top of a line or a curve).

   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.

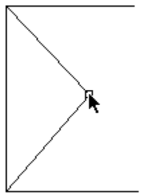
   The point is inserted into the figure at that position.



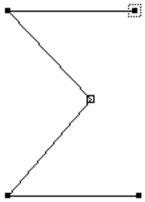
Click a path…



to place a new point.



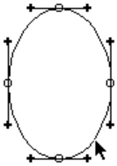
Drag the point…



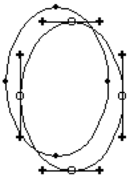
to a new location.

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.

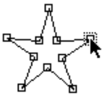


Select a path and choose ***Duplicate*** *from the* ***Edit*** *menu.*

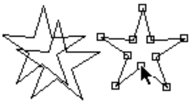


A duplicate path appears.

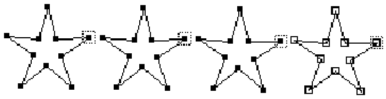
➲ ***Tip:*** If you use the Duplicate command, and move the first duplicated path immediately after the new image appears, all future duplicates will be spaced the same amount of distance from each other.



Select a path.



Duplicate the path and move it to a new position.

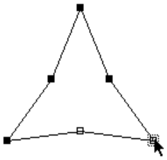


Each subsequent duplicated image is spaced the same distance.

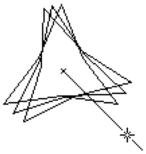
Power duplicating

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 character, and then try some combinations of your own.

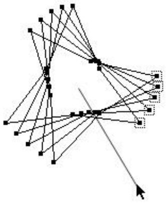
1. Create any shape.



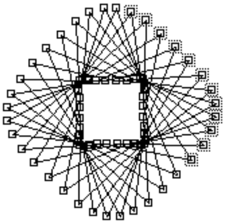
2. Choose **Clone** from the **Edit** menu.
3. Use the rotation tool on the cloned image.



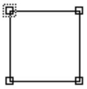
4. Select **Duplicate** from the **Edit** menu.



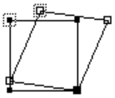
5. Repeat step 4 to make as many duplicates as you like.



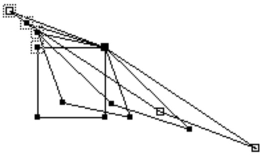
Try this with some of the other transformation tools to see what effects you can create.



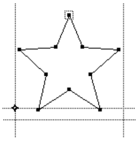
Create a square.



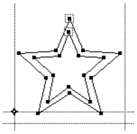
Clone and then skew it.



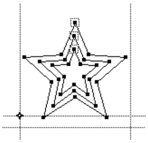
Duplicate it as many times as you like.



Create a star.



Clone and reduce it.



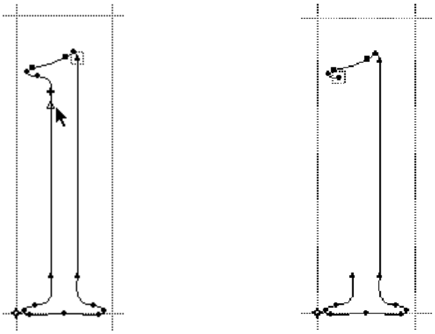
Duplicate it as many times as you like.

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.

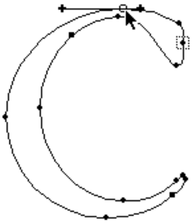
This removes active points, opening the path up if it was closed, or splitting it if it was open and the selected point was not an end point.

Splitting a path

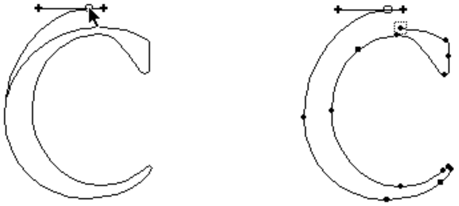
Sometimes, you'll need to split a path at a point to create either two open paths or, if you're working with a closed path, to create an open path. Simply click a point and then select **Split Points** from the **Points** menu. Fontographer splits the selected point into two separate points. Both of 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.



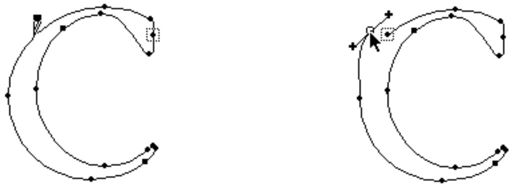
2. Choose **Split Points** from the **Points** menu. Fontographer splits the selected point into two points. The second point is hidden under the top point.
3. Press the TAB key to deselect any selected points.
4. Choose the selection pointer from the tool palette.
5. Click the split point and move it.



➲ ***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.

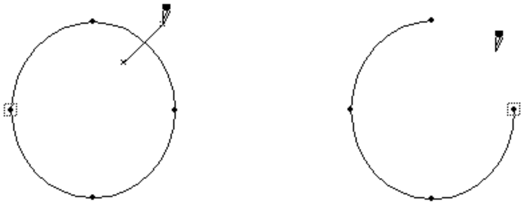
Splitting line segments

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.

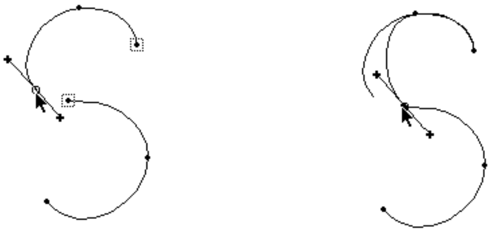
Note: Never mistake this method for **Merge Points**, found in the **Points** menu. Although related, the **Merge Points** command essentially removes any selected points on a path without breaking the path.

To join points from open paths:

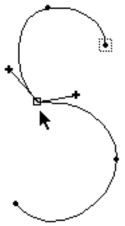
- Drag an end point of one of the paths over an end point of the other path.

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...



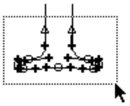
to join the paths.

Adding serifs

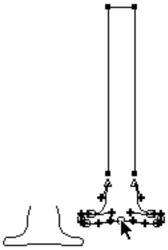
For those instances where you might want to combine the serifs from a serif typeface with your sans-serif typeface:

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 character (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.

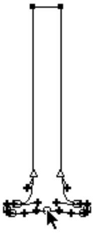
   Fontographer automatically joins the points and unites the paths.



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



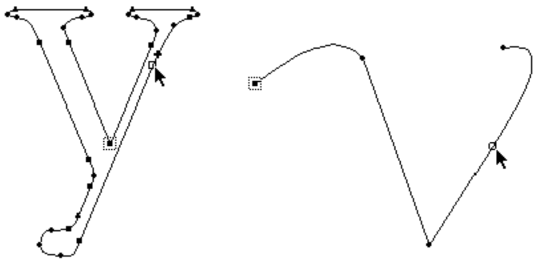
Drag the points over the points on the other path.



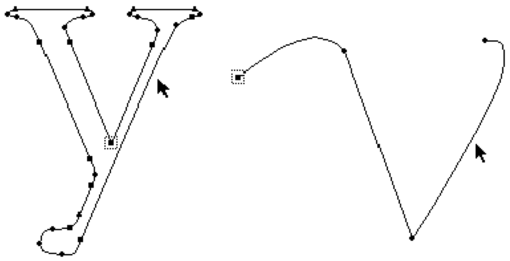
Fontographer automatically joins the points and paths.

Merging points

There will be many times when you'll want to remove excess points from a character. 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.

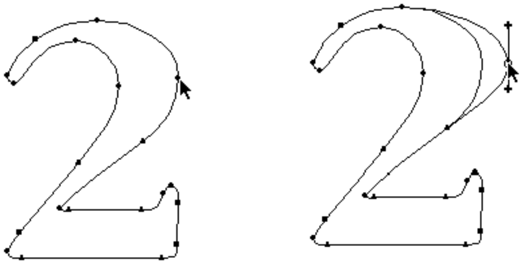
Moving a point

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.

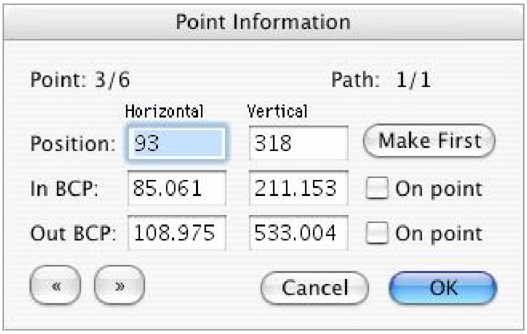
Keyboard commands to move points

- 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.



Point Information

Point: 3/6 Path: 1/1

	Horizontal	Vertical	
Position:	93	318	Make First
In BCP:	85.061	211.153	□ On point
Out BCP:	108.975	533.004	□ On point

Any changes you make in the Point Information dialog box are immediately reflected in the character 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 character 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 11, "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.

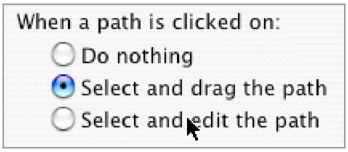
   You can also retract BCPs in the outline window with a menu command. See "Retracting BCPs" on page 203 for more information.

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 Behavior from the first pop-up in 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.



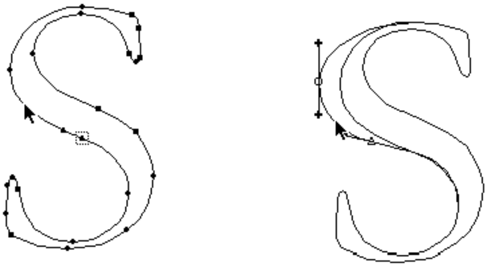
When a path is clicked on:

○ Do nothing

● Select and drag the path

○ Select and edit the path

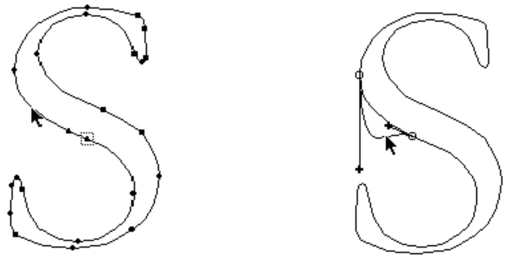
- You can drag paths as a whole. To choose this option, click the **Select and drag the path** radio button.



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.

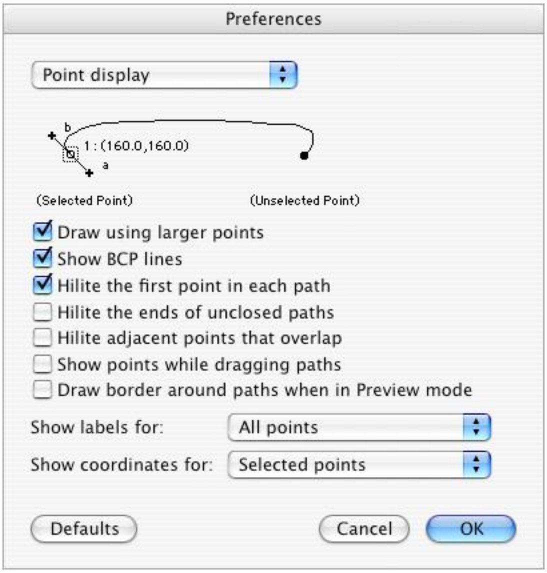
The outline of the original position of the line stays on the screen as you drag, enabling you to go back to your starting place. Of course, you can also undo these actions by choosing **Undo** from the **Edit** menu.

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 characters 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 11, "Expert Advice".



Preferences

Point display

☑ Draw using larger points
☑ Show BCP lines
☑ Hilite the first point in each path
□ Hilite the ends of unclosed paths
□ Hilite adjacent points that overlap
□ Show points while dragging paths
□ Draw border around paths when in Preview mode

Show labels for: All points

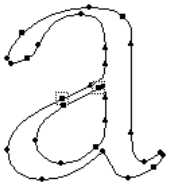
Show coordinates for: Selected points

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.

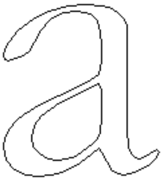
To show control points:

- 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

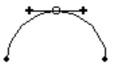
- BCPs control the shape of Bézier curves between points.
- Selecting a control point activates that point's Bézier points.
- BCPs that lie within control points may not be visible.
- There are two BCPs associated with each control point.

✐ ***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.

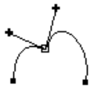
➲ ***Tip:*** If a BCP handle is not visible, you can press the **Option** key to drag the BCP out from the control point.

Dragging a control point’s BCPs

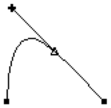
The BCPs for each of the three different kinds of points behave in different ways. A curve point’s BCPs align in a 180-degree angle, whereas a corner point’s BCPs move independently of one another. The tangent point’s BCPs move only along the line of its slope.



Curve point



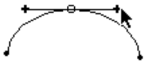
Corner point



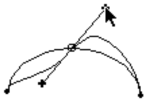
Tangent point

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.

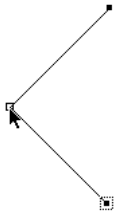
✐ ***Note:*** If a BCP and a control point lie on top of one another, any attempt to select the BCP will select the control point instead. In other words, the point is selected before the BCP is recognized (control points have selection priority over BCPs). To get around this problem, select a point, 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.

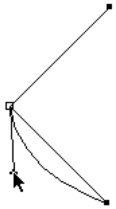
To select an incoming or outgoing BCP:

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.

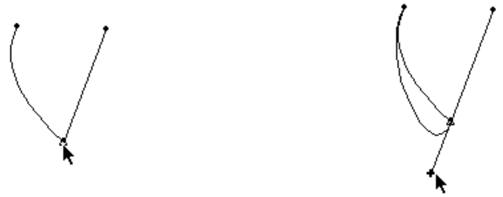


➲ ***Tips:*** Press the **Shift** key and drag to constrain BCP movement to the horizontal and vertical axes, or to 45-degree increments.

➲ 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

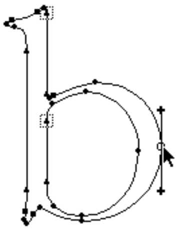
A tangent point's BCPs always constrains to the tangent (that is why they are named tangent points). Consequently, a tangent point's BCPs can never be moved away from the tangent line. Moving the BCP will not change the slope of the curve. The **Option** key can be used to select the BCPs that lie on top of a tangent point.



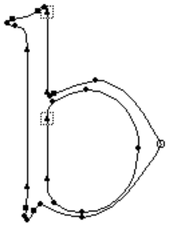
➲ ***Tip:*** If you want to extract the outgoing BCP from a tangent point, you will need to drag out the incoming BCP with the **Option**, then drag out the outgoing with the **Option**, and drag the incoming BCP back onto the point.

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.



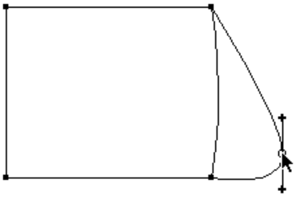
Select the control point and then choose Retract BCPs from the Points menu.



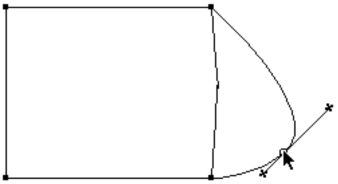
Fontographer retracts the BCPs into their point and changes the curve accordingly.

Auto curvature

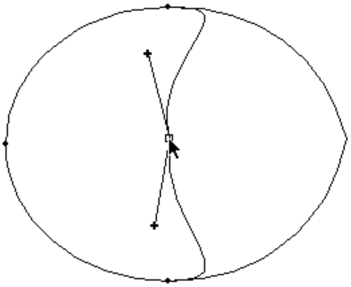
The **Auto Curvature** option instructs the curve point tool where to put the BCPs and how far to extend their handles. You can control auto curvature from the **Points** menu. Auto curvature is active primarily on curve points, but corner points can also have auto curvature if their adjacent points 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.



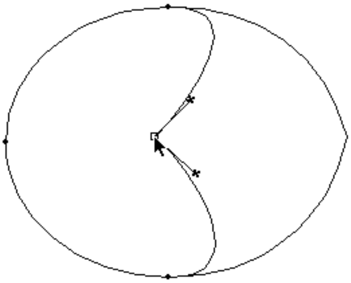
This curve point doesn't have auto curvature.



This curve point does.



This corner point doesn't have auto curvature.



This corner point does.

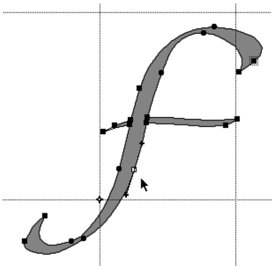
When you use the curve point tool to draw a new path, the curve points will default to have auto curvature on. To toggle it off, select **Auto Curvature** from the **Points** menu or move a BCP. When you edit a BCP, the **Auto Curvature** option turns itself off.

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**.

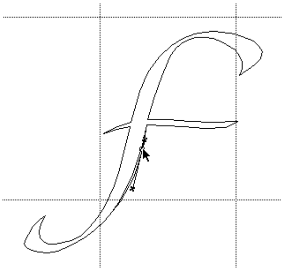
One of the most important uses for auto curvature is in tweaking paths. If you've copied a bitmap image into the template layer and autotraced it, auto curvature can make your work easier.

Here is a template image that was traced.

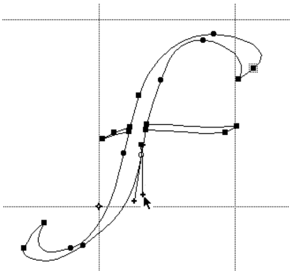
1. Move the curve point so that it will be at the extrema (the outermost edge of any curve).



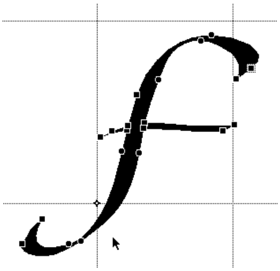
2. Select the point.
3. Select **Auto Curvature** from the **Points** menu.
4. Move the point, and the BCPs update automatically.



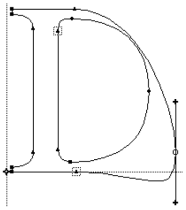
5. After you move the point, move the BCP with the **Shift** key to make it vertical (and conform to the template bitmap).



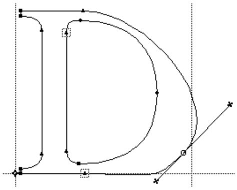
This will turn off auto curvature.



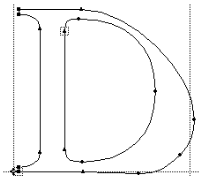
What else can you do with auto curvature? Suppose you want to make a Beer Belly font. Grab the east-most point of the bowl of letter D and pull it out and down. In the bad old days, before you could control auto curvature, it would look like this:



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 character (and font) will hint correctly.

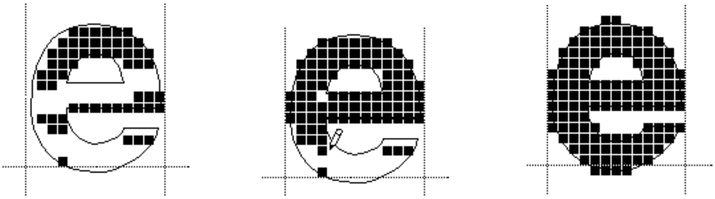
Editing Bitmaps

There are two reasons why you probably won't have to worry about bitmaps. The first is Adobe Type Manager (ATM), a program 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 advent of ATM and 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. Maybe you don't have ATM or don't want to switch to TrueType. Perhaps you'd like to create grayscale Type 3 fonts, which TrueType doesn't recognize and ATM can't contain; or you're creating professional fonts for distribution. Since ATM and TrueType don't render well at the smaller point sizes (say below 12 points), you might want to hand-edit them to achieve more readable character images. Or maybe you just like playing with pixels.



Create bitmap characters by turning pixels on and off.

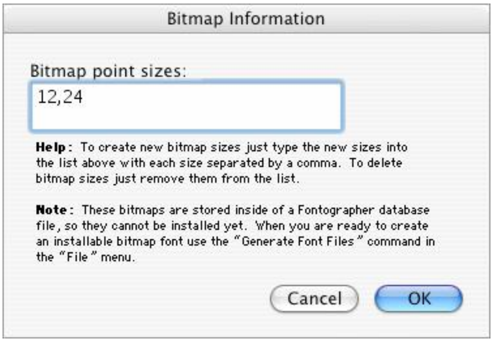
Using the bitmap window

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 character by clicking it in the font window, opening a character's outline window, or clicking a character in the metrics window.
2. Choose **Open Bitmap Window** from the **Windows** 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.



Bitmap Information

Bitmap point sizes: 12,24

Help: To create new bitmap sizes just type the new sizes into the list above with each size separated by a comma. To delete bitmap sizes just remove them from the list.

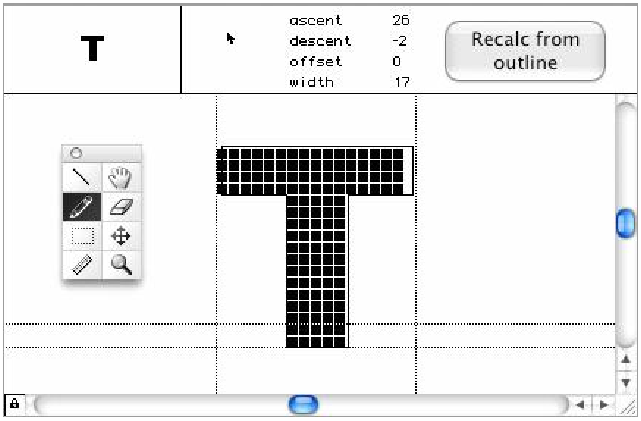
Note: These bitmaps are stored inside of a Fontographer database file, so they cannot be installed yet. When you are ready to create an installable bitmap font use the "Generate Font Files" command in the "File" menu.

   You can open multiple windows so that different characters (or even different point sizes of a single character) can be worked on at the same time. This is especially helpful when you are editing a character 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 character, and the character 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 character simply by typing the keystroke(s) of the new character. Having the lock icon closed, prevents the character from being changed to another character, if you accidentally press a keystroke.

➲ ***Tip:*** The **Enter** or **Return** key toggles the state of the lock on and off.

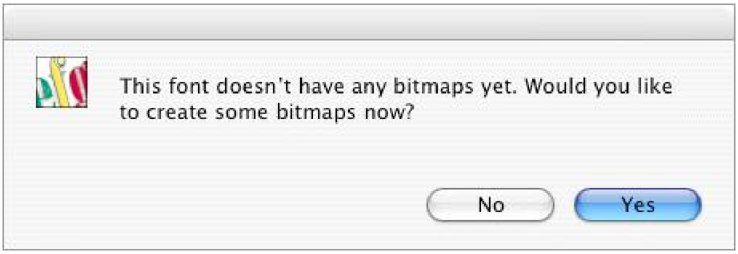
The info bar at the top of the bitmap window shows you the character's ascent, descent, offset, and width values, as well as listing the horizontal and vertical location of the cursor in pixels. These features give you the ability to precisely measure distances when you create a character. An actual point-size image of the character is displayed in the top left corner.

The **Recalc from outline** button allows you to recompute a new pixel image based on the character's outline. This button is useful when you have edited a particular outline character and don't want to recalculate the entire bitmap font. For more information, refer to "When should you recalculate bitmaps?" on page 227.

Editing a bitmap

To make changes to a bitmap:

1. Select the “A” in the Font window.
2. Choose **Open Bitmap Window** from the **Windows** menu.

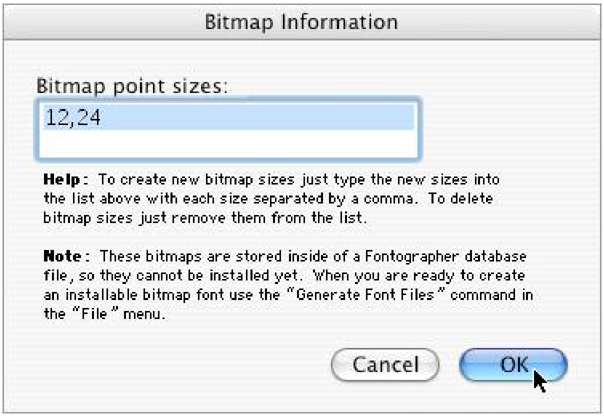


This font doesn't have any bitmaps yet. Would you like to create some bitmaps now?

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.



Bitmap Information

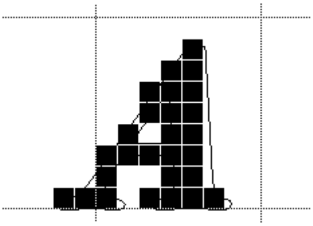
Bitmap point sizes: 12,24

Help: To create new bitmap sizes just type the new sizes into the list above with each size separated by a comma. To delete bitmap sizes just remove them from the list.

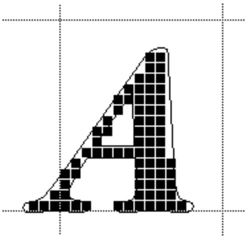
Note: These bitmaps are stored inside of a Fontographer database file, so they cannot be installed yet. When you are ready to create an installable bitmap font use the “Generate Font Files” command in the “File” menu.

4. Click **OK**.

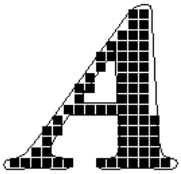
   Fontographer generates the bitmap files. The bitmap window of “A” appears.



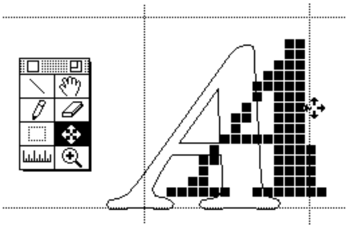
5. Choose **Next Point Size** from the **View** menu to view the next larger point size available.



6. Type **Option-G** to turn off the guidelines, if you find them distracting.

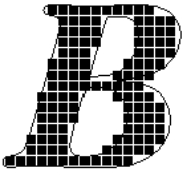


7. Select the bitmap and select the move tool to move the bitmap.

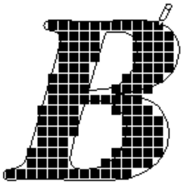


8. Choose **Next Character**.

   The character “B” appears.



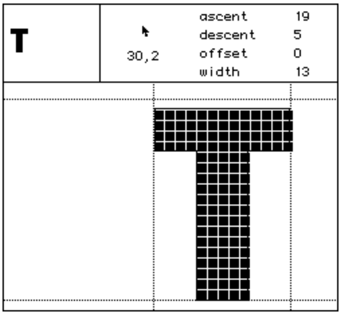
9. From this view decide which bits to change to improve the character’s looks, and then click with the pencil tool to remove or add them.



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 character to guide you in choosing the pixels that you want to turn on or off.



ascent	19
descent	5
offset	0
width	13

The Info bar shows the offset and width values for the character. The offset value will change to correspond to any changes made to the bitmap character’s offset. The width value is the character’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 **Recalc from outline** button.

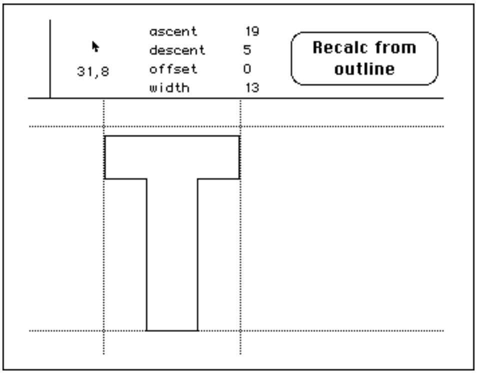
Ascent and descent values

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 19 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 five pixels.

The bitmap window in Fontographer 4.7 includes maximum ascender and descender lines. These lines represent the limits for bitmap placement based upon the outlines for all existing characters. You'll find these guides valuable when you are manually creating bitmaps. The maximum ascender line appears only when a character's outline extends below the descender line. However, if you have selected to preserve line spacing (in the Recalc Bitmaps dialog box), the maximum ascender and descender lines will not appear for that character; 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 character. If there are no points between these two, the offset is zero.



31,8

ascent	19
descent	5
offset	0
width	13

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 character in pixels: The "T" is 13 pixels wide.

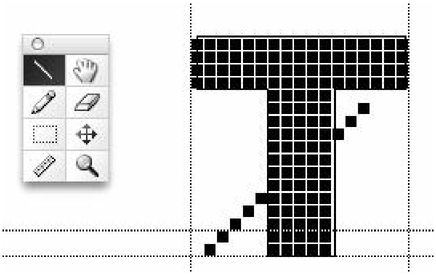
Visible layers

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 character will, however, display in the bitmap window.

Tools in the bitmap window

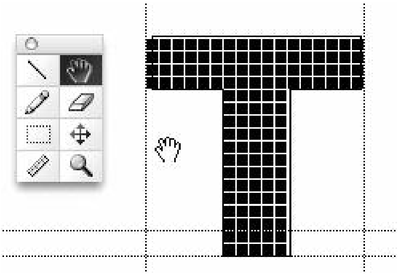
There are eight tools in the bitmap window's tool palette. The straight line tool, directly beneath the close box, allows you to draw straight bitmapped lines by holding down the mouse button and dragging. Hold down the **Shift** key to constrain the line tool to draw horizontal lines, vertical lines, 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.

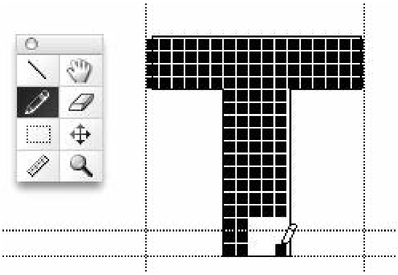
The hand tool is used to move the screen in any direction by clicking the window and dragging. The hand tool scrolls the entire character window. Move the bitmap image around to an optimum screen position before you begin to edit it. You'll need to do this if your characters are too large to fit into the central edit area.



When the lock icon is in the locked position, you can access the hand tool by typing 2.

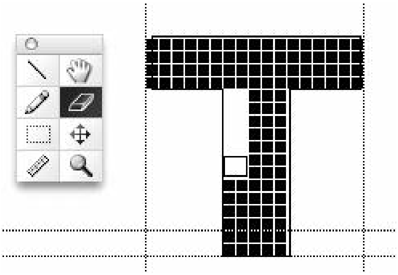
➲ ***Tip:*** You can temporarily change any of the bitmap tools to the hand tool by pressing the **SPACEBAR**.

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.

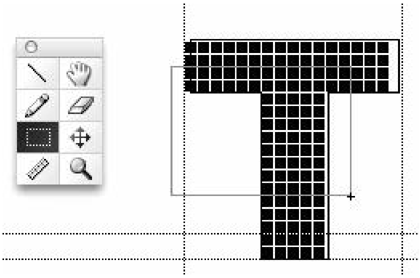
➲ ***Tip:*** Double-click the eraser tool to erase all pixels in the edit area.

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.

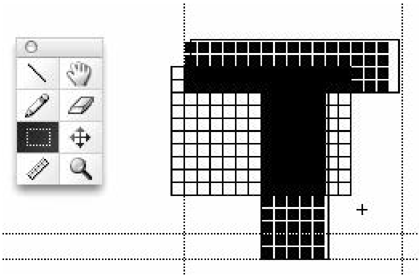
When the lock icon is in the locked position, you can access the marquee selection tool by typing 5.

➲ ***Tips:*** You can use the marquee selection tool with the standard **Cut**, **Copy**, and **Paste** commands to move or copy pixels between characters.

➲ To enclose all the bits automatically, choose **Select All** from the **Edit** menu when the marquee tool is selected.



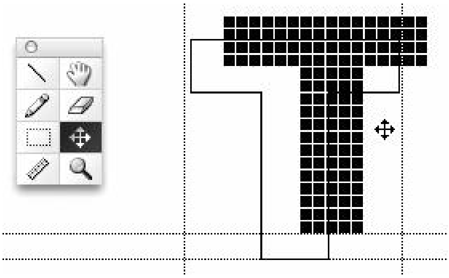
Use the marquee like a selection tool to outline an area to be moved.



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.

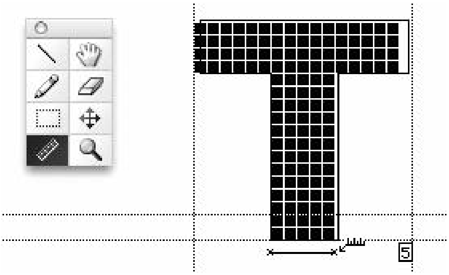
The move tool lets you move the bitmap character away from its outline in any direction.

When the lock icon is in the locked position, you can access the move tool by typing 6.



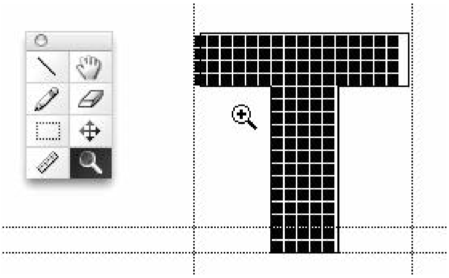
You can use the measuring tool to measure distances between pixels. When you position it and click the mouse and drag to another place, the measured distance in pixels will appear within a rectangular box. The measuring tool is automatically constrained to 45-degree angles, horizontal or vertical straight 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.

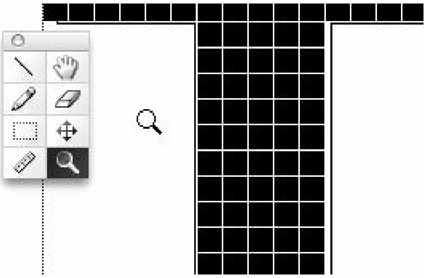
To reduce the size of the bitmap, hold down the **Option** key. The magnifying tool will display a minus sign.



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 eight, 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 does 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
Actual size	Command 1
200%	Command 2
400%	Command 4
800%	Command 8
1600%	Command 6

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, or 16 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 characters 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 character into that window. This prevents a character from being changed in case you accidentally press a character 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 character by typing that character's keystroke(s).

➲ ***Tip:*** You can toggle the lock on and off by pressing the **RETURN** or **ENTER** key.

Next and previous character

To display the next sequential character, choose **Next Character** from the **View** menu or use the shortcut **OPTION-RIGHT ARROW** To select the previous character choose **Previous Character** 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:

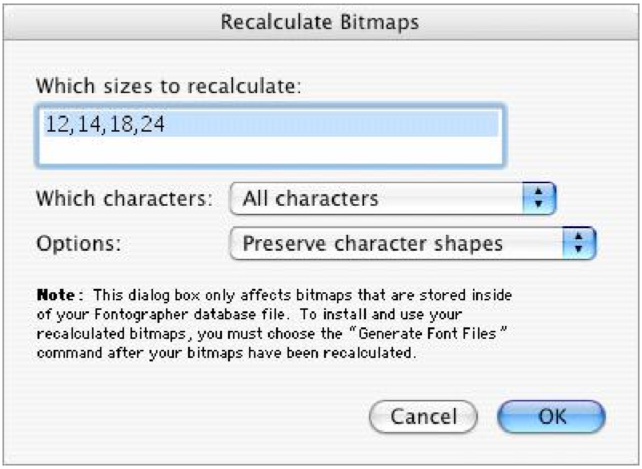
- Choose **Previous Point Size** from the **View** menu or press **COMMAND-OPTION-UP ARROW.** The next available smaller point size will automatically replace the character in the bitmap window.

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 character in the bitmap window.

When should you recalculate bitmaps?

Recalculate bitmaps from the outline when you have edited bitmap characters 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.



Which sizes to recalculate: 12,14,18,24

Which characters: All characters

Options: Preserve character shapes

Note: This dialog box only affects bitmaps that are stored inside of your Fontographer database file. To install and use your recalculated bitmaps, you must choose the "Generate Font Files" command after your bitmaps have been recalculated.

Preserving your original bitmaps

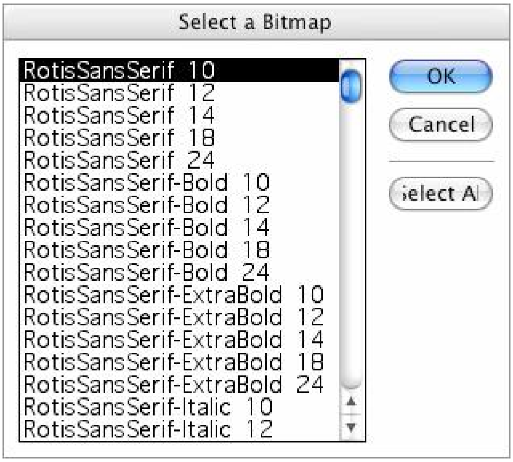
Even though you may have modified an existing font's outlines, you can still keep the original font's bitmaps if you like. This is useful in those instances where you've added some special characters to your font (fractions or accents, for example). You don't want to regenerate all the existing bitmaps, 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 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.



RotisSansSerif 10
RotisSansSerif 12
RotisSansSerif 14
RotisSansSerif 18
RotisSansSerif 24
RotisSansSerif-Bold 10
RotisSansSerif-Bold 12
RotisSansSerif-Bold 14
RotisSansSerif-Bold 18
RotisSansSerif-Bold 24
RotisSansSerif-ExtraBold 10
RotisSansSerif-ExtraBold 12
RotisSansSerif-ExtraBold 14
RotisSansSerif-ExtraBold 18
RotisSansSerif-ExtraBold 24
RotisSansSerif-Italic 10
RotisSansSerif-Italic 12

   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.

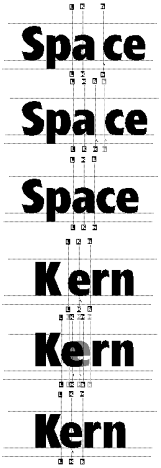
Note: You won't see the change in the font window, since the representation there is simply a rendition of the outline character.

3. Now you can go back and hand-edit your added characters 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 characters, not to provide characters 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 characters 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.

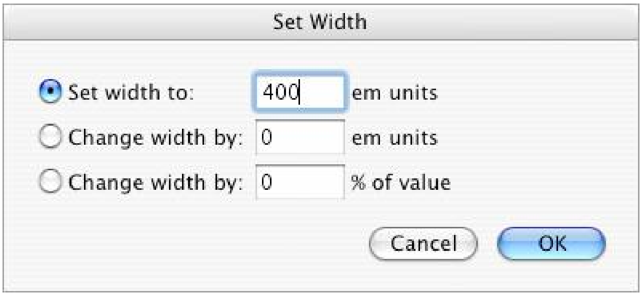
Font metrics is the term used to describe how letters are spaced when they are typed. This is easy to imagine if you remember how type used to be set. When characters were carved on the end of a piece of metal, each character's width was the width of the piece of metal on which it was carved. Setting 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 characters, 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:



Set Width

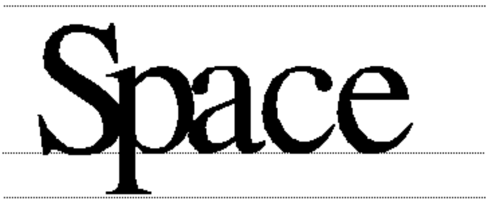
● Set width to: 400 em units

○ Change width by: 0 em units

○ Change width by: 0 % of value

Fontographer will automatically set the spacing of the characters to be a uniform width of 400 em units, producing an awful, monospaced font.

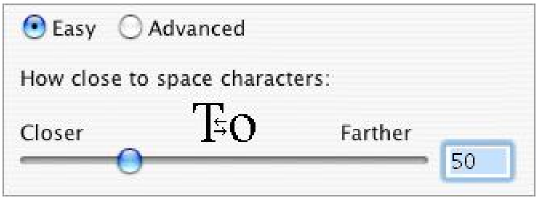
5. Click any character to deselect all.
6. Choose **Open Metrics Window** from the **Windows** menu.
7. Type the word “Space”.



Rather than manually setting different widths for each monospaced character, you can auto space the font.

8. Choose **Auto Space** from the **Metrics** menu.

   The Auto Space dialog box appears.



● Easy ○ Advanced

How close to space characters:

Closer Farther

50

   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 characters. 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.

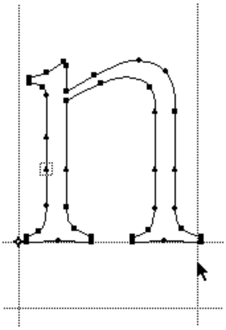
Fontographer also has auto kerning, which works in the same easy manner as auto spacing. The rest of this chapter covers both auto spacing and auto kerning in more detail.

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 character is supposed to be, so it will know how much to move over before placing the next character. 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 character to begin with respect to the one you are modifying.

➲ ***Tip:*** Usually, you should set the width of each character 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 characters, 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 characters in a font.



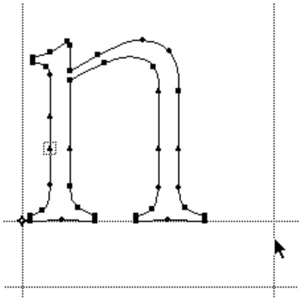
If you set the width of the “n” like this…

font

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.

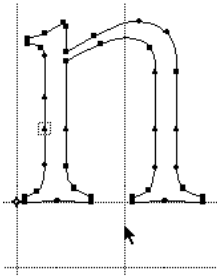
Some letters can never really be spaced correctly for all possible combinations. For example, the “T” usually presents some problems.



If you want a wider space given to the “n” and you change the width to this…

fon t

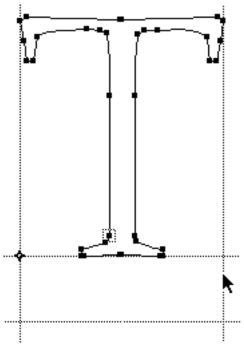
your result will look like this.



Of course, if you set the width too tight…

font

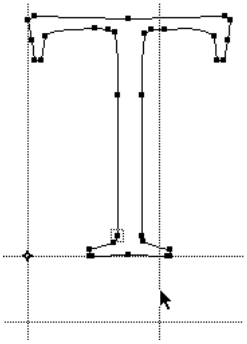
the computer will happily pile letters on top of each other, like this.



If you set the width of the “T” like this…

To

some combinations of letters might seem too loose.



Make the width narrower…

To

to fix the problem.

Now, of course, you have a new problem.

Th

Look at this…

TT

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 character 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 character'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 characters 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 character combinations which cause trouble for the spacing value you picked. Typical problem character pairs involving "T" are "Ta", "Te", "To", "Tu", "Tw", "Ty", and so on. Those character 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 character 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 characters 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 PageMaker®, 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).

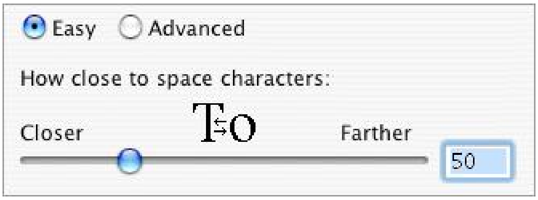
The other drawback to pair kerning is that the more kerning pairs you make for your font, the bigger and slower it becomes. Granted, it won't make the font tremendously slower or bigger, but it can become noticeable in extreme cases. Most commercial font vendors ship their fonts with between 100 and 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, 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 ªƒ, or °¢, 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. As of this writing, Fontographer is the only program we know of that is capable of doing it. 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, time-saving 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.



● Easy ○ Advanced

How close to space characters:

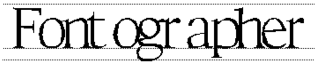
Closer

Farther

50

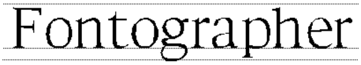
The only control you can modify in the **Easy** mode is the one that determines how close (or how far apart) the spacing should be.

Here’s an example: We opened up the font TFHabitat and demolished the spacing by setting each letter’s width to 450.

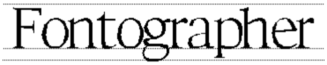


As you might expect, it looks kind of rough; however, this is what you get after having drawn a typeface without setting any spacing.

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:


Fontographer

That's a little loose for our taste; let's auto space again, this time with a value of 25:


Fontographer

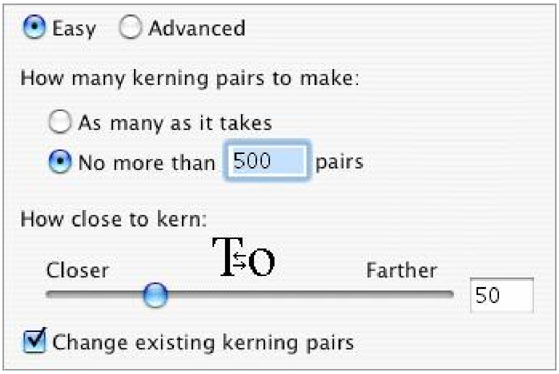
As you can see, auto spacing is as easy as choosing a value, and seeing if you like the result. Any characters that have spacing you don't like can easily be tweaked in the metrics window.

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.



● Easy ○ Advanced

How many kerning pairs to make:

○ As many as it takes

● No more than 500 pairs

How close to kern:

Closer To Farther 50

☑ Change existing kerning pairs

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.

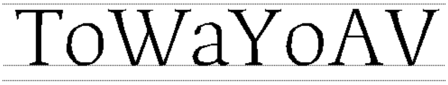
   This is a personal preference – some people like to have only 100 or 200 pairs in a font; others don't care. We recommend limiting the number of pairs to 1000 or so.
3. Next, choose a setting for how close together you want the kerning pairs to be.

   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.

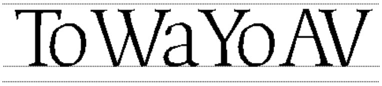
This is useful for when you have manually set some special kerning pairs, and want Fontographer to do the rest without changing your work.

➲ ***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.

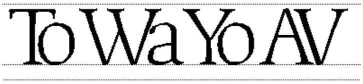


Now we’ll apply auto kerning, with a slider value of 60, which yields this:



Fontographer’s auto kerning has created a To, Wa, Yo, and AV pair. Remember, these are only the characters 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:

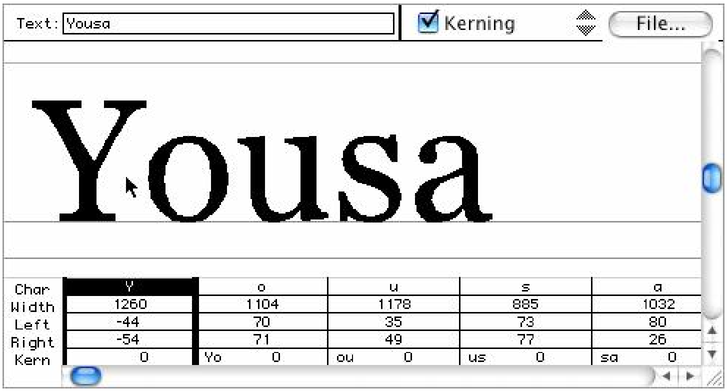


With Fontographer’s auto kerning, you can try different values and see what you get, just like with auto spacing. Also like auto spacing, you’ll find a ton of hidden ability, conveniently tucked away in the Advanced mode. The controls in the Easy mode are suitable for 90% of our customers. Don’t feel 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 characters look in relation to one another, you might expect a way to simultaneously view characters from within Fontographer. There is, and it is cleverly called the metrics window. To invoke it, simply choose **Open Metrics Window** from the **Windows** menu. The contents of the metrics window should look something like the one below.



Text: Yousa

☑ Kerning

Char	Y	o	u	s	a
Width	1260	1104	1178	885	1032
Left	-44	70	35	73	80
Right	-54	71	49	77	26
Kern	0	Yo 0	ou 0	us 0	sa 0

You enter text in the box at the top of the window labeled **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 characters; however, the practical maximum is probably considerably smaller than that. Basically, the fewer characters 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 characters at a time.

If you select some characters in the font window, and then open the metrics window, it will appear with those selected characters automatically showing in the window. To view even more characters 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 character display area, and that's where the actual characters 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 **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.

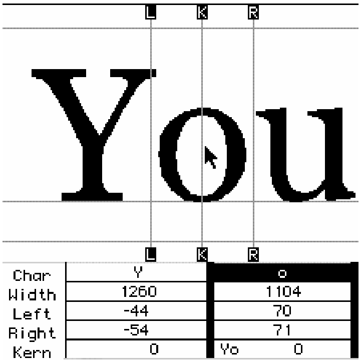
➲ ***Tip:*** This is a great way to judge how your font will look in different environments – with a quick mouse click, you can flip back and forth between modes, and see how the text will look in a word processor without kerning as compared to in a page layout program.

The **File** button, also in the upper right corner of the window, is an alternate way to enter text samples into the window. If you press the **File** button, 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 **File** button. 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.

Character display

The fun thing about the metrics character display is that all the attributes of the characters can be modified interactively with the pointer tool.



Char	Y	o
Width	1260	1104
Left	-44	70
Right	-54	71
Kern	0	Yo 0

The “L” guide is for adjusting the left sidebearing (the distance from the character origin to the beginning of the actual character outline). The guide itself is situated at the character 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 character outline to the width of the character). The guide itself is situated at the character 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 character with the previous character. That’s why, when you select the first character in the metrics window, no kerning guide appears: there is no previous character 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 character in the window. To move through a particular character’s set of undo levels, select that character and then choose **Undo** from the **Edit** menu.

➲ ***Tips:*** **OPTION**-dragging on the "L" bar keeps the left sidebearing constant as you move the character; **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 character itself around between the sidebearings is very straightforward. Simply click any character and move it left or right. If you hold down the **SHIFT** key, the character 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 character, "T" for example, and enter a character like "a" for the second character. Select character "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 Character** 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 Character** again until you see another pair you think needs kerning. This is an easy way to move quickly through all the combinations of characters.

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 character is selected, you can choose a new character 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 character will change the current selection by looking at the **Char** field in the spreadsheet. If it is black, as in the previous metrics window illustration, that means it's selected and you can change characters in this way. The general rule of thumb is that what you type from the keyboard will replace any black (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 characters with as little clutter as possible.
- If you are editing in the spreadsheet area, **OPTION-G** and **OPTION-H** will simply give you those characters rather than changing the visual states of the guidelines. To get out of the spreadsheet area, click an empty area of the character display.
- Holding down the **SPACEBAR** in the character 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 12, "Reference", for more information on magnification.

The spreadsheet area

The spreadsheet area has numeric displays for all the entities in the character display section: the character, 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 character (previous/ next), or click in another view.

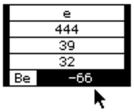
To move from cell to cell:

- 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.

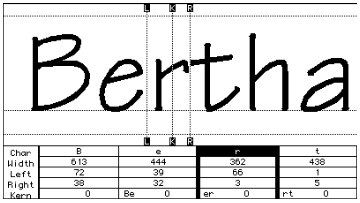
➲ ***Tip:*** To nudge values up and down, you can use the **arrow** keys: **Option-Up** or **Option-Right** or **Caps Lock+Right** moves the values up in increments of one; **Option-Down** or **Caps Lock+Down** or **Option-Left** or **Caps Lock+Left** decrements the values by one; **Shift-Option-Up** or **Shift+Caps Lock+Up** or **Shift-Option-Right** or **Shift+Caps Lock+Right** increments the values by ten; **Shift-Option-Down** or **Shift+Caps Lock+Down** or **Shift-Option-Left** or **Shift+Caps Lock+Left** decrements the values by ten. Use these shortcuts to tweak the existing values without constantly reentering numbers.

Sometimes, you will be examining a font whose characters are wider or narrower than the cell widths in the spreadsheet area, and so the character 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 character in the character display area, and its corresponding spreadsheet column will adjust itself to be directly underneath that character.

➲ ***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.

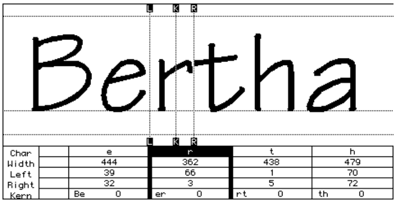


	e
	444
	39
	32
Be	-66



Char	B	e	r	t
Width	613	444	362	438
Left	72	39	66	1
Right	38	32	3	5
Kern	0	Be 0	er 0	rt 0

Here, the numbers pertaining to the "r" have drifted away from the visual representation of that letter.



Char		e	r	t	h
Width		444	362	438	479
Left		39	66	1	70
Right		32	3	5	72
Kern		Be 0	er 0	rt 0	th 0

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.

Importing metrics

When you open up an existing PostScript font, Fontographer automatically includes its spacing information, because it is located in the PostScript file. So the character 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.

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. TrueType fonts often have embedded kerning information. Fontographer will read in any existing kerning data as it opens a TrueType 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. 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
```

Characters for which there are no standard letters (like the first 20 characters or characters 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 characters in the font window, and then copy and paste their widths over a selection of other characters, without altering anything else in those other characters.

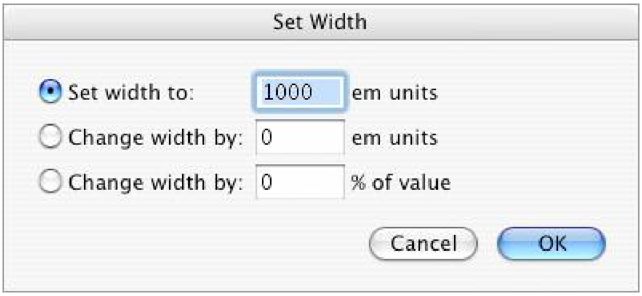
More powerful spacing and kerning commands

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 characters at once.

The **Set Width** command works with the current selection. So if you are in the font window, simply select all the characters you want to change, and then bring up the Set Width dialog box. You can simply replace each character'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.



Set Width

● Set width to: 1000 em units

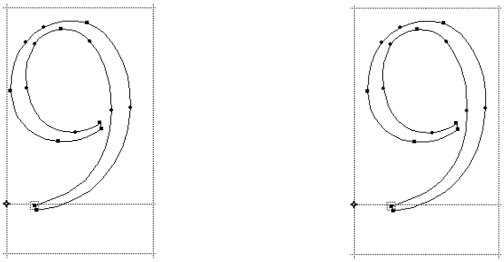
○ Change width by: 0 em units

○ Change width by: 0 % of value

Equalizing sidebearings

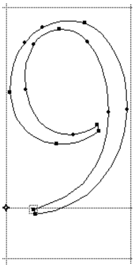
Sometimes you will want a character's left sidebearing to equal its right sidebearing, for the sake of vertical alignment. Numbers frequently appear in tabular form, and it's a lot nicer when the columns of numerals line up. In addition, some programs (like Macromedia FreeHand) can vertically align arbitrary text.

Equalizing the sidebearings is easy: choose **Equalize Sidebearings** from the **Metrics** menu.



Choose ***Equalize Sidebearings*** *from the* ***Metrics*** *menu to make this character (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.*

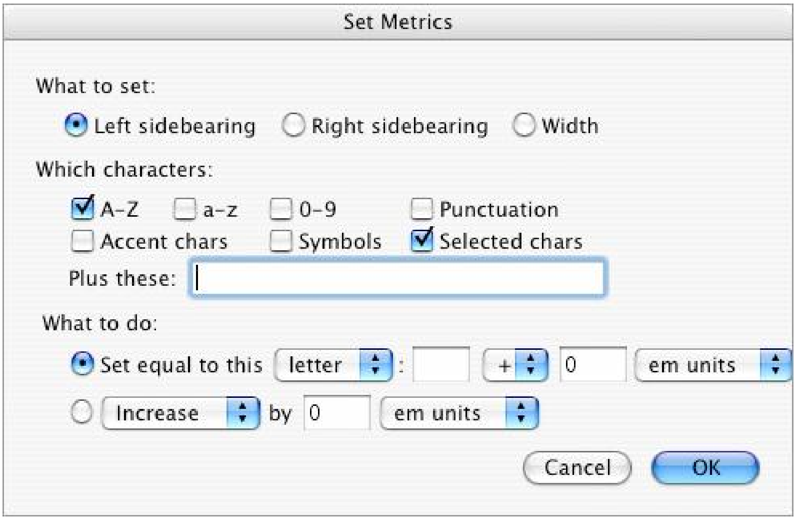
➲ ***Tip:*** If you have some points selected, Equalize Sidebearings will center just those points between the origin and the width, rather than the entire character. This makes for a useful centering command, but be aware that the character's sidebearings will probably not be equalized in this case.

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 characters, and there are lots of controls for incrementally increasing and decreasing various fields. Here is what the Set Metrics dialog box looks like:



What to set:

● Left sidebearing ○ Right sidebearing ○ Width

Which characters:

☑ A-Z □ a-z □ 0-9 □ Punctuation

□ Accent chars □ Symbols ☑ Selected chars

Plus these:

What to do:

● Set equal to this letter : + 0 em units

○ Increase by 0 em units

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 characters you wish to apply the changes to. We have found that **Selected chars** is what we normally choose. So, you can go to the font window, select just those characters you want to adjust, and then choose only **Selected chars** in the **Set Metrics** command.

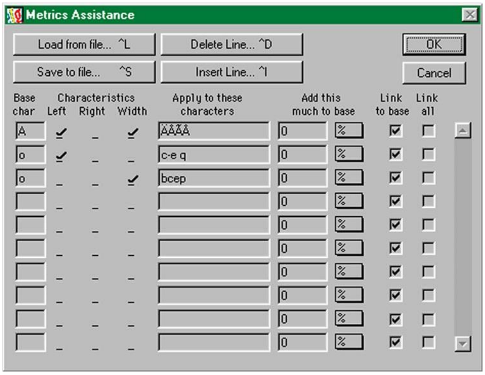
Finally, you choose what changes to make. The first line is for setting absolute values. You can set character's attributes equal to those of a different character, or equal to a specific value. You can then add or subtract either an absolute value or a percentage. For example, you could set the left 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.



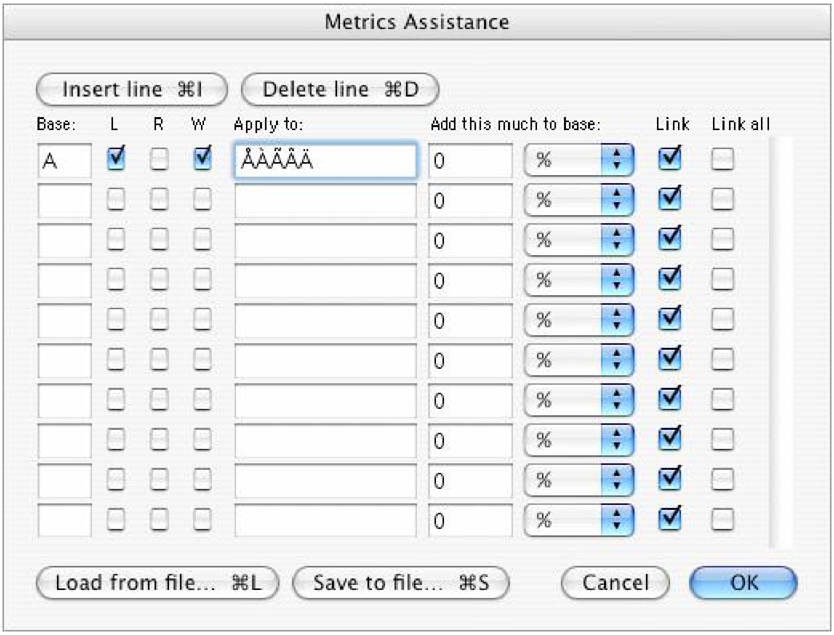
Base char	Characteristics Left	Right	Width	Apply to these characters	Add this much to base		Link to base	Link all
A	✓	_	✓	ÄÅÃÅ	0	%	☑	□
o	✓	_	_	c-e q	0	%	☑	□
o	_	_	✓	bcep	0	%	☑	□
	_	_	_		0	%	☑	□
	_	_	_		0	%	☑	□
	_	_	_		0	%	☑	□
	_	_	_		0	%	☑	□
	_	_	_		0	%	☑	□
	_	_	_		0	%	☑	□
	_	_	_		0	%	☑	□

What we call assisted metrics, other people might refer to as equivalence classes. An equivalence class is a rule that dictates what other characters' 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:



Base:	L	R	W	Apply to:	Add this much to base:		Link	Link all
A	☑	□	☑	ÅÀÃÂÄ	0	%	☑	□
	□	□	□		0	%	☑	□
	□	□	□		0	%	☑	□
	□	□	□		0	%	☑	□
	□	□	□		0	%	☑	□
	□	□	□		0	%	☑	□
	□	□	□		0	%	☑	□
	□	□	□		0	%	☑	□
	□	□	□		0	%	☑	□
	□	□	□		0	%	☑	□

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 characters to one another.

In the left-hand column, you choose the base character. This is the character 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 character.

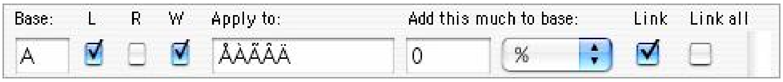
Now you can optionally set up some difference (in value) to be applied to the base character's attributes. For instance, you could create an equivalence class that means certain characters' widths will be equal to a base character's width, plus 10 percent.

Next, you specify whether the class should be linked. If the **Link to base** checkbox is checked, whenever the values in the base character change, the corresponding values in all the other characters 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 character's controlled attributes change, all other character's attributes will be automatically updated (including the base character).

Using these definitions, let's explore the sample equivalence class:

Our base character is "A". That means the attributes of the "A" will determine what the other character's attributes are. We chose to control both the left sidebearing and the width.



Base:	L	R	W	Apply to:	Add this much to base:		Link	Link all
A	☑	□	☑	ÅÀÃÂÄ	0	%	☑	□

✐ ***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 characters 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 character. 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 character “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 characters will update automatically.

You can get pretty imaginative in figuring out which characters 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 characters 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 characters. 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 characters.

Removing equivalence classes is quite straightforward: simply select the base character, 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 characters field. This happens when you had originally entered something like “adefghijklmz”. Fontographer will detect contiguous characters 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 characters.

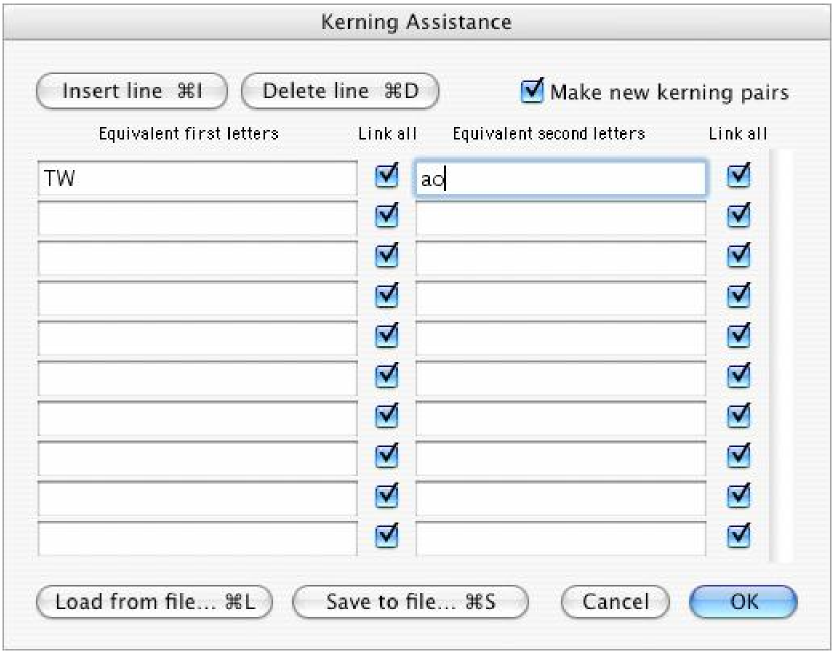
The same base character can be involved in different classes. For instance, one class might connect the left sidebearing of the "A" with a bunch of other characters, while another class could connect the width of the "A" to some different characters. If you need to connect "A" with more characters than will fit in the Apply to these characters field, it is perfectly legal to make another class, with the same base character and same characteristics, to continue the first class. Of course, base characters for some classes can be included in the Apply to these characters parts of other classes, but watch out – this can get confusing really fast.

If you include the base character in the Apply to these characters 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:



☑ Make new kerning pairs

Equivalent first letters	Link all	Equivalent second letters	Link all
TW	☑	ao	☑
	☑		☑
	☑		☑
	☑		☑
	☑		☑
	☑		☑
	☑		☑
	☑		☑
	☑		☑
	☑		☑

Just like in the Metrics Assistance dialog box, the **Insert Line** and **Delete Line** commands allow for intuitive grouping when relating characters to one another.

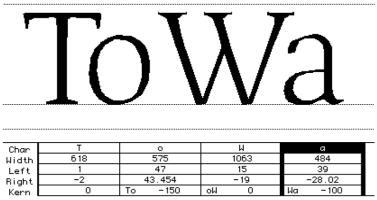
The first column of equivalence classes are those characters that kern the same when they are the first character of a kern pair. (In other words, they will each get all the same kerning companions that the others have – for instance, if the font has the two kerning pairs “Te” and “Wy”, it will have 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.

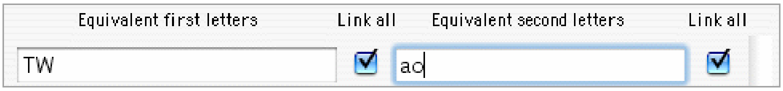
The **Link all** checkboxes perform a function similar to what they do in metrics assistance. When these boxes are checked, any changes to kern pairs involving the members of that class will cause the other kerning pairs derived from the class to update automatically.

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 already existing –“To” and “Wa:”



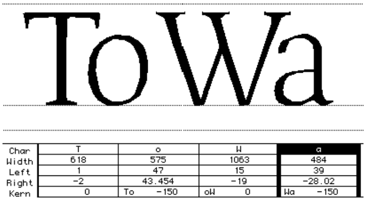
Char	T	o	W	a
Width	618	575	1063	484
Left	1	47	15	39
Right	-2	43.454	-19	-28.02
Kern	0	To -150	oW 0	Wa -100

When we apply this set of kerning equivalence classes...



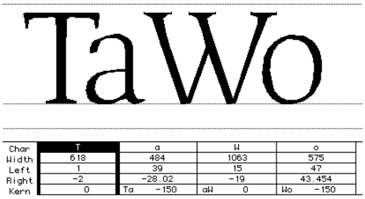
Equivalent first letters	Link all	Equivalent second letters	Link all
TW	☑	ao	☑

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

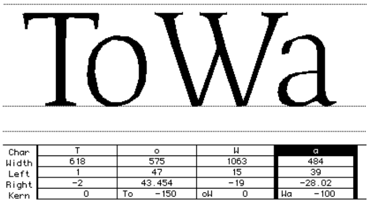
Fontographer has made the “To” and “Wa” kerning pair values equivalent. They are both now set at -150. In addition, Fontographer has also created the “Ta” and “Wo” kerning pairs below:



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

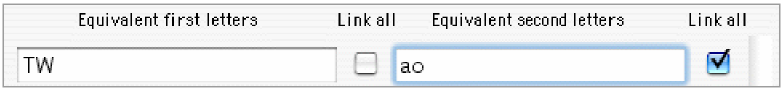
In summary, we began with To = -150, Wa =-100. We ended with To = Wa = Ta = Wo = -150. Since the characters were all linked, they all became equal.

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:



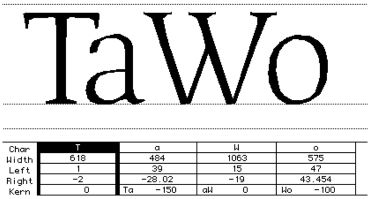
Char	T	o	W	a
Width	618	575	1063	484
Left	1	47	15	39
Right	-2	43.454	-19	-28.02
Kern	0	To -150	oW 0	Wa -100

Now, however, we will apply this slightly changed set of kerning equivalence classes:



Equivalent first letters	Link all	Equivalent second letters	Link all
TW	☐	ao	☑

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:

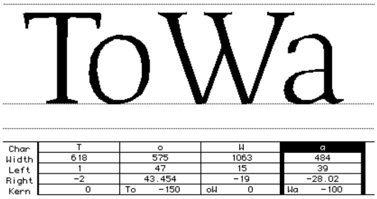


Char	T	a	W	o
Width	618	484	1063	575
Left	1	39	15	47
Right	-2	-28.02	-19	43.454
Kern	0	Ta -150	aW 0	Wo -100

Note that this time, the values of “Ta” and “Wo” are not the same.

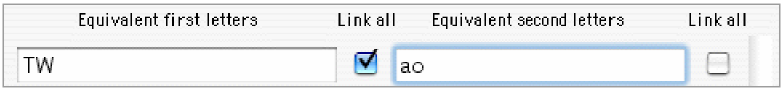
In summary, we began with To = -150, Wa =-100. We ended with To = Ta = -150, and Wa = Wo = -100. This class sort of says anything-with-o will be equal to that-same-anything-with-a.

Now let’s try one more permutation. We’ll again start from scratch, with our trusty old “To” and “Wa” pairs:



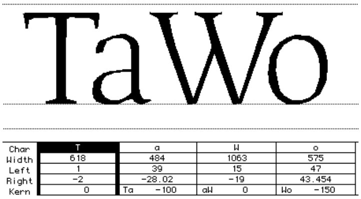
Char	T	o	W	a
Width	618	575	1063	484
Left	1	47	15	39
Right	-2	43.454	-19	-28.02
Kern	0	To -150	oW 0	Wa -100

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	aW 0	Wo -150

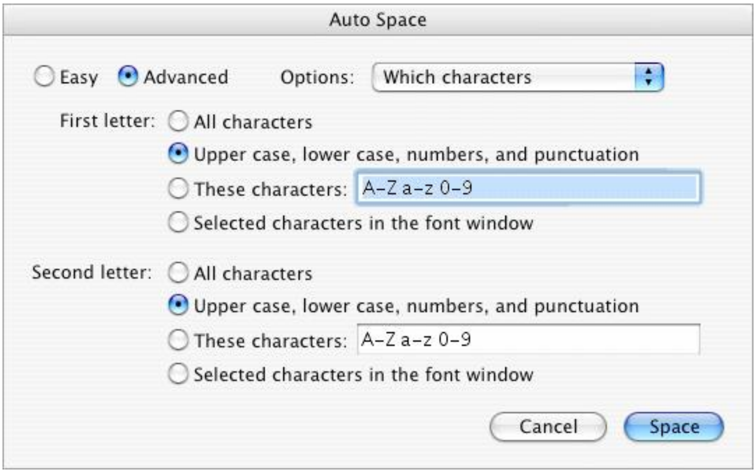
This time, however, the “Ta” got the value of -100 and “Wo” got -150. This is because the “T” and “W” are linked, but the “a” and “o” are not.

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 characters dialog box.



Auto Space

○ Easy ● Advanced Options: Which characters

First letter:
○ All characters
● Upper case, lower case, numbers, and punctuation
○ These characters: A–Z a–z 0–9
○ Selected characters in the font window

Second letter:
○ All characters
● Upper case, lower case, numbers, and punctuation
○ These characters: A–Z a–z 0–9
○ Selected characters in the font window

The **Which characters** dialog box allows you to tell Fontographer which characters should get their widths set and which characters should be considered when choosing optimal widths. The **First letter** field defines which characters should get new widths. The **Second letter** field describes which companion characters Fontographer should consider.

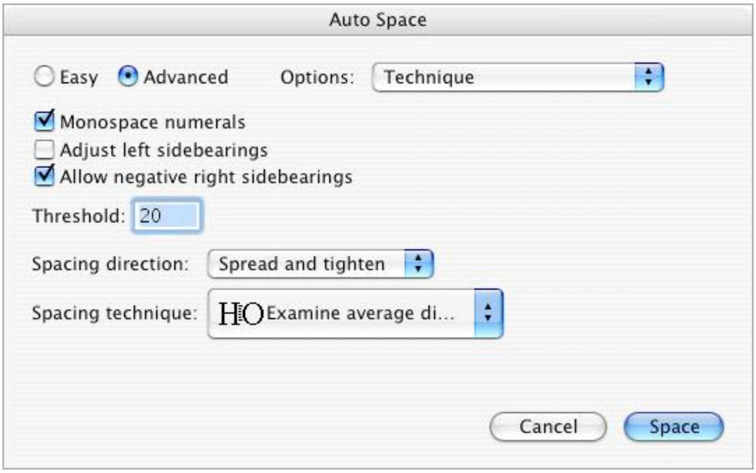
You may not want to have Fontographer set the widths of every character in the font. Suppose you have manually set the widths for the uppercase and lowercase characters. By selecting other characters in this field, you can have Fontographer space the rest of the font but not touch the characters you 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 character 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:



Auto Space

○ Easy ● Advanced Options: Technique

☑ Monospace numerals

□ Adjust left sidebearings

☑ Allow negative right sidebearings

Threshold: 20

Spacing direction: Spread and tighten

Spacing technique: HO Examine average di...

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 characters 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 character outlines extend to the right of the width. This will generally allow tighter spacing, but can in some cases cause certain characters to touch each other.

The **Threshold** field contains a value that influences the grouping of sets of character 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 characters narrower (tighten), only by making characters wider (spread), or by doing whatever is needed (spread and tighten).

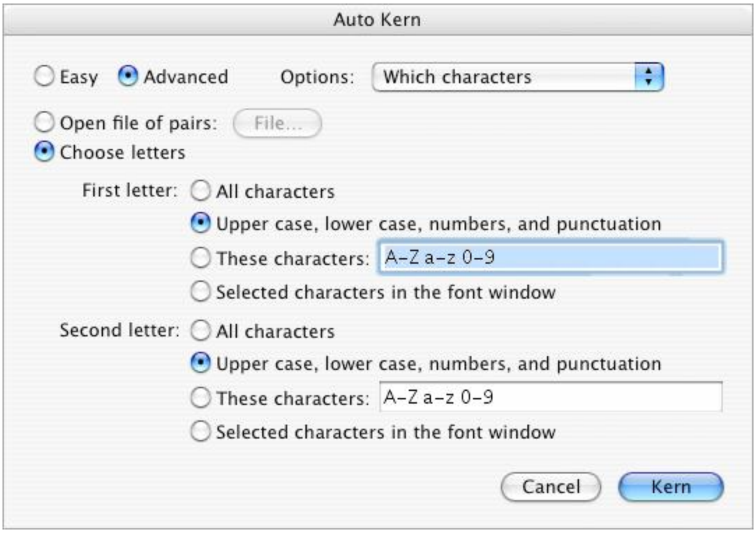
Spacing technique is another difficult-to-document feature. Examine minimum distance tells Fontographer to calculate spacing by looking only at the smallest distance two characters are from each other. For serif fonts, this can result in loose spacing because serifs frequently come close to touching 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 character 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 character 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 characters**, and it looks something like this:



Auto Kern

○ Easy ● Advanced Options: Which characters

○ Open file of pairs: File...
● Choose letters

First letter: ○ All characters
● Upper case, lower case, numbers, and punctuation
○ These characters: A-Z a-z 0-9
○ Selected characters in the font window

Second letter: ○ All characters
● Upper case, lower case, numbers, and punctuation
○ These characters: A-Z a-z 0-9
○ Selected characters in the font window

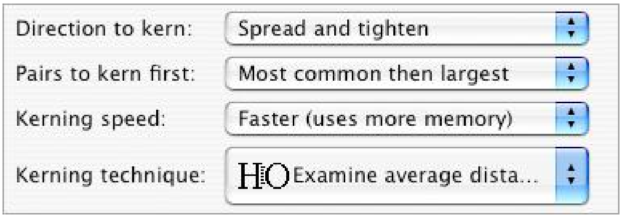
Use this dialog box to tell Fontographer which characters to auto kern. The first option, Open file of pairs, allows you to select a text file of kerning pairs. Once you do that, Fontographer will do its normal auto kerning functions, but it will only create kerning pairs for the ones specified in that 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 characters to kern. For instance, under First letter, you could simply enter “T” in the **These characters** field, and for the **Second letter** field, choose **All characters**. 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 characters (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 characters 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:



Direction to kern: Spread and tighten

Pairs to kern first: Most common then largest

Kerning speed: Faster (uses more memory)

Kerning technique: Examine average dista...

The **Technique** dialog box allows you to choose different kerning techniques.

Direction to kern 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.

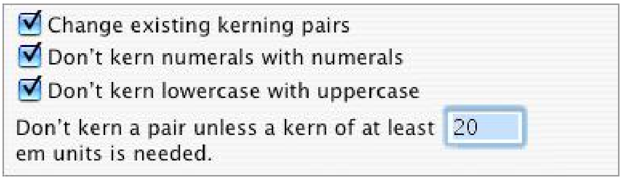
Largest pairs first will cause Fontographer to order the 2500 pairs it found from largest value to smallest, and output the 500 largest values. This is a "sure-fire" way to set the letter combinations that need kerning the most; however, you will find that many or most of these are often the goofy symbols characters, or punctuation characters (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 characters. Examine minimum distance tells Fontographer to calculate kerning by looking only at the smallest distance two characters 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 character 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 characters 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:



☑ Change existing kerning pairs
☑ Don't kern numerals with numerals
☑ Don't kern lowercase with uppercase
Don't kern a pair unless a kern of at least 20 em units is needed.

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 character is lowercase and the second character 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 characters 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 characters, but you also might want to have 10 or 20 pairs involving the accent characters 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.

Printing

You’ve just created a font, or perhaps just a few characters, and now you would like to see those characters 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.

7

Fontographer gives you the choice of printing a variety of samples:

- All characters in the font
- Individual characters
- Specific typed text
- Text from a file
- A PostScript sample with several lines of text
- A key map, showing all the characters in the font and their respective widths, codes, and offset specifications
- A complete list of all the kerning pairs in the font
- A character sample in assorted sizes or just one giant character
- A sample showing all of the points in a selected character, 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 character 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.

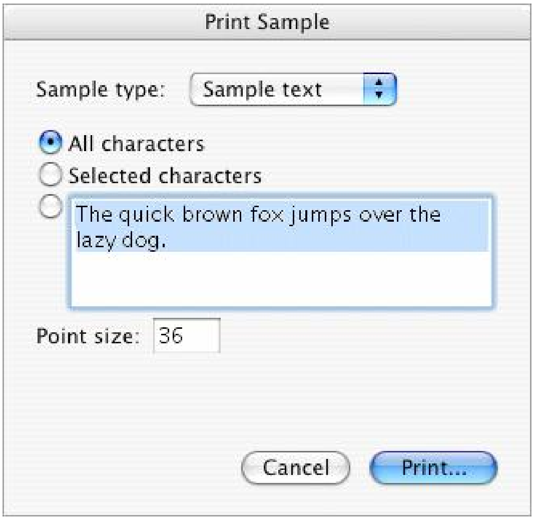
Sample text

The first choice in the **Sample type** pop-up is Sample text, which provides three printing options.

Choosing **All characters** prints all the characters of the font at the point size you designate.

To print a sample of all characters:

1. Choose **Print** from the **File** menu.
2. Choose Sample text from the **Sample type** pop-up.
3. Click the **All characters** radio button in the Print Sample dialog box.
4. Click **Print**.



Print Sample

Sample type: Sample text

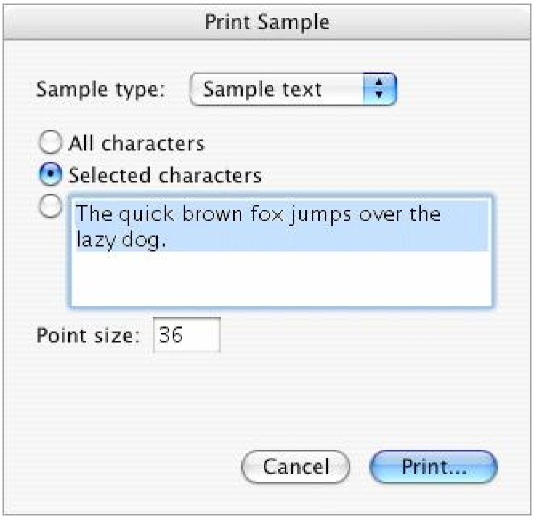
● All characters
○ Selected characters
○ The quick brown fox jumps over the lazy dog.

Point size: 36

Choosing **Selected characters** allows you to print characters selected in the Font window at a designated point size.

To print selected characters:

1. In the Font window, select the characters you wish to print.
2. Choose **Print** from the **File** menu.
3. Choose Sample text from the **Sample type** pop-up.



Print Sample

Sample type: Sample text

○ All characters

● Selected characters

○ The quick brown fox jumps over the lazy dog.

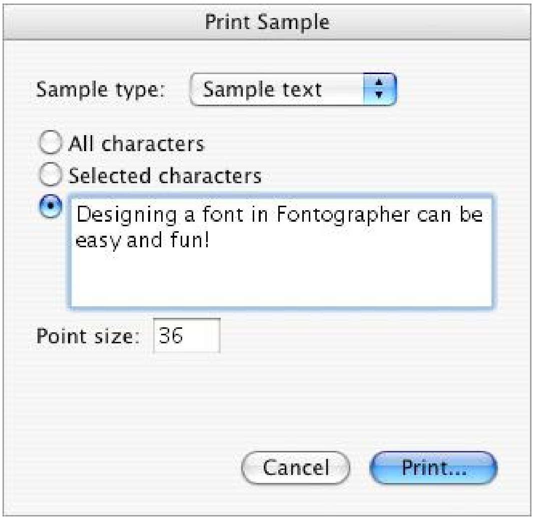
Point size: 36

4. Click the **Selected characters** 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 to the left of the text sample box.)



Sample type: Sample text

○ All characters

○ Selected characters

● Designing a font in Fontographer can be easy and fun!

Point size: 36

4. Type your text into the text box.

   Fontographer allows you to type up to 256 characters 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**.

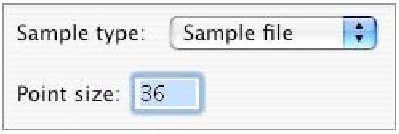
Note: The text box is a good place to test logos, kerning pairs, or character spacing in specific word and sentence combinations.

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.



Sample type: Sample file

Point size: 36

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.

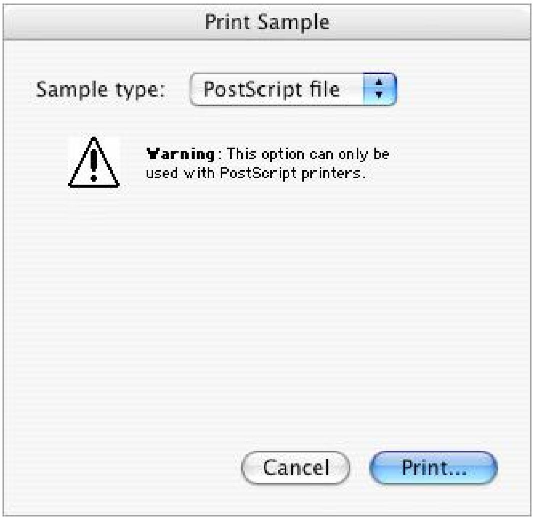
Note: Fontographer can print only text files as samples. This means Fontographer cannot print normal word processor files. You must first resave these files as Text files before Fontographer can print them.

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 4.7. You may use them as is, or you may edit them with any text editor to define your own custom print sample.



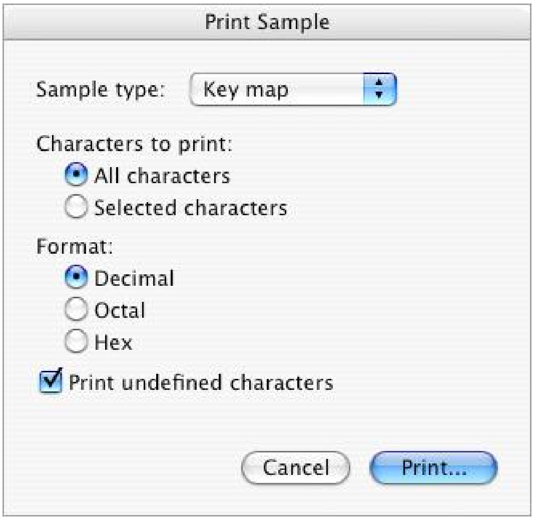
Print Sample

Sample type: PostScript file

Varning: This option can only be used with PostScript printers.

Key map

The Key map option allows you to print a sample for the entire font or only characters that you have selected.



Print Sample

Sample type: Key map

Characters to print:

● All characters
○ Selected characters

Format:

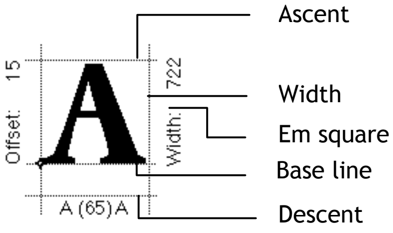
● Decimal
○ Octal
○ Hex

☑ Print undefined characters

Cancel Print...

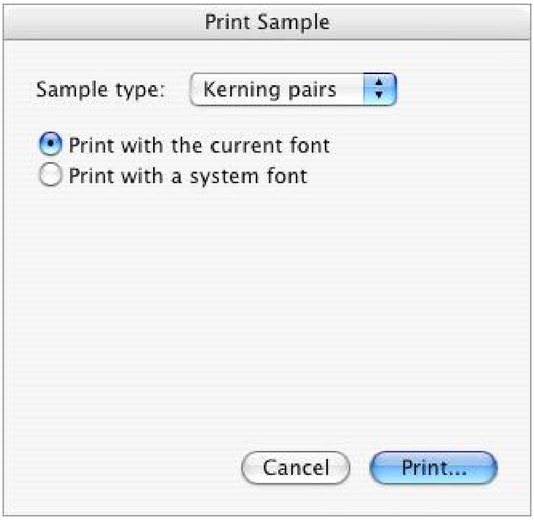
If you select the All characters option, Fontographer prints pages containing all the characters in the font. The printed pages consist of rows of characters, along with their offsets, widths, and corresponding key stroke codes. Fontographer gives you the option of showing decimal, octal, or hexadecimal locations. The Print undefined characters option allows you to print or omit undefined characters from your printout. If printed, these characters will be surrounded by a gray box, their decimal, octal, or hexadecimal code, and the word Undefined.

A character 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.

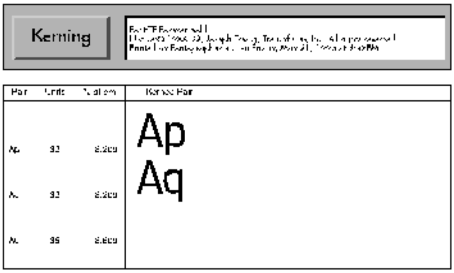


Print Sample

Sample type: Kerning pairs

● Print with the current font
○ Print with a system font

Choose Kerning Pairs...



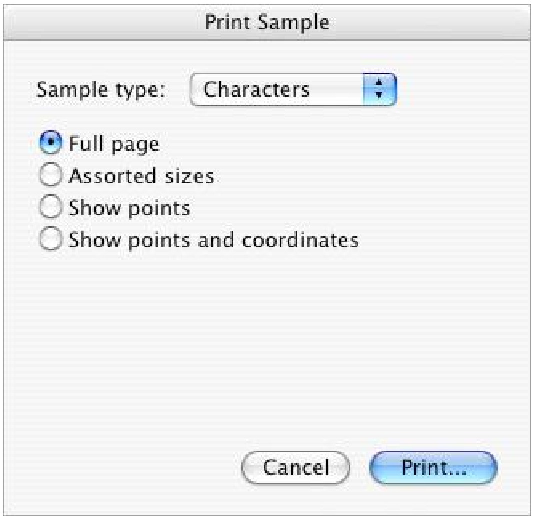
to print a kerning pairs list.

Characters

The Characters sample type option provides you with four different choices: printing one full-page-sized filled character, several assorted sizes of the filled character, a sample that shows the points of the character, or a sample that shows both the points and x/y coordinates for each point.

To print a full-page character sample:

1. Select the character(s) you want to print from the Font window, or open an Outline window for the character you want to print.
2. Choose **Print** from the **File** menu.
3. Choose Characters from the **Sample type** pop-up.

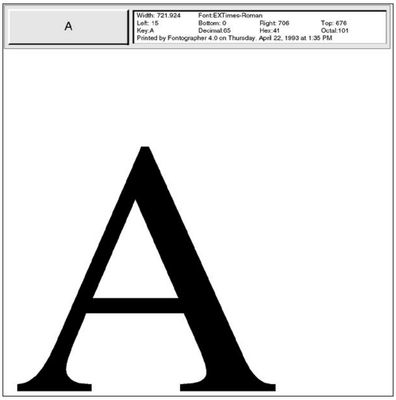


Sample type: Characters

● Full page
○ Assorted sizes
○ Show points
○ Show points and coordinates

4. Click the **Full page** radio button.
5. Click **Print**.

This is the quickest way to print a single character.



A

Width: 721.924	Font:EXTimes-Roman		
Left: 15	Bottom: 0	Right: 706	Top: 676
Key:A	Decimal:65	Hex:41	Octal:101

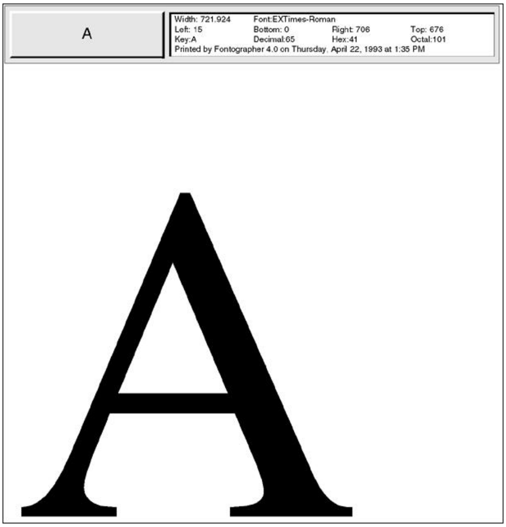
Printed by Fontographer 4.0 on Thursday, April 22, 1993 at 1:35 PM

To print a text sample in assorted sizes:

1. Select the characters you want to print from the Font window. Use any letters you want to see printed or open an Outline window for the character you want to print.
2. Choose **Print** from the **File** menu.
3. Choose Characters 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



A

```
Width: 721.924    Font:EXTimes-Roman
Left: 15          Bottom: 0          Right 706     Top: 676
Key:A             Decimal:65         Hex:41        Octal:101
Printed by Fontographer 4.0 on Thursday, April 22, 1993 at 1:35 PM
```

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 characters, the header will read Selected, and so on. An individual character such as the letter “S” will only have the character 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.

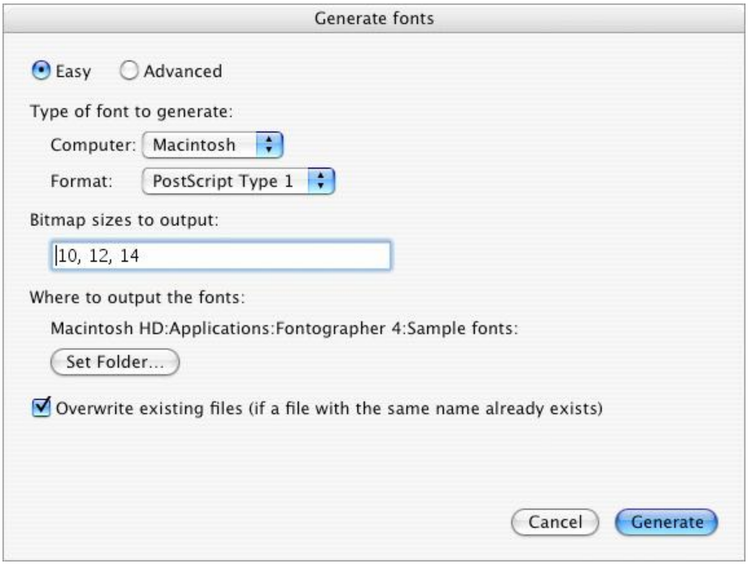
Before you do anything…

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 4.7 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 windows by** option to File name. This enables you to know at a glance which file you're working on.

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.



Generate fonts

● Easy ○ Advanced

Type of font to generate:

Computer: Macintosh

Format: PostScript Type 1

Bitmap sizes to output:

10, 12, 14

Where to output the fonts:

Macintosh HD:Applications:Fontographer 4:Sample fonts:

☑ Overwrite existing files (if a file with the same name already exists)

Time for a pop quiz: What encoding do Sun fonts require? What em square does? Why Quark™/Win prefer for TrueType fonts? 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 computer for which you are preparing the font.
2. Determine the format (PostScript or TrueType).
3. Choose the bitmap sizes – if any.

✐ ***Note:*** Windows fonts do not need any bitmaps, nor do Macintosh TrueType fonts. Macintosh PostScript fonts, however, require bitmaps.

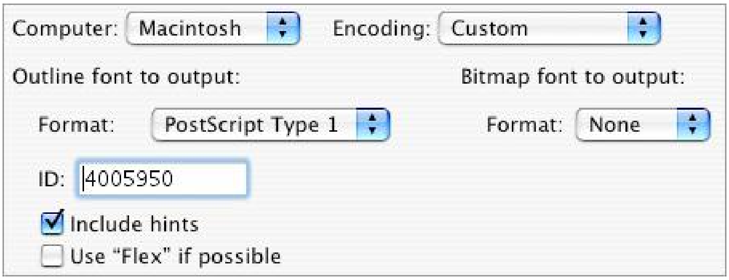
4. Set the folder into which you want the font files saved.
5. Decide whether you want to overwrite existing files with the same name.

Fontographer will then decide the sticky issues; for instance, if an AFM file should be generated for the NeXT fonts (yes), which encoding to select for Sun fonts (ISO Latin 1), or even how to set up the em square 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.

Macintosh fonts

Macintosh fonts require two particular font resources: the outline font and the FOND. In the case of 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 10, "Installing and Removing Fonts".



Computer: Macintosh

Encoding: Custom

Outline font to output:

Format: PostScript Type 1

ID: 4005950

☑ Include hints

□ Use "Flex" if possible

Bitmap font to output:

Format: None

PostScript Type 1

PostScript Type 1 fonts – also known as ATM fonts – are the fonts supported by Adobe Type Manager. 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 will render the Type 1 PostScript fonts on-screen, thus removing the need for bitmaps for every point size.

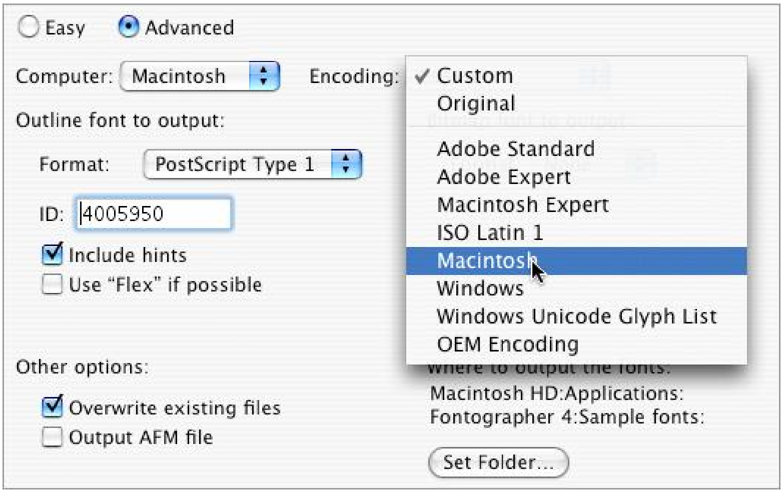
Type 3 fonts aren't widely supported, most operating systems cannot work with them. You will probably never have the need to generate Type 3 fonts but these fonts will be discussed in detail in the next section.

To generate Macintosh PostScript Type 1 fonts:

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 Macintosh from the **Computer** pop-up.



○ Easy ● Advanced

Computer: Macintosh

Outline font to output:

Format: PostScript Type 1

ID: 4005950

☑ Include hints

□ Use "Flex" if possible

Other options:

☑ Overwrite existing files

□ Output AFM file

Where to output the fonts:

Macintosh HD:Applications:
Fontographer 4:Sample fonts:

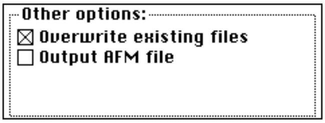
In the **Encoding** pop-up, you have several choices. If you feel adventurous, read the story on encoding options later in this chapter. If you're creating fonts for Western languages (English, German, French, Spanish etc.), then in ninety-five percent of the cases, you'll want to use the Macintosh encoding option.

4. Select PostScript Type 1 from the **Format** pop-up.

5. The **ID** field is useful in PostScript Type 1 fonts. Generally, just accept the number Fontographer proposes. It is randomly generated in the range of 4,000,000 to 4,999,998. If you want more information on this field, then read the PostScript ID field section on the next page.

6. We recommend that you accept our default settings for the **Hints** and **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 302 later in this chapter.

7. 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 bullet (•) or dollar sign ($) to the file name.



Other options:

☒ Overwrite existing files

□ Output AFM file

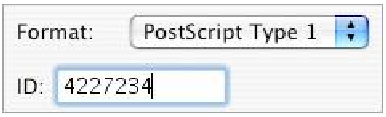
8. AFM files are not used on the Macintosh. So, if you're using a Macintosh, the output ATM file option is turned off by default. If you want an AFM for your Macintosh font, just turn the checkbox on.

9. The default directory is indicated in the first line of the **Where to output the fonts** checkbox. If you press the **Set Folder** button, you are presented with a standard file dialog box. Select the destination folder in the dialog box and press **OK**.

Many users find it extremely useful to be able to save the generated files to various folders on their disk. The **Set Folder** button displays a standard Save File dialog box that enables users to select any folder on any mounted volume as the destination folder for their fonts. The currently specified path will appear in the Generate Font Files dialog box.

The PostScript ID Field

Type 1 fonts use this information, so if you are generating Type 1 fonts it might be to your advantage to use it. This ID is used for caching the stored bitmaps inside your printer.



Format: PostScript Type 1

ID: 4227234

When a page is sent to the printer for the print job, all the characters it uses are rasterized (turned into a bitmap at the specified point size). When the print job is finished, the bitmaps are flushed from the printer's RAM.

If you own an old PostScript printer or an imagesetter, you may have a hard drive attached to your printer for storage of outline fonts. In this case, the newly rasterized bitmaps are saved to it in a font cache and referenced not by font name, but by the PostScript ID and point size. The operating system does not save the image as PostScript, but as a bitmap matched to the output resolution of the printer. When you print the same font a second time at the same point size, the bitmaps print much faster.

➲ ***Tip:*** If you aren't sure you're done with your PostScript font, but you want to generate it and test it in an application, write down the ID Fontographer gives you and set the PostScript ID to 0 (zero) so that it won't cache in the printer. When you are sure you have it how you want it, change the PostScript ID back to the number you wrote down earlier.

Font IDs for new fonts are automatically generated in the range of 4,000,000 to 4,999,998. If you are a commercial font developer, you should contact Adobe Systems, Inc. for an allocated font ID.

The PostScript ID Field is optional. If you do not want your font to have a unique PostScript ID, set this field to 0 and Fontographer won't put a unique ID in your font.

➲ ***Tip:*** You've been modifying your font's characters and you know the changes are in the database, but the printer keeps giving you the same old characters. You've probably fallen victim to the PostScript ID feature. Just regenerate your font with a PostScript ID of zero and leave it that way till you're through adjusting your font.

✐ ***Note:*** Fontographer-generated Macintosh fonts are fully compatible with current versions of font management applications such as Extensis Suitcase, Insider Font Agent Pro, Linotype FontExplorer X or Master Juggler. When using these programs, refer to their manuals for instructions on installing fonts. These applications have their own ways of dealing with font numbering conflicts.

TrueType

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 em square of 1000 built into Fontographer. In some specific cases, you may want to use a different em square, for example 2000 or 2048. This may give you finer control of the character outline.

If you use a non-1000 em square, you can set your ascent and descent in **Font info** under the **Element** menu to adjust for the larger em square, if you wish. You might try values of 1638 for ascent and 410 for descent to approximate the default em square’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 Macintosh from the **Computer** pop-up.
4. Select TrueType from the **Format** pop-up. A TrueType Options dialog box appears when you click the **TrueType Options** button.

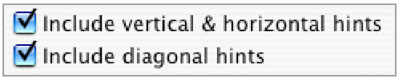
You can select the type of hints that will be used to instruct the character, select the amount of dropout control that is to be used when the characters are imaged, and select whether you would like standard or sequential character ordering. Each of these preferences is explained in the sections that follow.

Include vertical and horizontal hints

This checkbox can override options set in Hint Parameters and Hint Info. Refer to “Font hinting” in Chapter 11, “Expert Advice”, for more information.

Include diagonal hints

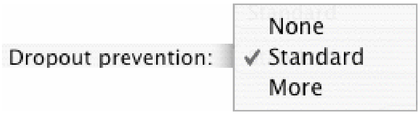
This checkbox can override options set in Hint Parameters and Hint Info. Refer to "Font hinting" in Chapter 11, "Expert Advice", for more information.



☑ Include vertical & horizontal hints
☑ Include diagonal hints

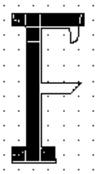
Dropout prevention

A dropout occurs when both sides of a character's outline or curve fall between pixel boundaries. A dropout is said to have happened when an important feature of a character fails to appear at all. Fontographer provides three levels of dropout control.

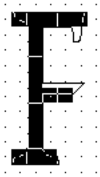


Dropout prevention:
None
✓ Standard
More

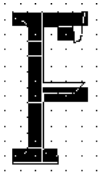
If you select None, the TrueType rasterizer will not attempt to find and correct dropouts. Standard dropout control will eliminate nearly all dropouts. More will eliminate all dropouts but will sometimes turn on bits that weren't wanted. The following illustrations show the results obtained using the three levels of dropout prevention. The first "F" character is missing its cross bar because it happens to fall between two pixel boundaries. With Standard dropout prevention, the feature is restored.



No dropout control



Standard dropout control



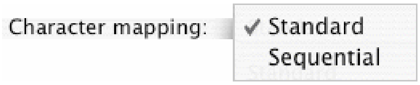
More dropout control

Notice that the outline at the right side of the top bar descends below the bar. This type of structure is called a stub. The character would conform to the intended outline better if a pixel were turned on in this area. Standard dropout prevention will not find stubs. The third character was drawn with More dropout prevention. It obviously has too many pixels turned on.

If you require a more precise control over the screen appearance of TrueType fonts, you can use Fontlab Studio 5 (http://www.fontlab.com/studio/) which includes more sophisticated mechanisms to control the TrueType hinting.

Character mapping

Most of the time, you will never change the default, which is Standard. Choosing this option gives you a normal font with all the characters you would expect, in the places defined by Apple and Microsoft for a standard TrueType font.



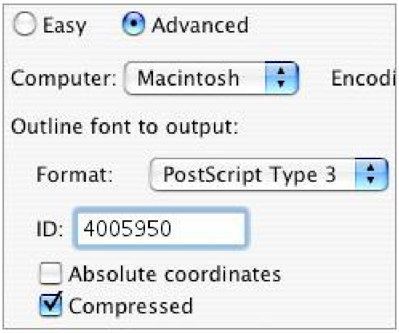
Character mapping: ✓ Standard

Sequential

Sequential character mapping, on the other hand, means that in the generated TrueType font, characters are stored in the same order as they appear in the font window. This option is potentially useful for non-Roman character sets, and nonstandard fonts. This is really of marginal use on the Macintosh; unless you are a real font guru, don't worry about it.

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.



○ Easy ● Advanced

Computer: Macintosh Encodi

Outline font to output:

Format: PostScript Type 3

ID: 4005950

□ Absolute coordinates

☑ Compressed

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.

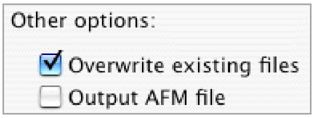
To generate Macintosh PostScript Type 3 fonts:

1. Choose **Generate Font Files** from the **File** menu.
2. Click the **Advanced** button, and then select the Macintosh option from the **Computer** pop-up.
3. You have several choices in the **Encoding** pop-up. If you feel adventurous, read "Encoding options" on page 300. Ninety-five percent of the time, you'll want to use the Macintosh encoding. There are only two encoding vectors that make no sense for the Macintosh: ISO Latin 1 and Windows.
4. In the **Format** pop-up, select PostScript Type 3.

5. The ID field is NOT necessary in PostScript Type 3 fonts. (Disregard it.)

6. For more information on the **Absolute coordinates** and **Compressed** checkboxes, see the section “Other Type 3 Formats” on page 299.

7. 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.

   Other options:

   ☑ Overwrite existing files
   □ Output AFM file

   If deselected, Fontographer will leave the older files alone, and change the name of the conflicting file by appending a bullet (•) or dollar sign ($) to the file name.

8. 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.

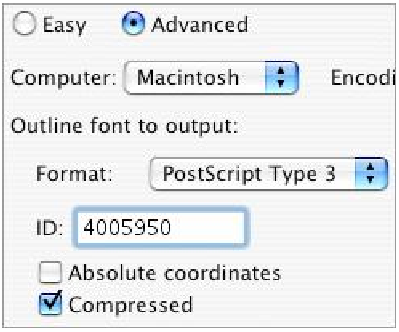
9. Use the **Set Folder** button to tell Fontographer where you’d like to place your fonts.

   The **Set Folder** button displays a standard Save File dialog box that enables users to select any folder as the destination folder for their fonts. The currently specified directory will appear in the Generate Font Files dialog box.

10. 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 **Compressed** option should also be checked.



○ Easy ● Advanced

Computer: Macintosh

Encodi

Outline font to output:

Format: PostScript Type 3

ID: 4005950

☐ Absolute coordinates

☑ Compressed

If you deselect the **Compressed** 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.

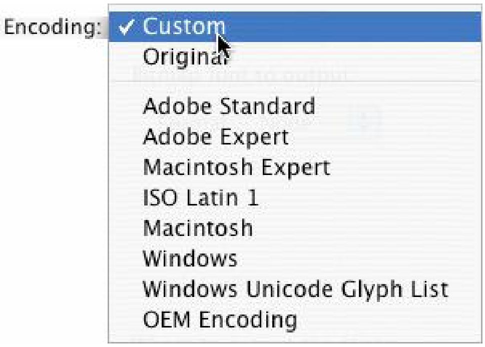
Compressed is the standard Type 3 font format. This font file can be used for both automatically downloadable fonts or fonts that are to be downloaded to the printer's hard disk. Compressed PostScript is the preferred form for generating Type 3 fonts. PostScript is stored very inefficiently inside the 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.

None

This option is provided to disable outline font generation. In the remote instances when you only need a fresh bitmap file, it is not necessary to regenerate the outline font.

Encoding options

The encoding of the font is the ordering of its characters. Version 4.7 of Fontographer has two dialog boxes that allow you to set the encoding. The **Encoding** pop-up in the Font Info dialog box will reorganize the layout of the font window to display the key positions of the encoding you selected. On the other hand, when you select a different encoding in the Generate Font Files dialog box, Fontographer will output the font to match the encoding you select – it won't change the encoding displayed in the font window.



The most commonly used encodings for the Macintosh are Macintosh and Adobe Standard.

Custom encoding allows for the custom naming of characters. The Character Information dialog box (which is opened by selecting a character and choosing **Selection Info** from the **Element** menu) contains a name field that allows you to change the character name. Custom encoding becomes the default encoding option any time you change a character name. Custom character 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.

Adobe Standard encoding (ASE) is Adobe's default encoding. In a Macintosh font, ASE will allow for the substitution of 16 characters from the Symbol font (see the tip on this page).

Adobe Expert is the encoding required for fonts known as expert sets. These typically include small caps, swash caps, old style numerals, superior and inferior characters, and special text symbols. You would want to use this option only if you are creating an expert font set.

➲ ***Tip:*** In the Adobe Standard encoding, these characters will be used from the Symbol font instead of from your font so you won't have to bother drawing them: notequal, infinity, plusminus, lessequal, greaterequal, partialdiff, summation, product, pi, integral, Omega, radical, approxequal, Delta, lozenge, and apple. Visually, those characters are: ≠ ∞ ± ≤ ≥ ∂ ∑ ∏ π ∬ Ω √ ≈ ∆ ◊ □. If you don't want these characters substituted (you prefer to draw your own versions of these characters), use the Macintosh encoding instead of the Adobe Standard encoding.

ISO Latin 1 is the encoding preferred by most UNIX based systems. Select this option if you are generating a font for Sun workstations.

Macintosh encoding lets you use your font on a Macintosh and maintain keystroke compatibility with other Macintosh fonts. This is very similar to Adobe Standard encoding, but it will put something in every spot (assuming you have enough characters for that) and won't substitute the symbol characters. Also, characters can appear in the lower 32 slots.

If you need more information about encoding vectors, refer to:

http://partners.adobe.com/public/developer/opentype/index_char_sets.html

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. For the thorny technical details of hinting, turn to Chapter 10, “Installing and Removing Fonts”. 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.

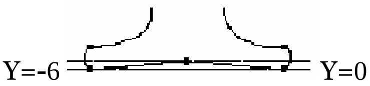
A word about flex

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 would need 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.

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 a range of font ID numbers from 16384 through 32767. In other words, if you wish to design a Thai font for use in the Thai system, by simply entering a font ID of 26625, you have linked the font to the Thai scripting system. Perhaps you need a right-to-left font. Simply use an ID number in the range for the Arabic or Hebrew systems, switch to those systems, and your font will type right to left in those operating systems. 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. Also see “KCHR resources” in Chapter 11, “Expert Advice”.

Script system	Font ID range
Japanese	16384 – 16895
Traditional Chinese	16896 – 17407
Korean	17408 – 17919
Arabic	17920 – 18431
Hebrew	18432 – 18943
Greek	18944 – 19455
Cyrillic	19456 – 19967
Simplified Chinese	28672 – 29183
Vietnamese	31232 – 31743

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 XP 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.

Windows PostScript fonts consist of two mandatory files: PFB and PFM. In addition, the font can include optional files: AFM and INF.

Five encoding vectors are accepted in Windows: Custom, Windows 3.1, Windows 95, Adobe Standard, and Adobe Expert. 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.

For Windows, the prevailing choices for the encoding vectors are Windows/Windows 95 and Adobe Standard.

Windows lets you create a Windows 3.1 PostScript or TrueType .fon. You also have the option of Windows 95.

➲ ***Tip:*** If you choose Windows 3.1 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.

Windows Unicode Glyph List gives you access to the full character set for Windows 95. Your database will change to the appropriate size (2,147 on the Macintosh), and you can begin filling your character slots with characters from Greek, Cyrillic, Hebrew, Arabic, Eastern European, and other languages.

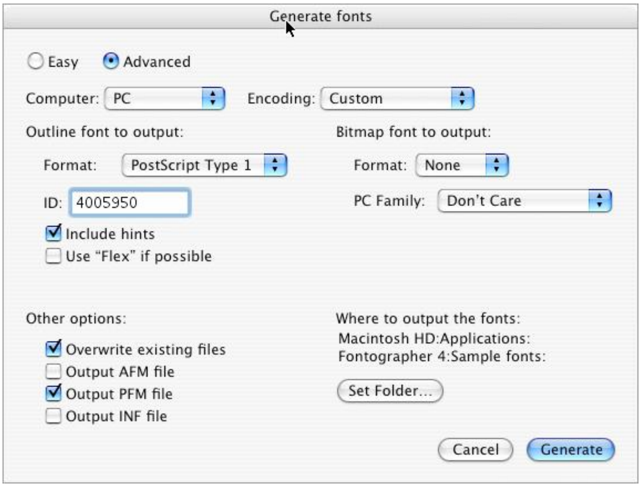
OS/2 Unicode Glyph List gives Windows users access to the full OS/2 character set.

OEM encoding is a pretty plain encoding vector that lists character names and Unicodes.

To generate Windows PostScript Type 1 fonts:

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

2. Select PC from the **Computer** pop-up.



Generate fonts

○ Easy ● Advanced

Computer: PC

Encoding: Custom

Outline font to output:

Format: PostScript Type 1

ID: 4005950

☑ Include hints

□ Use "Flex" if possible

Bitmap font to output:

Format: None

PC Family: Don't Care

Other options:

☑ Overwrite existing files

□ Output AFM file

☑ Output PFM file

□ Output INF file

Where to output the fonts:
Macintosh HD:Applications:
Fontographer 4:Sample fonts:

3. Select the encoding option appropriate for your font. If you feel adventurous, read the section "Encoding options" on page 300. Ninety-five percent of the time, you'll want to use Windows or Windows 95 encoding. There are only two encoding vectors that don't make sense for Windows: ISO Latin 1 and Macintosh.

4. Select the PostScript Type 1 option in the **Format** field.

5. The option **Include hints** is checked by default. Generally leave this on; only turn it off if your font exhibits hint-related printing problems.

6. **Output PFM file** should be checked; the **AFM** and **INF** options should be deselected in most situations.

7. Do not generate a bitmap font. They are not useful for Windows. If you will be using your font in DOS, check the requirements of the host application to see if BDF or FON files will be helpful.

8. Point Fontographer to the destination folder, via the **Set Folder** button.

9. 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 bullet (•) or dollar sign ($) appended to its name.

Output AFM file

The AFM file is not used by ATM/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 ATM/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 ATM/Windows.

The CFG file

CFG files are used by Adobe's Font Foundry DOS program for installing PostScript Type 1 fonts into major DOS programs, such as WordPerfect. Since this is the only use of a CFG file, we decided to show you how to make one, rather than generate a dummy file every time you generate fonts.

If you were to create a CFG file for the Nova family, it would look like the following example. (We have included comments to help explain things.)

If you must install fonts into WordPerfect DOS and the like, then use this as a model for writing your CFG. All lines preceded by "#" are comments and do not need to be in the actual file. When you are finally ready to run the Installer program, copy the PFB, PFM, INF, AFM, and the CFG files to their own disk.

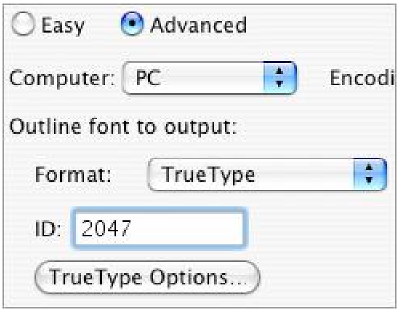
```
# Installer Configuration File
# Contains package specific information used
# by the install program. DISKSIZE 713
# size in kbytes of floppy disk used
FILE INSTALL.CFG 1
# the CFG file must be named "install.cfg" FILE README 1
# It looks for a Readme file to install
# Font entry fields:
# Fontname - PostScript name of the typeface
# Filename - 5 character PC filename base
# Number of the disk this filetype sits on or
# zero if not present.
# One entry for each filetype PFB, INF, CTF,
# PFM, ABF, and AFM.
FONT Nova-Normal NON_1 1 0 1 0 1
FONT Nova-Oblique NOO_1 1 0 1 0 1
FONT Nova-Bold NOB_1 1 0 1 0 1
FONT Nova-BoldOblique NOBO_1 1 0 1 0 1
DISKNAME 1 ’Font Disk’
# Give the name -label- of the floppy
PKGNAME novafam
```

TrueType for Windows

To use Windows TrueType fonts, you must have Microsoft Windows 3.1 or later.

When generating TrueType for Windows, Fontographer makes use of a dialog box that allows you to determine the level of hinting, dropout, encoding, and character mapping options. Activate this dialog box by pressing the **TrueType Options** button. (Choices in this dialog box may override the selection made in the **Encoding** pop-up.)

1. For **Computer**, select PC.
2. Select the TrueType option in the **Format** field.



○ Easy ● Advanced

Computer: PC

Encodi

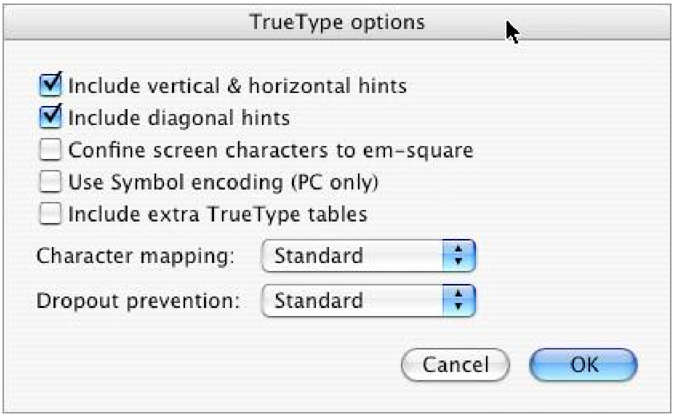
Outline font to output:

Format: TrueType

ID: 2047

3. You can disregard the ID field as that pertains only to PostScript Type 1 fonts.
4. If you are curious, press the **TrueType Options** button; otherwise, skip to Step 6.

   The default settings are shown below. Generally leave them alone, but turn off hinting if your font exhibits printing problems. The **Use Symbol encoding** and **Sequential Character mapping** 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. Confine screen characters to em-square will not change the way characters print. Include extra TrueType tables is provided for future use by Windows applications.



TrueType options

☑ Include vertical & horizontal hints
☑ Include diagonal hints
☐ Confine screen characters to em-square
☐ Use Symbol encoding (PC only)
☐ Include extra TrueType tables

Character mapping: Standard
Dropout prevention: Standard

5. Press **OK** to close the TrueType Options dialog box.

6. Windows users may generate a bitmap font (.fon) for use with Windows TrueType fonts, or for use when printing to a non-PostScript printer, such as dot matrix.

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 bullet (•) or dollar sign ($) appended to its name.

8. Point Fontographer to the folder where you wish to save your fonts, via the **Set Folder** button.

9. For more on choosing the Windows family, see “FON” on page 318.

10. When all the options have been selected, press **Generate** to close the dialog box and generate the font.

Symbol encoded Windows fonts

When you generate a Windows TrueType font, a checkbox allows for a special symbol encoding option. This is the same encoding format used in the dingbats font in Windows 3.1, 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.

✐ ***Notes:*** Symbol is not available for Windows PostScript Type 1 fonts. Also, Symbol encoded fonts do not use kerning, and may display bogus Unicode numbers.

✐ Be sure to select Decorative from the **Windows Family** pop-up in the Advanced Generate Font Files dialog box for your Symbol encoding to work.

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 and use some conversion tool in DOS to make it into a usable format.

NeXT and Sun PostScript fonts

Whether you are running NeXTSTEP on a sleek, black, discontinued slab or a 486 machine, your NeXT Type 1 PostScript fonts will behave similarly. For that matter, using your Sun fonts on a SPARCStation requires very similar font generation. Both formats generate only two files: the AFM and the PostScript file. The AFM (Adobe Font Metrics) contains the metrics information. The PostScript file – the one with the ".next" or ".sun" extension – is a text file that contains a small header followed by the encrypted PostScript routines. The difference in the files for the two operating systems is indeed small.

➲ ***Tip:*** NeXTSTEP prefers Adobe encoding while Solaris prefers ISO Latin 1 encoding. Use these for standardized document interchange, but you may use other encoding options if need be.

The naming conventions required for both platforms are the same. Always insert a hyphen between the family name and style type (this is highly recommended for all platforms). Thus, the italicized face from the Nova family will be Nova-Oblique. If you get this correct, then all the fonts should line right up by their lineage.

Unless you really need the special features of the **Advanced** mode, you are better off keeping to the **Easy** font generation routine for UNIX fonts. This will ensure that the encoding is optimum and that the AFM file is generated. If you generated fonts in the **Advanced** mode, but failed to generate the AFM file, then do not proceed – go back to the Generate Font Files dialog box and generate the AFM file. You won't get anywhere installing the font without it.

For more about hints, flex, and other options, see "Encoding options" on page 300, "When should you use hints?" and "A word about flex" on page 302, and "Font hinting" in Chapter 11, "Expert Advice".

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.

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/

Fontographer and OpenType

Fontographer 4.7 **cannot** open or generate OpenType PS fonts.

However, Fontographer 4.7 does open and generate PC TrueType fonts. The PC TrueType fonts that are produced by Fontographer are compatible with the OpenType specification and will work in any system that requires OpenType.

Fontographer 4.7 **cannot** import or generate OpenType Layout features in the fonts. If you open an OpenType TT font in Fontographer, the glyphs and simple kerning will import fine, but all OpenType Layout features and other OpenType extensions will not be retained.

If you're interested in opening or generating OpenType fonts, we recommend Fontlab Studio 5 (http://www.fontlab.com/studio/) – the next-generation digital font editor from Fontlab Ltd.

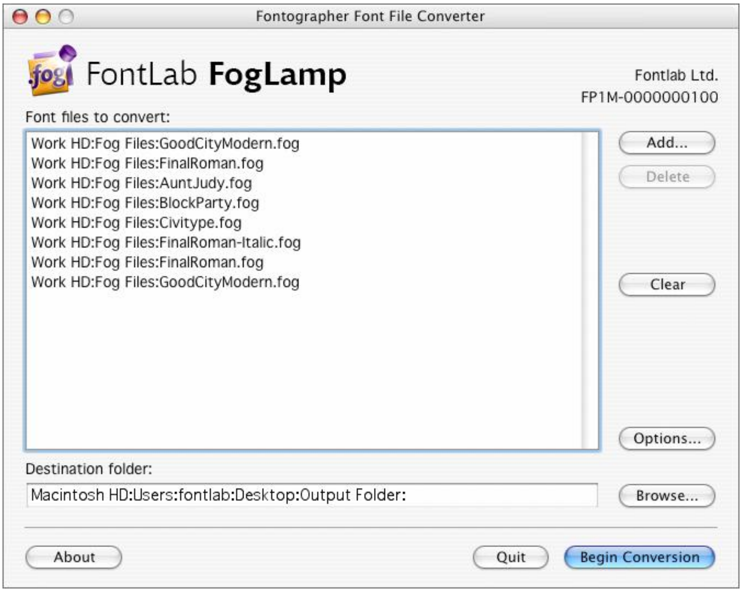
But still you have the opportunity to convert your .fog database files into fonts in OpenType format. This can be done with the help of the FogLamp (http://www.fontlab.com/foglamp/) and TransType Pro (http://www.fontlab.com/transtype/) utilities.

Using FogLamp

FogLamp is a small utility allowing you to convert Fontographer database font files (.fog) into FontLab VFB files (.vfb). FogLamp for Macintosh can convert files of both Macintosh and Windows versions of Fontographer of any version including version 4.7 for Mac OS X.

Converted elements**:** outlines, template/mask, guidelines, hints, font naming and other font header information are converted from .fog to .vfb format. Only one (last) bitmap per glyph is converted but bitmaps are not needed for further production of OpenType font.

Easy drag-and-drop batch conversion of multiple font files at a time allows you to quickly port all your designs to Fontlab Studio or TransType Pro and do the final font production in Type 1, TrueType or OpenType format in minutes.



Fontographer Font File Converter

FontLab **FogLamp**

Fontlab Ltd.
FP1M-0000000100

Font files to convert:

Work HD:Fog Files:GoodCityModern.fog
Work HD:Fog Files:FinalRoman.fog
Work HD:Fog Files:AuntJudy.fog
Work HD:Fog Files:BlockParty.fog
Work HD:Fog Files:Civitype.fog
Work HD:Fog Files:FinalRoman-Italic.fog
Work HD:Fog Files:FinalRoman.fog
Work HD:Fog Files:GoodCityModern.fog

Destination folder:

Macintosh HD:Users:fontlab:Desktop:Output Folder:

Pack your suitcase: bitmap fonts

Bitmaps vs. outlines

In these days of Type 1 and TrueType 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.

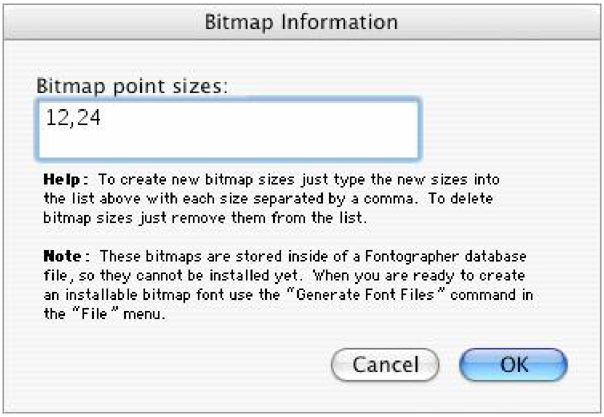
Think about your screen image. It is represented by a lot of pixels, each either turned on or turned off. When the font is rendered to the screen, the font display driver maps out the bits it wants turned on. That is essentially a bitmap created on the fly.

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 4.7 generates three varieties of bitmap fonts: one specifically for use as a screen image on the Macintosh, the .bmap (called NFNT in the bitmap selection box), one used as a screen image in Windows, the .fon – 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.



Bitmap point sizes: 12,24

Help: To create new bitmap sizes just type the new sizes into the list above with each size separated by a comma. To delete bitmap sizes just remove them from the list.

Note: These bitmaps are stored inside of a Fontographer database file, so they cannot be installed yet. When you are ready to create an installable bitmap font use the "Generate Font Files" command in the "File" menu.

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 sizes to output field and click Generate. 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.

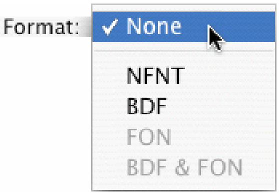
Deleting bitmap sizes

Unwanted bitmap sizes can be removed from the database file by removing the size from the Bitmap point sizes field in the Bitmap Information dialog box. Fontographer will display a warning message before it deletes the bitmaps.

Bitmap format

None

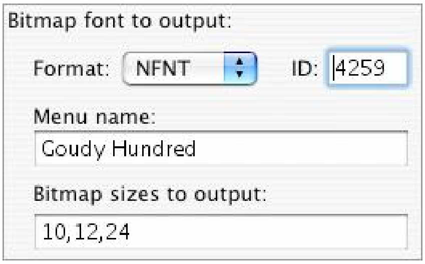
This option is provided to disable bitmap generation. In the instances where simple changes have been made to the outline font (like adjusting a BCP, which does not always affect the bitmap), it is not necessary to regenerate the bitmaps. Also, when you have no need of a bitmap, there is no reason to take the time to generate one. You can also avoid the bitmap generation process by having no sizes in the Bitmap sizes to output field.



Format: ✓ None

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.



Bitmap font to output:

Format: NFNT ID: 4259

Menu name:
Goudy Hundred

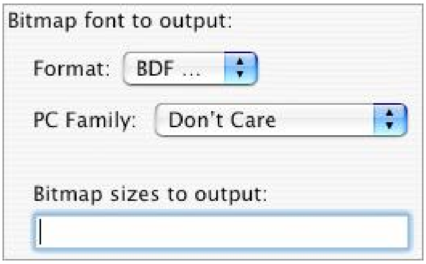
Bitmap sizes to output:
10,12,24

Fontographer generates NFNTs (New FoNTs) for the Macintosh. NFNT ID numbers can range up to 32,767. 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.

BDF

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



Bitmap font to output:

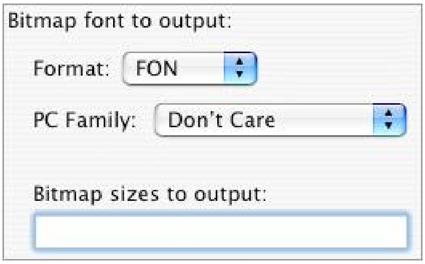
Format: BDF ...

PC Family: Don't Care

Bitmap sizes to output:

FON

This option gives you FON file generation for use in Windows. FONs allow you to have hand-tuned bitmaps that you can use as screen fonts in Windows. Fontographer 4.7 generates FONs at the standard Windows VGA resolution of 96 dpi. In order for the bitmaps from Fontographer on the Macintosh to match up to the higher resolution bitmaps in Windows, you must apply a 4/3 factor to the source bitmap. In other words, if you want the bitmap to be 12 points in Windows, then start with a 16-point bitmap on the Macintosh. The formula to use would be Windows bitmap size X ÷ 3 = bitmap size in Fontographer. (For more information, refer to "Creating bitmaps for cross-platform use" on page 320.)



Bitmap font to output:

Format: FON

PC Family: Don't Care

Bitmap sizes to output:

✐ ***Note:*** The **PC Family** pop-up in the “Bitmap font to output” portion of the Generate Font Files dialog box allows Windows computers users to choose the family type of their fonts. Fontographer stores the data in the TrueType, PFM, and FON files only. Use Roman for a serif font, Swiss for a sans serif, Modern for stylistic fonts, and script for cursive-type fonts. Use the default setting, Don’t Care, if you are unsure. Few Windows programs currently refer to this, but it is expected that more will in the future. PC Family information will be useful in providing Windows with basic data about the style of the font.

The ID Field

The ID field is provided to assign the font family ID. Bitmap font families (essentially, FONDs) are identified internally by a unique number ranging between 1 and 16,383. Numbers between 1 and 1023 have already been used by Apple and other font developers, and they should not be used. Apple has reserved font numbers between 3071 and 16,383 for commercial use. However, when creating a font, you must try to choose a value that does not conflict with those assigned to existing fonts. This is especially important with respect to your printing devices, like those at your service bureau. Macromedia recommends that you assign font numbers ranging from 1024 to 3070 to try to avoid those conflicts.

Conflicting font IDs are automatically renumbered during font installation, but this renumbering can cause confusion when the font is used. Many Macintosh programs identify fonts by ID number, not by name. Thus, if you move a document from one computer to another, the displayed fonts may change if numbering conflicts occurred during font installation. The best way to avoid this problem is to number all your fonts uniquely so the conflict won’t occur. Again, we recommend you start with font family ID 1024 and count up; this is the best compromise we can suggest for now.

The menu name

The menu name is the name that will be used in the font menu on the Macintosh. This field becomes the FOND resource name, so we don’t recommend making the name radically different. For instance, don’t enter a menu name of Skipper for a font named Gilligan.

Creating bitmaps for cross-platform use

Because of the difference in monitor display resolution between the Macintosh and Windows computers, you'll need to use the table below to create bitmaps at sizes that will display properly on both platforms. You'll notice that a few NFNT font sizes will not be available for cross-platform use. The table isn't all-inclusive, so you can create larger bitmaps in the same proportions as those shown in the table.

FROM the Macintosh… Create a _ point .FON	TO Windows computers to get a _ point .FON
5	4
7	5
8	6
9	7
11	8
12	9
13	10
15	11
16	12
17	13
19	14
20	15

Exporting files

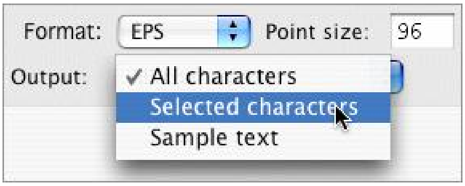
Fontographer is equipped with an **Export** menu item. From within its submenu you may export EPS, PICT or metrics files. In this chapter we will discuss exporting your font characters as graphics for use in applications such as Macromedia FreeHand or Adobe Photoshop. For more about exporting metrics, 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 Macromedia FreeHand and 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 **Output** option.

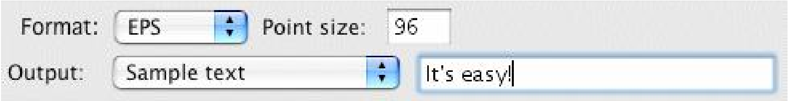


Format: EPS
Point size: 96

All characters will export all the characters in your font.

Selected characters will export only the characters currently selected in the font window or from the outline window (the active character).

Choosing **Sample text** will bring up a text entry box for you to type the text string you would like to export.



Format: EPS Point size: 96

Output: Sample text It's easy!

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 **Save** button and the font will be exported without further comment.

Exporting Macintosh PICT files

Fontographer provides the PICT file option to generate PICT format files. Fontographer creates an object-oriented PICT that stores the points and paths of the characters.

To export a PICT file:

1. Select the **PICT format** from the **Export** pop-up in the **File** menu.
2. Select the point size necessary for your intended use.
3. Determine the **Output** option.

   All characters will export all the characters in your font.

   Selected characters will export only the characters currently selected in the font window; or from the outline window (the active character).

   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 a PICT file, Fontographer allows you to choose your own file name.

This is a PICT file opened in MacDraw and edited, with fills applied.

Once you've chosen the folder to export the file to, press the **Save** button. Fontographer will export the font.

Programs such as Adobe Photoshop can read this file format directly for editing purposes. So now you can reshape, distort, or enhance characters; fill them with specific colors or patterns; and even change their line sizes.

✐ ***Note:*** PICT outlines are not absolutely identical to their PostScript counter parts; therefore, there will be some approximation involved in the conversion.

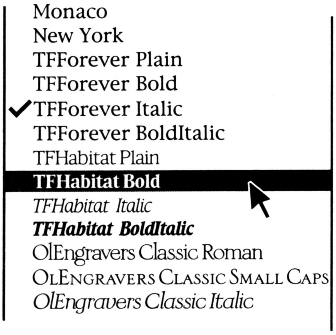
Creating a Font Family

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 posture – that is, Roman and Italic.

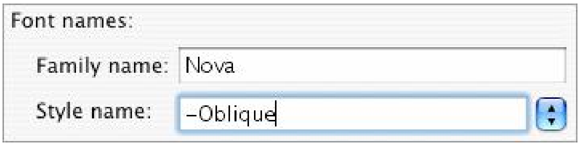
9

A typical computer (font) family unit frequently contains four fonts – Plain, Italic, Bold, and BoldItalic – in all point sizes.



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.



Font names:

Family name: Nova

Style name: -Oblique

When you name your font in Fontographer's Font Information dialog box be sure to name each font properly. Make sure the Family Name text box has the same name in it for all the styles of the family you're building. This is important to remember for making families on all platforms.

Families: Windows, Sun, NeXTSTEP

Making the 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 8, “Generating and Exporting Fonts”.

✐ ***Note:*** Windows 3.1/95/98 only recognizes Regular, Normal, or Plain (no style name) as the plain member of the family; Italic or Oblique for the italic member; Bold or Black for the bold member; and combinations of the last two for the bold-italic member of the font family. Font families for use in Windows 3.1 can have no more than four members and can only contain members named from the choices listed above.

Font families on the Macintosh

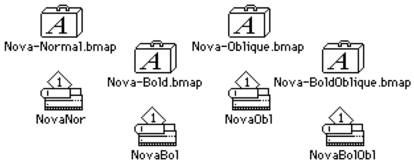
If your font is a logo typeface or one that contains special symbols, you probably don't need to make a family of fonts. In fact, many people use Fontographer to make only one version of a typeface. In these cases, family information is not important. But if you are ambitious, and have made more than 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.

Note: Style Merger, a 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. Style Merger and TransType Pro will work with both PostScript and TrueType fonts. They never affects 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 8, "Generating and Exporting Fonts"). For this example, we started with these font files:

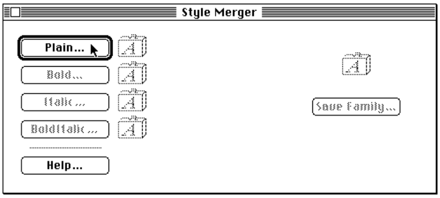


Now let's run Style Merger. It is so easy to use that the complete process takes only four steps.

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.

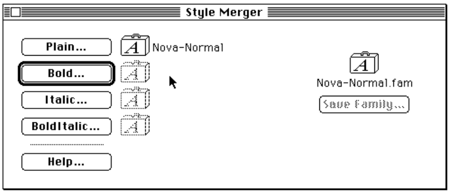
How Style Merger works

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.

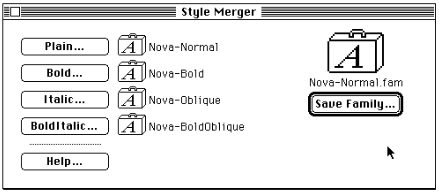


Once you've chosen the Plain face and clicked ***OK****, you will see the name of your Plain font appear to the right of the* ***Plain*** *button.*

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.



Nova-Normal
Nova-Bold
Nova-Oblique
Nova-BoldOblique
Nova-Normal.fam

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.

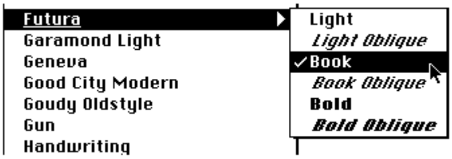
✎ ***Note:*** Please, do yourself a favor and always make sure the fonts you use to build families with are not installed. Always close any fonts installed with Suitcase, Master Juggler, Font Porter or whatever you installed them with; then build your family and reinstall your fonts. If you are using 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.

Adobe Type Reunion (ATR) compatibility

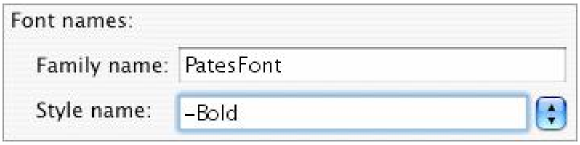
Style Merger builds a completely functional Macintosh font family: there is only one entry in the font menu for the font, and all the appropriate styles are referenced automatically when different styles are applied to the font. This is the recommended method for creating font families.

However, some people like to create font families that work in conjunction with a product called Adobe Type Reunion™, which makes your fonts display in the font menu in a manner similar to the Adobe typefaces. Each of these faces has an entry for each style that appears in a hierarchical menu off of the font menu. For example, the Futura family has a single entry in the font menu, and to the right of the name is an arrow indicating that there are multiple fonts under that listing. If you press on the Futura font listing, you will see a menu of four styles off to the right: Light, Light Oblique, Book, Book Oblique, Bold, and Bold Oblique.



It is easy to create font families that behave like this. All you need is Style Merger, ResEdit, and a couple of minutes. Be aware, however, that if you create ATR-style families, everyone who uses that font without Adobe Type Reunion will have the penalty of seeing four entries in their font menu (one for each style).

The first step in creating ATR-compatible font families is to make sure you have named each font properly. Type the name you want to see in the font menu in the Family Name field of the Font Information dialog box. In the Style Name field specify the name you want to appear for that particular style, prefixed with a hyphen.



Font names:

Family name: PatesFont

Style name: -Bold

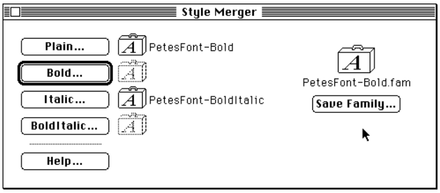
In the next example, we create a font family called PetesFont, which consists of four styles. For all four fonts, enter PetesFont in the Family Name field. Then for the plain font, leave the Style Name field blank. For the other fonts, enter -Bold, -Italic, and -BoldItalic in the Style Name field for the bold, italic, and bolditalic fonts, respectively. Of course, you don't have to use these names; Wacko, Slanto, and ClapDappo would work just as well – maybe better.

The thing that's complicated about doing this stuff is linking together the subfamilies. For example, suppose you were using PetesFont-Italic in an application, and you changed to the bold style. The selected font should automatically change to PetesFont-BoldItalic, not merely make a fake bold of the italic face. In addition, if you were using PetesFont-Bold, and you selected the italic style, that should also switch to PetesFont-BoldItalic since it was already bolded.

So the first thing to do is build the two subfamilies. PetesFont-Bold has to link to PetesFont-BoldItalic for the italic, and PetesFont-Italic has to link to PetesFont-BoldItalic for the bold. This is easy to do.

To build subfamilies:

1. Open Style Merger and choose the bold font for the Plain style, and the bolditalic font for the Italic style.



This makes a family with the bold font as the base font, and the bolditalic font as the italic style. We don't add any more fonts here, because there isn't a bolder version of the bold, nor is there a boldbolditalic.

2. Simply choose **Save Family** and keep the name PetesFont-Bold.fam.

This builds one of the subfamilies.

3. To build the other subfamily, choose PetesFont-Italic as the Plain style, and then PetesFont-BoldItalic as the Bold, and create a family called PetesFont-Italic.fam.

   This builds the other subfamily.

Now you are ready to build the main family, which links the bold, italic, and bolditalic with the plain.

4. Run Style Merger on the PetesFont fonts as described in the initial example with the Nova family. This is the main family, so save the family with the name you would like to see in the font menu.

Note: In Step 4, make sure you use the ".bmap" files rather than the ".fam" files – this is an extremely easy mistake to make if you're not careful.

The final step involves ResEdit. What you want to do is merge the three families you just created and the bolditalic font into one file.

5. Copy each FOND resource from PetesFont-Bold.fam, PetesFont-Italic.fam, and PetesFont-Bold-Italic.bmap into the file called PetesFont.fam. This is easy to do in ResEdit; just launch ResEdit, open up these four files, and copy and paste the FONDs from the three other suitcases into PetesFont.fam.

To use ResEdit:

1. Open ResEdit 2.0 or a later version.
2. Open the three font families you just created plus the bolditalic bitmap file.
3. Using the PetesFont.fam file as your repository, copy each FOND in the other fonts into it.

When you have done that, there should be a total of four FONDs in PetesFont.fam. This can easily be checked in ResEdit by double-clicking the little FOND icon in the window for the *PetesFont.fam* file. Note that you don't need to copy any of the NFNT resources; the construction of the main font family in Style Merger copied in all the NFNT resources for you. All you need are the four FONDs brought together into the main family.

4. Close ResEdit, save your changes, and install the file *PetesFont.fam*. Now you have an ATR-compatible font family.

Installing and Removing Fonts

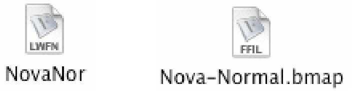
You’ve devoted a lot of time to getting your font ready to use. You’ve tweaked, scaled, and skewed the characters; adjusted the kerning and spacing; printed out a sample to see what the characters look like; and finally generated the font. And since you’re perfectly happy with your creation, you’re ready 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.

10

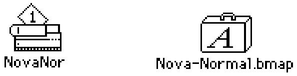
Installing Macintosh PostScript fonts

Macintosh PostScript 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).



In Mac OS 9 and earlier Classic versions, the printer font file icon defaults to a little printer icon while the bitmap font file icon usually looks like a little suitcase. However, both icons may be customized by the font vendor or the user.



Each of these files must be installed in its proper place for a PostScript font to work.

Installing PostScript 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.

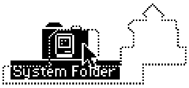
After you install a font, it appears in the All Fonts collection.

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.

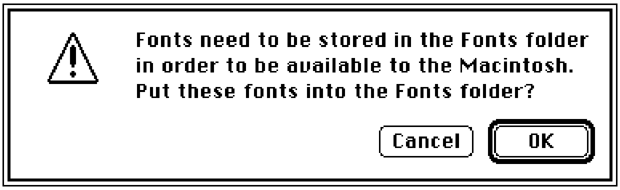
Installing PostScript fonts in Mac OS 7-9

To install fonts in Mac OS 7.1-9:

1. Select both PostScript and bitmap files by dragging the pointer around each.
2. Drag both files on top of your System Folder and drop them there.



You will be asked if you want it to put the fonts into the Fonts folder inside your System Folder. Click **OK** to install the font.

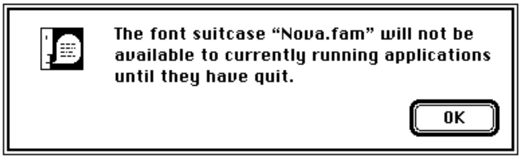


Fonts need to be stored in the Fonts folder in order to be available to the Macintosh. Put these fonts into the Fonts folder?

Both PostScript and bitmap files are moved into the Fonts folder.

Your font is now installed. The font name will appear in the **Font** menu for each application that has a **Font** menu.

Notes: Don't put your Fontographer database file in System 7.1's Fonts folder. Doing so could cause problems for Fontographer when you generate your fonts.

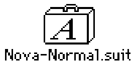


The font suitcase "Nova.fam" will not be available to currently running applications until they have quit.

If other programs were open and you don't get this message, then your font is not installed properly in the System file. Make sure you are copying the font into the System Folder (the folder with the small picture of a Mac on it). If you did and you are still having a problem, then you may need to rebuild your desktop so that the Finder will recognize that your fonts belong in the Fonts folder.

Installing Macintosh TrueType fonts

TrueType fonts exist all in one file. There are no separate screen and printer files like there are for PostScript fonts. Instead, everything comes in a single suitcase file.



Note: When you install the font, both the TrueType outline and any bitmaps you have 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.

Installing TrueType fonts in Mac OS X

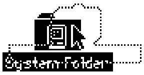
Installing TrueType fonts in Mac OS X is the same as installing PostScript fonts. You may refer to Mac OS X help for details.

Note that Mac OS X not only supports “Mac TrueType” fonts but also data-fork “PC TrueType” fonts (with the file extension .ttf).

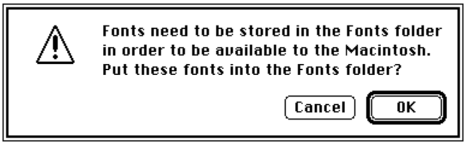
Installing TrueType fonts in Mac OS 7-9

To install TrueType fonts in System 7.1:

1. Drag the suitcase file containing your font on top of the System Folder icon and release the mouse button.

   You will be asked you if you want to store your fonts in the Fonts folder.

   Fonts need to be stored in the Fonts folder in order to be available to the Macintosh. Put these fonts into the Fonts folder?

2. Click **OK** to install your fonts.

   The system software moves your suitcase into the Fonts folder, where it is automatically installed.

Your font is now installed. The font name will appear in the **Font** menu in any application that has a **Font** menu.

Note: Be sure you don't put your Fontographer database file into System 7.1's Fonts folder. Doing so could cause problems for Fontographer when you generate font files.

Installing Windows fonts

Windows fonts generated by Fontographer are intended for use in Windows 3.x, Windows 95, Windows NT, Windows 2000 and Windows XP. PostScript fonts may be used in 3.0 or later versions with the help of Adobe Type Manager. Windows 2000 and XP directly support PostScript fonts so ATM is not required. TrueType fonts require Windows 3.1 (or later versions), which has increased functionality in its Control Panel to enable TrueType font installation. There is a difference in the implementation of PostScript and TrueType fonts in Windows, but they do have one similarity: neither depends upon a bitmap file for the screen image.

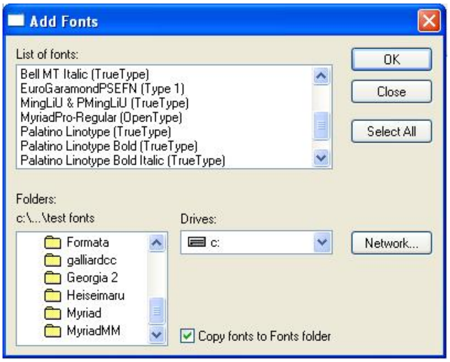
Installing fonts in Windows 2000 or Windows XP

Installing PostScript fonts in later versions of Windows doesn't require that you have Adobe Type Manager (ATM).

To install both PostScript and TrueType 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.



List of fonts:

Bell MT Italic (TrueType)
EuroGaramondPSEFN (Type 1)
MingLiU & PMingLiU (TrueType)
MyriadPro-Regular (OpenType)
Palatino Linotype (TrueType)
Palatino Linotype Bold (TrueType)
Palatino Linotype Bold Italic (TrueType)

Folders:
c:\...\test fonts

Formata
galliardcc
Georgia 2
Heiseimaru
Myriad
MyriadMM

Drives:
c:

☑ Copy fonts to Fonts folder

You'll see three windows labeled 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 PostScript fonts in Windows 3.1/95/98/ME with Adobe Type Manager

Installing PostScript fonts in Windows requires that you have Adobe Type Manager (ATM). Windows 95 requires ATM 3.02 or later. You can download the most current version of ATM Light from:

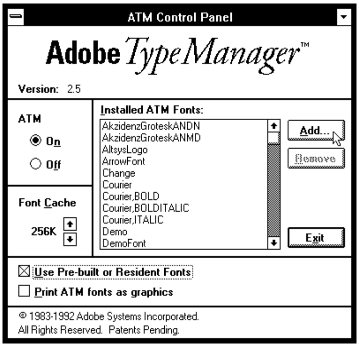
http://www.adobe.com/products/atmlight/main.html

ATM creates screen fonts from the PostScript outlines and allows you to print your PostScript fonts to any printer attached to your Windows computer. PostScript fonts in Windows consist of two parts, a .PFB (PostScript Font Binary) file and a .PFM (PostScript Font Metrics) file. Both files are necessary. Make sure you have generated both files from Fontographer before proceeding.

1. Double-click the ATM Control Panel icon in the Program Manager to start it.

   ATM Control Panel

   The ATM Control Panel displays.



ATM Control Panel

Adobe *Type Manager*™

Version: 2.5

ATM
● On
○ Off

Font Cache
256K

Installed ATM Fonts:
AkzidenzGroteskANDN
AkzidenzGroteskANMD
AltsysLogo
ArrowFont
Change
Courier
Courier,BOLD
Courier,BOLDITALIC
Courier,ITALIC
Demo
DemoFont

☒ Use Pre-built or Resident Fonts
□ Print ATM fonts as graphics

© 1983-1992 Adobe Systems Incorporated.
All Rights Reserved. Patents Pending.

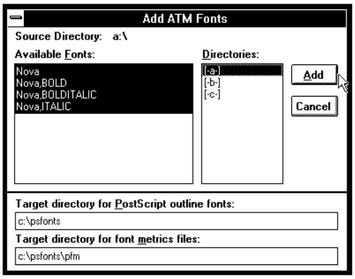
2. Click the **Add** button.
3. Scroll the **Directories** list until you find the drive/directory containing your fonts.

4. Double-click the drive/directory to show the names of the fonts on the disk.

Note: If you expected more (or different) font styles to appear in the Add ATM Fonts dialog box, make sure you've followed the naming conventions for Windows font families. If the family and style names aren't set up correctly, then the fonts you want to add may not appear in the list.

5. Select the fonts you want to install from the Available Fonts list. You can drag through the list to select multiple fonts.

6. Click the **Add** button to install your font(s).



Add ATM Fonts

Source Directory: a:\

Available Fonts:

Nova
Nova,BOLD
Nova,BOLDITALIC
Nova,ITALIC

Directories:

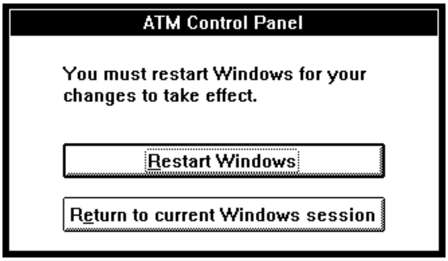
[-a-]
[-b-]
[-c-]

Target directory for PostScript outline fonts: c:\psfonts

Target directory for font metrics files: c:\psfonts\pfm

7. Click the **Exit** button in the ATM Control Panel.

   ATM 2.0 will display a dialog box asking you to restart Windows.



ATM Control Panel

You must restart Windows for your changes to take effect.

8. Press **Restart Windows**.

   If you didn't see this dialog box, everything's okay; you're just using ATM 2.5 or later, so this doesn't apply to you.

Your font is now installed. The font name will appear in the **Font** menu in every application that has a **Font** menu.

Windows TrueType and .fon installation

TrueType and .fon fonts for Windows consist of a single file with the extension .ttf (TrueType Font) or .fon. You must be running Windows 3.1 (or later versions) to use TrueType or .fon fonts, since no previous version of Windows has TrueType or .fon capability.

✐ ***Note:*** If you are installing both a .ttf and a .fon, you must install the .ttf first.

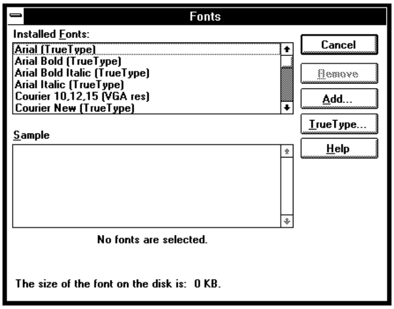
Installing TrueType and .fon fonts in Windows 3.1/95/98/ME

1. Open the Windows Control Panel by double-clicking its icon.

   Control Panel

2. Open the Fonts panel by double-clicking its icon.

   The Fonts panel appears.



Fonts

Installed Fonts:

Arial (TrueType)
Arial Bold (TrueType)
Arial Bold Italic (TrueType)
Arial Italic (TrueType)
Courier 10,12,15 (VGA res)
Courier New (TrueType)

Sample

No fonts are selected.

The size of the font on the disk is: 0 KB.

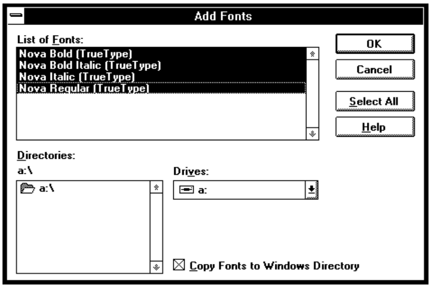
3. Click the **Add** button.

   The Add Fonts panel appears.

4. Choose the appropriate disk drive from the **Drives** list.

5. Choose the appropriate directory from the **Directories** list.

Note: If you expect more (or different) font styles to appear in the Add Fonts panel, make sure you've followed the naming conventions for Windows font families described in Chapter 3, "Creating New Fonts". If the family and style names aren't set up correctly, then the fonts you want to add may not appear in the list.



List of Fonts:
Nova Bold (TrueType)
Nova Bold Italic (TrueType)
Nova Italic (TrueType)
Nova Regular (TrueType)

Directories:
a:\
a:\

Drives:
a:

☒ Copy Fonts to Windows Directory

6. Select the names of the fonts you want to install from the List of Fonts or click **Select All**.

7. Click **OK** to install your font(s).

   The Add Fonts panel closes.

Your font is now installed. The font name will appear in the **Font** menu in every application that has a **Font** menu.

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

In Mac OS X

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 fonts managing utility then use this utility to disable or completely remove the font.

In Mac OS 7.1-9

1. Quit any open applications.
2. Open the Fonts folder (in the System Folder).
3. 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.

Removing a Windows font

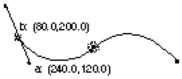
If you were using ATM or some other font utility for font installation then use it to disable or completely remove the font.

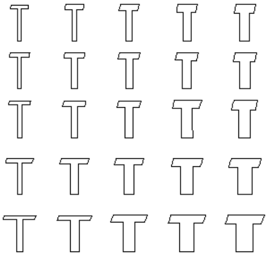
Otherwise

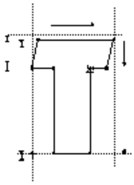
1. Open the Fonts control panel by double-clicking its icon or choosing it in the **Start** menu.
2. Select the font you want removed.
3. Move it to some other place on your disk.
4. If you want the font permanently removed from your disk, move it to the Recycle Bin or choose **Delete** in the **File** menu.

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.







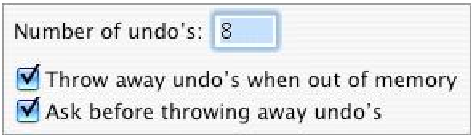
11

Choose **Preferences** from the **Fontographer** menu.

A dialog box will appear, and like some of the others in Fontographer, a pop-up at the top of the dialog box allows you to navigate through the various Preferences screens. Preferences are divided into four areas:

- General preferences
- Editing behavior
- Point display
- Windows and dialog boxes

General preferences



Number of undo's: 8

☑ Throw away undo's when out of memory

☑ Ask before throwing away undo's

Options for undo

This is where you can set the number of undo levels and redo levels. You can have Fontographer remember up to 101 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 20, depending upon how often you like to undo (and how far back you like to go).

The other two items in the undo area have to do with memory considerations. 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. For instance, if you have been running Fontographer for a couple of hours, and edited two hundred characters with an average of ten edits per character, you have 200 x 10 = 2000 undo levels floating around. If you let Fontographer get rid of these when it runs out of memory, you can sometimes reclaim a fair amount of memory. In addition to the **Throw away undo's when out of memory** item, there is an **Ask before throwing away undo's** checkbox. This is useful so you will know when Fontographer is running low on memory; an alert box will appear, asking if you want Fontographer to throw away the undo levels.

Using the keyboard to choose a character

This area has to do with typing in things from the font window. When you are looking at the font window and you type a key from the keyboard, Fontographer has to automatically choose the cell in the font window that corresponds to that key click. However, there are a couple of choices to be made to decide which character should be highlighted. Basically, Fontographer can either choose the character that corresponds to the font's encoding vector, or it can choose the character that corresponds to the encoding vector of the computer currently running Fontographer. The following example will clarify this description.

Suppose you have opened an Adobe Standard encoded font, and this preference item has been set to **Use the font's encoding to choose the character**. What is going to happen now can be a little confusing if you're not used to thinking in this manner. Fontographer, even though it is running on a Macintosh, will not use the Macintosh-standard character encoding. It is instead going to use the character ordering of the Adobe Standard encoding. For most of the alphabet and normal characters, you won't notice a difference. If you choose "a" from the keyboard, "a" is going to get highlighted in the font window. However, if you press **Option-2**, the quotedblleft character will get highlighted instead of the expected trademark character that Macintosh users expect to get when they key in **Option-2**. This is because **Option-2** from the keyboard maps to character number 170, and character number 170 in the Adobe Standard encoding is not "trademark" like it is in the Macintosh encoding, but is quotedblleft. To get trademark when you type **Option-2**, change the preference so Fontographer will use the machine-specific encoding vector.

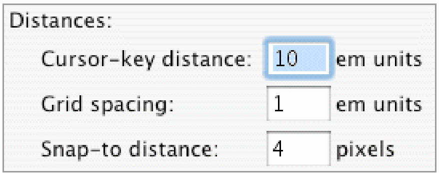
The main point of all this is, unless you are doing a lot of cross-platform font development or have some other reason to be using different encodings, keep this preference set to use your computer's encoding vector. That way, you'll get what you expect.

Note: You won't see actual Windows keystrokes (for example, Alt+0173) on the Macintosh regardless of encoding.

When reading an outline font

Before reading this section, read the previous section, “Using the keyboard to choose a character” on page 354. These two preferences are very closely related. Essentially, this preference item allows you to either see the font in its original encoding (which will typically be Adobe Standard encoding for most PostScript fonts), or to see the font in the encoding of your particular computer. We recommend, unless you have some particular need to see the font’s native encoding, that you have Fontographer re-encode each character to your computer’s encoding vector so the font window will place each character where you are used to seeing it.

Editing behavior



Distances:

Cursor-key distance: 10 em units

Grid spacing: 1 em units

Snap-to distance: 4 pixels

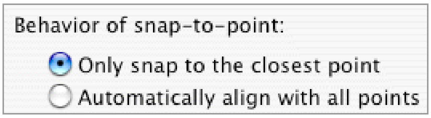
Distances

The **Cursor-key distance** 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).

Snap-to 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.

Behavior of snap to point



Behavior of snap-to-point:

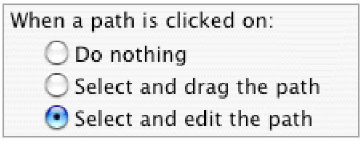
● Only snap to the closest point

○ Automatically align with all points

Snap to Point is a mode that makes aligning points much easier. If **Snap to Point** is on, 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.

When a path is clicked on

This 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.



When a path is clicked on:

○ Do nothing

○ Select and drag the path

● Select and edit the path

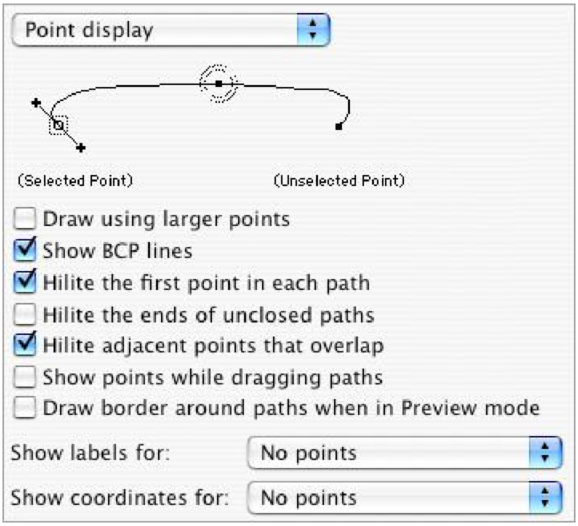
Do nothing is a good setting if you want to prevent yourself from modifying a path by clicking it accidentally.

Select and drag the path is a Macromedia 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



Point display

☐ Draw using larger points
☑ Show BCP lines
☑ Hilite the first point in each path
☐ Hilite the ends of unclosed paths
☑ Hilite adjacent points that overlap
☐ Show points while dragging paths
☐ Draw border around paths when in Preview mode

Show labels for: No points

Show coordinates for: No points

The picture area at the top 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.

Hilite 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.

Hilite the ends 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 characters in the font, see if any have this problem of paths being left open, and then turn off the preference when you have finished.

Hilite 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.

Show 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.

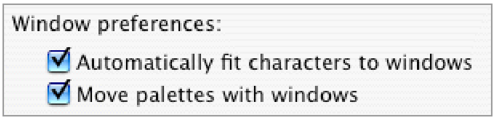
Draw border around paths when in Preview mode, when checked, causes a dashed border to appear around lines during edits in Preview mode. This lets you always see your lines even when dragging them over black areas, which is not an uncommon occurrence in Preview mode. This preference makes editing somewhat slower, however, and can distort the image of the character while you are editing it.

Show labels for 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.

Show coordinates for 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

Window preferences



Window preferences:

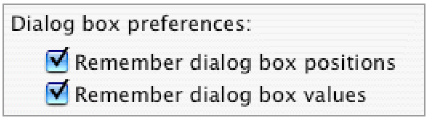
☑ Automatically fit characters to windows

☑ Move palettes with windows

Automatically fit characters 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.

Dialog box preferences



Dialog box preferences:

☑ Remember dialog box positions

☑ Remember dialog box values

Remember dialog box positions, 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 dialog box 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.

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 **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, 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 characters 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 character will be interpolated (unless you have chosen two very disparate fonts). Characters that don't get blended are frequently some of the seldom-used symbol characters, because the upper 128 characters can vary greatly from font to font. If this is the case, you can decide you don't care because you never use those characters, and go on and use the font successfully. Or you can try to figure out why those several characters couldn't be interpolated like the rest, which is what this section is about.

The blending process

Font Blend is very much like the Macromedia 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.

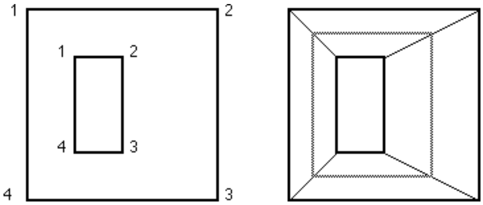


Figure 1

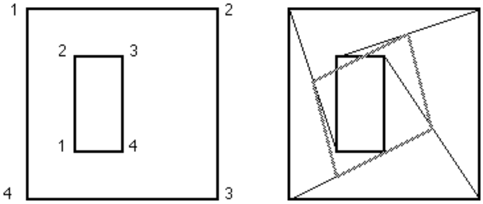
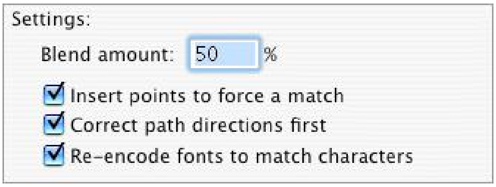


Figure 2

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.


Settings:
Blend amount: 50 %
☑ Insert points to force a match
☑ Correct path directions first
☑ Re-encode fonts to match characters

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.

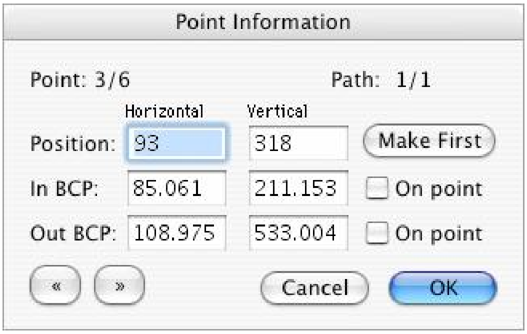
To make characters interpolate more easily, Fontographer reorders the points (and the paths) if you click the **Correct path directions** checkbox. If you correct path directions on all characters of each font you are trying to interpolate, Fontographer will almost always arrange the points so that they match up, and interpolating will give you the right result. On rare occasions, due to the way some characters might have been drawn, you can confuse Fontographer's point ordering technique. If this occurs, you will know it from the interpolate results (characters 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 character, choose **Selection Info**, then move to the next or previous point until you are at the desired starting point. Check the **Make 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 **Show labels for** 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 characters are likely to interpolate well or not.

➲ In some characters, 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.



Point Information

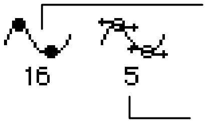
Point: 3/6 Path: 1/1

	Horizontal	Vertical	
Position:	93	318	Make First
In BCP:	85.061	211.153	□ On point
Out BCP:	108.975	533.004	□ On point

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 character in different fonts. You will have to figure out the best technique for getting the points to match on a character-by-character basis. Sometimes it is better to remove extra points from a character with too many points, and sometimes it is better to add points to a character 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.



total number of points

number of selected points

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 successfully use Fontographer.

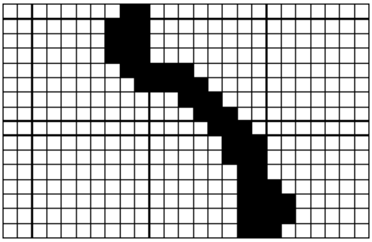
Are you still with us?

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.

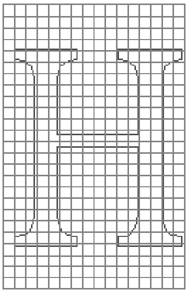
The reality of laser printing is that the smooth lines you get out of your printer aren't really smooth at all. They are composed of little dots (called pixels) which, like a mosaic, are all assembled on a grid to form a picture. At high enough resolution, your eye perceives smoothness even though there really is jaggedness.



Close-up of a printer's output.

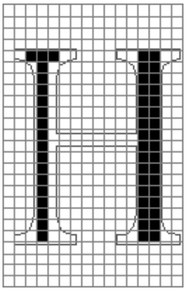
When the printer prints, it has to choose which pixels to turn on or off to best achieve the desired image. As you can imagine, it isn't quite as nice as just drawing smooth curves on paper with a pen.

To see how that relates to characters 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:

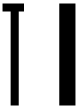


Remember that the printout you get is composed entirely of pixels, so if we were to print now, we would get a blank sheet because the computer hasn't yet turned on any pixels. That's the hard part; drawing the pure mathematical outline was easy.

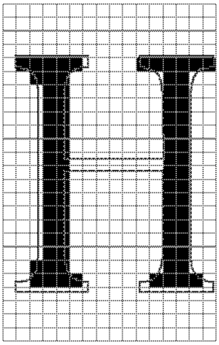
The trick is to figure out, by examining the outline, which pixels to turn on and which to leave off. A reasonable start is to simply turn on all the pixels that are entirely contained by the character outline. That gives this result:



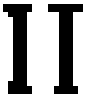
Or, shrunk down to size, this:



Since that was less than an ideal "H", we obviously need to be more clever. 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:

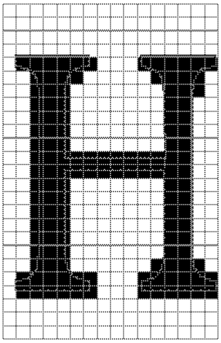


Or, scaled down, this:

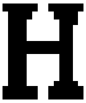


That didn't turn out very well either. Note that, although we have at least a semblance of the four serifs, we still don't even have a cross bar.

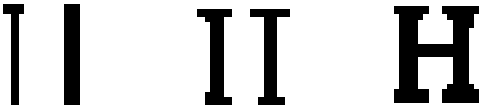
Let's try one more way: we'll turn on all the pixels that have any portion whatsoever of the outline touching them:



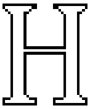
When scaled down, it looks like this:



To review, here are the three H's thus far:

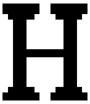


And here is what the mathematical outline looks like:



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.

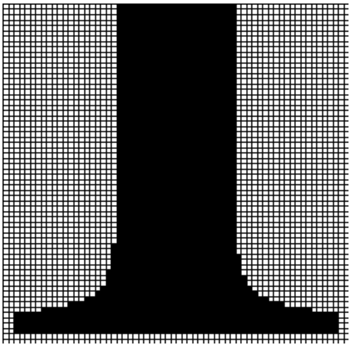
What we would like is something like this:



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 character. 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 characters 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 character 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.



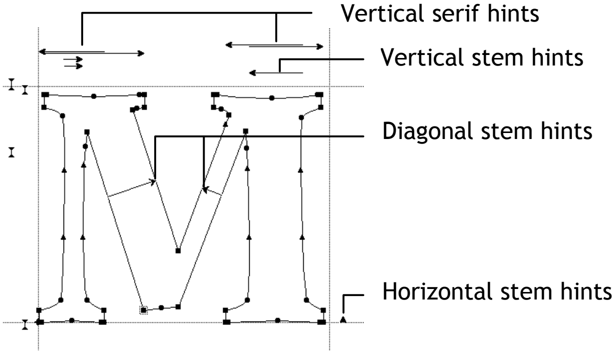
Since hinting can control at most one pixel’s difference either way, the human eye can’t usually perceive the difference hinting makes at high resolutions.

To sum up: A hint is special information placed into a character’s outline definition that causes the character’s outline to be adjusted in a way that improves the character’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 character 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 character 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 character to character. 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.

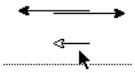
You can tell if Fontographer is on automatic by looking in the **Hints** menu. This menu contains an item called **Autohint**; a checkmark indicates that Fontographer will recalculate hints for the selected character every time the outline gets edited. If you don’t want that, choose **Autohint** from the menu. The checkmark will go away, and autohinting will be turned off for that character. Be careful, because if you then turn **Autohint** back on by reselecting it from the menu, Fontographer will recalculate new hints for that character.

When autohinting is on, you can watch Fontographer come up with new hints as you edit a character. Just make the Hints layer visible, check to see that **Autohint** is on, and edit the character 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.

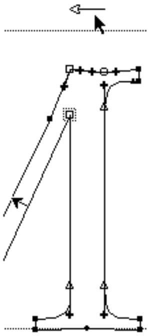
You can select hints with the mouse, just like you select points. Selected hints have a hollow looking arrowhead:



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 character outline in the case of overlap). Hints may also be reordered from the Hint Parameters dialog box, as described in "Hint parameters" on page 386.

Changing the direction of a hint can be done by selecting the hint, and then choosing **Flip Hint Direction** from the **Hints** menu or clicking the arrowhead of the hint. See "Hint parameters" on page 386 for more information on hint direction. To see what points were responsible for creating a particular hint, simply **Option**-click the hint, and those points will become selected:



Removing hints

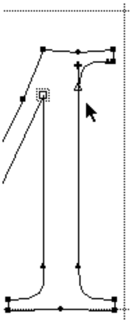
Removing a hint is simple. Just press the DELETE key, or choose **Clear** from the **Edit** menu. Be careful – if you have just OPTION clicked on a hint to study the points that made it, you will remove both the hint and those points if you press the DELETE key. Be sure to deselect the points, by pressing 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:

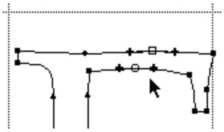
1. Select two points that define the stem.



2. Choose **Make Vertical Stem** from the **Hints** menu.

To make a new horizontal stem hint:

1. Select two points that define the stem:

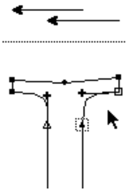


2. Choose **Make Horizontal Stem** from the **Hints** menu.

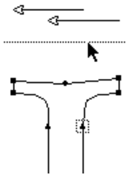
To make a new serif hint:

There are two ways to do this:

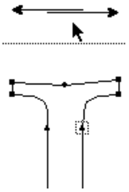
1. You can start off with two hints. In the case of a serif on a vertical stem, you need one hint from the right side of the stem to the left side of the serif; the other should be from the left side of the stem to the right side of the serif. Hint direction doesn't matter.



2. Select those two hints.



3. Choose **Build Serif** from the **Hints** menu, and those two hints will be replaced by a serif hint.



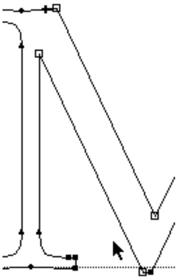
The other method is to select four points that define the serif and choose **Make Horizontal Serif** or **Make Vertical Serif** from the **Hints** menu.

To get the two separate hints back again, select the double-headed serif hint, and choose **Split Serif** from the **Hints** menu.

✐ ***Note:*** Serif hints are useful only in Fontographer's internal bitmap generation and for TrueType fonts. PostScript fonts do not use serif hints.

To make a new diagonal hint:

1. Select four points that define the diagonal stem.

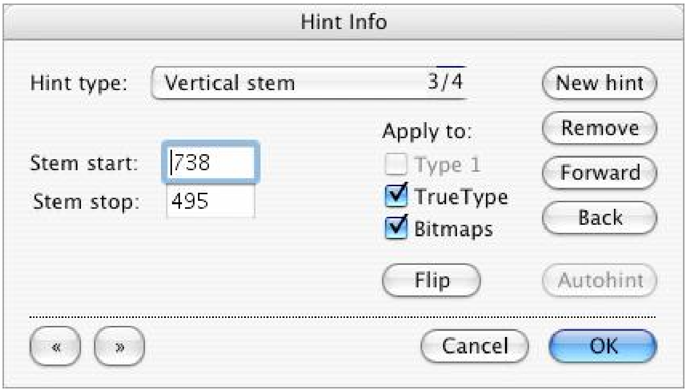


2. Choose **Make Diagonal Hint** from the **Hints** menu.

✐ ***Note:*** Diagonal hints are used only for TrueType fonts. PostScript fonts do not use diagonal hints.

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 character and modify them.



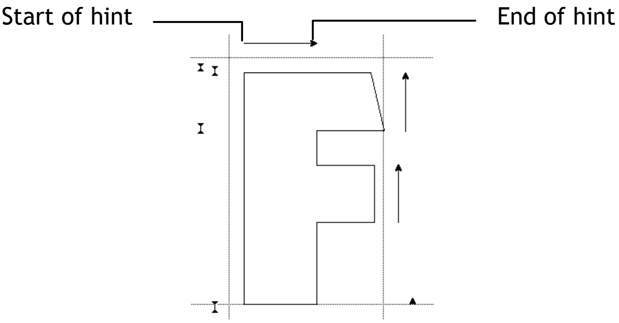
Hint Info

Hint type: Vertical stem 3/4

Stem start: 738

Stem stop: 495

Apply to:
☐ Type 1
☑ TrueType
☑ Bitmaps

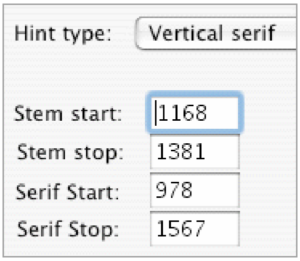


Hint type

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.

If we change to Vertical serif, for instance, we get corresponding information about that particular serif.



Hint type: Vertical serif

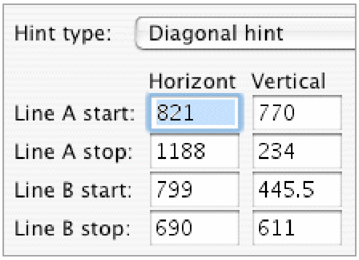
Stem start: 1168

Stem stop: 1381

Serif Start: 978

Serif Stop: 1567

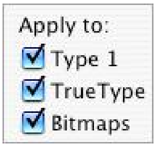
Changing to Diagonal hint displays this screen:



Hint type: Diagonal hint

	Horizont	Vertical
Line A start:	821	770
Line A stop:	1188	234
Line B start:	799	445.5
Line B stop:	690	611

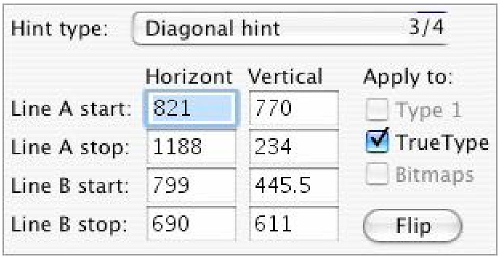
Apply to



Apply to:
- ☑ Type 1
- ☑ TrueType
- ☑ Bitmaps

These three checkboxes influence how the particular hint gets output. You can control, on a hint-by-hint basis, the font types that should receive the hint. For example, there may be some hints that work well for PostScript fonts but not for TrueTypes. By setting these flags, you can have one Fontographer database that is tuned to different font formats. The **Bitmaps** checkbox refers to Fontographer's internal bitmap generation algorithms – if you discover some hints that seem to throw off Fontographer's automatic bitmap generation, you can suppress those hints. That may save time over having to hand edit the poorly created bitmap images.

Of course, you don't always have the option of choosing hint format type. If you choose Diagonal hint from the **Hint type** pop-up, Type 1 and Bitmaps both become dimmed, indicating that neither of those two choices are able to accept diagonal hints.



Hint type: Diagonal hint 3/4

	Horizont	Vertical
Line A start:	821	770
Line A stop:	1188	234
Line B start:	799	445.5
Line B stop:	690	611

Apply to:
- □ Type 1
- ☑ TrueType
- □ Bitmaps

The buttons

Flip will change the direction of the current hint. See “Hint parameters” on page 386 for more information on hint direction.

New 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 character into an alignment zone).

Remove 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.

Autohint performs the same task as the **Autohint** item in the **Hints** menu – it will turn on autohinting and rehint the character.

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.

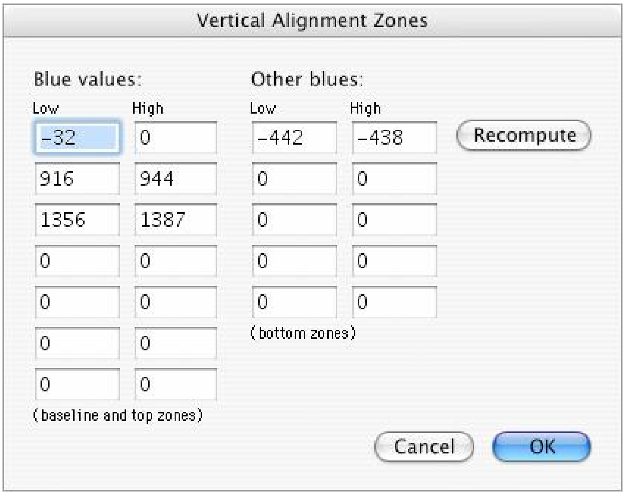
Vertical Alignment Zones

In addition to stem width hints, Type 1 font hints also control the vertical positions of characters. 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 characters, they apply to the entire font. However, it is helpful to think of them as corresponding to certain groups of characters. When Fontographer computes or recomputes the vertical alignment zones, the first pair of values is based on characters 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 (characters such as "C" and "O"). The second value is the normal baseline.

The remaining blue zones are computed from the tops of characters and the x-height (characters 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 characters such as the descender line in characters "g" and "y".

Choosing **Vertical Alignment Zones** from the **Hints** menu brings up this dialog box.


Vertical Alignment Zones

Blue values: Low	High	Other blues: Low	High
-32	0	-442	-438
916	944	0	0
1356	1387	0	0
0	0	0	0
0	0	0	0
0	0		
0	0		

(bottom zones)

(baseline and top zones)

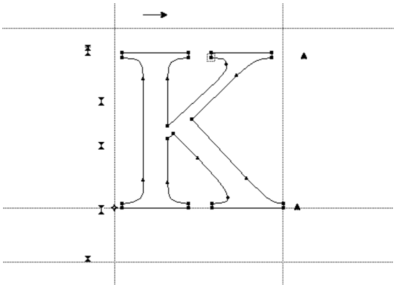
We don't really know why the PostScript hinting variables these fields were named after have anything to do with the color blue. But they do, and once again, you are encouraged to look at *Adobe's Type One Font Specification*, which covers PostScript hinting quite admirably. Be aware that, although the 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 **Recompute** button. If that button is inactive, that means you haven't edited anything yet, so the recomputed values would be the same as those which 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 characters $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.

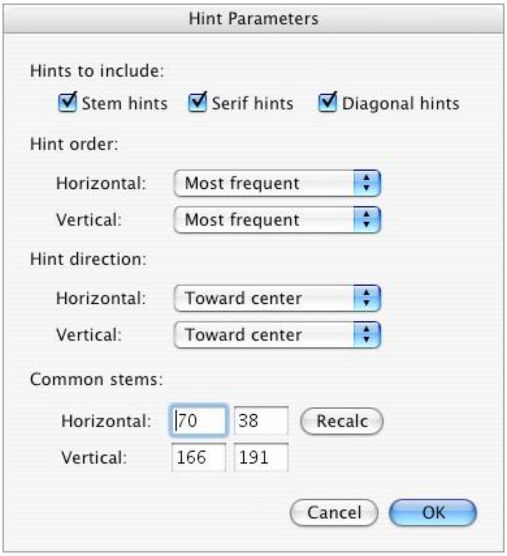


To remove a zone, you can select it and then press the **Delete** key. You may also go into the Vertical Alignment Zones dialog box and enter "0" for both the low and high value for a particular zone.

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.

Hint parameters

Choosing **Hint Parameters** from the **Hints** menu brings up this dialog box.



Hint Parameters

Hints to include:

☑ Stem hints ☑ Serif hints ☑ Diagonal hints

Hint order:

Horizontal: Most frequent

Vertical: Most frequent

Hint direction:

Horizontal: Toward center

Vertical: Toward center

Common stems:

Horizontal: 70 38

Vertical: 166 191

Hint Parameters, like most of the other dialog boxes, will affect different groups of characters, depending upon which window you are in when you bring it up. If you are in the outline window, it will only affect the hints of that character. If you bring it up from the font window, it will affect all the currently selected characters.

Note: The Common stems values are an exception – they always apply to the entire font.

Type 1 fonts have a restriction that hints cannot overlap. There is a way to get around this, known as hint replacement, but the technique is not implemented in Fontographer. Maybe someday the Type 1 font renderer will be changed to handle overlapping hints. This mainly affects serifs, which require overlapping hints to control the serif width on each side of the stem. Without overlapping hints or hint replacement, serif widths are uncontrolled. Fortunately, this effect is not generally a serious problem in practice.

Fontographer applies heuristics to pick which hints to keep and which to ignore when sorting through overlapping hints. You can override this by changing the pop-ups in Hint order so you can control whether the most frequent hints get applied first, or the largest, or some other option.

Hints to include

This is a way to suppress entire groups of hints. If you are developing PostScript fonts, for instance, you could decide that you never want to look at any serif hints, or diagonal hints, since those kinds of hints can't be applied to PostScript fonts. By deselecting Serif hints and Diagonal hints, those hint types will never appear in the outline window.

Hint order

Changing hint ordering is useful for PostScript Type 1 and TrueType fonts. Since PostScript doesn't handle overlapping hints, you sometimes need to specify which of the overlapping hints should get used – the hints closest to the outline in the window are the ones that will get used in the case of overlap. Typically, you will want to apply the most frequent hints first, on the grounds that they will work best for that character in comparison to the rest of the font. (Hint frequency is based on examining all the hints in the entire font and sorting them according to frequency of occurrence.) Usually, changing these pop-ups will result in changes in hint ordering which you can see in the Hints layer of the outline window.

➲ ***Tip:*** Many people like to have their hints ordered according to frequency. To determine frequency of hints, Fontographer has to look over all the characters in your font, count up the hints, sort them, figure out which are the most common, and then reorder all the hints. If you do substantial editing to a font, chances are that it will need different hints, and so hint frequency may have changed. In other words, the hint frequency Fontographer found the last time it ordered the hints is probably out of date. Before generating fonts, you should select all the characters from the font window, bring up the Hint Parameters dialog box, and choose to reorder all the hints. This only impacts you if you care about hint ordering; if you don't, don't worry about it.

Hint direction

The hint direction tells which way the outline will be adjusted during gridfitting (aligning the outline to the pixel grid used by the screen and/or printer). Generally, you want all your hints to point inward (Toward center); but that is not always the case, so we included these pop-ups as a way to customize this behavior easily for entire characters. (Individual hints can have their direction changed on a hint-by-hint basis, in the Hints layer of the outline window.) Hint direction makes a difference for TrueType, but if it affects PostScript, we haven't been able to detect it.

Common stems

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 **Recalc** button.

Changing the default Hint Parameters

In the font window, Hint Parameters works with the selected characters. For example, if you wanted to modify some of the parameters for an entire font, you would simply **Select All** prior to bringing up Hint Parameters.

OPTION-**Parameters**, on the other hand, changes the defaults for new characters.

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 character 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 characters and choose **Autohint** from the **Hints** menu, so serif and diagonal hints can be found. This will remove the original stem hints and find new ones.

Autohinting is on by default, so when you edit a character, 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 **Autohinting** 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 characters. 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 resource editor to tweak Fontographer 4.7 on your Macintosh

The tips and notes in this section are highly technical and are for people experienced with how resources work on a Macintosh. If you want to try this, we recommend you learn to use Resorcerer or ResEdit before proceeding.

Adding custom encoding vectors

What is a custom encoding vector? Different languages have their own encoding vectors. It represents the layout of the script or alphabet. In previous versions of Fontographer, it was difficult to set up a foreign language script. It required setting new character names to each of the cells in the font. Starting with Fontographer 4.1 we have added two special resources to the STR# resource to enable you to set up a custom encoding vector with custom names. To create your own custom encoding vector, you will need to be familiar with the use of a Macintosh resource editor such as Resorcerer or Apple's own ResEdit. The particular resources we will begin looking at are id 384 and 386.

Before you begin, make a copy of Fontographer 4.7 to work with. When Fontographer is open in ResEdit, you will only be able to run one copy of Fontographer at once. Therefore, you should work on a copy so as not to damage your original copy of Fontographer.

This example will not necessarily set up the Unicode numbers for your new character names. If the names you use are listed in STR#400, then the standard Unicode number will match up with it. Otherwise, refer to step 11 for further instructions.

To begin:

1. Open a copy of Fontographer in your resource editor of choice. Our example uses ResEdit.
2. Double-click the STR# icon.
3. Open id 386. WAIT, first take a look in id 384… *Don't make any changes yet!* You will see a list like the one on the next page. Each line is a separate field in the resource.

```
7
21317
Adobe Standard
256
1
22597
Adobe Expert
256
1
19800
Macintosh Expert
256
1
```

The first line (7) refers to the number of strings in the resource. This resource has 7 strings, which in this case means seven encoding vectors. The next four lines comprise the first string in this resource. They set up the Adobe Standard encoding vector.

The 21317 is the STR# id in which you will find the list of character names in sequential order, starting at 1 and ending at 256.

The name of the encoding vector is next. It appears in the **Encoding** pop-up found in Font Info and Generate Font Files dialog boxes.

The next field, 256, represents the number of character names in the encoding vector. You can have more than 256 names, but never less.

The final entry in the string is the "1". This indicates that you want the encoding name to appear in the dialog boxes. If you do not want it to appear, set this to 0. We have done this for TrueType glyph encoding.

Now that you have seen the format for setting up your encoding, open id 386 to make your own custom encoding vector. Do not make any changes to id 384. This was just a rehearsal so that you can understand the format required.

4. Back in id 386, the resource opens with a 0 in one field. To add new fields in ResEdit, select the last line (in this case "2) ******) and press **COMMAND-K**. You will need to add 4 lines (type **COMMAND-K** four times).
5. After adding all the fields, go back to string 1, select the 0, and begin entering your data. The first field will be 1, for the number of encoding vectors in this resource.
6. The second field will contain the id of the new resource you will make, which will contain all the character names. Select a resource id that has not been used.
7. Here is an example of an encoding vector (STR# 386):

1	number of encoding vectors
21321	id for resource containing character names
Cyrillic	name of encoding vector
256	number of chars
1	makes the encoding name show up in the dialog boxes

8. Close the resource id 386.

Creating the names resource

Now you need to decide how to make your resource of character names. If you have a font that already has your custom names, the easiest way to get a list of the names is to export an AFM file from Fontographer. Open the AFM file in a text editor and then copy and paste the names into the ResEdit resource.

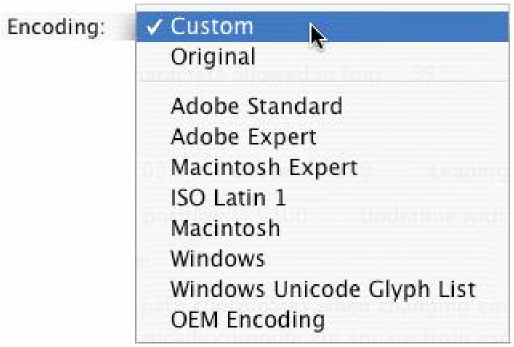
Rather than create your own resource by typing **COMMAND-K** 256 times, you'll find it faster to copy one of the existing resources and replace your names with the ones there.

9. Select id 21317 and duplicate it. It will likely be given an id of 129. To change an id in ResEdit, select the resource id and type **COMMAND-I**. Then type in the number you want. Make sure it is the same number you entered in step 6.

This is now a copy of the Adobe encoding vector. To begin changing your names, just start replacing the names in the strings. You will notice that the first 32 fields are .notdef. Most operating systems do not reliably access characters in this range. However, certain encoding vectors, such as Arabic, require that characters be mapped in that range. As was mentioned before, if you have your custom encoded font, then you already know where the characters belong in the font. If you don't have your custom encoding font, then you must find out what the sequential order of the character names is.

Note: Mistakes in name entries will manifest themselves in the font, so be careful and accurate in your editing.

10. When you have entered all the character names, close the resource, **Save**, and **Quit**. Fontographer is now ready to supply your custom encoding at the click of a mouse.



Encoding: ✓ Custom

11. Open Fontographer and create a new font. Choose **Font Info** from the **Element** menu and select your own encoding from the encoding pop-up. Now copy your characters into their proper locations in this font database.

    While in the font window, select Unicode Glyph list from the **View by** pop-up. If a ** appears in the character label over the custom characters, then you know that the custom names did not map to known Unicode numbers. All of the Unicode data in Fontographer is stored in STR# 400. It is a large resource; so large that ResEdit can't open and edit it in the template view. 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 STR# 400. It's better to make further changes to the specific character in the Character Info dialog box.

12. When your font is ready to generate, just choose **Generate Font Files** from the **File** menu and select your custom encoding from the encoding pop-up.

Setting developer IDs

If you are a font developer, who has been issued an unique developer ID by Microsoft, then we have provided a means for you to hard code that ID into Fontographer by editing the STR# 5140 resource.

Open the resource in Resorcerer or ResEdit. The fourth string is the TrueType (Vendor) Developer id.

The default selection is Pyrs. This is the ID Microsoft supplied to Pyrus North America Ltd. Replace the Pyrs four-character string with your own unique vendor ID.

Customizing Sample Text printout

Yet another ResEdit trick is to edit the STR# resource to change the font used to label coordinates in the character 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. Open STR#6021.
2. 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 resident in your Adobe PostScript printer.

Customizing Fontographer sounds on your Macintosh

Fontographer 4.7 has the ability to play sounds during certain actions. It plays sounds when you snap to a point or snap to a guide in the outline window, and it plays a sound whenever the Please Wait dialog box closes. The sound file *Fontographer Sounds* is installed in your Fontographer folder. Don't change the name of the file, because Fontographer looks for that name specifically.

If you're comfortable using Resorcerer or ResEdit, you can easily customize your sounds.

1. Get the sound you wish to use.
2. Convert the sound into a "snd" resource if necessary.
3. Use Resorcerer or ResEdit to paste the snd resource into the *Fontographer Sounds* file.
4. Then name the resource (the spelling must be exact) one of the following names:

   Snap to point
   Snap to guide
   Operation done
5. Save and close the file.

That's it. Run Fontographer and you'll now hear the sounds.

If you find the noises emanating from your Macintosh annoying, remove the file *Fontographer Sounds* from the folder where you keep your Fontographer application.

Macintosh Expert Advice

KCHR resources

KCHRs (Keyboard CHaracter Resources) are a seldom-heard-of Macintosh resource that can do some unusual things with fonts. Basically, KCHR resources are system-level resources (meaning they aren't part of your fonts – but separate things that live in your system software) that allow you to set up unusual keyboard sequences for accessing characters. For instance, if you edited a font, and put a drawing of a hotdog in the character that is usually accessed by typing **Shift-Option-W**, you could set up a KCHR that would allow **Option-H-D** to access the hotdog character (giving you the benefit of a more easily remembered keyboard sequence).

It is conceivable that you would want to make a KCHR resource that would find fractions you put into your font. You can create fractions, and place them anywhere in your font. Then you could set up a KCHR resource such that **Option-1-2** would give the one-half character, **Option-1-4** would give one-fourth, **Option-5-8** would give five-eighths, and so on.

KCHRs are important in foreign character handling. Most, if not all, of the foreign Apple systems use various KCHRs to do the different keyboard mappings required for some languages.

Basically, KCHR combinations are a lot like the dead-key modifiers. In the American Macintosh systems, for example, the character "é" is accessed by typing **Option-e-e**. The first character, **Option-e**, doesn't do anything when you type it; it's waiting for the next key to decide what character to show (**Option-e-i**, for instance, brings up a different character). Other dead-key modifiers that exist on the American system are **Option-i**, **Option-u**, **Option-o**, and **Option-n**. By making your own KCHR resources, you can have practically any character involving a modifier (such as **Option**) be a dead-key. KCHRs allow you to use up to 16 modifier combinations involving **Shift**, **Option**, **Caps Lock**, **Control**, and **Command**.

Fontographer's font window (and sample printouts) support KCHRs. So if you set up some custom ones, they will work as you would expect when you enter characters from the keyboard in the font window.

ResEdit is the only tool we know of to actually edit KCHRs. It has a difficult interface, but can be made to work. You will need to find a reference book about how to best use ResEdit in order to fully understand how to edit and create KCHRs. But for those interested, you can do some pretty interesting stuff with KCHRs.

Beginning with version 10.2 (Jaguar), Mac OS X supports an XML format for keyboard layouts (.keylayout files). You can edit and create new keyboard layouts in this format with Ukelele or Keyboard Builder shareware utilities that can be found on the web.

Refer to Apple's Technical Note 2056 to learn how to install new keyboard layouts (KCHR or XML-based) in Mac OS X:

http://developer.apple.com/technotes/tn2002/tn2056.html

Multiple master fonts

Use Fontographer to create multiple master fonts that can be used with Adobe’s Font Creator utility. Font Creator 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 Font Creator to make a wider version of your existing multiple master font, while still retaining the size and weight of the original typeface. Font Creator 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 you create with Font Creator.

✐ ***Note:*** You must have Adobe Type Manager® and Font Creator. If you need information about any of these 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.

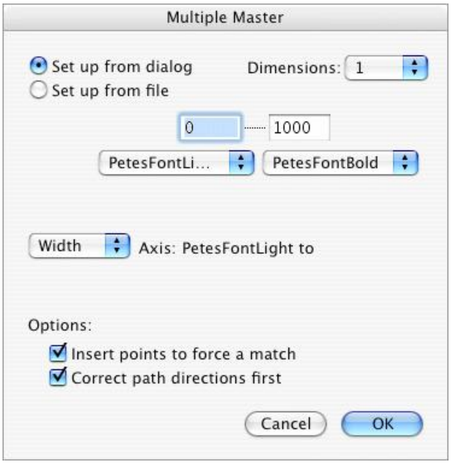
`Light⟵ Normal⟶` **`Bold`**

`R          R          `**`R`**

In our example, we made a one-dimensional font that Adobe Font Creator 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.



● Set up from dialog   Dimensions: 1
○ Set up from file

0 ········ 1000

PetesFontLi...   PetesFontBold

Width   Axis: PetesFontLight to

Options:

☑ Insert points to force a match
☑ Correct path directions first

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.

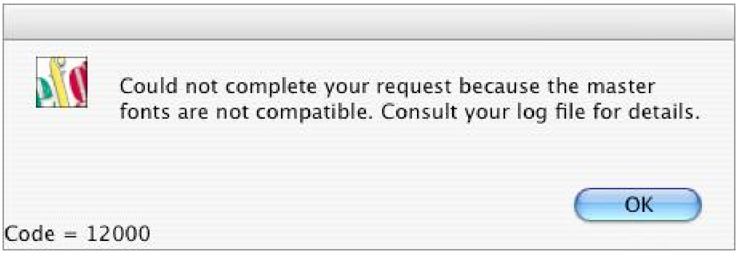
4. Deselect the options **Insert points to force a match** and **Correct path directions first**.
5. Type in your values.

   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 *Fontog.log* file.



Could not complete your request because the master fonts are not compatible. Consult your log file for details.

Code = 12000

The *Fontog.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 *Fontog.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.

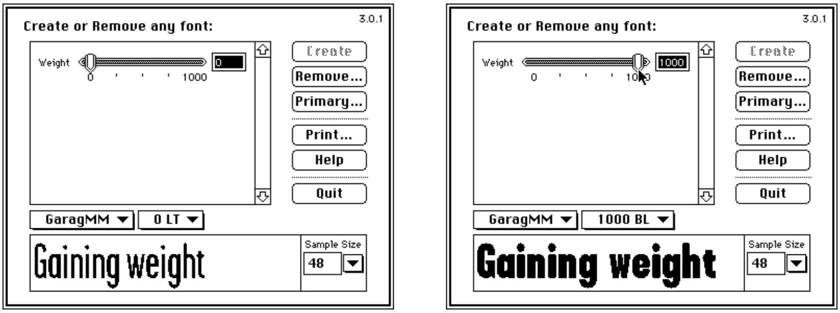
```
>>>>>>>>>>>>>>>>><<<<<<<<<<<<<<<<<

>>>>>>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 (this is the folder where Font Creator resides). 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 in Font Creator.

Now you can open your font in Adobe Font Creator and create as many new versions of your font as there are combinations between your preset width values of 50 and 500.



The Garage Font developed by Tobias Frere-Jones of Font Bureau is a one-dimensional multiple master font with weight changing capability. The font was based on a font found on a parking garage ticket, hence the name Garage Font. To change the weight you select any weight, from zero to one thousand using the slider, and watch as the text sample changes.

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 independence.

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**.

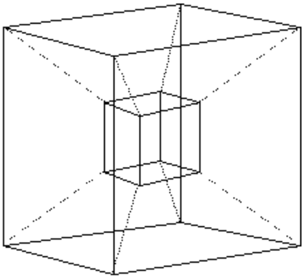
Multiple master font technology is really just interpolate and name technology. Why not try letters that bounce up and down, letters that flip around, or a little man that jumps around the screen?

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.

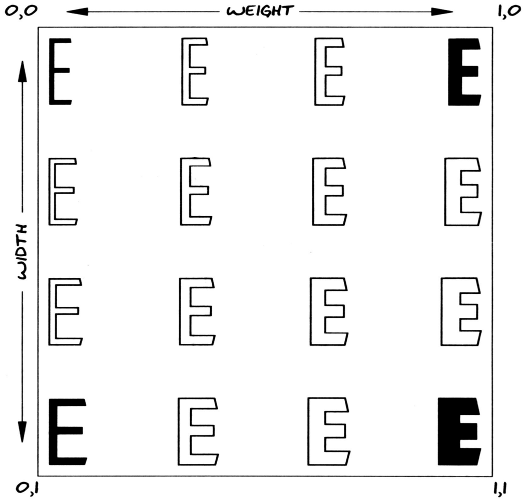


As we stated in the overview, for one-dimensional fonts, the two fonts are at opposite ends of a one-dimensional line segment. Typically, this single dimension could be weight, width, contrast, or optical size.

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 coordinates) 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 labeling 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 labeling. 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

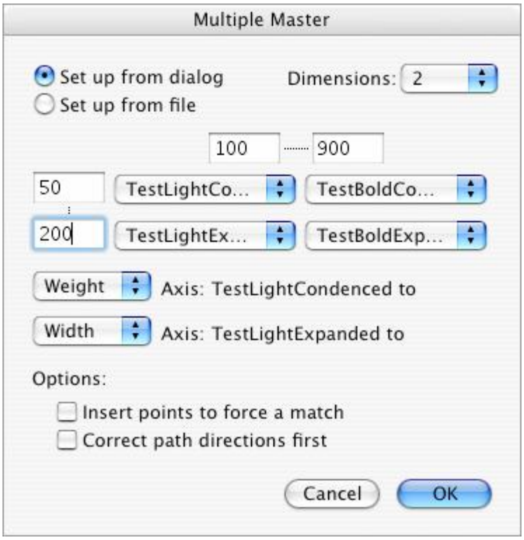
Including an optical size axis in your multiple master font provides a unique opportunity to optimize legibility for the entire range of sizes at which the font will be used. Traditionally, punch-cutters developing metal type would alter various aspects of a design depending on what looked correct for 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.)



Multiple Master

● Set up from dialog
○ Set up from file

Dimensions: 2

100 — 900

50 — TestLightCo... — TestBoldCo...

200 — TestLightEx... — TestBoldExp...

Weight — Axis: TestLightCondenced to

Width — Axis: TestLightExpanded to

Options:

□ Insert points to force a match
□ Correct path directions first

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 31 characters in a file name, and other platforms may have their own limitations for font names.

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 **Set up 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 **File** 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) for use in the Font Creator utility. 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 *Fontog.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 **Set up 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 **Set up 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 **Set up from file**.
4. Click the **File** 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 *Fontog.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. Restart your system so that ATM 3.0 sees the new multiple master font, and you should be ready to explore your font coordinate space with the Font Creator program. (If this is the first time you’ve installed this font, you probably will not have to restart, but if you have changed an existing version, you should restart after installing it.)

Of course, there are still a few things that can go wrong. Not only must the points of corresponding master characters be compatible, but the same type of hints must be defined in each character. If you were unlucky enough to create a font that automatically hinted differently in the different characters, you will have to manually edit the characters that are different. See “Font hinting” on page 368 for details on how to do this.

The log file produced by Fontographer gives precise information about where the hints differ between characters. When it finds a mismatch, it tells you how many it found, the character(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 character 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 character. These are called compatibility check errors. You can look at this list to see how the characters 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 characters 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.

12

Windows in Fontographer

Font window

The font window shows you all the characters in your font at 24 points. The View by menu at the top of the window allows you to change the label over each character 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 character, it means that you cannot access the character from the keyboard using the currently selected encoding. 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 character.

✐ ***Note:*** When you see a double-asterisk (**) above a character, it means that you cannot access the character from the keyboard using the currently selected encoding. You can change encoding in the **Font Info** dialog box in the **Element** menu.

Decimal

The Decimal item shows the decimal value of a character'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 character'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 character, it means that the slot is undefined. You can type the contents you place in the slot.

Left sidebearing

The left sidebearing mode shows the measurement of the character's left sidebearing – the distance between the origin line and the left edge of the character's outline – in em units.

✐ ***Note:*** A negative number indicates that some part of the character'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 character, it means that the character is undefined. You can type the contents you place in the slot.

Right sidebearing

The Right sidebearing mode shows the measurement of the character's right sidebearing (the distance between the width line and the right edge of the character's outline) in em units.

✐ ***Note:*** A negative number indicates that some part of the character'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 character, it means that the character is undefined. You can type the contents you place in the slot.

Fill Tint

The Fill Tint item shows the percentage of black that will fill the character 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 character is unfilled or that it is undefined using currently selected encoding. You can type the contents you place in the slot.

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 character, it means that the character is not stroked or that it is undefined. You can type the contents you place in the slot.

Weight

The Weight item shows the character'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 character, it means that the character slot is empty, includes no weight in em square units, or is undefined. You can type the contents you place in the slot.

Outline window

The outline window shows you the outlines (or strokes) of the character 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 character in the outline window to some other character by merely typing the keystroke(s) of that character. If you click the lock, it will open and you will then be able to change characters simply by typing the keystroke(s) of the new character. 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.

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 a check 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 character.

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 character 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 character. Dragging from the origin and baselines creates horizontal and vertical guidelines that will appear behind every character in the font. Like guidelines, any outlines drawn here will not be used for PostScript creation. They will appear gray behind the characters.

Hints layer

The Hints layer displays hints that define features like vertical stems, horizontal stems, serifs, and cap-heights. Here you can adjust the hinting of individual characters and edit character 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.

Switching characters

If you need to change the character in the outline window to the next sequential character, you can use the **View** menu's **Next Character** item – **Command-]**.

To change to the previous sequential character, use the **View** menu's **Previous Character** item. When the lock icon is off, you can change characters simply by typing the character – **Command-[**.

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 character'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 character'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 character, 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 characters by merely typing the keystroke(s) for the new character. If you click the lock, it will open and you will be able to change characters by simply typing the keystroke(s) of the new character.

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 character 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 character 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 character 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.

Ascent/Descent/Offset/Width

In the info bar at the top of the window, you find the character’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 character. The width indicator shows us the width of the character in points.

From outline

When you press this button, the bitmap will be recalculated in order to correspond with the outline of the character. This is useful when you change a character’s outlines after having created bitmaps. To recalculate bitmaps for more than one character at a time use the **Recalc Bitmaps** command in the **Element** menu.

Magnification

You can change the magnification of this window with the **Magnification** items in the **View** menu or you can use keyboard shortcuts. To enlarge the bitmap’s magnification, hold down the **COMMAND** key and **SPACEBAR** and click anywhere in the window. To reduce the bitmap’s magnification, hold down the **OPTION-COMMAND-SPACEBAR** and click anywhere in the window. You can also use the magnifying tool.

Scrolling

The **SPACEBAR** switches the current tool to a hand tool that will allow you to scroll the character back and forth in the window.

Switching characters

If you need to change the character in the bitmap window to the next sequential character, you can use the **View** menu's **Next Character** item – **COMMAND-]**.

To change to the previous sequential character, use the **View** menu's **Previous Character** item. When the lock icon is off, you can change characters simply by typing in the character – **COMMAND-[**.

Changing point sizes

To select the Next or Previous bitmap point size choose one of these menu items from the **View** menu. **COMMAND+=** will give you the next point size up. **COMMAND+-** 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.

File

Use this button to load text from a file into the textbox and display area. Click the **UP** and **DOWN ARROWS** beside the file button to scroll vertically within the file.

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.

Kerning

To display the characters in the textbox with or without kerning information, click the **Kerning** checkbox. When the option is switched on, 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.

Menus

The Fontographer menu

About Fontographer

This dialog box shows you information about your current setup. You can see the version number below the program title. 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 101), 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 11, “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 **Close All**, **Save All**, **Cancel**, or **Review**.

The File menu

New Font

In Fontographer when you select **New Font**, 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 443.)

Open Font

Open Font shows you a standard file dialog box that allows you to open a Fontographer outline file, a PostScript font, or a TrueType font file. The dialog box also allows you to display files according to their formats.

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

The **Revert** 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 computer type 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 encoding vectors, outline font IDs, and so forth.

Import

Importing bitmap fonts, EPS, 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: Fontographer database files, TrueType files, AFM files, PFM files, and Macromedia metrics files. 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. Import TrueType characters also works from double-byte TrueType fonts.

Export

When you want to export metrics information, an EPS graphic or a PICT image, just select **Export** from the **File** menu. You can choose **Metrics**, **EPS or PICT** from the submenu. When you choose **EPS or PICT**, a dialog box appears allowing you to choose the format type for the exported material, 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 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: Macromedia metrics, AFM, or PFM.

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 Characters.

Printing Sample text offers three choices. All characters will print all the characters in the font you are modifying. Selected characters will print only those characters 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 or directory. 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 characters in the font including offsets, widths and key codes, and offers you the choice of Selected or All characters, and format types.

Kerning pairs prints the kerning pairs in the current font or in a monospaced System font.

Characters prints four kinds of print samples: a single full-page sized character, assorted sizes of the same character, a sample showing the points of a character, or one that shows points and x/y coordinates.

The Edit menu

Undo

Fontographer has 101 possible levels of undo. The default setting is ten. This means that you can undo the last ten things you changed 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 101 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 character width of a selected character to the Clipboard.

Copy Reference

Copy Reference copies a reference of a selected character into the Clipboard, which can then be pasted to allow layered objects. You must use **Copy Reference** instead of **Copy** if you want to create a composite character.

Unlink Reference

Unlink Reference replaces a referenced character 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 character 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%, and 200%.

Next Character

This item switches the currently selected character to the next character 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 Character

This item switches the currently selected character to the previous character 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 character(s) or selected parts of a character. 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 character 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 character(s) or selected parts of a character. You can choose for the rotation to occur relative to the basepoint, the character 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 characters. If 100% scaling is selected, Fontographer does not change the size of the character(s). If you wish to make selected characters 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 characters 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 character 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 characters uniformly, with 100% representing the current size. Scale relative to the basepoint, the character origin, or the center of selection.

Skew

The **Skew** item allows you to skew any character(s) or selected parts of a character. The skewing will occur relative to the basepoint, the character 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 character to the left; negative degrees slant right.

Arrange

Arrange allows you four choices when you are working with referenced or composite characters. 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

In the top of the dialog box you will see the name of the current font and its style. Next you will see the encoding options. If you choose Adobe® standard character encoding from the **Encoding** box, Fontographer displays and stores a font with Adobe encoding. Do not check this option if your fonts have all the upper 128 characters defined.

All the numbers in this dialog box are in em square units. The ascent and descent default to a sum of 1,000 units for a PostScript font, or 2048 for a TrueType font (refer to Chapter 8, "Generating and Exporting Fonts").

➲ ***Tip:*** Leading is normally left at 0; but if you must set it, a good setting is 10% of the sum of the ascent and descent.

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.

You can put any text in the **Notice** you like, but it must be limited to 100 characters or less. Normally you will put a copyright notice and the **Embedding** flag here (see the next section).

You would deselect the **Automatically compute em square from ascent + descent** if you want an em square that is not equal to the sum of the ascent plus the descent.

With the **Retain path coordinates when changing em square** checked, the entire font will look smaller if you make the em square larger.

✐ ***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. If you do not do this, any characters of the font above 127 will be incorrect.

Font Embedding

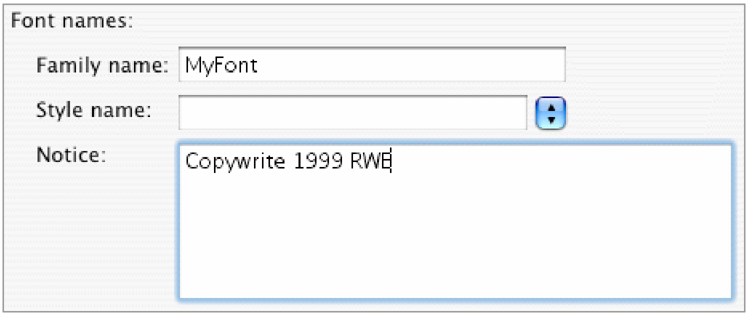
This setting controls how the font may be embedded into documents. Embedding is a feature of the Windows operating system and some applications that allow programs to include fonts into documents (PDF, for example) to guarantee that they will be reproduced correctly. However, this feature may cause problems 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	**NOE**	Embedding is not allowed for this font
Only printing and previewing is allowed	**ROE**	The font may be embedded, but editing of the document it contains is not allowed
Editing of the document is allowed	**RWE**	The font may be embedded and the document that contains the font may be viewed, printed and edited.

You can set your font embedding parameter in Fontographer, as follows:

1. Choose **Element** > **Font Info**.
2. Inspect the **Notice** text box at the top of the dialog box.
3. If copyright notice appears in the **Notice** text box, position the cursor one space after the notice and type **NOE**, **ROE** or **RWE** after the copyright, as indicated in the illustration below.



Font names:

Family name: MyFont

Style name:

Notice: Copywrite 1999 RWE

Note: If copyright information does not appear in the **Notices** text box, then type only **NOE**, **RWE** or **ROE**.

If you do not intend to restrict the font usage then just do not put anything after the copyright notice. This will mean "Everything is allowed" and the font will work as if it was installed in the system.

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 character information, and in the outline window either character 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 character information.

Character Information

Use the Character Information dialog box to change the fill and stroke characteristics of individual characters for Type 3 fonts. Normally a character will be either filled or stroked, but you can use this dialog box to create characters that are both filled and stroked. The **Tint**, **Weight**, **Cap**, and **Join** options allow you to set the characteristics of a stroked font.

The character **Name** textbox allows you to change the name of the character – 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.

- **Fill**

  If the **Fill** 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** 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 character'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 Information

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 character'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 characters (such as old versions of Courier) into contoured (outline) or filled characters. 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 characters in your font, selected characters, or just the characters that are changed. You may also choose to preserve the line spacing, or the character shapes. Recalculating bitmaps is useful when you change a character'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 characters 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 character. You can select the Easy or the Advanced mode. With the Easy mode, you just select the spacing between characters 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 characters 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 characters 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 characters as bases or prototypes for the rest of the characters 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 characters. Choose the characters you want to apply the metrics settings to, from the categories under Which character. These categories include: A–Z, a–z, 0–9, Punctuation, Accent characters, Symbols, or Selected characters. Add any other characters 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 character's measurement, or you can pick an em-unit value to type in to set the width, and/or sidebearings for the selected characters. You may also add an amount to that specification – either in em units, as a percentage of the distance represented by the character's metrical measurements, or the value you chose.

Set Width

This dialog box allows you to set the width of any selected character(s). If you are creating a monospaced font, select all the characters and set their width to a chosen number of units. Fontographer will set them all at once.

If the spacing of your characters is too tight or too loose, you can use the Change width by em units option to change the width of selected characters by a specified number of em square units.

If you need to increase or decrease the width of selected characters 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 characters 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. 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 Window(s) 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 character.

View Windows by

Use **View Windows by** to choose whether you want to display your font windows by **Font name**, **File name**, or **Both**.

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 **Windows** menu, there is a list of all open windows. Selecting a window in the list will bring that window to the front.

The Hints menu

Make Vertical Stem

To make a new vertical stem hint, select two of the points that define the stem and choose this item from the menu.

Make Horizontal Stem

To make a new horizontal stem hint, select two points that define the stem and choose this item from the menu.

Make Vertical Serif

To make a vertical serif (a serif that attaches to a vertical stem), select four of the serif's points and choose this item from the **Hints** menu.

Make Horizontal Serif

To make a horizontal serif (a serif that attaches to a horizontal stem), choose four of the serif's points and select this item from the **Hints** menu.

Make Diagonal Hint

Select four of a diagonal stem's defining points and choose **Make Diagonal Hin**t from the **Hints** menu. Diagonal hints are used only in Fontographer's internal bitmap generation and for TrueType. PostScript doesn't use diagonal hints.

Build Serif

When you choose **Build Serif** from the **Hints** menu with two hints selected, one from each side of the vertical stem at its upper end, a serif hint will replace those two hints. Or you can select four points and choose **Build Serif**.

Split Serif

Select a double-head serif hint and choose this item to split it into two separate hints.

Flip Hint Direction

This item will change the direction of the current hint. Generally, hints should point inward but there are some exceptions. The direction of the hint tells you which way the outline will be adjusted when the pixels are fit to the grid during outline rasterization.

Hint Parameters

You can change the hint parameters for selected characters using the Hint Parameters 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 11, "Expert Advice", for more details on hints.

Vertical 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 character 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.

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 character or to the next character. The **UP** and **DOWN ARROWS** select the character in the row above or below the present character.

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.

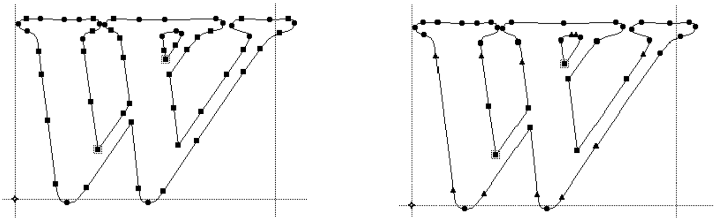
SHIFT-arrows move the selected points in the direction of the arrows by 10 times the default setting.

In the metrics window, use the **LEFT** and **RIGHT ARROW** keys to move to the next or previous character in the screen display when a character is selected.

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.

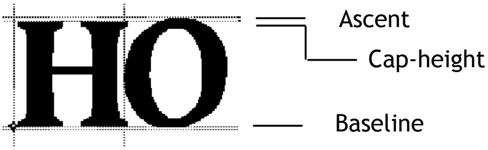


- Images that are imported or pasted into the Template layer are automatically sized to fit the em square.

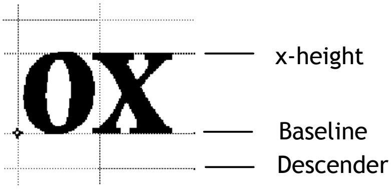
  To maintain the original size of the image when pasting into the layer, hold down the Option key when you paste or import the image. The size of the image you see depends on the magnification level in the window.

- 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."

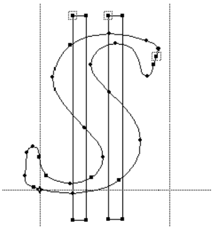


Lowercase letters' top coordinates should be between the top of the "x" and the "o." Lower coordinates should be between the baseline and the bottom of the "O."

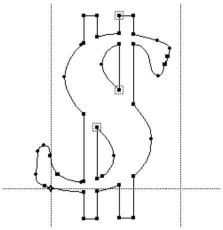
All characters' upper and lower coordinates will be automatically aligned if they fall into these zones.

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 centre 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 4.7 has a limit of 8192 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 4.7 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. Bibliography of Typography and Allied Subjects

by David S. Rose

Fontographer gives you the ability to create new digital typefaces from your imagination, to recreate classic faces that are otherwise unavailable in digital form, and to adapt existing faces for your particular needs. As with any powerful tool, the more you know of the history behind it, the better able you will be to utilize it. The books listed here are just a few of the many hundreds that have been written on the subject of typography over three centuries, but they will provide an excellent start for background 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. A number of such dealers are listed at the end of this bibliography.

Overviews of printing types

Printing Types: An Introduction by Alexander Lawson [Boston: Beacon Press, 1971] is a short (120 pages) easy-to-read overview that is exactly as advertised: an introduction. For over thirty years, Lawson has taught a course in 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.

Letters of Credit: A View of Type Design by Walter Tracy [Boston: David Godine, 1986. 224 p, ill.] A beautiful and profusely illustrated step-by-step demonstration of type-design aesthetics that traces the beginnings and the path of modern-day typesetting.

Printing Types: Their History, Forms, and Use by Daniel Berkeley Updike (1860–1941) [New York: Dover, 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."

Type and Typefaces by J. Ben Lieberman [New Rochelle: The Myriade Press, 1978] is an alternative to the Lawson book, but much less accurate, bigger (142 pages, 8 1/2 x 11, hardcover) and much harder to find. Lieberman was an enthusiastic amateur printer, and this book is an exuberant look at the history, classification, identification, and personalities of typography. It includes examples of over 1000 type faces, and is well illustrated. Lieberman was not a scholar, but if you like unabashed "boosterism," you might find this book fun to read, despite its errors of both omission and commission.

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: Bedford Arts, 1988] is an excellent selection of articles from *Fine Print* magazine, the one indispensable periodical that anyone concerned with type should subscribe to. Each issue is designed by a different typographer, and is printed by letterpress. In addition to scholarly articles, typographic overviews, reviews, and notices of new books on typography, a supplement included with each issue contains a complete calendar of everything of typographic interest that is happening nationally and internationally. 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. Information might still be available from *Fine Print* in San Francisco.

Typographical periodicals between the wars; a critique of the Fleuron, Signature, and Typography by Grant Shipcott [Oxford: Oxford Polytechnic Press, 1980. xiv, 111 p. :ill.]. These classic periodicals (particularly the Fleuron) were to their time what *Fine Print* was to typography and book design today, but because of the ferment in the world of design during the 20s and 30s and because of their illustrious contributors, they had a much greater effect on the typography of the time.

History and development of lettering and letter forms

The 26 Letters by Oscar Ogg [New York: Thomas Y. Crowell, 1948]. A nicely done book by a well known American calligrapher, tracing the evolution of the alphabet from prehistoric times to the invention of printing. 250 pages, well illustrated.

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 covers some of the same material as the Ogg book, but includes much more information on the development of letter forms 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 falls somewhere between the Ogg and Nesbitt books, from Goudy's unique perspective as the most prolific type designer of the twentieth century.

Roman Lettering by L.C. Evetts [New York: Taplinger, 1979] includes a character-by-character analysis of the letters on Trajans 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: Pentalic, nd ca. 1976]. 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 Johnson [New York: Taplinger, 1980]. The comprehensive calligraphy manual by the man who led the twentieth century revival of calligraphy. Johnson's influence on English, American and German lettering and design was immense.

History of Lettering by Nicolete Gray [Boston: David Godine, 1987. 256 p].

Type designs from various periods

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, (1897–) [New York: Pierpont Morgan Library; 1984. Boston, MA: D.R. Godine, xiv, 192 p. : ill.]. Available both in hardcover and paperback, this collection by one of the great printer/scholars of the century is a must have for anyone interested in original source material. More than a hundred full pages 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 Morrison 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) Oxford, Clarendon P., 1969. xii, 137 p. 45 plates. illus., facsims., col. map].

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, 1852–1893 and A. F. Johnson [Folkestone: Dawsons, Reissued 1974 xiv, 400 p., fold. leaf : ill., facsims].

Notes on a Century of Typography at the University Press, Oxford, 1693–1794 by Horace Hart [Oxford, Clarendon Press, Reissued 1970 (1st ed. of 1900 reprinted) with an introduction and additional notes by Harry Carter. ix, 16, xvi, 203 p., plate. illus. facsims]. 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 a classic from 1938, this large format 240 page work is the definitive book on its subject.

American Wood Type, 1828–1900 by Rob Roy Kelly [New York: Da Capo Press]. 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.

The Typographic Book 1450–1935 by Stanley Morrison and Kenneth Day [Chicago: University of Chicago Press, 1963]. A lush, expensive, visual treasury of almost 500 years of typography, including 357 plates.

American typography today by Rob Carter [(New York: Van Nostrand Reinhold, 1989. 159 p. : ill. (some col.)].

The Liberated Page Edited by Herbert Spencer [San Francisco: Bedford Arts, 1987]. An anthology of the major typographic experiments of the 20th century, as recorded in *Typographica* magazine.

Typography

The Crystal Goblet; sixteen essays on typography by Beatrice Warde [Cleveland and New York: World Publishing Company, 1956]. From a major expert in the field of typography come some thought-provoking pieces, including the famous analogy that gave the collection its name. Out of print, but available, and mandatory reading.

Better Type by Betty Binns [New York: Watson-Guptill, 1989. 192 p]. An excellent contemporary book that shows by copious examples the subtle differences in relationships between typefaces, letters, and spaces. From the preface: "This book systematically trains designers to make these fine discriminations, with the aim of specifying text type that is not only readable, but also beautiful and expressive." Highly recommended.

Introduction to Typography by Oliver Simon [London: Faber & Faber, 1945]. Not a bad place to start. This edition is out of print, but there has been at least one reprint in recent years. Simon's introduction is designed for the layman, and discusses many of the basic principles and theories of designing with type.

First Principles of Typography by Stanley Morrison [Cambridge: at the University Press, 1951]. An important book from the man who designed Times Roman for the *London Times*.

Asymmetric typography by Jan Tschichold [(Translated by Ruari McLean) New York, Reinhold Pub. Corp. 1967. 94 p. illus. (part col.) facsims]. Jan Tschichold (1902–1974), a well-known typographer, caused many people to rethink "conventional" theories of typography when this seminal work was published in the mid-60s. Whether or not you agree with his approach, this book is required reading and will widen your typographic horizons.

An essay on typography by Eric Gill [1st U.S. ed. Boston: D.R. Godine, 1988]. A classic typographic manifesto on the art and craft of letterforms from the designer of Gill Sans and the famous typography of London Underground.

Typography, A Manual of Design by Emil Ruder [Niederteufen, Switzerland: Arthur Niggli Ltd, 1977. 3rd Edition]. A fascinating, disciplined, and very Swiss analysis of typography and letterforms. Ruder's discussion and illustration of the importance of white space in letter forms and graphic designs is excellent background reading.

Report on the typography of the Cambridge University Press by Bruce Rogers [Cambridge, Cambridgeshire: University Printer, 1950. viii, 32, (3) p. : ill.]. Bruce Rogers (1870–1957) 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 and practice in both Britain and the U.S.

*Designing with type; a basic course in typograph*y by James Craig and Susan E. Meyer [Rev. ed. New York: Watson-Guptill Publications, 1980. 176 p]. A modern "how-to" book, available at many large bookstores and graphic arts dealers.

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, Morrison, Zapf, and so forth.

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 (1865–1947). [Reissued 1977. Berkeley: University of California Press, xviii, 170 p.: ill.; 24 cm.]. Written by the most prolific type designer of the 20th century [creator of, among others, the eponymous Goudy Oldstyle], this reprint of the 1940 edition 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 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 Herman Zapf [Frankfurt, New York: Z-Presse, 1968]. Herman Zapf is known to most desktop typographers primarily for giving his name to the Zapf Dingbat font. He is, in addition, one of the most respected and creative typographers and type designers of the century, who created not only the Dingbat and Zapf Chancery fonts, but also Optima, Lydian, 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. Superb as a source of inspiration and example.

Herman Zapf and His Design Philosophy by Herman 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 Johnson by Priscilla Johnson [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 translation of 1927)]. Similar to the Dürer book, but this is the one with the letters shown against naked human bodies in addition to the grid system.

Pioneers of modern typography by Herbert Spencer [Cambridge, Mass.: MIT Press, 1983. Rev. ed. 160 p. : ill.].

Typeface reference works

Graphics master; a workbook of planning aids, reference guides, and graphic tools for the design, estimating, preparation, and production of printing, print advertising, and desktop publishing by Dean Phillip Lem [Los Angeles, Calif.: D. Lem Associates, 1988. 4th ed. 153 p. : ill. (some col.)]. Although it covers much more than just type design, and is fairly expensive, this is probably the most important and continually useful reference work that a desktop designer and/or publisher should have.

A.T.A. Type Comparison Book by Frank Merriman [Advertising Typographers Association of America, 1965]. An indispensable handbook for identifying typefaces. Hundreds of faces are grouped together by design, making it easy to find the one you want. Still in print, possibly in a more recent edition.

The Encyclopedia of Type Faces, 4th Edition by W. Pincus Jaspert, W. Turner Berry, and A. F. Johnson [Poole, Dorset: Blandford Press, 1983]. The standard reference in the field. A detailed listing of over 1000 faces, arranged by name, with full information on their history, designers, and so forth.

Graphic Arts Encyclopedia, 3rd Edition by George A. Stevenson, Revised by William O. Pakan [New York: McGraw-Hill, 1990, 624 pp, 167 illus.] is a complete, readily available, and pricey ($58) reference to everything from basic block printing to desktop publishing. While not as Macintosh-specific as some of the other works available, it covers the hows, whens and whys of a whole range of graphic tools, including typography, art preparation, copyfitting and layout, film and platemaking, traditional photocomposition and typesetting, papers and papermaking, and so forth. A good general purpose book, but overkill for the beginning desktop publisher.

Type Terminology on the Desktop [Richardson, Texas: Altsys Corporation, 1992] is already in your possession, in hypertext form (included with Fontographer on its CD.] A hard copy of this award-winning booklet defining typographic terms is also in the Fontographer package.

Font & Function [Mountain View, California: Adobe Systems] is Adobe's biennial catalog of their latest font offerings. But this tabloid size, four-color publication is also quite a bit more. It includes articles on typographic history, the background to many Adobe PostScript fonts, technical information and a graphic listing of over 1500 Adobe fonts. There's no reason not to get this, especially since it is free for the asking from: Adobe Systems Incorporated, Attention: Font & Function, P.O. Box 7900, Mountain View, CA 94039–7900.

The typEncyclopedia; a user's guide to better typography by Frank J Romano [New York: R.R. Bowker Co., 1984. xii, 188 p. : ill.].

Type and typography; the designer's type book by Ben Rosen [New York: Van Nostrand Reinhold Co., 1976 Rev. ed. 406 p. : ill.].

History of printing

A Short History of the Printed Word by Warren Chappell [Boston: Nonpareil Books (David Godine), 1980]. A once-over-very-lightly in 240 pages of large type, hitting the highlights in the development of type, printing and bookmaking.

Five Hundred Years of Printing by S. H. Steinberg [Baltimore: 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.

A Dictionary of Book History by John Feather [New York: Oxford University Press, 1986, 278 pp] is a concise one-stop reference, in alphabetical order, to topics including bibliography and bibliographical terminology, the history of printing, the physical history of the book (including typography, binding, and so forth) and book collecting. It has over 650 articles ranging from a few lines to several pages, and covers the ground pretty thoroughly. Although not a classic work (and, indeed, poorly designed itself as a book), it serves as a very handy reference to the history of books. An expensive purchase at the original price of $45, it is often available on remainder for about $10.

The Making of Books by Seán Jennett [New York and Washington: Frederick A. Preager, 1967]. A good overview of the entire art and craft of the book, including a little history and a fairly detailed examination of every stage of the process. If you are interested in books in general, this is a good place to start. Out of print, but rather ubiquitous at second-hand and antiquarian dealers.

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.

Electronic typography

The Macintosh Font Book 2nd Edition by Erfert Fenton [Berkeley: Peachpit Press, 1993. 350 pp]. Everything you want to know about Macintosh fonts and type faces, including type terminology, building a typeface library, font management, printing options, and a near-complete list of available PostScript typefaces (with vendor addresses) and typesetting service bureaus. The second edition of this award-winning book covers TrueType fonts, System 7, and much more. The best all-around introduction to the world of Macintosh-specific font handling.

Desktop Publisher's Survival Kit by David Blatner [Berkeley, California: Peachpit Press, 1991, 172 pp, 800K disk] An excellent overview of the various facets of desktop publishing on the Macintosh. Blatner writes informally and well, and provides a good general grounding in file formats, fonts, typography, scanning and printing. The book includes a helpful section on troubleshooting when things go wrong, as well as a Macintosh disk full of DTP utilities, clip art and two PostScript fonts. While not an in-depth study of any particular area, this book is a very good first place to start for beginning desktop publishers.

Collier's Rules for Desktop Design and Typography by David Collier [New York: Addison-Wesley, 1992, 135 pp, illus.] David Collier of DeCode Design is a Young Turk of the British graphic design field, who has produced, an interesting, useful, graphically dazzling and extremely self-absorbed reference for Macintosh typography. This affordable paperback is chock-full of the latest on Macintosh-based typography, page-layout, design conventions and the like. While it has a distinctly British flavor, and goes overboard in its layout and graphic examples, Collier's Rules is not a bad way for a budding typographer to absorb some of the basic principles and arcana of computer-based typography.

The Mac is not a typewriter by Robin Williams [Berkeley, California: Peachpit Press, 1993, 72 pp.] A slim, elegant guide to typesetting on the Macintosh. Quick and easy chapters cover the "top twenty" tips for effective Macintosh-based typographic design. Friendly in tone and easily accessible, most of the included information will be known by experiences designers but provide a very good primer for the novice.

Fontographer: Type by Design by Stephen Moye [MIS: Press a subsidiary of Henry Holt and Company, Inc, 1995. 275 pp]. The first and only third-party manual on using Macromedia Fontographer.

Learning PostScript: A Visual Approach by Ross Smith [Berkeley, California: Peachpit Press, 1993, 350 pp.] For those who are interested in manipulating PostScript directly, this hefty but accessible guide provides the best introduction to the industry-standard page description language. Clear and comprehensive, it alternates PostScript concepts and demonstration programs with full-page examples of printouts. Topics include drawing lines, arcs and curves, repeat loops and special effects.

Macintosh desktop typography by John Baxter [Sunnyvale, CA, U.S.A.: Baxter Group, 1986 ill.]. One of the first books of this genre. A well done introduction to typography on the Macintosh, including useful suggestions about dealing with Postscript typefaces. Not as up to date as some of the books from Peachpit Press, though.

Digital typography; an introduction to type and composition for computer system design by Richard Rubinstein [Reading, MA: Addison-Wesley Pub. Co., 1988. xi, 340 p.: ill.].

Bibliographies

A Typological Tally compiled by Tony Appleton [Brighton, (T. Appleton, 28 Florence Rd., Brighton, Sussex BN1 6DJ), 1973. 94 p. ill.]. Thirteen hundred writings in English on printing history, typography, bookbinding, and paper-making, compiled by one of the world’s top 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.

Book dealers/publishers specializing in typography and the book arts

Nijhof & Lee international booksellers
Staalstraat 13a
1011 JK Amsterdam
The Netherlands
http://www.nijhoflee.nl/

Oak Knoll Books
414 Delaware Street
New Castle, DE 19720
USA
http://www.oakknoll.com/

Mark Batty Publisher, LLC
6050 Boulevard East, Suite 2H
West New York, NJ 07093
USA
http://www.markbattypublisher.com/

Hyphen Press
115 Bartholomew Road
London NW5 2BJ
England
http://www.hyphenpress.co.uk/

Verlag Hermann Schmidt Mainz GmbH und Co. KG
Robert-Koch-Str. 8
55129 Mainz
Germany
http://www.typografie.de/

Many thanks to Howard Gralla, Alvin Eisenman, Robert Fleck, Kathy Schinhofen, Chuck Rowe, Earl Allen, Susan Lesch, Kathleen Tinkel and Michael J. Boyle for their suggestions before and during the compilation of this bibliography.

Type Designers

The type designers listed below have contributed their fonts, their typographic expertise, and their time to the development of this manual and we thank them. To contact them write to the following addresses:

Richard Beatty
Hendersonville, NC, USA

David Berlow
The Font Bureau
Boston, MA, USA
http://www.fontbureau.com/

Peter Fraterdeus Alphabets, Inc.
Evanston, IL, USA
http://www.alphabets.com/

Jonathan Hoefler
Hoefler & Frere Jones
New York, NY, USA
http://www.typography.com/

Dennis Ortiz-Lopez
New York, NY, USA
http://www.ortiz-lopez.com/

Judith Sutcliffe
The Electric Typographer
Audobon, IA, USA

Paul Sych
Faith
Toronto, Canada
http://www.faith.ca/

Joe Treacy Treacyfaces
West Haven, CT, USA
http://www.treacyfaces.com/

Appendix C. General Information

Type terminology

Ascent 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.

Baseline

The baseline is the imaginary line upon which capital 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.

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.

Em square

Each character is described in terms of a rectangle called an em square, or the UPM 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.

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

When speaking about a group of related fonts, we use the word family to describe a related group of faces. For example: Goudy Oldstyle-Plain, Goudy Oldstyle-Bold, Goudy Oldstyle-Italic, and Goudy Oldstyle-BoldItalic.

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.

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 Fourniér, a Parisian typefounder. Fourniér'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.

Sans serif

This manual's section headings are sans serif (without serifs). 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.

Serif

Serif letters can be distinguished by the line that crosses the main strokes of a character. Serifs are the short cross-strokes at the ends of letters, drawn there to improve the readability of the type.

Style

Styles of a face include modifications such as bold, italic, condensed, and extended. Each style should be considered a separate face. The combination of styles makes up a font. A typestyle provides a way to add emphasis to written text. 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.

Text type

Text type is used for larger masses of text and should be highly readable.

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, faces with taller x-heights are perceived as larger and more readable than those with small x-heights.

Fontographer background

Em square

Each character is described in terms of a rectangle called an em square, or the UPM 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.

Unique PostScript ID

This ID is used for caching character images in your printer, thereby decreasing printing time. Font IDs for new fonts are automatically generated in the range 4,000,000 to 4,999,998. If you are a commercial font developer, you should contact Adobe Systems, Inc. for an allocated font ID. If you do not want your font to have a unique PostScript ID, enter 0 in this field, and Fontographer will not assign a unique ID to your font.

Bitmap background

Bitmap fonts (Macintosh)

The Macintosh uses bitmap fonts for screen display. Bitmap fonts are selected from any application which understands fonts. Bitmap fonts are stored in the System file, where they are accessible to all programs. Bitmap fonts cannot be smoothly scaled, so each outline font usually will have several bitmap fonts in different sizes for display purposes.

✐ ***Note***: Mac PostScript fonts require at least one bitmap size, but TrueType fonts or Windows PostScript fonts don't require any.

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 bitmap font 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 System 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 System Folder or in the same folder as the printing application 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.

Windows FON (PC)

Windows FON is a Windows-compatible bitmap font that Fontographer 4.7 allows you to generate. Since these are standard bitmap fonts, they are limited to sizes between 1 and 255 points. The width of a bitmap character is limited to 254 points. At the larger point sizes, this may require you to limit the width of an outline character. Fontographer avoids trouble by automatically limiting wide bitmap characters to 254 points, even if the character is calculated to be wider.

Generating bitmaps doesn't necessarily mean that Fontographer will generate a .FON file. Fontographer will keep those bitmaps in memory and in its database file until you decide to remove them.

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 which 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 which 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 which 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 lowercase "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.

Metrics

PostScript stores auxiliary information in the character metrics table. This contains information such as the width of the character, its origin, and its bounding box. This table is generated automatically by Fontographer as you edit the character. The only direct user interaction with the character metrics is specification of the character width. The width is the distance by which the drawing pen is advanced before starting the next character. The origin point of the current character is aligned with the width point of the previous character before drawing the current character.

TrueType font

This is an outline font format developed by Apple Computer (for use with System 7) and supported by Microsoft Corporation. TrueType is the primary font format for Windows 3.1 and Windows 95.

Type 1 font

Type 1 fonts are the standard type of PostScript fonts. These may be sent to PostScript printers or viewed onscreen with Adobe System’s program, Adobe Type Manager (ATM).

Glossary

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.

Application: A computer program, designed to perform a specific function such as word processing or illustrating.

Ascent / Ascender: A font's maximum distance above the baseline.

ASCII: (American Standard Code for Information Interchange). A numbering scheme used for identifying printing characters.

Adobe Type Manager (ATM): The program that improves your screen font display by eliminating jagged edges on Type 1 fonts.

Baseline: The line upon which the letters of a font rest.

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.

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.

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.

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.

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: A font's maximum distance below the baseline.

Descender line: The lowest line that a character's descender extends to, like the line beneath the lowercase "j" and "y."

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).

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 contain 72, 75, or 90 dpi.

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 indentions. Traditionally, the em space was created by non-printing blocks of metal used to add space between words.

Em square: A square the size of a capital letter "M." Also the height of the ascent added to the height of the descent.

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.

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.

Face: Short for typeface; the style of a font or set of character images.

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.)

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: In modern usage the term font is often confused with typeface and family. Traditionally, the term font represents a complete set of characters or symbols, which share the same size and style. For example, 12 point Goudy Oldstyle Bold is a font. Fonts can be as small as the basic alphabet or up to hundreds of characters. Some languages, like Japanese, can exceed these numbers, which make them more difficult to access from the standard keyboard. Originally derived from the word found as in typefoundry.

Font attributes: Characteristics which apply to the font as a whole (such as the ascent, descent, leading, etc.).

Font Window: The graphic display of a character set in Fontographer from which individual character slots may be accessed.

Fontographer: Macromedia's outline font editor designed to simplify the creation of high-quality fonts, logos, and other intricate PostScript artwork.

Generate: To create a character or set of characters so that they can be viewed on the screen or printed. Usually a bitmap font and a PostScript font is generated for each font, but with outline font formats, you only need to generate one font file.

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.

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.

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.

Linotype 100/300: High-resolution typesetting systems.

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).

NFNT (New FoNT): The Macintosh font resource which contains your bitmap screen font. NFNTs have a wider range of font identification numbers than the older FONTs, NFNTs opened up another 16,000 FOND ID numbers when they were introduced by Apple with the Macintosh Plus.

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.

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.

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.

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: The height of the type body. A standard type measurement system was originally developed by the Parisian type founder Piérre Fournier Le Jeune in 1737. In the days of metal type, the point size was the total number of points in the height of metal type, including the ascent and descent of the letters, and the metal above and below the letters (for example, built-in leading).

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.

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 dpi.)

Right justified: Type aligned with its right margin. Also known as flush right.

Sans serif: A typeface with no counterstrokes at the end of each stroke.

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 typeface with counterstrokes at the end of each stroke.

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.

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.

Stroke: A line which may be expanded in width; or the width of the linear elements that compose characters.

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.

Style Merger: Macromedia's utility that merges Macintosh styled fonts into one font family.

Swash capitals: Uppercase letters that have flourishes added to them. Originally designed to go with Italic typefaces.

Template layer: The layer of the Outline Window where background images and scans are inserted.

Textbox: Within a dialog box any rectangular outline that includes text.

TTF: A file in an outline font format (TrueType Font) that is used both in screen display and in printing.

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. (See letterspacing.)

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.

TrueType: Apple's outline font format designed to be used with their System 7 operating software and Microsoft Windows 3.x.

Type 1 PostScript font: Genuine Adobe PostScript or Fontographer Type 1 fonts with hints, which provide high quality resolution at any size.

Type 3 PostScript font: Also referred to as user-defined fonts, these are non-Adobe encrypted fonts. They will not render on-screen if you are using ATM.

Typeface: A set of characters which share a similar appearance.

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 height of a face's lowercase letters, or the size from mean line to baseline of the lowercase "x." The lowercase "x" is used for measurement since it usually sits squarely on the baseline and has no ascenders or descenders.

Index

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.