Lecture: Week 7 (continued)

VIII. Screen shots

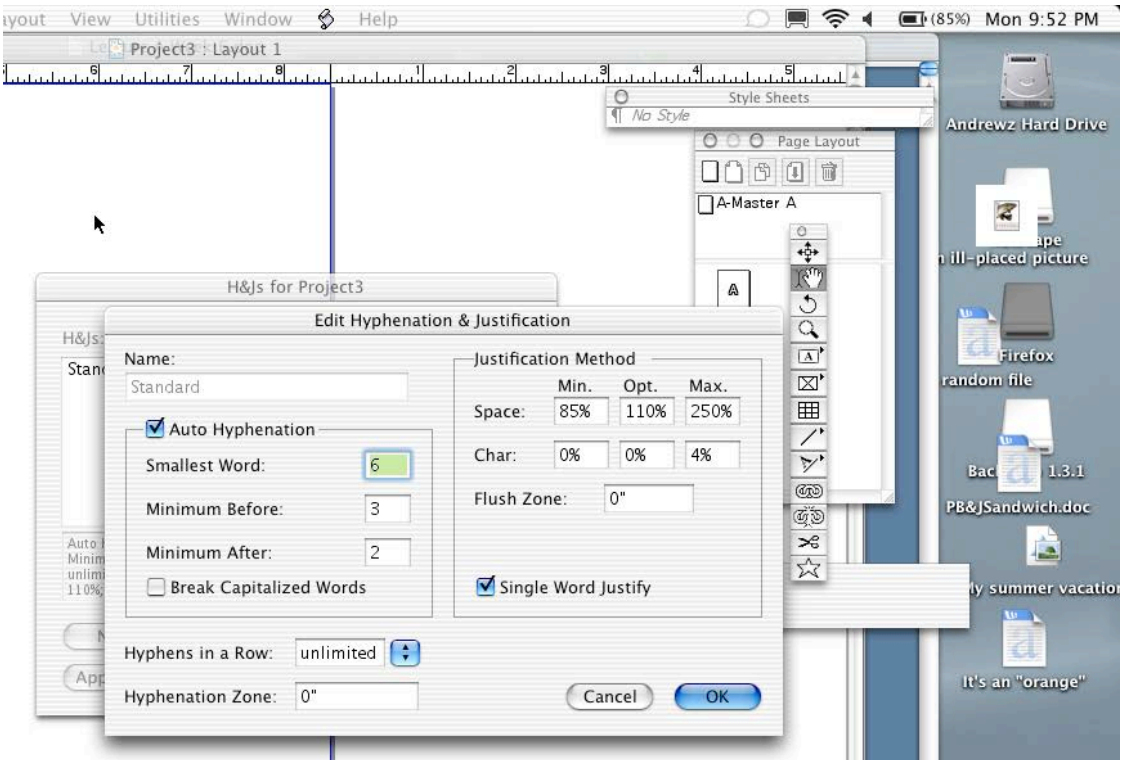


Fig. 7: Don't let this happen to you.

Fig. 7 is less a screen shot than a cry for help. Readers hoping to learn something about QuarkXPress will be hopelessly distracted by the layers of irrelevant information they are faced with instead: the Finder window sticking out underneath the QuarkXPress document; the various palettes which may or may not be relevant, some of which are obscured; the amount of battery power left on the laptop on which the screen shot was taken; the messy desktop full of old assignments and family photos; and what time it was when the whole mess was immortalized.

Keep these ten rules in mind when creating your own screen shots:

1. Screen shots should be clear, clean, and focused. No one wants to see that desktop picture of your pooch—no matter how cute it is—in the background.

Fig. 8: *Gossos no!*: A no dogs sign on the beach in Sitges, a sleepy town just south of Barcelona, where Catalan, not Spanish, is the preferred language. No matter how you say it, keep dogs, and other irrelevancies, out of your screen shots.

2. Screen shots should contain only the information that you are trying to convey to your reader.

3. If appropriate, include just enough extra visual information to situate the portion of the interface with which you are dealing within the larger application interface.

4. Sometimes, more is more: it is often helpful to include a picture of the cursor itself within the screen shot if you are illustrating actions like selecting, choosing, or clicking (see Fig. 9).

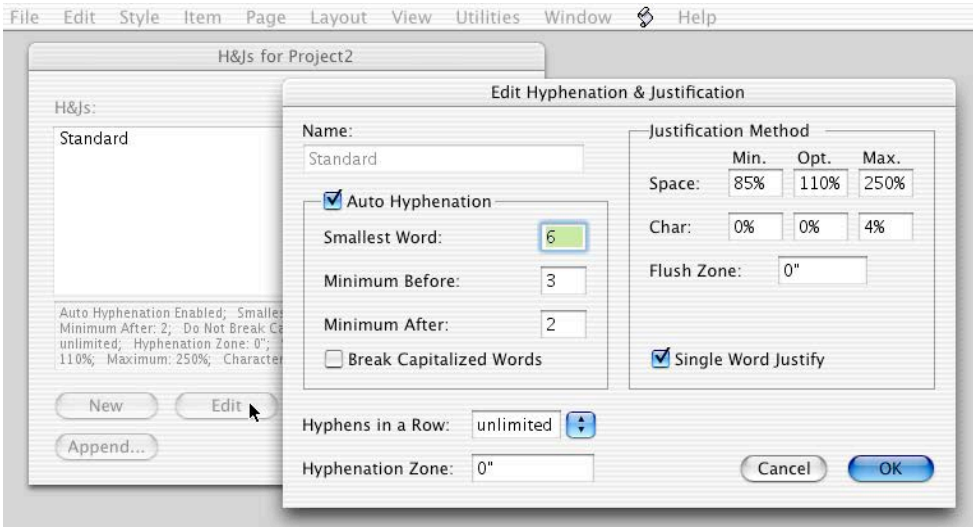
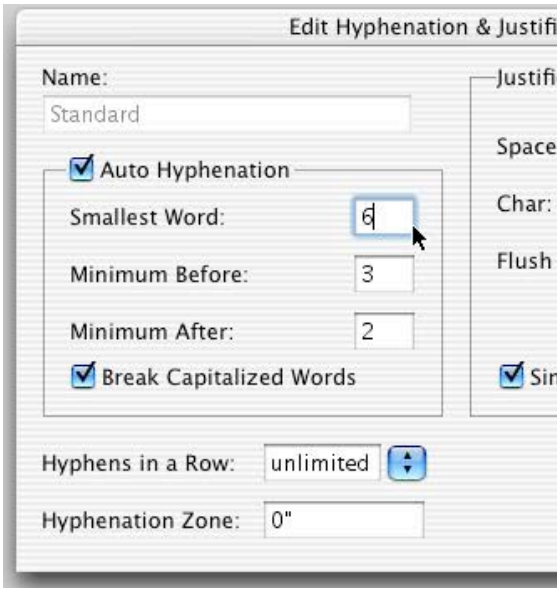
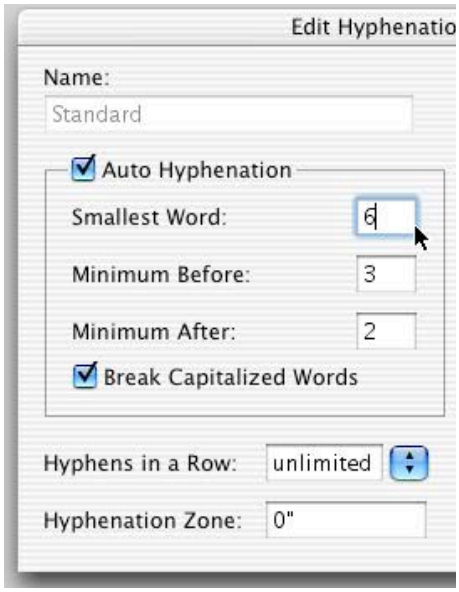


Illustration A. *Click on the Edit button, and the* Edit Hyphenation and Justification *dialog box will appear.*

Fig. 9: A screen shot in recovery: notice that background is a soothing gray. All but the relevant information—in this case, QuarkXPress' Edit Hyphenation and Justification dialog box—has been eliminated from view: the toolbar, the palettes, even the QuarkXPress document itself. I've chosen to keep the QuarkXPress menu bar to orient reader to the environment of the application, but I could have just as easily eliminated it, depending on the context of the screen shot. Finally, I've included both dialog boxes and the cursor to illustrate the clicking action in the instructions, and its results.

5. But sometimes, less is more. If you are discussing a specific part of a dialog box or a very long menu, after situating the reader within the interface, enlarge the area on which you wish to concentrate, and eliminate the rest. Be clear that your screen shot represents only a portion of the actual interface element that the reader will be seeing on his or her own computer.

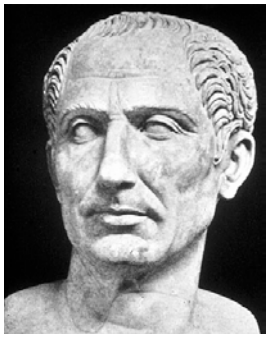




Illustrations B & C. *In the* Edit Hyphenation & Justification *dialog box, you can manipulate the way the application breaks words at the ends of lines. Select the* Break Capitalized Words *checkbox to allow the application's hyphenation and justification algorithm to work to the best. The values in the* Smallest Word, Minimum Before, and Minimum After *value fields affect the length of the smallest word that can be hyphenated, the minimum number of letters prior to a hyphen allowable, and the minimum number of letters after a hyphen allowable by the program, respectively. The smaller the values, the better the type will look.*

Figs. 10 & 11: Less is more: Once you have oriented the user, a partial screen shot allows you zoom in on a small area of a larger dialog box, enlarging its details. Cutting off words or other details (left) makes it obvious that it is a partial screen shot, further serving to situate the reader within the interface, and is preferable to the cleaner version on the right.

Fig. 12: *"Gallia est omnis divisa in partes tres."—Julius Caesar* (pictured)



6. Screen shots must always be captioned. Screen shots without captions are worse than useless. How will the reader know what they represent? Along those lines, all of Gaul was divided into three parts—and so are all captions: captions have a name, a number or letter, and a description. No caption ever has fewer than three parts. Ever. Hail Caesar! Those of us who are about to die salute you.

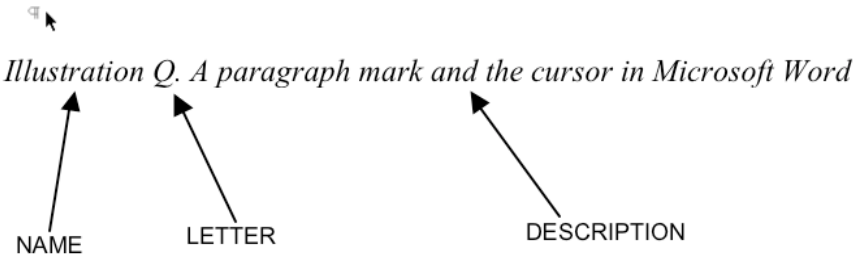


Fig. 13: The proper, tripartite way to caption an illustration

7. Once you establish a naming convention, use it consistently. If you call your screen shots *figures*, don't switch to *illustrations* halfway through. If you start numbering them, don't mix in letters. You can have parallel systems at work if, for example, you are counting illustrations separately from tables, but be sure to keep track of them carefully. Similarly, make sure that your typographical conventions are consistent: if you choose to set off the name and number of the illustrations from the description by using a sans serif bold font, do so consistently; otherwise, it becomes very confusing to read.

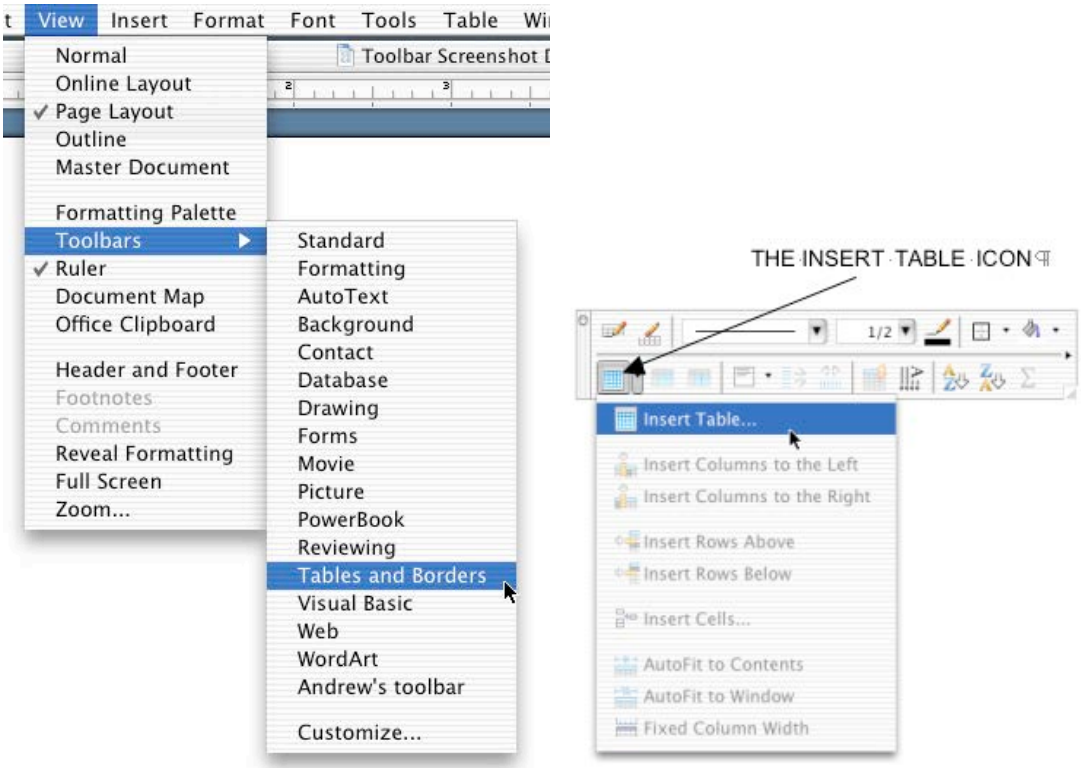
8. Always refer to your screen shots, by name and enumerator, within the text of your documentation. For example:

   For an example of a paragraph mark and the cursor, see Illustration Q, above.

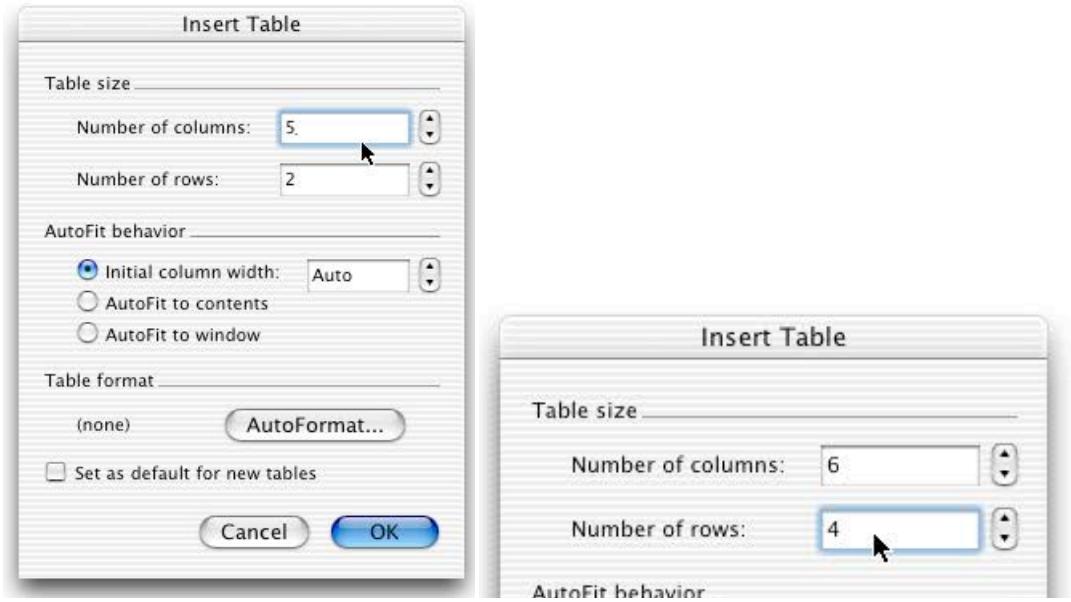
   The name of your screen shot, whether it is *illustration*, *figure*, or whatever you choose, is always *upper case*.

9. Always refer to your screen shots *in order*. Don't start with a reference to Figure 5, followed by a reference to Figure 2, followed by one to Figure 16. If necessary, reorder your screen shots.

10. Screen shots by themselves sometimes don't do the trick: it's quite often necessary to label the relevant portions of them to draw the reader's attention to what is most important about them (see Fig. 13 above, for example.) If labeling the screen shot becomes visually intrusive and obscures important information, use arrows to point out the feature you are discussing, and put the label off to the side. Make sure the label and the arrow line up clearly. And don't put boxes around the labels just because Microsoft Word puts them there automatically—they're ugly and distracting.



Illustrations D & E. *To insert a table into a Microsoft Word document, select* Tables and Borders *from the* Toolbars *submenu in the* View *menu (left). The* Tables and Borders *toolbar will appear (right). Click on the* Insert Table *icon, and select the first option,* Insert Table….



Illustrations F, G, & H. *The* Insert Table *dialog box will appear (left). In the* Table size *section, enter the number of columns and rows for your table in the appropriate fields (right) and click* OK. *A six-column, four-row table will be inserted into your document (below).*

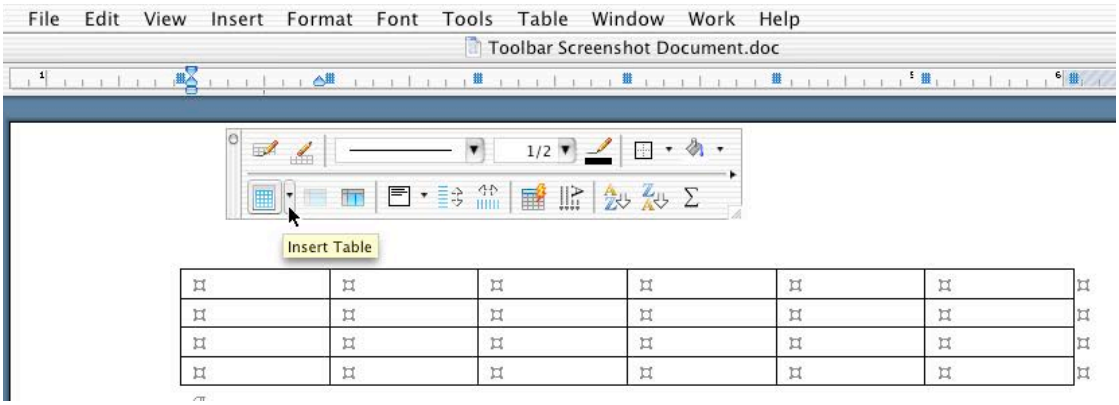


Fig. 14: The perfect screen shots: a picture is *not* worth a thousand words—but the perfect interaction of pictures and text is priceless.

To situate the menus in Illustration D, I included Word's main menu bar. To show that the command can only be executed if a document is open, there is a document open on the desktop, although it adds a bit of clutter (notice that I named it appropriately).

Illustration E has an arrow pointing to the Insert Table icon because the interface is not entirely intuitive, and the arrow is labeled. The arrow is placed as unobtrusively as possible: it does not obscure other parts of the toolbar that the user might be interested in looking at. Illustration G is a detail view of Illustration F. Illustration H brings it all together, showing the context of the document.

And of course, the illustrations are captioned, and the caption has three parts: a name (*illustration*) an enumerator (*R*) and a description. It can be helpful to offset the name and enumerator from the description by using different typography (see Illustrations A–H).

Finally, the series of screen shots tells a story: what the readers should do, what happens because they do it, and what the results are. It mirrors the story told in the text.

The goal of any screen shot is to document, as accurately as possible, what the readers will see on their screens. Anomalies that occur through the use of customized software should be avoided. When that isn't possible, the anomaly should be acknowledged in procedural and reference documentation (see Fig. 15).

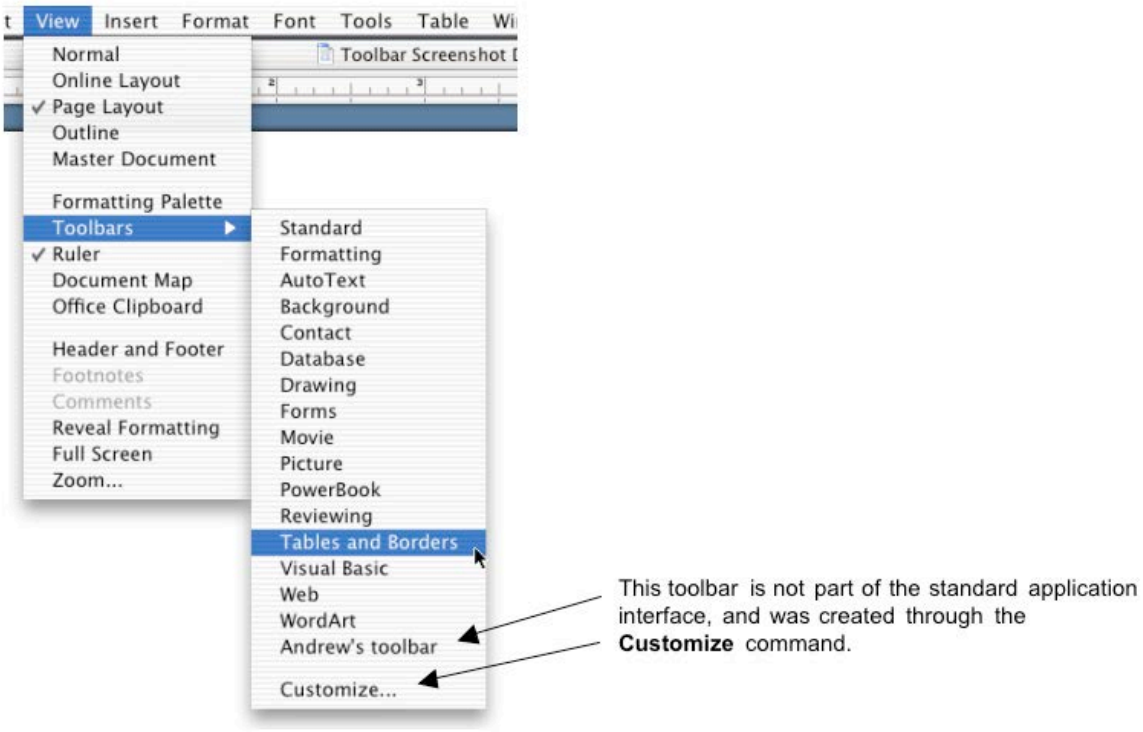


Fig. 15: Coming clean about your dirty interface…

IX. Typographic conventions

Software documentation describes how software works by helping its readers interact with the software's interface. In order to do that, you have to find a way to describe that interface with precision. There are several hurdles. Most are linguistic. Some involve using the correct terminology consistently: a *dialog box* is different from a *value field*, for example (we'll get out the technical dictionary next week…).

One obstacle to communicating clearly with your readers, however, is essentially graphical or typographic, and involves how you choose to write about words, commands, menu selections, names of data and value fields, and other named items within the interface itself. There is a difference between this sentence:

Ex. 1.
Copy the text from one document and paste it into another.

And this one:

Ex. 2.
Highlight the text you wish to transfer, and select **Copy** *from the* **Edit** *menu. Then click in the target document, and select* **Paste** *from the* **Edit** *menu; the text will appear.*

In both cases, I've used the words *copy* and *paste*. In the first example, because I used them to describe an action, no special typographic convention was needed. But in the

second example, I used them to refer to interface elements—the copy and paste commands in this application's edit menu. For that reason, they required a distinguishing typographic convention—in this example, initial caps, roman type instead of italics, and boldface to differentiate the words *copy* and *paste* (and *edit*) from those surrounding them. That differentiation also functions as a signal to the reader that the words carry a dual connotation: both their primary semantic one that tells the user about a function *(what they do),* and a secondary, nonverbal one that places them within a context *(where they do it).*

Figs. 16, 17, 18, & 19: People, places, and things: *Paris Hilton* vs. *overexposure*; *Paris, France* and *Frankfurt, Germany* vs. cities; and *Oscar Mayer Wieners* vs. *frankfurters*—in each case, the former is a proper noun, the latter a common noun (no matter how common you think Paris Hilton may be).

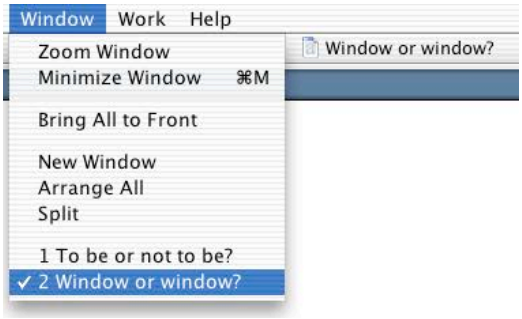
The distinction is similar to the one between common and proper nouns. Common nouns—*overexposure, cities, frankfurter*—are not capitalized; proper nouns—Paris Hilton; Paris, France; Frankfurt, Germany; and Oscar Mayer Wieners—are. That's a typographic convention, if you will. Only when describing the interface of a piece of software, things become a bit more complex. Sure, the examples above are clear enough. But in the following example, I've purposely left out the all of the typographic conventions, including capitalization. Can you tell when the word *window* refers to the application's interface, and when it refers only to the concept of a window?

Ex. 3.
Use the commands in Microsoft Word's window menu to affect how the application's windows behave. Zoom window makes the active window a standard size and places it in a standard location on your desktop, minimize window hides the active window, and new window places the active document in a new window on your desktop without closing the old document. Remember, a window differs from a document, although the two go hand-in-hand—windows display documents. But they have other features, depending on your operating system, including scroll bars and close boxes or buttons.

Not so easy to tell what's what, is it? Now take a look at Ex. 4, below:

Ex. 4.

Use the commands in Microsoft Word's **Window** *menu to affect how the application's windows behave.* **Zoom Window** *makes the active window a standard size and places it in a standard location on your desktop,* **Minimize Window** *hides the active window, and* **New Window** *places the active document in a new window on your desktop without closing the old document. The last items in the menu are the names of all open documents; the one that is active checked.*



A window differs from a document, although the two go hand in hand—windows display documents. But they have other features, depending on your operating system, including scroll bars and close boxes or buttons.

Illustration I: *The* **Window** *menu*

Much less ambiguous, yes? Accompanied by the proper screen shot, the words in a sans serif, bold typeface recognizably refer to a particular menu and commands in that menu. So **New Window** and *new window* may say the same thing semantically, but they tell the reader two very different things.

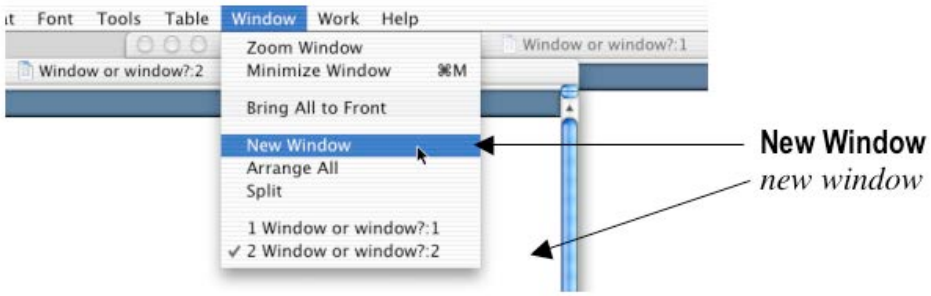


Fig. 20: Typographic conventions give technical writers an added tool for describing the niceties of software interfaces. Here, the same words refer to two different things: a menu item, and the resulting window it creates.

But it is equally important to look at the words to which no special typography has been applied. Let's go back to Paris Hilton and frankfurters for a moment. Proper nouns are *proper* because they refer to specific people, places and things. Similarly, those words in software documentation that refer to **specific** parts of the application interface require a distinguishing typographic convention. Those that refer to interface elements in a general way do not. So even though *scroll bars* and *close boxes* refer to things that one might find in an application interface—even in the very window the documentation is discussing—the same typographic conventions are not applied to those words. Those words do not, in this context, refer to specific things. Similarly, in the sentence in Ex. 4 that begins, "A window differs from a document," the word *window* is referring to windows in general.

But the distinction goes beyond one between specificity and generality. Consider the following examples:

Ex. 5.
Use the window's scroll bars to navigate through the document.

Ex. 6.
Use the window's **scroll bars** *to navigate through the document.*

Which is correct? If you voted for Ex. 5…you're right. Why? Typographic conventions also signal singularity—something **unique** within the interface about what the word or words describe. A user will come across *many* scroll boxes while using Microsoft Word, but there is only *one* **Window** menu.

Let's look at one last example—part of a sentence we've seen already.

Ex. 7.
New Window *places the active document in a new window on your desktop.*

The second *new window* really *does* refer to something specific: the new window that appears on the desktop after the command is executed. And it is arguably unique: each time the command is executed, it creates a window with the name of the active document and the next number in a series, so no two can ever really be alike—at least not during the same computing session. If I create a window 2, close it, and create a new one, it will be numbered 3, not 2.

But if I quit the application, reopen the document, and create a new window, window 2 will be back. This relates to the final, peculiar quality of words that require typographic conventions. It is one of duration. Relative to the application itself, the new document I've created has less **permanence**. It is not a fixture within the application interface itself. Therefore, it does not require a distinguishing typographic characteristic.

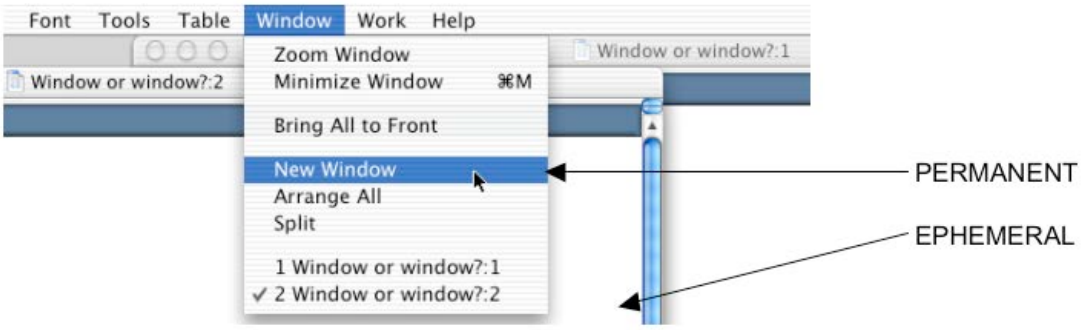


Fig. 21: The New Window command is permanent; a new window is ephemeral.

Ultimately, the difference between **New Window** and *new window,* is ultimately a semiotic one. So while a graphic designer may be picking which typeface is used for what, it's the job of the technical writer—not the documentation's art director—to figure out how to

apply them correctly. Until you have the benefit of working with a designer, you're on your own, so keep the following rules in mind.

1. Explain how your typographic conventions work up front: don't keep your readers guessing about what they mean.
2. Along the same lines, explain any shorthand you invent for things like choosing submenus from menu options from the menu bar at the beginning of the documentation. Each piece of writing is its own universe: don't assume your readers are familiar with conventions you have seen elsewhere, or that are commonly used (see Fig. 27).

View > Toolbars > Drawing

Fig. 22: This kind of shorthand for *Select the* Drawing *submenu from the* View *menu's* Toolbars *menu* is not acceptable for a first usage.

3. Always duplicate the interface's way of spelling, capitalizing and punctuating, even if it isn't consistent.
4. Contrasts work best. If your running text has serifs, try a bold, sans serif font to refer to the interface, and vice versa.
5. Keep it simple, and keep it consistent. The more easily the readers can follow your conventions, the more successful you have been.
6. Quotation marks are not typographic conventions—they are punctuation marks. The only thing that should be between quotation marks are something someone said, or a word you are using in an odd or self-conscious fashion.
7. Use italics, not quotation marks, when you need to call attention to a particular word or words within running text—either to emphasize them, or if you need to speak about them (see Ex. 8).

Ex. 8.
The word *word* is a noun.

In summary, when deciding whether a word or term requires the typographic convention you have established to refer to the interface you are describing, determine whether it meets the following criteria:

1. Specificity
2. Uniqueness
3. Permanence

