

---

### E.3 Using Word for Windows to format large, technical documents

To call MicroSoft Word for Windows a word processor is like calling the *Mona Lisa* a picture. The features of this program handle all the details of compiling a technical book, such as:

- Mathematical symbols, including the whole Greek alphabet, with a powerful equation editor for generating integrals, matrices, tensors, etc.
- Importing graphics, rescaling it, and positioning it anywhere on the page. If desired, only the horizontal position can be specified, so that the figure will move from page to page as the preceding text is modified.
- Generating text by performing arithmetic and logical calculations on other text. You can build an active spreadsheet into a document!
- Making tables, with automatic resizing of rows to fit the text.
- Automatic cross-referencing by page number, section number, figure number or selected text.
- Definition of paragraph styles with any combination of formatting commands, and with the freedom to adjust any of those commands in an individual paragraph without needing to define a new style.
- Viewing the document in outline form.
- Generation of various tables of contents, with customizable formats.
- An impressive collection of tools for selecting, cutting, pasting and archiving segments of text and graphics.
- A dictionary that not only checks spelling, but also gives definitions and synonyms.
- Annotation by footnotes, endnotes, marginal notes, hidden text and initialled reviewer's comments.
- Customizing the whole system by defining macros and changing the menus.

What follows is a collection of tricks and pointers that I have collected through over a year of using Word for Windows to prepare my Ph.D. dissertation and other technical documents, along with some complaints about problems encountered.

This is not a tutorial on how to use Word for Windows. A knowledge of its commands and basic operating procedures is assumed. Anyone who has ever used a word processor will have little trouble getting started in Word<sup>2</sup>. As for the advanced features, there is no question that learning them will take time and effort. Word is an amazingly complex program, and after a year of intense use there are still many features that I have never tried. Word comes with a tutorial, as well as a comprehensive documentation set, including separate instructions written for current users of various other major word processors. The *User's Reference* will become your main tool for exploring advanced capabilities. Perhaps these results of my own experience will also be of some help to you.

---

<sup>2</sup> Word is used here to refer to *Word for Windows*. Some of what is said here may apply to the similar *Word for Macintosh*, but this has not been verified. None of what is said here applies to the very different, text-based *Word for DOS*.

**Mathematical symbols**

The Word documentation is, on the whole, excellent. But amazingly, it leaves out the most important information you need to have in order to begin using it for technical documents: the mathematical symbols, including the Greek alphabet. It's not that they are not supplied. They are. The problem is that you have to do some detective work to find them. Here are the results of that search.

When you choose a font in Word, the characters in that font depend not only on the typeface that the font was made in, but also on the character set represented. There are two different character sets represented in the fonts supplied with Word, and those made for Windows by the BitStream font making software. All of the fonts except symbols represent what is known as the *ANSI Windows* character set. Here is a table of this character set in the BitStream Dutch typeface:

ANSI Windows character set in Dutch typeface, 10 point										
code	0	1	2	3	4	5	6	7	8	9
30	-			!	"	#	\$	%	&	'
40	(	)	*	+	,	-	.	/	0	1
50	2	3	4	5	6	7	8	9	:	;
60	<	=	>	?	@	A	B	C	D	E
70	F	G	H	I	J	K	L	M	N	O
80	P	Q	R	S	T	U	V	W	X	Y
90	Z	[	\	]	^	_	`	a	b	c
100	d	e	f	g	h	i	j	k	l	m
110	n	o	p	q	r	s	t	u	v	w
120	x	y	z	{		}	~			
140						'	,	"	"	o
150	—	-								
160		ı	ç	£	¤	¥	ı	§	¨	©
170	ª	«	¬	–	®	–	°	±	²	³
180	´	µ	¶	·	¸	¹	º	»	¼	½
190	¾	¿	À	Á	Â	Ã	Ä	Å	Æ	Ç
200	È	É	Ê	Ë	Ì	Í	Î	Ï	Ð	Ñ
210	Ò	Ó	Ô	Õ	Ö	–	Ø	Ù	Ú	Û
220	Ü	Ý	Þ	ß	à	á	â	ã	ä	å
230	æ	ç	è	é	ê	ë	ì	í	î	ï
240	ð	ñ	ò	ó	ô	õ	ö	•	ø	ù
250	ú	û	ü	ý	þ	ÿ				

This table is also given, although in a less handy format, in Appendix C of the Word *User's Reference*. The codes for the characters are obtained by adding the numbers in the far left column to the numbers in the top row. For example, "½" is character number 189.

As you can see, this set has all of the ordinary English letters and the Arabic numerals as well as all the standard punctuation symbols in the top half. These are the ordinary ASCII characters, and are inserted into a document just by pressing a single key or a shifted key on the keyboard. The second half of the set contains special symbols such as the curly apostrophe, ' , the curly quotes, "", the section symbol, §, the plus-or-minus, ±, and others. These

characters are entered by toggling the NumLock key to on and then holding down the Alt key while entering on the numeric keypad the *four-digit* (i.e., the first digit is “0”) integer for the character code.

The other character set is the one that turns your laser printer into a technical typesetter. This is the *PS Math* character set, and is represented in the Windows *symbol* font.

PS Math character set in Windows “symbol” typeface, 10 point										
code	0	1	2	3	4	5	6	7	8	9
30	–			!	√	#	∃	%	&	∞
40	(	)	*	+	,	–	.	/	0	1
50	2	3	4	5	6	7	8	9	:	;
60	<	=	>	?	≅	A	B	X	Δ	E
70	Φ	Γ	H	I	∂	K	Λ	M	N	O
80	Π	Θ	P	Σ	T	Y	ζ	Ω	Ξ	Ψ
90	Z	[	∴	]	⊥	—		α	β	χ
100	δ	ε	φ	γ	η	ι	φ	κ	λ	μ
110	ν	ο	π	θ	ρ	σ	τ	υ	ω	ω
120	ξ	ψ	ξ	{		}	~			
140										ο
150	–	–								
160		∓	'	≤	/	∞	f	♣	♦	♥
170	♠	↔	←	↑	→	↓	°	±	"	≥
180	×	α	∂	•	÷	≠	≡	≈	...	
190	–	←	≠	∫	℔	∅	⊗	⊕	∅	∩
200	∪	∩	⊇	∩	⊂	⊆	∈	∉	∠	∇
210	®	©	™	∏	√	·	¬	∧	∨	↔
220	←	↑	⇒	↓	◇	⟨	®	©	™	Σ
230										
240		}	}	}	}	}	}	}	}	
250								.	.	.

You access these symbols in exactly the same way that you do those from the other set, except that you choose the *symbol* font. So, for example, in the ANSI set, the letter “p” has code 122. Code 122 in the Math set is “π”. So all you have to do to get a “π” is to select the symbol font and type “p”. Similarly, you access the symbols in the second half of the set using the same method as for the second half of the ANSI set. For example, to insert the infinity symbol, “∞”, into your text, you toggle the NumLock key to on, hold down the Alt key, and enter “0165” on the numeric keypad.

Compound symbols not found above can be constructed using the overstrike and other features of the equation editor. Here are some useful examples:

Making compound symbols		
Symbol		Code
$\hbar$	$\hbar$	{eq \o(h,/)}
minus-or-plus	$\mp$	{eq \o(\s\do1(+),\s\up1(-))}
much greater than	$\gg, \gg$	{eq >\d\ba5(>)}, {eq >\d\ba2(>)}
vector	$\vec{x}, \vec{X}$	{eq \o(x,\s\up4(\rightarrow))}, {eq \o(X,\s\up7(\rightarrow))}
dagger	$A^\dagger$	{eq A\s\up5(o(i,\s\up1(-)))}

Notice that in the minus-or-plus and dagger symbols, the components are reduced to 6 point size. It would be helpful if MicroSoft would implement these two symbols directly in their symbol fonts. Also notice a problem with the vector symbol. This is made by raising the right arrow symbol to a superscript position and the line spacing algorithm makes extra room for the superscript by moving down the line that the vector appears on, as it does to this line for example:  $\vec{x}$ . But as you can see, the line is moved down far more than is necessary. This is because Word fails to notice that the arrow is not a full-height character.

A note on the minus sign: The character obtained from the keyboard, “-”, is really designed to serve as a hyphen. It is too short for a minus sign. I find that character number 173, “-”, looks much better in equations.

For symbols which you use very often, it is handy to define macros that enter them for you and to assign these macros to single keystrokes.

Enlargement of special symbols like brackets and integrals are handled very well by the equation editor, as in:

$$I = \frac{4\sqrt{mF}}{\pi} \int_0^{\pi/2} \left[ \sqrt{1 - \frac{E}{2F} \sin^2 \left( \frac{y}{2} \right)} - \frac{1 - \frac{E}{2F}}{\sqrt{1 - \frac{E}{2F} \sin^2 \left( \frac{y}{2} \right)}} \right] d \left( \frac{y}{2} \right),$$

although, as you can see, the square root symbols don't come out very pretty.

Sometimes the equation editor messes up and does not expand a set of brackets properly:

$$A_M (e^{i\omega t} |n+M\rangle + e^{-i\omega t} |n-M\rangle).$$

A fix for this situation is to push up and down on some space-characters somewhere inside the brackets. In the above example, if a space is added before and after the central plus sign, and if the one of those spaces is subscripted by four points, and the other is superscripted by ten, then the brackets expand nicely:

$$A_M \left( e^{i\omega t} |n+M\rangle + e^{-i\omega t} |n-M\rangle \right).$$

## Headings and sequential numbers

It was not easy to figure out how to use the sequence fields to obtain systematic numbering and referencing of chapters, sections and figures. The description of the *seq* field in the *User's Reference* is not very helpful. (In fact, I think it's wrong.) A method that was found to work very well is given here.

Chapter headings are entered in the following format:

```
{set nickname "{seq chapter}"}{ref nickname}{ eq section \r 0 \* CharFormat}{ eq figure \r 0 \* CharFormat}→FullName
```

The fields here work as follows:

- The first *seq* field, {seq chapter}, sets a sequential chapter number, which changes if the chapter is moved or another chapter is inserted in order before it.
- Next, the *set* field, {set ...}, assigns the text of that chapter number to the bookmark *nickname* for use in cross references. *Nickname* should not be more than 20 characters long.
- The *ref* field, {ref *nickname*}, becomes the first such reference, inserting the chapter number at this location.
- The next field is *not* an *eq* field. It is a *seq* field in which the formatting of the first character, the “s”, has been changed so that its color is white! This field, because of the “\r 0”, specifies that section numbering should be restarted at zero. It then inserts that section number, “0”, in the text. But the CharFormat switch tells Word to format the section number the same as the first letter of “seq”, so the section number is invisible. The net effect of this field then is that the next section number will be a “1.”
- The next field, in just the same way, restarts figure numbering also at “1.”
- The arrow signifies a tab.
- *FullName* is the name of the chapter, typed in normally.

Appendix headings have exactly the same syntax, except that a switch is added to the first *seq* field to obtain numbering in capital letters instead of Arabic numerals:

```
{set nickname "{seq chapter \* Alphabetic}"} . . .
```

In addition, the first appendix uses the reset switch to start the numbering of appendices at “A” instead of continuing in the sequence where the chapter numbers left off:

```
{set nickname "{seq chapter \r 1 \* Alphabetic}"} . . .
```

Section headings use the field structure:

```
{set nickname "{seq chapter \c} . {seq section}"}{ref nickname}→FullName
```

Here, the sequential number is set for the section by {seq section}, while the field {seq chapter \c} inserts the most recent chapter number. A period is inserted between these values, and the resulting composite section number is set to the bookmark *nickname*. The *ref* field then inserts this composite number at this location.

Figure numbers use a similar structure:

Figure {set nickname "{seq chapter \c}·{seq figure}"}{ref nickname \\* CharFormat}

The character format switch, “\\* CharFormat,” specifies that the result should appear in the character format of the first character of the field instruction “ref,” which in the actual figure numbers is Swiss Bold 10 point. If the CharFormat switch were not used, then the number would appear in the default format of the paragraph, which in the case of the captions where figure numbers are set is Dutch 8 point.

A chapter, section or figure number is referenced with a *ref* field. So, for example, the field {ref Stn\_CmptrWinWord} inserts the number of this section: E.3. But since figure numbers are created in bold face, they are reproduced by default in that same style. For example, the field {ref Fig\_IntroLabData} inserts the figure number of the laboratory data in the introduction: 1.1. To make this appear in ordinary type, use the CharFormat switch, as in {ref Fig\_IntroLabData \\* CharFormat} to get 1.1. Another way this switch is used here is to produce section references in 6 point type, as in {§{ref Stn\_CmptrWinWord \\* CharFormat}} TO get {§E.3}.

Section numbers and titles are reproduced in the page headers using a *styleref* field. This field copies the text of any preceding paragraph of a particular style. So the text of the headers is:

§{styleref 4} →[tab] {page}

The “4” is shorthand for “heading 4”. But here is a sticky problem: What if a chapter begins with a long introduction that is not given a section number, as is the case for Chapter 1? Or what about a chapter that does not have any section divisions at all, like Appendix F? The second page of such a chapter will be headed with the number and title of the last section of the previous chapter, which is definitely not desirable. There is a way to get around this problem which is similar to the use of the *seq* fields with the “\r 0” switches in the chapter headings, mentioned above. The trick is to follow every chapter heading with a paragraph with the following properties:

- It contains only the field {styleref 1}, which reproduces the chapter number and title, including any changes made later.
- It has paragraph style “heading 4” (but with Border Above turned off), so that its contents are reproduced in the page headers. And (here’s the trick:)
- It is made invisible by changing its character color to white (Alt-T-C, Alt-L, W, ←). When *styleref* copies the text to the header, it does not copy the text characteristics, so the text is visible. It does not work to *hide* the text (Alt-T-C, H, ←) because then *styleref* wouldn’t find it.

There has been one problem arise in the use of the *styleref* field. For some strange reason, *styleref* translates curly quotes into straight ones. This happens in Appendix F, as you can verify by looking at the second page thereof. But, on the other hand, if I hadn’t pointed it out, I’ll bet you would never have noticed.

## Bookmarks

In general, the bookmark facility in Word is tremendous. I use it for moving around in the text, cross referencing section and page numbers, and even reproducing entire equations and tables when I wish to repeat them exactly.

There are two ways to set a bookmark, and each is best used only for a separate set of circumstances:

- Use the *Insert Bookmark* command at a particular cursor location, or on selected text. This method is best used only for setting a *temporary* bookmark with a name that you set aside only for such use, such as “temp”.
- Insert a *set* field, and if the bookmark contains text, also a *ref* field. This method is demonstrated above in setting and showing all of the chapter, section and figure numbers. This is the method of choice for setting permanent bookmarks because it allows you to see where you have put your bookmarks, what you have called them, and exactly what text is included in them, all by turning *View Codes* on.

It can actually be dangerous to use the *Insert Bookmark* command for permanent bookmarks. If you insert text immediately in front of (i.e., to the left of) a bookmark set in this way, then the bookmark will expand to include this text. This can cause several deleterious effects which are best not experienced.

There is a problem common to both types of bookmarks. If you copy a selection of material (text and/or graphics) to the clipboard and then paste it somewhere else, any bookmarks in the material are moved to the new location. Usually this is not what is wanted in a copy and paste procedure. Be careful to remove any unwanted bookmarks from the pasted material, and then to update any *set* fields in the original material. It would be nice if MicroSoft would arrange it so that bookmark settings remain where they were originally set unless the *set* field is actually cut from the text.

Here is an undocumented trick for setting an equation as a bookmark. This is useful if you want to repeat the equation in several places without retyping it. The following code:

```
{set Einstein "{eq E=mc\sup5(2)}"}{ref Einstein}
```

does not work. I don't know why, but the *ref* field inserts nothing. Here is what you use instead:

```
{set Einstein "E=mc\\s\\up5(2)"}{eq {ref Einstein}}
```

This works, and yields  $E=mc^2$ . Note that the formula codes must now be preceded with two backslashes instead of one.

### The *sic* field

This is a field that MicroSoft forgot to implement, but you can invent it on your own. The idea of the *sic* field is to specify a particular occurrence of a word that the spelling checker should skip over, as in:

Don't tell me I've made a spelling mistake [*sic*].

You can get the spelling checker to ignore this intentional mistake by using an *eq* field:

Don't tell me I've made a spelling {eq mistake} [*sic*].

This works because the spelling checker does not look inside of equations. It does not work to use the *quote* field, inside of which the checker does look.

### Dealing with long documents

The *Sampler* that comes with Word warns against creating very large document files. The *User's Reference* has an entry that explains that you should break long documents into smaller pieces, and manage these parts with *include* and *RD* fields. I have some different advice for you: Type away to your heart's content.

I have one file that is over 1½ megabytes in size. It is an electronic dictionary under development by Ennex Technology Marketing, Inc. and, printed out, it would be 443 pages long. That's long. I find it most convenient to be able to keep all of that data in one unit, and I see nothing in Word that prevents me from doing so. In fact some features make it especially practical to work with long documents:

- *FastSave* is most important. It only saves changes so it only takes a few seconds to complete, no matter how big the document is.
- Going to a bookmark usually only takes a second or two.
- Even searching is quite fast, taking only a few minutes to go through a very long document.

This ability to work with long documents does not come from having a large memory machine. For the last six months I have been working on a notebook 386SX with three megabytes of RAM and a 20 megabyte hard disk, most of which is full.

There are, of course, some drawbacks to working with long documents:

- Every five or ten saves or so will be a full save and these can take several minutes. To make Word more efficient with long documents, MicroSoft should consider extending the amount of changes that can be saved with *FastSave*.
- There is a great deal of time wasted on repeated pagination whenever anything is printed. Even if you paginate the document first, and then go directly to print it out, Word will repaginate it again before printing. If you ask Word to print a single page, even just the first page, Word will repaginate the document before giving you that page. Even if you print out a single line of selected text, repagination will always take place first. In a long document, repagination can take a long time, so this is a terrible nuisance.
- Hyphenation is a problem if you have graphics, tables, or other items given specific positioning on the page, with other text flowing around them. That text cannot be hyphenated correctly in the normal editing mode; you must be in *Page View*. Unfortunately, trying to hyphenate a long document in *Page View* will exhaust the memory. So you can either hyphenate the document in segments, all in *Page View*, or you can hyphenate the whole document in normal editing mode, and then go into *Page View* to hyphenate the text flowing around the positioned items. However, hyphenation in *Page View* suffers from the same sickness as printing: it repaginates too much. (Actually it is worse than printing, because it repaginates more slowly and more often, once when you enter *Page View*, once every time you start hyphenating a selection, then again after the selection has been hyphenated). The results of hyphenation may



not be worth the wait. Unfortunately, it is in flowing text that hyphenation is most important, because it comes in narrower-than-usual columns. I would say that fixing the repagination routine should be the Number One priority for improvements to Word, at least with respect to making it easy to work with long documents.

- Printing a long document can crash the system if the computer runs out of disk space for the temporary files generated in printing. There is no warning. Processing simply stops and the computer is hung. When you reboot, you find that your hard disk is full, and the culprit is a mess of files in the \temp directory. The solution is, after first clearing the \temp directory, to either free up some more disk space or print the document in segments.

I find that, despite these problems, I enjoy the convenience of dealing with a large document as a single unit. I should point out, however, that a colleague of mine disagrees, preferring to work with smaller, more manageable files. It is a matter of personal preference.

## Problems

### *The errant spelling checker*

The worst part of Word for Windows is the spelling checker. It has many fairly severe deficiencies:

- There needs to be an option for *Ignore Once*. For example, if I allow one occurrence of *dir*, that should not automatically mean that I accept every occurrence.
- There needs to be an option for *Change Once*. For example, one occurrence of *essentiall* may need to be corrected to *essential*, but the next may need to be *essentially*.
- Some words continue to be flagged even after being added to the dictionary. This has been the case with *adjoint*, *conjugacy*, *counterexample*, *Davydov*, *eigenbasis*, *eigenstates*, *Hamiltonian* and *Hermitian*. However these errors are not consistent.
- *Orthogonal* is not in the main dictionary!
- Capitalization of a word in a custom dictionary needs to indicate required capitalization. For example, *Hamilton* should not be equivalent to *hamilton*. The current situation seems to be that capitalization in the custom dictionary is ignored. Another symptom of the problem: I have *GHz* in my dictionary, with that form of capitalization. The spelling checker highlights *GHz* in a document and complains of “Uncommon Capitalization”.
- There is no documentation on the checker messages *Uncommon Capitalization*, *Repeated Word*, *All Letters Not Capitalized* and *First Letter Not Capitalized*. These last two are strange. I get *All Letters Not Capitalized* when Word finds “gm” (for “gram”) in a document, and Word suggests (even with *Always Suggest* off) changing it to “GM.” Is that supposed to be for General Motors? I get *First Letter Not Capitalized* when Word finds the C keyword “goto,” and it suggests changing this to “Goto.”
- The usual default action for the checker is *Ignore*. But in that last example, when Word suggests changing “gm” to “GM” and “goto” to “Goto,” the default is *Change*. This is a dangerous inconsistency that at least once led me to make a change inadvertently.
- During the spelling check, the status line should show the current page and position.

- The suggestion algorithm needs some work. For example, the first suggestion for *permissible* is *perishable*. For *coumpound*, Word suggests *southbound* and *soupcon* (which, by the way should be *souçon*) and does not suggest *compound*.
- When a questionable word is displayed, it should be positioned in the middle of the visible part of the screen, not at the top, to allow examination of the context. The visible part of the screen is that part not taken up by the dialog window.
- The checker has an undocumented message: *Out of global memory*. This should be documented in the *Messages* appendix of the *User's Reference*. When the message comes up, the status line invites me to press F1 for help, but doing so only brings a beep from the computer.

In conclusion, it looks like the quality control team went to sleep during this part of the project.

While they are at work fixing those problems, here are some desirable features that Microsoft should consider implementing as improvements:

- Common abbreviations, including punctuation, should be accepted as words. For example, *Prof.* should be accepted if capitalized and with the period, but *prof* should not. Other examples: Mr., Mrs., Ms., Dr., Inc., Corp., Ltd., etc., e.g., i.e., 1st, 2nd, 3rd, 4th, ....
- Common contractions, including the apostrophe, should be supported for addition to the custom dictionary. For example: it's, don't, won't, wouldn't. Currently, adding these items to the dictionary does not seem to have any effect.
- A possessive formed with an apostrophe should be recognized as a word if the root is recognized. At the very least, it must be possible to add special possessives to the dictionary, such as *Hamilton's* and *Schrodinger's*.
- Certain words should be allowed only in specified combinations. For example, I would like to add *Runge-Kutta* to my dictionary, but not *Runge* and *Kutta* individually. *de Broglie* should be recognized, but *Broglie* alone should not.
- Words added to the dictionary should automatically imply their cognates. For example, adding any one of *vectorize*, *vectorized*, *vectorizes* or *vectorizing* should imply the others.
- A nice option would be *background spell checking* that would check spelling as text is entered.

#### *Problems with tables*

- There is no documented way to keep a table from being split by a soft page break. I have found that it usually works to format all the rows of the table except the last as *Keep with next*, but that seems to fail sometimes. This is an important feature that should be provided and documented.
- It would be nice to be able to merge cells vertically.
- It would be nice to be able to automatically center the contents of a cell vertically, instead of having to do it manually with *Space Before* in *Format Paragraph*.

*Other complaints*

- It would be helpful if hyphenation were done in real time, instead of requiring a special operation. As the program is now, the view of the document is not really WYSIWYG until hyphenation is run.
- Some printer setup options, especially orientation and resolution, should be in the document format with a section or page override available. For example, it would be nice to be able to include the occasional chart in landscape mode.
- As is suggested for the spelling checker above, the search facility (and likewise the search and replace facility) should display the found text in the middle of the (visible part of the) screen, not at the top.