JAMES FELICI

FOREWORD BY FRANK ROMANO

The Complete Manual *of* Typography

SECOND EDITION

A GUIDE TO SETTING
PERFECT TYPE

"The ultimate typographic tool: a concise, beautiful book that pulls together everything you need to produce great typography."

FRANK ROMANO
ROCHESTER INSTITUTE OF TECHNOLOGY, SCHOOL OF PRINT MEDIA

"Dangerously good book on typography. 'Dangerous' because there is enough well-presented information in this volume to set you on the path to typography snobbery. This book is an excellent read and reference volume for any designer, print or web."

—NORA BROWN, Nora Brown Design

"Felici elegantly and painstakingly sets out to demonstrate how to set type 'perfectly' in a digital age. This is the book that answers all the questions you wanted to ask, but also demonstrates all the steps you need to pursue to achieve a kind of typographic perfection."

—MARGARET RICHARDSON, FontShop

"Buy this book, read it cover-to-cover, then keep it handy. You'll be surprised at what a difference it can make in the appearance of your work, both print and web."

—PETER BAUER, *Photoshop User*

"*The Complete Manual of Typography*, by James Felici, condenses timeless wisdom and timely technology into one complete guide. It explains everything about type designs and usage. If you had only one book on typography, this should be it."

—JAY NELSON, *Design Tools Monthly*

"Reading this book is like sitting down with a longtime typesetter and going over the details of a complex job. Most people will use it as a reference—which it is—but reading any section straight through is rewarding. The writing is clear and straightforward, and Felici has obviously thought long and hard about everything he deals with here."

—JOHN D. BERRY, CreativePro.com

"This excellent book discusses how type should look and how to set type like a professional."

—LINDA BUSHYAGER, *HiTech Review*

while all earlier Mac OS versions used the original Mac imaging system called QuickDraw. For text, Windows uses a system called DirectWrite, which builds on the earlier GDI (Graphical Device Interface) and GDI+. In either case, the image of a page is drawn first to the screen and then possibly translated into another language when it's time to print. Because most fine type is set on high-resolution imagesetters, and most imagesetters are driven by PostScript RIPs, the path from screen to final page is most direct under the PostScript-driven Mac OS X.

How Fonts Are Used for Screen Display

The latest versions of popular operating systems are still capable of using bitmapped screen fonts for displaying type onscreen, particularly in small sizes. But by and large, the type that you see displayed onscreen has been scaled from the same font outlines used for printer output but rasterized for the resolution of the screen.

Most applications rely on the operating system to render screen type, but this is not always the case. Applications that emphasize typography—notably drawing and page layout programs—often rasterize their own screen type. Because each application vendor has its own technology for this, screen type as rendered by various programs will look different, especially in small sizes, where the placements of individual pixels have relatively more impact on the image of a character. This is shown in Figure 7.2.

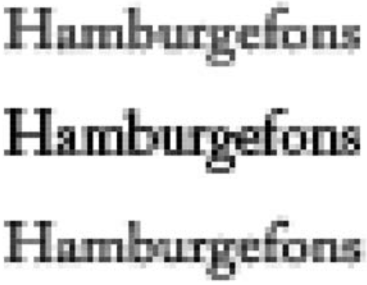

FIGURE 7.2 Screen type is rendered differently from one program to the next. These enlarged samples show the same 10′ Caslon type as displayed in QuarkXPress (top), Adobe InDesign (middle), and by Mac OS 10.6 (bottom).

Also visible in Figure 7.2 is the effect of anti-aliasing technology, which renders screen type using both black and nonblack (either gray or color) pixels. (An alias in this context is an artifact of poor image rendering, such as the jagged contours of black-and-white type rendered in small sizes.) This technology is commonly called *font smoothing,* a misnomer because the fonts themselves are not affected. Adding nonblack pixels to a character bitmap has the effect of making the character shapes smoother and typefaces more recognizable. It has the unfortunate side effect, because it makes the type slightly blurry, of contributing to eyestrain and fatigue as the reader's eyes struggle in vain to draw the type into sharper focus. Ultimately, only higher-resolution screens will solve this problem. The challenge of representing type accurately and effectively onscreen is examined in more detail in Chapter 18.

TYPE AND THE "STYLE" MENU

Word processing programs introduced the concept of switching from regular weight to bold or from roman to italic by issuing a generic keyboard command. These commands have become part of the standard user interface for virtually all application programs that use fonts. These usually act as toggles: Issuing a command to italicize roman text applies the italic member of that font family. Issuing the command again toggles the face back to roman. The same technique allows you to switch from regular weight to bold and back again.

This system works best when you're using a four-typeface font family: regular, italic, bold, and bold italic. When you issue a program command to convert regular weight text to bold, for example, the program looks to the regular-weight font for information about which font it should apply. These relationships are defined within the fonts. Likewise, once that text is set in bold, if you issue a command to italicize it, the program will consult the bold font to see what its italic complement is so it can properly set the type in bold italic.

In large typeface families, the effects of these commands is not standardized and quickly becomes confusing. This is illustrated in Figure 7.3. Sometimes the "bold" command simply causes the type to be set in the next heavier weight in that font family, for example changing regular to semibold. In other font families, the same command is taken more literally and will cause regular-weight type to be set in bold, even though an intermediate semibold weight exists. In yet other families, using a "bold" command on type set in a heavy weight (one step heavier than bold) will apply the extra bold font. Likewise making extra light or thin type "bold" may move it up one weight to light.

Confusing things further is that the "bold" command does not always act as a predictable toggle. In the Neue Helvetica samples in Figure 7.3, for example, the "bold" command may turn a medium weight to bold, but using the command again does not return you to medium: It changes the type to the slightly lighter regular weight, a surprise that could easily be overlooked.

When working with font families with more than four members, the only sure way to get the typeface you expect is to choose it explicitly from the Font menu rather than using a style command.

Linotype Neue Helvetica

Ultra Light + "bold" = **Bold**
Light + "bold" = **Bold**
Regular + "bold" = **Bold**
Medium + "bold" = Bold
Bold + "bold" = Regular

Adobe Garamond Pro

Regular + "bold" = **Semibold**
Semibold + "bold" = Regular
Bold + "bold" = no change

Adobe Warnock Pro

Light + "bold" = no change
Regular + "bold" = **Bold**
Semibold + "bold" = no change
Bold + "bold" = Regular

Bitstream Incised 901

Light + "bold" = no change
Roman + "bold" = Bold
Bold + "bold" = Roman
Black + "bold" = Roman

FIGURE 7.3 When using extended font families, the result of using a keyboard or style command to make type bolder is unpredictable. Each of the four families shown here handles the situation differently. Likewise, applying the command again may not simply toggle the type back to its original lighter weight.

SCREEN RENDERING WHEN FONTS ARE MISSING

To get an accurate screen preview of how type will look, it's important to have all the necessary fonts installed on your system (even though it may be possible to compose type without them). Operating systems are very obliging about giving you what you ask for in the way of type, and the results can be deceptive. As noted above, if you apply the bold command to make a typeface heavier, the OS will probably display a bold version of the type whether or not the bold font is available, or whether or not it exists at all. The latter possibility is illustrated in Figure 7.4, in which a titling face has been made heavier with the bold command. It looks bold onscreen, but because there is no such

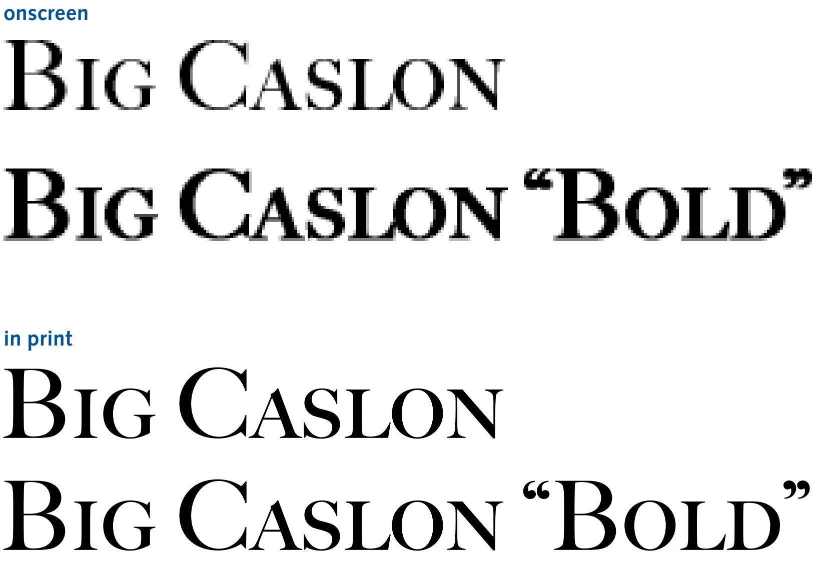

FIGURE 7.4 The couplet on top shows the effect of the bold style command. There is no such thing as Big Caslon Bold, though, so when printed, the lines at top will translate into the lines below, with the printer using the only font available.

thing as a bold version of this font, the regular-weight font will appear when it's printed, and the client will be in for an unpleasant surprise.

Operating systems will perform the same service when you use the italic command on roman type. If the italic version of the font isn't available, the os will create an electronic oblique by simply slanting the characters. This bit of fakery is easier to detect than a synthesized bold because the letters—especially in the case of seriffed faces—are usually clearly not true italics, as seen in Figure 7.5.

It is also possible with certain fonts that the operating system will image the various members of a typeface family correctly onscreen but print them incorrectly, using synthesized bolds and italics. This can happen when the outline fonts are physically present on the computer but not activated or installed by a font-management program. When installing fonts, then, it is a good policy always to install complete families, not just selected members.

How Operating Systems Manage Fonts

In order for an operating system to serve font information to applications and create images of their typefaces onscreen, the fonts have to be "installed." In other words, they have to be stored in the computer in a place where the operating system knows to look for them. In Windows, this storage place is the Windows/Fonts folder. This is not a typical Windows folder in that it contains special tools in the File menu specifically for installing and de-installing fonts. You can also add fonts to this folder by dragging them from their source or by using the File menu's Copy To Folder command.

The Macintosh OS stores fonts in Fonts folders inside of the multiple Library folders found on a typical Mac. One such Library is inside the System folder and contains fonts the OS needs to operate and display its interface correctly. Another Library is in the root hard disk folder, and fonts stored here are available to all users and all applications. In addition, each user of the Mac can create a unique account with personalized settings, and this creates a Library/Fonts folder whose contents are available only to that user. Font files can be dragged or copied into any of these folders.

With both Windows and the Mac OS, applications may also maintain their own font folders to store fonts that are intended for use only with that application.

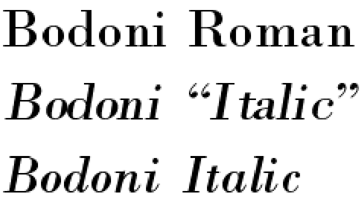

FIGURE 7.5 For fonts with cursive italic complements, it's easy to tell true italics from the italics generated from a roman screen font. Here, the character shapes in the false italics are simply inclined versions of the roman ones they're derived from. System-generated italics of sans serif faces may not be so obvious.

With a font-management program (see "The Basics of Font Management" in Chapter 4), fonts can be stored in folders other than those specified by the operating system. These programs work collaboratively with the OS to make sure that any fonts you specify are made available to your applications: that is, to make sure they appear in your programs' Font menus. The Macintosh OS includes its own: Font Book. Font-management programs for Windows are available only from third-party vendors.

Unless you're using a font-management program, merely having fonts stored on your computer's hard disk is not enough to make them accessible. Only when they're properly installed will they be made available to your applications. On the Macintosh, this may not happen immediately. Page layout programs typically feature *dynamic font updating,* which means that as soon as a new font is installed, it becomes available in the application's Font menu. Some other programs assemble the contents of the Font menu only when they are started up, so to update the Font menu after a new font has been installed, the application needs to be shut down and restarted. Windows applications poll the operating system every time you drop down a Font menu or open a Font dialog box, so you can add fonts to your system at any time, and they'll be available immediately.

Problem: Corrupted Fonts

Because they're read so often by the operating system, font files sometimes become damaged—*corrupted* is the official term—and a damaged font file can create an amazing array of problems. Not surprisingly, it can create printing errors, typically preventing a file from printing at all. Because fonts are intimately involved in the operating system, a corrupted one can bring the system to its knees, causing it to crash for no apparent reason. It can also cause individual programs to suddenly freeze.

Corrupted fonts are an unusual problem, but they do pop up from time to time, and they're hard to diagnose. It's always a good idea to keep a log of everything that changes on your computer—new programs, alterations to os settings, application preferences changes, and so forth. When things start to go wrong repeatedly, looking at the log may help to pinpoint a change that may have triggered the mayhem. When there's been no change to account for a series of problems, a corrupted font should be among your list of suspects. Not high on the list, but on the list nonetheless.

The easiest way to check for corrupted fonts is with a font-management program, as most are capable of scanning all the fonts in your system and looking for miscreants. If you don't have a font-management program and aren't inclined to pay for one, you can remove half the fonts from your system. If the problem goes away, you've located the problem font in the de-installed group. If the problem continues to occur, swap the "in" half for the "out" half. If the problem goes away, you know where the problem font can be found. If the problem persists, the source is not a font. If you determine that one of your fonts is to blame, take that half of your font collection and divide it in half again and repeat the same test. Do this until you identify the problem font. Keep in mind that fonts are being read by your operating system and application programs even though you may not be using them in a document.

Problem: Missing Fonts

As noted earlier, when you start up your computer, the operating system makes an inventory of all the installed fonts. At this point only a few fonts are required to be installed. These are the ones the operating system uses to create its interface, the typefaces in menu titles, dialog-box text, and so forth. The Macintosh os will not let you de-install these fonts, and it will give you an error warning if you try. Windows will let you remove any fonts, but if you remove the ones required for creating the interface—Marlett, for example—you will make a mess.

The problem of missing fonts arises most commonly when application-specific files—text documents, graphics, spreadsheets, etc.—are opened, especially when those files come from another computer where different fonts may have been installed. When you try to open a document that uses fonts that are not installed on your computer, your application program will probably warn you about the problem. Some will offer you the option of substituting an installed font for the missing one. With this latter option, the document won't look the same as the original—line endings and page breaks will change—but it will at least be readable.

When a document is printed, the same fonts used to create it have to be available to the printer. When you use a desktop printer, the printer *driver* (the software that translates the document into instructions that the printer needs to image the pages) will generally warn you if a font required for printing is not available. Depending on the settings of the RIP, however, an imagesetter may not issue a warning or cancel a job if a font is missing, instead substituting another font for the missing one. To make sure the substitution is obvious, the replacement font is usually Courier.

Problem: Duplicate Fonts

Many popular fonts exist in more than one font format, and there can be problems if multiple versions of a font are installed on the same computer. It's possible, for example, for you to have both TrueType and PostScript versions of faces—especially older ones, including Times, Helvetica, Palatino, Symbol, and Courier—installed in the same system and not know it, because the font name appears only once in an application's Font menu. When you select such a font, you don't know which font you're using, and when you send that file to an imagesetter or to another computer, it could make a big difference. Font-management programs may alert you to such duplications, but vigilance is the best solution. Be aware that many applications and operating system upgrades install fonts into your computer without warning you first.

It is also possible to have installed on your computer identically named fonts from different vendors. In such cases, the typefaces may be similar, but the font files are different. If both fonts indicate to the OS that they should be listed in the Font menu with the same name, confusion reigns. You may not be able to tell one from the other, and the OS may follow suit, causing it to list only one of the fonts in the menu and leaving it unclear which one you're getting when you select that font. Although font-management programs may be able to weed out duplicate fonts, they may not be able to cull identically named fonts.

Font Embedding

When a document travels from one computer to another, there is no assurance that the receiving machine will have the fonts necessary to compose and display it properly. For that reason, it's increasingly common for applications to offer the ability to embed in a document file the fonts used to format it. Embedding takes place when you save a file. Embedding offers an ironclad guarantee that the document will appear on any computer the way it looked on its creator's machine. The font data is embedded in the document in a platform-neutral form; that is, it will work on a Mac file opened on a PC, and vice versa.