

## Chapter 9

### Macro Examples

Links: The **Chapter Name**, above, moves to the next chapter. All page header links go to the contents menu. Within the chapter, **Topic Titles** returns here. **Other Red Links** are to other locations in this paper. **Blue Links** are to web sources.

Top	Contents	Glossary	Index	Release Notes
<a href="#">Chapter 1 Additional Resources</a>	<a href="#">Chapter 2 Understanding Macros</a>	<a href="#">Chapter 3 First Things</a>	<a href="#">Chapter 4 Controlling Macro Flow</a>	<a href="#">Chapter 5 Using the Dialog Editor</a>
<a href="#">Chapter 6 Math Routines</a>	<a href="#">Chapter 7 Date Routines</a>	<a href="#">Chapter 8 DialogShow/Callbacks</a>	<b>Chapter 9 Macro Examples</b>	<a href="#">Chapter 10 Glossary</a>

Macros In This Chapter		
<a href="#">Math.wcm</a>	<a href="#">ConvertFE.wcm</a>	<a href="#">Wp9select.wcm</a>

The purpose of this chapter is to present you with fully developed macros which utilize many if not most of the routines described in Chapters 3-8, one way or another. Aside from various **// comments like this** included in the macros, little expository information is provided. For that, read the parts of this manual which relate to the techniques used. In this manual's release, both macros below were substantially modified from their former versions. Many, not all, changes are shown with **//green comments**.

Dialogs in these macros were made in WordPerfect's Macro Dialog Editor. So, you won't see any commands which create the various dialogs. You can download these macros at my website as shown below and then you will have the full macro(s), including all their dialogs. Wp9select.exe works in WordPerfect 6.1 or later, but the other two work only in WordPerfect 8.0 or later.

The macros should be seen as illustrating some of the various means available to accomplish what they do – I'm sure that each can be made "better", and, doubtless, I'll be making changes in them as I find better ways to write macros – or fix any problems which may exist in the macros in their present form, just as I've done in the June 2004 version. You shouldn't see the code used as "limiting" or "exclusive" – other, perhaps better, means exist to do what these macros do.

**Math.wcm** is a number and date calculator which performs various types of computations while running WordPerfect and which can insert calculated results or formula and results into a WordPerfect document. Pictures of some Math.wcm dialogs are shown in Chapter 9.

**ConvertFE.wcm** converts either endnotes or footnotes to plain text at the end of a document, having the appearance of endnotes, but they really aren't. It is useful in preparing a document for a publication of some type in which "real" endnotes/footnotes are not desirable, as in a journal article. The primary dialog in ConvertFE.wcm looks like the picture here. Other dialog illustrations are in **Chapter 5**, above.

**Wp9select.wcm** turns on WordPerfect 9's text selection method if it isn't already on when run in WordPerfect 10 through 12.

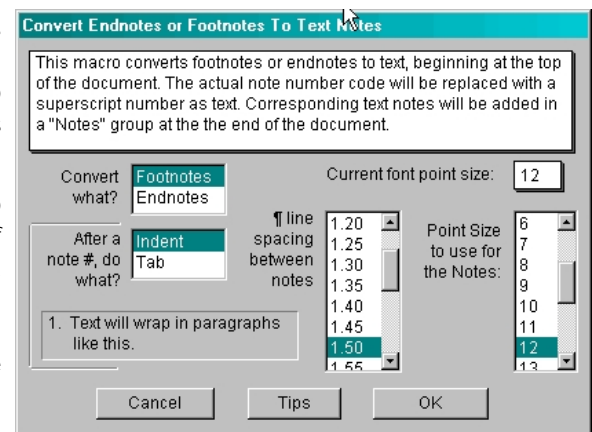
Each macro is downloadable in a self-extracting Winzip.exe file, as follows:

Math.wcm: <http://www.dougloudenback.com/wp/Math.exe>

CovertFE.wcm: <http://www.dougloudenback.com/wp/ConvertFE.exe>

Wp9select.wcm: <http://www.dougloudenback.com/wp/wp9select.exe>

Feel free to use and edit these macros as you want. Please **DO NOT FEEL FREE** to post the macros on any website without my prior written consent or to sell the macros to anyone.



```

1 //MATH.WCM
2 //Math.wcm © 2002-2004 by Doug Loudenback, Oklahoma City, All Rights Reserved
3 //A part of v5.5 and v6.0 Grande Macros, but without the QDRO date calculator element in the lawyer
4 commercial version of the Grande Macros program
5 //This macro may be distributed or edited as you want, but it may not be sold or posted
6 on any website without my express written permission.
7 //Revision Date: June 4, 2004
8 Application (WordPerfect; "WordPerfect"; Default!; "EN") H=NTOC(0F90Ah) OnError(vError)
9 QuickCorrectQuickBulletsSet(Off!) QuickCorrectQuickIndentSet(Off!) //added 6/4/2004
10 vSpecial=0 useSpecial=0 e=0
11 VarErrChk(Off!) // this turns off variable existence error checking
12 Display(On!) // this allows a user to "see" what's typed into a document, when the "Type" button is clicked
13 Today=?DateMonth+"/"+?DateDay+"/"+?DateYear
14 RegionSetWindowText("vMath";"Grande Macros Math & Date Calculator Today: "+Today)
15 vLoop=False DialogShow("vMath";"WordPerfect";cbMath)
16 tempN1="" tempN2="" tempL1=Today tempL2="0" tempC=""
17 tempD1=Today tempD2="0"
18 RegionSetWindowText("vMath.B1";"")
19 RegionShowWindow("vMath.S10";Hide!) RegionShowWindow("vMath.S11";Hide!)
20 RegionShowWindow("vMath.2006Bbtn";Hide!)
21 RegionShowWindow("vMath.sPer";Hide!)
22 vMethod="N" // this sets the initial vMethod as "N" for "Numbers"
23 Call(setMethod) RegionSetWindowText("vMath.S6";"Welcome to GrandeMath! Enter numbers in
24 1 or 2 of the top 2 boxes.") RegionSetWindowText("vMath.G1";"Info Message")
25 Repeat Until(vLoop) DialogDismiss("vMath";1) DialogDestroy("vMath") Quit
26 Label(cbMath) Switch(cbMath[3]) // this callback label does the work of the macro
27 Caseof "B0": // clicking a control causes action; B0 is the primary list (Numbers, Dates, etc.)
28     resetList setMethod
29     If(vCalc<>"C")
30         SwitchLists
31         If(Exists(q))
32             Discard(q)
33         EndIf
34     EndIf
35 Caseof "B3": // control B3 is the middle list box
36     setMethod
37     If(vCalc<>"C")
38         SwitchLists
39         If(Exists(q))
40             Discard(q)
41         EndIf
42     EndIf
43 Caseof "B4": // control B4 is the decimal combo box
44     setMethod
45     If(vCalc<>"C")
46         SwitchLists
47         If(Exists(q))
48             Discard(q)
49         EndIf
50     EndIf
51     RegionSetFocus("vMath.B1")
52 Caseof "OKBbtn": // control OKBbtn is the "Compute" button
53     SwitchLists
54     If(Exists(q))
55         Discard(q)
56     Else

```

```
57     If(vCalc="C")
58         RegionSetFocus("vMath.B2")
59     Else
60         RegionSetFocus("vMath.B1")
61     EndIf
62 EndIf
63 Caseof "TypeBttn": // the "Type" button
64     If(exists(vResult))
65         vType
66     Else
67         RegionSetWindowText("vMath.S6"; "Nothing to type - enter numbers in any needed boxes")
68     EndIf
69     RegionSetFocus("vMath.B1")
70 Caseof "CancelBttn": vLoop=True // the "Stop" button
71 Caseof "HelpBttn": // the "Help" button rev 6/4/2004 to deal with scientific notation, adding a new "help" dialog
72     Switch(vMethod)
73         Caseof "N": x=RegionGetSelectedText("vMath.B3") x=substr(x; 1; 1)
74         Switch(x)
75             Caseof "C": var="vHelp7"
76             Default:
77                 If(vSpecial< 2)
78                     var="vHelp"
79                 Else
80                     var="vHelp8"
81                 EndIf
82             EndSwitch
83         Caseof "D": var="vHelp2"
84         Caseof "L": var="vHelp3"
85     EndSwitch
86     // var (above) is the name of the dialog to open, depending on the value of variable vMethod
87     DialogShow(var; "vMath") RegionSetFocus("vMath.B1")
88     If(vCalc="C")
89         RegionSetFocus("vMath.B2")
90     Else
91         RegionSetFocus("vMath.B1")
92     EndIf
93 Caseof "AboutBttn": // the "About" button
94     DialogShow("vHelp4"; "vMath")
95     If(vCalc="C")
96         RegionSetFocus("vMath.B2")
97     Else RegionSetFocus("vMath.B1")
98     EndIf
99 Caseof "2006Bttn": DialogShow("vHelp6"; "vMath") RegionSetFocus("vMath.B1")
100 Caseof "ClearBttn": // the "Clear" button
101     setMethod vSet
102     If(exists(q))
103         Discard(q)
104     EndIf
105     If(vCalc="C")
106         RegionSetFocus("vMath.B2")
107     Else
108         RegionSetFocus("vMath.B1")
109     EndIf
110 Caseof "FlipBttn": // the "Flip #s" button
111     vReverse SwitchLists
112     If(exists(q))
```

```

113     Discard(q)
114     EndIf
115     RegionSetFocus("vMath.B1")
116 Caseof "PageBttn": // the "New Page" button
117     OnError Call(vError) FileNew OnError
118     If(vCalc="C")
119         RegionSetFocus("vMath.B2")
120     Else
121         RegionSetFocus("vMath.B1")
122     EndIf
123 Caseof "UseBttn1": var=1 setUse // new on 6/4/2004: inserts result in top box
124 Caseof "UseBttn2": var=2 setUse // new on 6/4/2004: inserts result in bottom box
125 EndSwitch Return
126     Label(setMethod) // this routine shows/hides/resets some items when the upper left list box is clicked
127 var=RegionGetSelectedText("vMath.B0") vMethod=SubStr(var;1;1)
128 If(vMethod<>"N") vSpecial=0 EndIf
129 Switch(vMethod) // the user's selection will be Numbers, Dates, or Lawyer dates N, D, L
130     Caseof "N":
131         If(exists(saveT2)) //new on 6/4/2004: uses stored value when switching around between groups
132             RegionSetWindowText("vMath.B2";saveT2) Discard(saveT2) EndIf
133         RegionShowWindow("vMath.S10";Hide!) RegionShowWindow("vMath.S5";Show!)
134         RegionShowWindow("vMath.B4";Show!) RegionShowWindow("vMath.FlipBttn";Show!)
135         RegionShowWindow("vMath.2006Bttn";Hide!)
136         RegionSetWindowText("vMath.S0";"1st number:")
137         RegionSetWindowText("vMath.S1";"2nd number:")
138         Nsel=RegionGetSelectedText("vMath.B3") vCalc=substr(Nsel;1;1)
139         Switch(Nsel)
140             Caseof "Cumul total":
141                 saveT2=tempn2 //new on 6/4/2004: uses stored value when switching around between groups
142                 RegionShowWindow("vMath.Ssel";Hide!)
143                 RegionSetWindowText("vMath.S5";"Round ALL NUMBERS to how many decimals?")
144                 RegionShowWindow("vMath.S3";Hide!)
145                 RegionShowWindow("vMath.UseBttn1";Hide!) // new on 6/4/2004
146                 RegionShowWindow("vMath.UseBttn2";Hide!) // new on 6/4/2004
147                 RegionShowWindow("vMath.SUse";Hide!) // new on 6/4/2004
148                 RegionShowWindow("vMath.B1";Hide!)
149                 RegionShowWindow("vMath.S11";Show!) RegionShowWindow("vMath.S0";Hide!)
150                 RegionShowWindow("vMath.FlipBttn";Hide!)
151                 RegionSetWindowText("vMath.S1";"New number:")
152                 RegionSetWindowText("vMath.B2";"") RegionSetFocus("vMath.B2")
153                 If(exists(Ttxt2))
154                     RegionSetWindowText("vMath.S6";Ttxt2)
155                 Else
156                     RegionSetWindowText("vMath.S6";"Nothing to compute - Select the decimal
157                     desired for this session and then enter a number in the top box")
158                 If(exists(vResult)) Discard(vResult) EndIf
159             EndIf
160         Default:
161             RegionShowWindow("vMath.Ssel";Show!)
162             RegionSetWindowText("vMath.S5";"Round the RESULT to how many decimals?")
163             RegionShowWindow("vMath.S3";Show!)
164             RegionShowWindow("vMath.UseBttn1";Show!) // new on 6/4/2004
165             RegionShowWindow("vMath.UseBttn2";Show!) // new on 6/4/2004
166             RegionShowWindow("vMath.SUse";Show!) // new on 6/4/2004
167             RegionShowWindow("vMath.FlipBttn";Show!)
168             RegionShowWindow("vMath.B1";Show!)

```

```

169         RegionShowWindow("vMath.S11";Hide!) RegionShowWindow("vMath.S0";Show!)
170         RegionSetWindowText("vMath.S0";"1st number:")
171         RegionSetWindowText("vMath.S1";"2nd number:") RegionSetFocus("vMath.B1")
172         RegionSetWindowText("vMath.S6";"")
173         tempN1=RegionGetWindowText("vMath.B1")
174         tempN2=RegionGetWindowText("vMath.B2")
175     EndSwitch
176     If(Nsel="Ratio to total" or Nsel="Percentage of 1st")
177         RegionShowWindow("vMath.sPer";Show!)
178     Else
179         RegionShowWindow("vMath.sPer";Hide!)
180     EndIf
181     Caseof "D": //revised 6/4/2004
182         RegionShowWindow("vMath.sPer";Hide!)
183         RegionShowWindow("vMath.S10";Show!) RegionShowWindow("vMath.S5";Hide!)
184         RegionShowWindow("vMath.2006BttN";Hide!)
185         RegionShowWindow("vMath.UseBttN1";Hide!) // new on 6/4/2004
186         RegionShowWindow("vMath.UseBttN2";Hide!) // new on 6/4/2004
187         RegionShowWindow("vMath.SUse";Hide!) // new on 6/4/2004
188         RegionShowWindow("vMath.B1";Show!)
189         RegionShowWindow("vMath.S11";Hide!) RegionShowWindow("vMath.S0";Show!)
190         RegionShowWindow("vMath.B4";Hide!) RegionShowWindow("vMath.FlipBttN";Hide!)
191         RegionShowWindow("vMath.2006BttN";Hide!)
192         RegionSetWindowText("vMath.S10";"Calculations use the 1st date to the last value and are
193         ""true"" calculations. No adjustment is made for dates ending on weekends. For that, use
194         Lawyer dates.") RegionSetWindowText("vMath.S0";"Date value:")
195         var=RegionGetSelectedText("vMath.B3") var=substr(var;1;1)
196         If(var="D" or var="W" or var="M" or var="Y")
197             RegionSetWindowText("vMath.S1";"Date value:")
198         Else
199             RegionSetWindowText("vMath.S1";"Second value:")
200     EndIf
201     Dsel=RegionGetSelectedText("vMath.B3") x=substr(Dsel;1;1)
202     tempD2=RegionGetWindowText("vMath.B2")
203     y=strpos(tempD2;"/")
204     Switch(x)
205         Caseof "D";"W";"M";"Y":
206             If(y>0)
207                 RegionSetWindowText("vMath.B2";tempD2) Else
208                 RegionSetWindowText("vMath.B2";"")
209             EndIf
210         Default:
211             If(y>0)
212                 RegionSetWindowText("vMath.B2";"0") Else
213                 RegionSetWindowText("vMath.B2";tempD2)
214             EndIf
215     EndSwitch
216     RegionSetFocus("vMath.B2")
217     Caseof "L":
218         RegionShowWindow("vMath.sPer";Hide!)
219         RegionShowWindow("vMath.S10";Show!) RegionShowWindow("vMath.S5";Hide!)
220         tempL1=RegionGetWindowText("vMath.B1")
221         tempL2=RegionGetWindowText("vMath.B2")
222         RegionShowWindow("vMath.2006BttN";Show!)
223         RegionShowWindow("vMath.UseBttN1";Hide!) // new on 6/4/2004
224         RegionShowWindow("vMath.UseBttN2";Hide!) // new on 6/4/2004

```

```

225     RegionShowWindow("vMath.SUse"; Hide!) // new on 6/4/2004
226     RegionShowWindow("vMath.S10"; Show!)
227     RegionShowWindow("vMath.B1"; Show!)
228     RegionShowWindow("vMath.S11"; Hide!) RegionShowWindow("vMath.S0"; Show!)
229     RegionShowWindow("vMath.B4"; Hide!) RegionShowWindow("vMath.FlipBttn"; Hide!)
230     RegionShowWindow("vMath.2006Bttn"; Show!)
231     RegionSetWindowText("vMath.S10"; "The date is day 'zero' for computing purposes. If an
232     end date falls on a weekend or legal holiday, the end date is usually the next business day.
233     12 OS §2006. Weekends are adjusted.") RegionSetWindowText("vMath.S0"; "Date value:")
234     RegionSetWindowText("vMath.S1"; "Number value:")
235     Lsel=RegionGetSelectedText("vMath.B3")
236     EndSwitch
237     Return
238     Label(resetList) // this routine is called when the Numbers/Dates/Lawyer dates list item is clicked and
239     completely resets control B3's list and some other stuff
240     var=RegionGetSelectedText("vMath.B0") var=Substr(var; 1; 1)
241     If(vMethod<>"N") vSpecial=0 EndIf
242     If(vMethod=var) // if vMethod is already the same value, no need to reset the stuff
243     Return
244     EndIf
245
246     RegionResetList("vMath.B3")
247     Switch(var)
248     Caseof "N": // this resets the list box items if "Numbers" is selected in the list box
249     RegionAddListItem("vMath.B3"; "Cumul total")
250     RegionAddListItem("vMath.B3"; "Add to 1st")
251     RegionAddListItem("vMath.B3"; "Subtract from 1st")
252     RegionAddListItem("vMath.B3"; "Multiply by 1st")
253     RegionAddListItem("vMath.B3"; "Divide into 1st")
254     RegionAddListItem("vMath.B3"; "Percentage of 1st")
255     RegionAddListItem("vMath.B3"; "Ratio to total")
256     RegionSelectListItem("vMath.B3"; Nsel)
257     Switch(Nsel)
258     Caseof "Cumul total": RegionSetWindowText("vMath.B2"; "")
259     If(exists(Ttxt2))
260     RegionSetWindowText("vMath.S6"; Ttxt2)
261     Else
262     RegionSetWindowText("vMath.S6"; "")
263     EndIf
264     RegionSelectListItem("vMath.B3"; "Cumul total")
265     RegionSetWindowText("vMath.B2"; "")
266     Default:
267     If(Exists(Nsel))
268     var=Nsel Else var= "Add to 1st"
269     EndIf
270     RegionSelectListItem("vMath.B3"; Nsel)
271     RegionSetWindowText("vMath.B1"; tempn1)
272     RegionSetWindowText("vMath.B2"; tempn2)
273     EndSwitch
274     RegionShowWindow("vMath.S5"; Show!) RegionShowWindow("vMath.B4"; Show!)
275     RegionShowWindow("vMath.S5"; Hide!)
276     RegionShowWindow("vMath.Ssel"; Show!) RegionShowWindow("vMath.B3"; Show!)
277     RegionShowWindow("vMath.B4"; Show!) RegionShowWindow("vMath.S10"; Show!)
278     RegionShowWindow("vMath.2006Bttn"; Hide!)
279     RegionShowWindow("vMath.B1"; Show!) RegionShowWindow("vMath.S11"; Hide!)
280     Caseof "D": // this resets the list box items if "Dates" is selected in the list box

```

```
281     RegionShowWindow("vMath.B1"; Show!)
282     RegionShowWindow("vMath.S11"; Hide!) RegionShowWindow("vMath.S0"; Show!)
283     RegionSetWindowText("vMath.B1"; tempd1)
284     RegionSetWindowText("vMath.B2"; tempd2)
285     RegionShowWindow("vMath.B3"; Show!)
286     RegionAddListItem("vMath.B3"; "Days between")
287     RegionAddListItem("vMath.B3"; "Weeks between")
288     RegionAddListItem("vMath.B3"; "Months between")
289     RegionAddListItem("vMath.B3"; "Years between")
290     RegionAddListItem("vMath.B3"; "Add days")
291     RegionAddListItem("vMath.B3"; "Add weeks")
292     RegionAddListItem("vMath.B3"; "Add months")
293     RegionAddListItem("vMath.B3"; "Add years")
294     RegionAddListItem("vMath.B3"; "Subtract days")
295     RegionAddListItem("vMath.B3"; "Subtract weeks")
296     RegionAddListItem("vMath.B3"; "Subtract months")
297     RegionAddListItem("vMath.B3"; "Subtract years")
298     If(Exists(Dsel))
299         var=Dsel
300     else
301         var="Add days"
302     EndIf
303     RegionSelectListItem("vMath.B3"; var)
304     RegionShowWindow("vMath.S5"; Hide!) RegionShowWindow("vMath.B4"; Hide!)
305     RegionShowWindow("vMath.2006Bbtn"; Hide!)
306     Caseof "L": // this resets the list box items if "Lawyer dates" is selected in the list box
307         RegionShowWindow("vMath.UseBbtn1"; Hide!) // new on 6/4/2004
308         RegionShowWindow("vMath.UseBbtn2"; Hide!) // new on 6/4/2004
309         RegionShowWindow("vMath.SUse"; Hide!) // new on 6/4/2004
310         RegionShowWindow("vMath.B1"; Show!)
311         RegionShowWindow("vMath.S11"; Hide!)
312         RegionShowWindow("vMath.S0"; Show!)
313         RegionShowWindow("vMath.2006Bbtn"; Show!)
314         RegionShowWindow("vMath.B1"; Show!)
315         RegionShowWindow("vMath.S11"; Hide!) RegionShowWindow("vMath.S0"; Show!)
316         RegionSetWindowText("vMath.B1"; tempL1)
317         RegionSetWindowText("vMath.B2"; tempL2)
318         RegionShowWindow("vMath.S5"; Hide!) RegionShowWindow("vMath.B4"; Hide!)
319         RegionAddListItem("vMath.B3"; "Add days")
320         RegionAddListItem("vMath.B3"; "Add weeks")
321         RegionAddListItem("vMath.B3"; "Add months")
322         RegionAddListItem("vMath.B3"; "Add years")
323     If(Exists(Lsel))
324         var=Lsel
325     Else
326         var="Add days"
327     EndIf
328     RegionSelectListItem("vMath.B3"; var)
329     RegionShowWindow("vMath.S5"; Hide!)
330     RegionShowWindow("vMath.S3"; Show!) RegionShowWindow("vMath.Ssel"; Show!)
331     RegionShowWindow("vMath.B3"; Show!) RegionShowWindow("vMath.B4"; Hide!)
332     RegionShowWindow("vMath.S10"; Hide!) RegionShowWindow("vMath.2006Bbtn"; Show!)
333 EndSwitch
334 Return
335     Label(SwitchLists) // this tells the macro what to "Call" - the shorthand method for Call is used
336 Switch(vMethod)
```

```

337 Caseof "N": DoMath
338 Caseof "D": DoDates
339 Caseof "L": DoLawyer
340 EndSwitch
341 Return
342 Label(setUse) // new routine on 6/4/2004: inserts the result into the top or bottom math box, if a result exists
343 If(exists(vResult))
344   x=StrPos(vResult,"%")
345   If(x>0)
346     tempn=StrTransform(vResult,"%";"") // new on 6/4/2004: this removes the % symbol, if present
347     tempn=StrTransform(tempn;",";"") // new on 6/4/2004: if ",", symbol is present, removes commas
348     tempn=.01*tempn // new on 6/4/2004: this converts % values to regular numeric values
349   EndIf
350   Switch(var)
351     Caseof 1:
352       If(x=0)
353         RegionSetWindowText("vMath.B1";vResult) tempn1=vResult
354       Else
355         RegionSetWindowText("vMath.B1";tempn) tempn1=tempn
356       EndIf
357     Caseof 2:
358       If(x=0)
359         RegionSetWindowText("vMath.B2";vResult) tempn2=vResult
360       Else
361         RegionSetWindowText("vMath.B2";tempn) tempn2=tempn
362       EndIf
363   EndSwitch
364   Else DialogShow("NoResult";"vMath") DialogDestroy("NoResult")
365 EndIf
366 Return
367 Label(setResult)
368 If(exists(q))
369   RegionSetWindowText("vMath.G1";"Error / Info Message")
370   Else
371     If(vSpecial>1) // revised 6/4/2004 to deal with scientific notation results or digits >= 15
372       RegionSetWindowText("vMath.G1";"APPROXIMATE Result * * * Click Help for more detail")
373     Else
374       RegionSetWindowText("vMath.G1";"Result")
375   EndIf
376 EndIf
377 Return
378 Label(doMath) // the called routines here are for "numbers", not the "dates" or "lawyer dates"
379 vFinal=0 e=0
380 If(NSel<>"Cumul total")
381   num1=RegionGetWindowText("vMath.B1")
382 Else num1=saveResult
383 EndIf
384 num2=RegionGetWindowText("vMath.B2")
385 varD=RegionGetSelectedText("vMath.B4") varD=StrNum(varD)
386 vCalc=RegionGetSelectedText("vMath.B3") Nsel=vCalc vCalc=Substr(vCalc;1;1)
387 If(num1="" and num2="")
388   q=1
389   RegionSetWindowText("vMath.S6";"Nothing to compute - enter numbers in any needed
390   boxes")
391   If(vCalc="C")
392     RegionSetFocus("vMath.B2")

```



```
393     else
394     RegionSetFocus("vMath.B1")
395 EndIf
396 setResult Return
397 EndIf
398 If(num1="")
399     num1="0"
400 EndIf
401 var=num1 Call(chkNum)
402 If(exists(q))
403     RegionSetWindowText("vMath.S6";"1st number, "+var+", is not a number. Fix it.")
404     RegionSetFocus("vMath.B1")
405     RegionSetEditSelection("vMath.B1") setResult Return
406 EndIf
407 var=num1
408 var=StrTransform(var;";";"") // replacement routine for getting rid of commas
409 chkMax // error routine new 6/4/2004
410 If(exists(q))
411     RegionSetWindowText("vMath.S6";"1st number, "+var+", is too long. Grande Math cannot
412     work with numbers exceeding 15 digits - unreliable results may occur. Use a different
413     number.") RegionSetFocus("vMath.B1") RegionSetEditSelection("vMath.B1") Return
414 EndIf
415 num1=var // num1 is now a legal raw number, w/o commas
416 Call(NewNum) //var will now have commas; num1 is original var, without commas
417 If(Nsel<>"Cumul total")
418     FirstN=var tempn1=FirstN //FirstN is declared, with commas; FirstN is used when typing
419 EndIf
420 If(num2="")
421     num2="0"
422 EndIf
423 var=num2 Call(chkNum)
424 If(exists(q))
425     RegionSetWindowText("vMath.S6";"2nd number, "+var+", is not a number. Fix it.")
426     RegionSetFocus("vMath.B2")
427     RegionSetEditSelection("vMath.B2") setResult
428     Return
429 EndIf
430 var=num2
431 var=StrTransform(var;";";"") // replaces previous routine to get rid of commas
432 chkMax // error routine new 6/4/2004
433 If(exists(q))
434     RegionSetWindowText("vMath.S6";"2nd number, "+var+", is too long. Grande Math cannot
435     work with numbers exceeding 15 digits - unreliable results may occur. Use a different
436     number.") RegionSetFocus("vMath.B2") RegionSetEditSelection("vMath.B2")
437     Return
438 EndIf
439 num2=var
440 Call(NewNum) //new in 6/4/2004; corrects display format
441 If(Nsel<>"Cumul total")
442     SecondN=var tempn2=var
443 EndIf
444 num2=StrNum(num2)
445 UsePer=0 // this variable declares the default intention of not using percentage signs when typing in doc
446 Switch(vCalc)
447     Caseof "A": var=num1+num2 // simple addition
448     Caseof "C": // this does the cumulative total using workarounds for floating cell errors
```

```

449     If(exists(T[]))
450         Else Declare(T[1000])
451     EndIf
452     If(varD=0)
453         num2= Integer(roundoff(num2; 1/(10**varD))) Else
454         num2=numstr(num2) x=strPos(num2; ".") If(x>0) num2=num2+"000001"
455     EndIf
456     num2=strnum(num2)
457     num2 = RoundOff (num2; (1/(10**varD))) EndIf
458     var=num2 Call(newNum) SecondN=var
459     If(exists(yResult))
460         var=num2+yresult yresult=var
461     Else var=num2 yResult=num2
462     EndIf
463     RealTotal=var
464     Caseof "S": var=num1-num2 // this does simple subtraction
465     Caseof "M": var=num1*num2 // this is simple multiplication
466     Caseof "D": // this is ordinary division
467         If(num2=0)
468             q=1
469             RegionSetWindowText("vMath.S6"; "2nd number is 0. Division by 0 cannot be done.")
470             RegionSetFocus("vMath.B2") setResult
471             Return
472         EndIf
473         var=num1/num2
474     Caseof "P": // this is division, but, with the eventually called Label (not here), will reset values to % values
475         If(num1=0)
476             q=1
477             RegionSetWindowText("vMath.S6"; "1st number is 0. Division by 0 cannot be done.")
478             RegionSetFocus("vMath.B1") setResult
479             Return
480         EndIf
481         var=num2/num1 UsePer=1
482     Caseof "R": // essentially the same as the above, but uses different values for computation
483         If(num1+num2=0)
484             q=1
485             RegionSetWindowText("vMath.S6"; "The total is 0. Division by 0 cannot be done.")
486             RegionSetFocus("vMath.B1") setResult
487             Return
488         EndIf
489         var=num2/(num1+num2) UsePer=1
490     EndSwitch
491     vSpecial=0 // added routines 6/4/2004 to deal with scientific notation and other >=15 digit issues
492     x=strpos(var; "-")
493     If(x>0)
494         var=var*-1 negnum=1
495     else negnum=0
496     EndIf
497     If(var<1) // for results with values less than 1
498         vSpecial=1
499     EndIf
500     x=strpos(var; "e") If(x>0) e=1 vSpecial=2 EndIf // for results that are in scientific notation
501
502     x=var
503     If(vSpecial=0)
504         tVar=numstr(x; 0) xvar=numstr(x)

```

```

505   If(0+tvar=xVar) var=tVar EndIf // eliminates any decimal and trailing zeros & var becomes string
506   x=strpos(var;".")
507   If(x>0)
508       vMainL=substr(var;1;x-1) vMainL=strlen(vMainL)
509       vRemL=substr(var;x+1;strlen(var)) vRemL=strlen(vRemL) vAll=vMainL+vAll
510   Else
511       vRemL=0
512       vMainL=substr(var;1;strlen(var)) vAll=vMainL vRemL=0
513   EndIf
514   num1F=FractionalPart(num1) num2F=FractionalPart(num2)
515   vAll=StrTransform(var;".";"") vAll=strlen(vAll)
516   rMax=15-(vAll) // rMax is the maximum rounding capacity
517   If(Num1F>0 or Num2F>0) NumF=1 Else NumF=0 EndIf // testing for either's fractional value
518 EndIf
519 //... Identifying the 3's after this point ... otherwise, all others will be vSpeical will be 0
520 If(vAll=15 and NumF=1)
521     If(varD>vRemL)
522         vSpecial=3 e=1
523     EndIf
524 EndIf
525 If(vSpecial=0)
526     If(varD>rMax)
527         If(vRemL<varD)
528             If(vAll<15)
529                 Else
530                     If(vRemL<>0)
531                         vSpecial=3
532                     EndIf
533                 EndIf
534             EndIf
535         EndIf
536     EndIf
537 EndIf
538 // new & revised 6/4/2004: replaces earlier version's RoundOff method: deals w/ floating cell & scientific notation issues
539 Switch(vSpecial)
540     Caseof 0:
541         If(negnum=1) // modified 6/4/2004: negnum set above
542             RealTotal="-"+var // RealTotal added 6/4/2004
543         Else
544             RealTotal=var
545         EndIf
546     Switch(varD)
547         Caseof 0: If(usePer=0) var=RoundOff(var;1) else var=RoundOff(var;0.01) EndIf
548         Caseof 1:
549             If(usePer=0) var=RoundOff(var;0.01) else var=RoundOff(var;0.001) EndIf
550             var=RoundOff(var;0.1)
551         Default:
552             vPoints=varD
553             x="0."
554             z=1
555             Repeat
556                 x=x+"0"
557                 z=z+1
558             Until (z=vPoints)
559             x=x+"1"
560             x=strnum(x)

```

```
561         var=RoundOff(var;x)
562     EndSwitch
563     If(negnum=1)
564         var="-"+var
565     EndIf
566     Caseof 1: // for results less than 1, greater than 0; negnum was set above
567         If(negnum=1)
568             RealTotal="-"+var // RealTotal added 6/4/2004
569         Else
570             RealTotal=var
571         EndIf
572         var=numstr(var;varD) var=strnum(var)
573     Caseof 2: // for results containing scientific notation
574         x=strpos(var;"e")
575         y=substr(var;x+2;3) y=strnum(y) y=y+1
576         var=substr(var;1;x-1)
577         var=var-"."
578         varq=strlen(var)
579         If(varq<y)
580             Repeat
581                 var=var+"0"
582             Until (strlen(var)=y)
583         EndIf
584         x=strpos(var;".")
585         If(x=0)
586             var=var-"."
587         EndIf
588         If(negnum=1)
589             RealTotal="-"+var // RealTotal added 6/4/2004
590         Else
591             RealTotal=var
592         EndIf
593     Caseof 3: // for results greater than 1; negnum was set above
594         If(negnum=1) RealTotal="-"+var
595         Else
596             RealTotal=var
597         EndIf
598 EndSwitch
599 If(vSpecial<>2)
600     var=strnum(var)
601     tVar=numstr(var;0) xvar=numstr(var)
602     If(0+tvar=xVar)
603         var=tVar
604     EndIf // added 6/4/2004; eliminates trailing zeros in displayed result for non-scientific notation results
605 EndIf
606 NewNum // this converts the result into a string
607 If(UsePer=1)
608     Call(MakePer)
609 EndIf
610 vResult=var
611 If(vSpecial=2 or vSpecial=3)
612     x=0
613     negnum=strpos(var;"-")
614     If(negnum>0)
615         x=x+1
616     EndIf
```

```
617 y=strpos(var; ".")
618 If(y>0)
619     x=x+1
620 EndIf
621 v15=substr(RealTotal; 1; 15+x)
622 EndIf
623 If(vCalc="C")
624     If(exists(vN))
625         vN=vN+1 Else vN=1
626     EndIf
627     T[vN] = SecondN
628     saveResult=vResult
629     RegionSetWindowText("vMath.B2"; "")
630     RegionSetFocus("vMath.B2")
631     If(exists(Ttxt))
632         Ttxt=Ttxt+ " + "+SecondN Else
633         Ttxt=vResult
634     EndIf
635     If(Ttxt=vResult)
636         Ttxt2=vResult Else
637         Ttxt2=Ttxt + " = "+vresult
638     EndIf
639     RegionSetWindowText("vMath.S6"; Ttxt2)
640     Else
641     RegionSetWindowText("vMath.B1"; FirstN)
642     RegionSetWindowText("vMath.B2"; SecondN)
643     RegionSetWindowText("vMath.S6"; vResult)
644 EndIf
645 setResult //resets info box top line
646 Return
647 Label(doDates) // this routine is called for "Dates", not "Numbers" or "Lawyer dates"
648 var=RegionGetWindowText("vMath.B1") tempD1=var
649 var2=RegionGetWindowText("vMath.B2") tempD2=var2
650 If(var="")
651     q=1
652     RegionSetWindowText("vMath.S6"; "Nothing to compute - enter date in top box")
653     RegionSetEditSelection("vMath.B1") setResult Return
654 EndIf
655 x=substr(dsel; 1; 1)
656 Switch(x)
657     Caseof "D"; "W"; "Y"; "M":
658         If(tempD2="")
659             RegionSetWindowText("vMath.S6"; "Nothing to compute - enter date in second box")
660             RegionSetEditSelection("vMath.B2") setResult Return
661         EndIf
662     Default:
663 EndSwitch
664 Call(dateCnv)
665 If(Exists(q))
666     RegionSetWindowText("vMath.S6"; vMsg)
667     RegionSetFocus("vMath.B2")
668     RegionSetEditSelection("vMath.B2") setResult Return
669 EndIf
670 FirstN=var tempd1=mo+ "/" + day+ "/" + year
671 var=RegionGetSelectedText("vMath.B3") Dsel=var vCalc=substr(dSel; 1; 1)
672 Switch(var)
```

```
673 Caseof "Days between": vCalc=1 getDays
674     If(Exists(q))
675         Return
676     EndIf
677     vBetween
678     var=days Call(NewNum) vDays=var vResult=vDays+ " days"
679 Caseof "Weeks between": vCalc=2 getDays
680     If(Exists(q))
681         Return
682     EndIf
683     vBetween
684     Repeat
685         saveback=t
686         w=DateAddWeeks(saveback;0)
687         x=DateAddWeeks(t;1)
688         y=Integer(ConvertType(x;Float!)) t=y z=z+1
689     Until (y>=s)
690     If(y>s) // if so, we want to go back 1 mo and count days from that to end of 2nd date
691         z=z-1 // and this subtracts 1 week from total years if y's value exceeds s's value
692         x=Integer(ConvertType(saveback;Float!)) vDays=s-x
693     Else vDays=0
694     EndIf
695     If(vDays=1)
696         dword="day" Else
697         dword="days"
698     EndIf
699     var=z Call(NewNum) vWeeks=var
700     If(vWeeks=1)
701         wword="week" else wword="weeks"
702     EndIf
703     vResult=vWeeks+ " "+wword+ ", "+vDays+ " "+dword
704 Caseof "Months between": vCalc=3 getDays
705     If(Exists(q))
706         Return
707     EndIf
708     Call(vBetween)
709     Repeat
710         saveback=t
711         w=DateAddMonths(saveback;0)
712         x=DateAddMonths(t;1)
713         y=Integer(ConvertType(x;Float!)) t=y z=z+1
714     Until (y>=s)
715     vMonths=z
716     If(y>s) // if so, we want to go back 1 mo and count days from that to end of 2nd date
717         vMonths=z-1 // and this subtracts 1 month from total months in y's value
718         x=Integer(ConvertType(saveback;Float!)) vDays=s-x
719     Else vDays=0
720     EndIf
721     If(vDays=1)
722         dword="day" else dword="days"
723     EndIf
724     If(vMonth=1)
725         mword="month" else mword="months"
726     EndIf
727     var=vMonths Call(NewNum) vMonths=var
728     vResult=vMonths+ " "+mword+ ", "+vDays+ " "+dword
```

```
729 Caseof "Years between": vCalc=4 getDays
730   If(Exists(q))
731     Return
732   EndIf
733   If(d=s)
734     vYears=0 vMonths=0 vDays=0 Else
735     Call(vBetween)
736     Repeat
737       saveback=t
738       w=DateAddYears(saveback;0)
739       x=DateAddYears(t;1)
740       y=Integer(ConvertType(x;Float!)) t=y z=z+1
741     Until (y>=s)
742     vYears=z
743     If(y=s)
744       vMonths=0 vDays=0
745     Else
746       z=0
747       vYears=vYears-1
748       t=Integer(ConvertType(saveback;Float!))
749       Repeat
750         saveback=t
751         w=DateAddMonths(saveback;0)
752         x=DateAddMonths(t;1)
753         y=Integer(ConvertType(x;Float!)) t=y z=z+1
754       Until (y>s)
755       If(y>s)
756         vMonths=z-1
757         x=Integer(ConvertType(saveback;Float!))
758         vDays=s-x
759       Else vDays=0
760     EndIf
761   EndIf
762 EndIf
763 If(vYears=1)
764   yword="year" else yword="years"
765 EndIf
766 If(vMonths=1)
767   mword="month" else mword="months"
768 EndIf
769 If(vDays=1)
770   dword="day" else dword="days"
771 EndIf
772 vResult=vYears+" "+yword+", "+vMonths+" "+mword+", "+vDays+" "+dword
773 Caseof "Add days": vCalc=5 getV
774   If(Exists(q))
775     Return
776   EndIf
777   d=DateAddDays(DateAndTime (Day;Mo;Year;0;0;0;0); var) set6
778 Caseof "Add weeks": vCalc=6 getV
779   If(Exists(q))
780     Return
781   EndIf
782   d=DateAddWeeks(DateAndTime (Day;Mo;Year;0;0;0;0); var) set6
783 Caseof "Add months": vCalc=7 getV
784   If(Exists(q))
```

```
785     Return
786     EndIf
787     d=DateAddMonths(DateAndTime (Day;Mo;Year;0;0;0;0); var) set6
788     Caseof "Add years": vCalc=8 getV
789     If(exists(q))
790         Return
791     EndIf
792     d=DateAddYears(DateAndTime (Day;Mo;Year;0;0;0;0); var) set6
793     Caseof "Subtract days": vCalc=9 getV
794     If(exists(q))
795         Return
796     EndIf
797     d=DateAddDays(DateAndTime (Day;Mo;Year;0;0;0;0); -var) set6
798     Caseof "Subtract weeks": vCalc=10 getV
799     If(exists(q))
800         Return
801     EndIf
802     d=DateAddWeeks(DateAndTime (Day;Mo;Year;0;0;0;0); -var) set6
803     Caseof "Subtract months": vCalc=11 getV
804     If(exists(q))
805         Return
806     EndIf
807     // DateAddMonths is broken in Wp9, 10 and 11 insofar as subtracting months if the year would be earlier than the year
808     containing the date value; this work-around code draws from suggestions by Ken Hobson in a WordPerfect Universe
809     thread on 9/17/2002; use the following structure when subtracting months, where "var" is the number of months
810     x=DateAndTime(Day;Mo;Year;0;0;0;0)
811     yMonths= var Mod 12 yYears= (var-yMonths) /12
812     d=DateAddYears(DateAddMonths(x; 12-yMonths); -1-yYears) set6
813     Caseof "Subtract years": vCalc=12 getV
814     If(exists(q))
815         Return
816     EndIf
817     d=DateAddYears(DateAndTime (Day;Mo;Year;0;0;0;0); -var) set6
818 EndSwitch
819 RegionSetWindowText("vMath.S6";vResult) setResult
820 Return
821 Label(doLawyer) // this routine is called for "Lawyer dates", not "Numbers" or "Dates"
822 var=RegionGetWindowText("vMath.B1") templ1=var
823 var2=RegionGetWindowText("vMath.B2") templ2=var2
824 If(var="")
825     var="0"
826 EndIf
827 If(var2="")
828     var2="0" RegionSetFocus("vMath.B2")
829 EndIf
830 If(var="0")
831     q=1
832     RegionSetWindowText("vMath.S6";"Nothing to compute - enter values in any needed boxes")
833     RegionSetFocus("vMath.B1") RegionSetEditSelection("vMath.B1") setResult Return
834 EndIf
835 Call(dateCnv)
836 If(Exists(q))
837     RegionSetWindowText("vMath.S6";vMsg) RegionSetFocus("vMath.B1")
838     RegionSetEditSelection("vMath.B1")
839     setResult Return
840 EndIf
```



```
841 FirstN=var tempL1=mo+ "/" +day+ "/" +year
842 var=RegionGetSelectedText("vMath.B3") Lsel=var
843 Switch(var)
844   Caseof "Add days": vCalc=1 getV
845     If(exists(q))
846       Return
847     EndIf
848     d=DateAddDays(DateAndTime (Day;Mo;Year;0;0;0;0); var) set6
849   Caseof "Add weeks": vCalc=2 getV
850     If(exists(q))
851       Return
852     EndIf
853     d=DateAddWeeks(DateAndTime (Day;Mo;Year;0;0;0;0); var) set6
854   Caseof "Add months": vCalc=3 getV
855     If(exists(q))
856       Return
857     EndIf
858     d=DateAddMonths(DateAndTime (Day;Mo;Year;0;0;0;0); var) set6
859   Caseof "Add years": vCalc=4 getV
860     If(exists(q))
861       Return
862     EndIf
863     d=DateAddYears(DateAndTime (Day;Mo;Year;0;0;0;0); var) set6
864   EndSwitch
865 RegionSetWindowText("vMath.S6";vResult) setResult
866 Return
867   Label(vBetween) // this called routine converts m/d/yyyy dates into serial date values
868   d=DateAndTime (day1; mo1;year1;0;0;0;0) d=Integer(ConvertType(d;Float!))
869   s=DateAndTime (day2;mo2;year2;0;0;0;0) s=Integer(ConvertType(s;Float!))
870   If(d>s)
871     x=d y=s s=x d=y day1=day2 mo1=mo2 year1=year2
872   EndIf
873   days=Integer(s-d)
874   t=d z=0 // z is number of months and t becomes d, and d remains unchanged
875   Return
876   Label(getDays) // this routine is called from Label(doDates): sets text appearing in the Results box, Region S6
877   mo1=mo day1=day year1=year
878   var=RegionGetWindowText("vMath.B2")
879   If(var="")
880     q=1
881     RegionSetWindowText("vMath.S6";"Nothing to compute - enter values in any needed boxes")
882     RegionSetFocus ("vMath.B1") RegionSetEditSelection("vMath.B1") setResult Return
883   EndIf
884   Call(dateCnv) SecondN=var
885   If(exists(q))
886     RegionSetWindowText("vMath.S6";vMsg) RegionSetFocus("vMath.B2")
887     RegionSetEditSelection("vMath.B2") Return
888   EndIf
889   day2=day mo2=mo year2=year
890   Return
891   Label(getV) // this is part of the date routines and can set text in the Results box, Region S6
892   var=RegionGetWindowText("vMath.B2")
893   If(var="")
894     var="0"
895   EndIf
896   chkNum
```

```
897 If(exists(q))
898   RegionSetWindowText("vMath.S6";"2nd value, "+var+", is not legal")
899   RegionSetFocus ("vMath.B2") RegionSetEditSelection("vMath.B2") Return
900 EndIf
901 If(NegNum = 1)
902   q= 1
903   RegionSetWindowText("vMath.S6";"2nd value, "+var+", is not legal. Use Subtract in regular
904   date list instead.")
905   Discard(NegNum)
906   RegionSetFocus ("vMath.B2") RegionSetEditSelection("vMath.B2") setResult Return
907 EndIf
908 var=StrNum(var)
909 SecondN=tempD2
910 Return
911   Label(set6) // this is part of the lawyer date routines in case the date falls on a weekend
912 If(vMethod="L")
913   x=DateWeekdayName(d;Short!)
914   If (x="Sat")
915     d=d+2
916   EndIf
917   If(x="Sun")
918     d=d+1
919   EndIf
920 EndIf
921 d=DateString(d; "dddd, MMMM d, yyyy") vResult=d Return
922   Label(vReverse) // this flip-flops the values in numbers 1 and 2
923 Num1=RegionGetWindowText("vMath.B1") Num2=RegionGetWindowText("vMath.B2")
924 RegionSetWindowText("vMath.B1";Num2) RegionSetWindowText("vMath.B2";Num1)
925 Discard(num1;num2) Return
926   Label(vSet) // this routine clears values in numbers 1, 2 and the results box
927 RegionSetWindowText("vMath.B1";"") RegionSetWindowText("vMath.B2";"")
928 If(exists(T[]))
929   While (exists(t[]))
930     Discard(t[])
931   EndWhile
932 EndIf
933 If(exists(vN))
934   Discard(vN)
935 EndIf
936 If(exists(vResult))
937   Discard(vResult)
938 EndIf
939 If(exists(yResult))
940   Discard(yResult)
941 EndIf
942 If(exists(Ttxt))
943   Discard(Ttxt)
944 EndIf
945 If(exists(Ttxt2))
946   Discard(Ttxt2)
947 EndIf
948 RegionSetWindowText("vMath.S6";"Nothing to compute - enter values in any needed boxes")
949 RegionSetFocus("vMath.B1") RegionSetEditSelection("vMath.B1") setResult Return
950   Label(vWrite) // this Label is called from Label(vType), above. It insures that decimal values are handled well
951 w=strPos(var;".")
952 Switch(w)
```

```

953 Caseof 0:
954     If(varD<>0)
955         var=var+ "." y=0
956         Repeat
957             var=var+"0" y=y+1
958         Until(y=varD)
959     EndIf
960 Default: y=substr(var;w+1;strlen(var)) z=strlen(y)
961     If(z<varD)
962         Repeat
963             var=var+"0" z=z+1
964         Until(z=varD)
965     EndIf
966 EndSwitch
967 Return
968 Label(eText) // new 6/4/2004: for results which involve scientific notation
969 Type("(after the 15th digit, the result is approximate due to computer chip limitations)") Return
970 Label(TypeM1) //new on 6/4/2004 merely to avoid typing redundancy
971 Type(t1+" "+mchar+" "+t2+" = "+vResult+" [rounded to "+varD+" "+dWord+"]") Return
972 Label(TypeDw) //new on 6/4/2004 merely to avoid typing redundancy
973 Type(Dw+" between "+SecondN+", and "+FirstN+" = "+vResult) Return
974 Label(TypeDa) //new on 6/4/2004 merely to avoid typing redundancy
975 Type(FirstN+Do+SecondN+" "+Dw+plural+" = "+vResult) Return
976 Label(vTypeE) //new on 6/4/2004: types text depending on var e
977 If(e=1)
978     Type("(actual computed result up to 15 digits)")
979     Else
980     Type("(actual computed result)")
981 EndIf
982 HardReturn
983 Return
984 Label(vTypeD) //new on 6/4/2004: part of number column typing
985 Type("=") dTabs Type(vResult) Tab Type("(rounded to "+varD+" "+dword+")") Return
986 Label(vType) // this routine types stuff into the open document! modified to add alternative on 6/4/2004
987 RegionResetList("vType.B1")
988 Switch(vCalc)
989     Caseof "C":
990         RegionAddListItem("vType.B1";"Result only")
991         RegionAddListItem("vType.B1";"Formula and result in a column")
992         If(exists(TypeWhat))
993             Switch(TypeWhat)
994                 Caseof "All the above and sample text"; "Formula and result";"Formula and result on
995                     same line":
996                     TypeWhat="Formula and result in a column"
997             EndSwitch
998         EndIf
999     Caseof "A";"S";"M";"D";"P";"R":
1000         RegionAddListItem("vType.B1";"Result only")
1001         RegionAddListItem("vType.B1";"Formula and result in a column")
1002         RegionAddListItem("vType.B1";"Formula and result on same line")
1003         If(exists(TypeWhat))
1004             Switch(TypeWhat)
1005                 Caseof "All the above and sample text"; "Formula and result":
1006                     TypeWhat="Formula and result in a column"
1007             EndSwitch
1008         EndIf

```

```
1009 Default:
1010 RegionAddListItem("vType.B1";"Result only")
1011 RegionAddListItem("vType.B1";"Formula and result on same line")
1012 If(exists(Whatever) and (Whatever="Result only" or Whatever="Formula and result on
1013 same line")) Else
1014     Whatever="Result only"
1015 EndIf
1016 EndSwitch
1017 If(exists(Whatever))
1018     RegionSelectListItem("vType.B1";Whatever) Else
1019     RegionSelectListItem("vType.B1";"Result only")
1020 EndIf
1021 DialogShow("vType";"WordPerfect") x=MacroDialogResult
1022 If(x=2)
1023     Return
1024 EndIf
1025 TypeMe=Strlen(Whatever) //substantially modified to add additional typing option on 6/4/2004
1026 x=?DocBlank
1027 If(x=false)
1028     PosLineEnd HardReturn
1029 EndIf
1030 If(vMethod="N" and varD=1)
1031     dWord="decimal"
1032 Else
1033     dWord="decimals"
1034 EndIf
1035 If(TypeMe=11) // substantially modified: result only: modified on 6/4/2004
1036     Switch(vMethod)
1037     Caseof "N":
1038         If(vCalc="C")
1039             var=yresult
1040             Type(var) RegionSetFocus("vMath.B2")
1041             Else Type(vResult) RegionSetFocus("vMath.B1")
1042         EndIf
1043     Default: Type(vResult)
1044     EndSwitch
1045     If(vSpecial>1)
1046         eText
1047     EndIf
1048 EndIf
1049 If(TypeMe=31) // substantially modified: formula & result on same line: modified 6/4/2004
1050     Switch(vMethod)
1051     Caseof "N": // what to type if numbers are being computed
1052         Switch (vCalc) // routine modified 6/4/2004 to avoid typing redundancy
1053         Caseof "A": mChar="+" t1=FirstN t2=SecondN TypeM1
1054         Caseof "S": mChar="-" t1=FirstN t2=SecondN TypeM1
1055         Caseof "M": mChar="x" t1=FirstN t2=SecondN TypeM1
1056         Caseof "D": mChar="÷" t1=FirstN t2=SecondN TypeM1
1057         Caseof "P": mChar="÷" t1=SecondN t2=FirstN TypeM1
1058         Caseof "R": Type(SecondN+" ÷ (" +FirstN+" + " +SecondN+")) = "+vResult+"
1059         [rounded to "+varD+" "+dWord+"]")
1060         EndSwitch
1061     Caseof "D": // what to type if regular dates are being calculated
1062         If(SecondN=1)
1063             plural="" Else plural="s"
1064         EndIf
```

```

1065     Switch(vCalc) // routine modified 6/4/2004 to avoid typing reduncancy
1066         Caseof 1: Dw="Days" TypeDw
1067         Caseof 2: Dw="Weeks" TypeDw
1068         Caseof 3: Dw="Months" TypeDw
1069         Caseof 4: Dw="Years" TypeDw
1070         Caseof 5: Dw="day" Do=" + " TypeDa
1071         Caseof 6: Dw="week" Do=" + " TypeDa
1072         Caseof 7: Dw="month" Do=" + " TypeDa
1073         Caseof 8: Dw="year" Do=" + " TypeDa
1074         Caseof 9: Dw="day" Do=" " TypeDa
1075         Caseof 10: Dw="week" Do=" " TypeDa
1076         Caseof 11: Dw="month" Do=" " TypeDa
1077         Caseof 12: Dw="year" Do=" " TypeDa
1078     EndSwitch
1079     Caseof "L": // what to type if Lawyer dates are being calculated
1080     If(SecondN=1)
1081         plural="" Else plural="s"
1082     EndIf
1083     Switch(vCalc) // routine modified 6/4/2004 to avoid typing reduncancy
1084         Caseof 1: Dw="day" Do=" + " TypeDa
1085         Caseof 2: Dw="week" Do=" + " TypeDa
1086         Caseof 3: Dw="month" Do=" + " TypeDa
1087         Caseof 4: Dw="year" Do=" + " TypeDa
1088     EndSwitch
1089     EndSwitch
1090     If(vSpecial>1)
1091         eText
1092     EndIf
1093 EndIf
1094 If(TypeMe=30) // column
1095     PosDocBottom
1096     sTabs
1097     Switch(vCalc)
1098         Caseof "C": n=1
1099         Repeat
1100             var=T[n] Call(vWrite) typevar=var
1101             dTabs Type(typevar) HardReturn n=n+1
1102         Until (T[n]="")
1103         var=vResult vWrite
1104         PosLineUp PosLineBeg vUnd Type("+ ") SelectMode(On!) PosLineEnd
1105         AttributeAppearanceToggle(Underline!) SelectMode(Off!) PosLineDown
1106         Type("=") dTabs Type(var) HardReturn
1107     Default: //substantially modified; formula & result in a column; added 6/4/2004
1108     Switch(vCalc)
1109         Caseof "A": vchar="+" x="Addition (rounded to "+varD+" "+dword+");"
1110         Caseof "S": vchar="-" x="Subtraction (rounded to "+varD+" "+dword+");"
1111         Caseof "M": vchar="x" x="Multiplication (rounded to "+varD+" "+dword+" "++);"
1112         Default: vchar="÷" x="Division (rounded to "+varD+" "+dword+");"
1113     EndSwitch
1114     Switch(vCalc)
1115         Caseof "R":
1116             var=num1+num2 newnum tResult=var
1117             Type("Ratio of the following number to the Total of 2 Numbers (rounded to
1118             "+varD+" "+dword+");") HardReturn
1119             vUnd dTabs var=SecondN vWrite Type (var) PosLineEnd HardReturn
1120             Type(vchar) HardReturn

```

```

1121     Type("[")dTabs var=FirstN vWrite Type(var) HardReturn
1122     vUnd // in Wp9 & higher, could add UnderLineStyle (Solid!) but that won't work in Wp8
1123     Type("+") dTabs var=SecondN vwrite Type(var) PosLineEnd HardReturn
1124     vUnd Type("=") dTabs var=tResult vWrite Type(var) PosLineEnd
1125     Type("]") HardReturn
1126     Type("=") dTabs var=RealTotal vWrite Type(var) PosLineEnd Tab
1127     If(e=1)
1128         Type("(actual computed result up to 15 digits)")
1129     Else
1130         Type("(actual computed result)")
1131     EndIf
1132     HardReturn
1133     Type("=") dTabs Type(vResult) Tab Type("(rounded to "+varD+" "+dword+"")")
1134 Caseof "P":
1135     Type("2nd number's percentage of 1st number (rounded to "+varD+"
1136     "+dword+"):") HardReturn
1137     dTabs Type(SecondN) HardReturn
1138     vUnd // in Wp9 & higher, could add UnderLineStyle (Solid!) but that won't work in Wp8
1139     Type(vChar) dTabs Type(FirstN) PosLineEnd HardReturn
1140     Type("=") dTabs var=RealTotal vWrite Type(var) PosLineEnd Tab
1141     vTypeE vTypeD
1142 Default:
1143     Type(x) HardReturn
1144     dTabs var=FirstN vWrite Type(var) HardReturn
1145     vUnd // in Wp9 & higher, could add UnderLineStyle (Solid!) but that won't work in Wp8
1146     Type(vchar) dTabs var=SecondN vWrite Type(var) PosLineEnd HardReturn
1147     Type("=") dTabs var=RealTotal vWrite Type(var) PosLineEnd Tab
1148     vTypeE vTypeD
1149 EndSwitch
1150 If(vSpecial>1)
1151     HardReturn eText PosLineBeg DeleteCharNext PosLineEnd
1152 EndIf
1153 EndSwitch
1154 HardReturn
1155 EndIf
1156 RegionSetFocus("vMath.B2")
1157 Return
1158     Label(dTabs) // new on 6/4/2004 to set up tabs for typing a math column
1159 x= Strlen(vResult)
1160 If(x<27) x=4 EndIf
1161 If(x>=27 and x<34) x=5 EndIf
1162 If(x>=34) x=6 EndIf
1163 RepeatValue(x) TabDecimal Return
1164     Label(sTabs)
1165 TabSet (Relative!; { 1.0"; TabLeft!; 1.5"; TabLeft!; 2.0"; TabLeft!; 2.6"; TabLeft!; 3.0"; TabLeft!;
1166 3.5"; TabLeft!; 4.0"; TabLeft!; 4.5"; TabLeft!; 5.0"; TabLeft!; 5.5"; TabLeft!; 6.0"; TabLeft!;
1167 6.5"; TabLeft!; 7.0"; TabLeft!; 7.5"; TabLeft!; 8.0"; TabLeft!})
1168 DecimalAlignmentCharacter (".") Return
1169     Label(vUnd) // new on 6/4/2004 to set up underlining when typing a math column
1170 UnderlineTabs (Yes!) UnderlineSpaces (Yes!) AttributeAppearanceToggle (Underline!) Return
1171     Label(chkMax)
1172 //new error routine 6/4/2004: the number of digits (excluding leading zeros) cannot be longer than 15 "digits"
1173 x=strtransform (var; "-"; "")
1174 x=strtransform (x; "."; "")
1175 x=strtrim(x; TrimLeft!; "0")
1176 x=strlen(x)

```

```
1177 If(x>15)
1178   q=1 DialogShow("vMax";"WordPerfect") DialogDestroy("vMax")
1179 EndIf
1180 Return
1181   Label(chkNum) // this routine determines if all characters are valid for numeric purposes. revised 6/4/2004
1182 x=StrToChars(var;Keep!;".") x=StrLen(x) //this routine insures that only one decimal is in the string
1183 If(x>1)
1184   q=1
1185   Return
1186 EndIf
1187 x=substr(var;1;1)
1188 If(x="-")
1189   NegNum=1 var=substr(var;2;strlen(var)) Else NegNum=0
1190 EndIf
1191 Ret=IsNumberString(var)
1192 If(Ret)
1193   Else q=1
1194 EndIf
1195 Return // if the variable contains non-allowed characters, q=1 says error is present!
1196
1197 Function IsNumberString (InString)
1198   RefString="1234567890.,"
1199   ForNext (Count;1;StrLen(InString);1)
1200     If (StrPos(RefString;SubStr(InString;Count;1))=0)
1201       Return (False)
1202     EndIf
1203   EndFor
1204   Return (True)
1205 EndFunc
1206   Label(NewNum) // this routine massages numeric values into new string values
1207 If(var=0)
1208   Return
1209 EndIf
1210 x=StrPos(var;"-")
1211 If(x=1)
1212   var=Substr(var;2;strlen(var)) NegNum=1
1213 EndIf
1214 x=StrPos(var;".")
1215 Switch(x)
1216   Caseof 0: tNum1=var Call(mainNum) tNum2=""
1217   Default: tNum1=Substr(var;1;x-1) Call(mainNum) tnum2=substr(var;x;strlen(var))
1218     x=strlen(tnum2)-1
1219     If(x<varD and vFinal=1)
1220       Repeat
1221         tnum2=tnum2+"0" x=strlen(tnum2)-1
1222       Until(x=varD)
1223     EndIf
1224 EndSwitch
1225 var=tNum1+tNum2
1226 If(NegNum=1)
1227   var="-"+var
1228 EndIf
1229 Discard(NegNum;tNum1;tNum2)
1230 Return
```

```
1231 Label(mainNum) // this routine creates commas in the main number from Label(NewNum)
1232 If(StrLen(tNum1) <= 3)
1233 Return
1234 EndIf
1235 z= Integer(StrLen(tNum1)/3)
1236 newvar=""
1237 Repeat
1238 y= SubStr(tNum1; StrLen(tNum1)-2; 3)
1239 tNum1= SubStr(tNum1; 1; StrLen(tNum1)-3)
1240 If(StrLen(tNum1) > 3)
1241 newvar=", "+y+newvar Else newvar=tNum1+", "+y+newvar
1242 EndIf
1243 If(StrLen(tNum1) > 3)
1244 z= Integer(StrLen(tNum1)/3) Else z=0
1245 EndIf
1246 Until(z=0)
1247 tNum1=newvar
1248 Return
1249 Label(MakePer) // this routine creates percentage string values. revised 6/4/2004 for scientific notation values
1250 e=0
1251 var=StrTransform(var; ", "; "")
1252 var=strnum(var)
1253 var=var*100 // a % value is always 100 times the non-percentage value
1254 x=strpos(var; "e") // scientific notation routine
1255 If(x > 0)
1256 e=1
1257 y=substr(var; x+2; 3) y=strnum(y) y=y+1
1258 var=substr(var; 1; x-1)
1259 var=var-". "
1260 varq=strlen(var)
1261 If(varq < y)
1262 Repeat
1263 var=var+"0"
1264 Until (strlen(var)=y)
1265 EndIf
1266 x=strpos(var; ".")
1267 If(x=0)
1268 var=var-". "
1269 EndIf
1270 EndIf
1271 If(e=0)
1272 var=numstr(var) // want to determine if that following the decimal is greater than 0, but not if scientific notation
1273 EndIf
1274 x=strpos(var; ".") testn=substr(var; x+1; strlen(var)) testn=strnum(testn) // testn is rem w/o the "."
1275 If(testn=0)
1276 var=substr(var; 1; x-1) // if the remainder is 0, drops the ".0" from var
1277 EndIf
1278 If(e=0) // not if scientific notation
1279 var=strnum(var)
1280 EndIf
1281 Call(NewNum)
1282 var=var+"%"
1283 If(negNum=1)
1284 var="-"+var
1285 EndIf
1286 Return
```



```
1287 Label(DateCnv) // this tests and sets the numeric parts of a m/d/yyyy date and makes string values
1288 Call(dateChk)
1289 If(exists(q))
1290 Return
1291 EndIf
1292 x=strPos(var;"/")
1293 If(x>0)
1294 q=1
1295 vMsg=var+" is not a legal date. Don't use consecutive '/' marks." setResult Return
1296 EndIf
1297 x=StrPos(var;"/")
1298 If(x=0)
1299 q=1
1300 vMsg=var+" is not a legal date. Use a m/d/yyyy date format." setResult Return
1301 EndIf
1302 If(x>0)
1303 mo=substr(var;1;x-1) rem=substr(var;x+1;strlen(var))
1304 x=StrPos(rem;"/")
1305 day=substr(rem;1;x-1) year=substr(rem;x+1;strlen(rem))
1306 If(strlen(mo)>2 or strlen(day)>2 or strlen(year)<>4)
1307 q=1
1308 vMsg=var+" is not a legal date. Use a m/d/yyyy date format." Return
1309 EndIf
1310 EndIf
1311 Mo=Strnum(Mo) Day=Strnum(Day) Year=strnum(Year)
1312 If(year<1601)
1313 q=1
1314 vMsg="Dates earlier than 1/1/1601 cannot be used. Change the date" Return
1315 EndIf
1316 If(Mo=0 or Day=0 or Year=0)
1317 q=1
1318 vMsg="Don't use zeros for months, days or years. "+Mo+"/"+Day+"/"+Year+" is not a legal
1319 number."
1320 Return
1321 EndIf
1322 If(Mo>12)
1323 q=1
1324 vMsg=Mo+" is not a legal month. Use 1~12." Return
1325 EndIf
1326 Switch(Mo)
1327 Caseof 1: vMo="January" Caseof 7: vMo="July"
1328 Caseof 2: vMo="February" Caseof 8: vMo="August"
1329 Caseof 3: vMo="March" Caseof 9: vMo="September"
1330 Caseof 4: vMo="April" Caseof 10: vMo="October"
1331 Caseof 5: vMo="May" Caseof 11: vMo="November"
1332 Caseof 6: vMo="June" Caseof 12: vMo="December" EndSwitch
1333 vLeap=DatIsLeapYear(year) // this is used for determining what to do with February, below
1334 Switch(vMo)
1335 Caseof "February":
1336 Switch(vLeap)
1337 Caseof False:
1338 If(day>28)
1339 q=1
1340 EndIf
1341 Caseof True:
1342 If(day>29)
```

```
1343         q=1
1344     EndIf
1345 EndSwitch
1346 Caseof "April":
1347     If(Day>30)
1348         q=1
1349     EndIf
1350 Caseof "June":
1351     If(Day>30)
1352         q=1
1353     EndIf
1354 Caseof "September":
1355     If(Day>30)
1356         q=1
1357     EndIf
1358 Caseof "November":
1359     If(Day>30)
1360         q=1
1361     EndIf
1362 Default:
1363     If(Day>31)
1364         q=1
1365     EndIf
1366 EndSwitch
1367 If(Exists(q))
1368     vMsg=vMo+" "+year+" does not contain "+day+" days." Return
1369 EndIf
1370 var=vMo+" "+Day+", "+Year
1371 Return
1372 Label(dateChk) // this tests for m/d/yyyy string legal date characters
1373 ret=IsNumberDate(var)
1374 If(Ret)
1375     Else vMsg=var+" is not a legal date. " q=1
1376 EndIf
1377 Return
1378 Function IsNumberDate (InString)
1379     RefString="1234567890/"
1380     ForNext (Count;1;StrLen(InString);1)
1381         If (StrPos(RefString;SubStr(InString;Count;1))=0)
1382             Return (False)
1383         EndIf
1384     EndFor
1385     Return (True)
1386 EndFunc
1387 Label(vError) // this is the initial OnError() Label
1388 x=?NumberOpenDocuments
1389 If(x=9)
1390     vMsg="You have attempted to make a new blank document."+h+h+ "WordPerfect can only
1391     have 9 open documents at a time, and you are at that point. "+h+h vMsg=vMsg+"You may
1392     continue to use the macro, but no new documents can be opened at this time. "+h+h+"To
1393     stop the macro, click the Stop button."
1394     Else
1395     vMsg="An error has occurred for unknown cause. "+h+h vMsg=vMsg+"If you receive this
1396     message, please send me an e-mail (loudenbk@swbell.net) or call me (405-236-4004) "
1397     vMsg=vMsg+"and describe what you were doing when the error occurred and I'll try to fix the
1398     problem."+h+h vMsg=vMsg+h+"It will help to know your WordPerfect version and your
```

```

1399   operating system, e.g., WindowsXP."+h+"The macro must now stop."
1400 EndIf
1401 DialogShow("verr";"WordPerfect") Return
1402 // End of Math.wcm

```

Top of Chapter 9

Top of Macro

Chapter 10

```

1 //CONVERTFE.WCM
2 // © 1998-2004 by Doug Loudenback, Oklahoma City, All Rights Reserved
3 //This macro may be distributed or edited, but may not be sold or posted on any website without my
4 express written permission.
5 //Revision Date: 6/11/2004
6 // This macro converts either endnotes or footnotes to plain text "pseudo" endnotes at the end of the
7 document, but preserves the appearance of endnotes – note numbers remain in superscript.
8 // This macro is downloadable at http://www.dougloudenback.com/wp/convertFE.exe
9 Application (WordPerfect; "WordPerfect"; Default!; "EN")
10 H=NTOC(0F90Ah) Display(On!)
11 If(?docBlank)
12   vMsg="This is a blank document. It contains no endnotes or footnotes so there's nothing for
13   this macro to do."
14   DialogShow("NoNotes2";"WordPerfect") Quit
15 EndIf
16 BookmarkCreate("x") // a Bookmark is created at the top; it will later be deleted
17 fsz=?FontSize // this is the font name at the top of the document
18 vfont=?font // this is the font size at the top of the document
19 fsz=ConvertType(fsz;Points!) fsz=ConvertType(fsz;String!)
20 fsz=substr(fsz;1;strlen(fsz)-1) vsz=fsz
21 vLoop=False
22 DialogShow("vType";"WordPerfect";cbType)
23 vtxt="1. Text will wrap in paragraphs"+h+" like this."
24 RegionSetWindowText("vType.S0";vtxt)
25 Repeat Until(vLoop) DialogDestroy("vType")
26 If(x=2) // Cancel Button is pushed in the dialog ...
27   BookmarkFind("x") BookmarkDelete("x") Quit
28 EndIf
29 Switch(vWhat)
30   Caseof "Endnotes": var="[Endnote]" // [Endnote] and [Footnote] are not text
31   Caseof "Footnotes": var="[Footnote]" // they are inserted using the Codes ... button on the Macro Toolbar
32 EndSwitch
33 Display(Off!)
34 Label(vTest) // this tests the document to see if it contains what you wanted to convert - Endnotes or Footnotes
35 PosDocTop // moves the insertion point to the top of the document
36 OnNotFound(NoNotes) // what happens if the document contains no notes
37 SearchString (var) MatchPositionAfter SearchNext
38 // the "OnNotFound" condition did not occur; the macro continues
39 PosDocBottom SearchString(var) MatchPositionAfter SearchPrevious
40 Switch(vWhat) // new on 6/11/04; gets last note number value for progress dialog (vWait)
41   Caseof "Endnotes":
42     y=?EndNote y=numstr(y) // gets last endnote value
43   Caseof "Footnotes":
44     y=?FootNote y=numstr(y) // gets last footnote value
45 EndSwitch
46 PosDocTop
47 Main=?DocNumber // this identifies the source document being worked on
48 FileNew Notes=?DocNumber // this identifies the temporary document containing pseudo notes
49 QuickCorrectQuickBulletsSet(Off!) //these 2 items were added 6/11/04;
50 QuickCorrectQuickIndentSet(Off!) //if "on", printing issues are present

```

```

51 Font(vFont) FontSize(strnum(vSz) + "p") // font size and other characteristics of the Notes document is set
52 ParagraphSpacing (vParSp) // revised 6/11/04: replaces former code with variable paragraph spacing
53 TabSet (Origin: Relative!) //revised 6/11/04: allows for tab/indent of ¶ numbers
54 TabSet (Relative!; { 1.0"; TabLeft!; 1.38"; TabLeft!; 1.6"; TabLeft!; 1.9"; TabLeft!; 2.2";
55 TabLeft!; 2.5"; TabLeft!; 2.8"; TabLeft!; 3.1"; TabLeft!; 3.4"; TabLeft!; 3.7"; TabLeft!;
56 4.0"; TabLeft!; 4.3"; TabLeft!; 4.6"; TabLeft!; 4.9"; TabLeft!; 5.2"; TabLeft!; 5.5";
57 TabLeft!; 5.8"; TabLeft!; 6.1"; TabLeft!; 6.4"; TabLeft!; 6.7"; TabLeft!; 7.0"; TabLeft!;
58 7.3"; TabLeft!; 7.6"; TabLeft!; 7.9"; TabLeft!; 8.2"; TabLeft!})
59 HLineCreate HardReturn Advance(AdvanceDown!; 0.101")
60 Center Type("NOTES") HardReturn BookmarkCreate("x")
61 SwitchDoc(Main)
62 Msg1="Finding and converting notes" Msg2="Note "+x+" of "+y+" Please wait . . . "
63 Call(WaitMsg)
64 Label(Begin)
65 OnNotFound(vNext) // what happens if no more notes are found
66 SearchString (var) MatchPositionAfter SearchNext
67 Switch(vWhat)
68   Caseof "Endnotes":
69     x=?EndNote x=numstr(x) EndnoteEdit(x) // sets found note value and enters it for copying
70   Caseof "Footnotes":
71     x=?FootNote x=numstr(x) FootnoteEdit(x) // sets note value and enters it for copying
72 EndSwitch
73 RegionSetWindowText("vWait.S0";"Note "+x+" of "+y+" Please wait . . . ") //new on 6/11/04:
74 shows progress
75 PosDocTop
76 Repeat //new on 6/11/04: makes sure leading "spaces" in the note are eliminated for copy/paste routine
77   c=?RightChar
78   If(c=" ")
79     PosWordNext EndIf
80   Until(c<>" ")
81 Repeat //new on 6/11/04: makes sure leading "LeftTab" codes in the note are eliminated for copy/paste routine
82   c=?RightCode
83   If(c=4576)
84     PosWordNext EndIf
85   Until(c<>4576)
86 // the code below copies a note's text, exits the note, switches to the Notes doc, pastes, and returns to the main doc
87 SelectMode(On!) // this positions the insertion point at the top of the note and turns on Select
88 PosDocBottom // this positions the insertion point at the bottom of the note
89 EditCopy SelectMode(Off!) Close // this copies the selection to Windows Clipboard and exits the note
90 SwitchDoc(Notes) PosDocBottom Type(x+".") //switches to the "notes" document, types note #
91 If(vIndent="Tab") Tab Else Indent EndIf //new on 6/11/04: uses Tab or Indent per user choice
92 EditPaste PosLineEnd HardReturn
93 // the above switches to the temporary Notes document, types a "note" number, and pastes the Clipboard value
94 SwitchDoc(Main) Go(Begin) // the search loops and continues until no notes are found
95   Label(vNext) // if no more notes are found, the macro continues here
96 SwitchDoc(Main) PosDocTop
97   Label(vNext2) //routine loops until no more notes are found
98 OnNotFound(vDone) SearchString (var) MatchPositionAfter SearchNext
99 Switch(vWhat)
100   Caseof "Endnotes": x=?EndNote x=numstr(x)
101   Caseof "Footnotes": x=?FootNote x=numstr(x)
102 EndSwitch
103 FontSuperscriptToggle HardSpace Type(x) FontSuperscriptToggle Go(vNext2)
104   Label(vDone) // these routines clean up the 2 documents, deleting all true footnotes or endnotes
105 PosDocTop // in the original document here
106 SearchString (var) ReplaceString ("") ReplaceForward (Extended!)

```

```
107 SwitchDoc(Notes) // in the Notes document here
108 BookMarkFind("x")
109 SearchString("[HRt][HRt]") ReplaceString("[HRt]") ReplaceForward //insert [HRt] with Codes... Bttn
110 BookMarkFind("x")
111 SearchString("[Font Size]") ReplaceString("") ReplaceForward //insert [Font Size] code with Codes... Bttn
112 BookMarkFind("x")
113 SearchString("[Tab Set]") ReplaceString("") ReplaceForward //insert [Tab Set] code with Codes... Bttn
114 BookMarkFind("x") BookMarkDelete("x")
115 // this copies the temporary Notes document and pastes it at the end of the original document
116 PosDocTop
117 PosDocVeryTop SelectMode(On!) PosDocBottom
118 EditCopy SelectMode(Off!) CloseNoSave
119 SwitchDoc(Main)
120 PosDocBottom HardReturn EditPaste
121 BookMarkFind("x") BookMarkDelete("x")
122 KillMsg // stops the progress dialog
123 Msg1="The operation is done." Msg2="To undo changes, click your Undo icon"
124 WaitMsg Wait(20) KillMsg
125 Quit
126 Label(NoNotes) // this sees if the original "type" of notes are in the main document: if not, it the other type IS
127 OnNotFound(vNone)
128 PosDocTop Switch(vWhat)
129 Caseof "Endnotes": var="[Footnote]" // insert the [Footnote] code with Codes... button
130 Caseof "Footnotes": var="[Endnote]" // insert the [Endnote] code with Codes... button
131 EndSwitch
132 SearchString(var) MatchPositionBefore SearchNext
133 Switch(vWhat)
134 Caseof "Endnotes":
135     vmsg="This document contains Footnotes, not Endnotes. Do you want to do Footnotes
136     instead?"
137 Caseof "Footnotes":
138     vmsg="This document contains Endnotes, not Footnotes. Do you want to do Endnotes
139     instead?"
140 EndSwitch
141 Display(On!)
142 DialogShow("NoNotes"; "WordPerfect") x=macrodialogresult DialogDestroy("NoNotes")
143 If(x="CancelBttn" or x=2) BookMarkFind("x") BookMarkDelete("x") Quit EndIf
144 Display(Off!)
145 Switch(vWhat)
146 Caseof "Footnotes": vWhat="EndNotes" var="[Endnote]" // insert the [Endnote] with Codes... button
147 Caseof "Endnotes": vWhat="Footnotes" var="[Footnote]" // insert the [Footnote] with Codes... button
148 EndSwitch
149 Go(vTest)
150 Label(vNone) // neither Footnotes or Endnotes are present in the main document
151 Display(On!)
152 vmsg="This document contains neither Footnotes nor Endnotes. So, there is nothing for it to do
153 and the macro will stop." DialogShow("NoNotes2"; "WordPerfect") DialogDestroy("NoNotes2")
154 BookMarkFind("x") BookMarkDelete("x") Quit
155 Label(cbType) Switch(cbType[3]) // this is the callback label for/from the main dialog
156 Caseof "OKBttn"; 1: vClose
157 Caseof "B2": //new routine on 6/11/04
158     var=RegionGetSelectedText("vType.B2") //changes "vType.S0" control to show effect of selection
159     If(var="Tab")
160         vtxt="1. Text will wrap in paragraphs"+h+"like this."
161     Else
162         vtxt="1. Text will wrap in paragraphs"+h+" like this."
```

```

163 EndIf
164 RegionSetWindowText("vType.S0";vtxt)
165 Caseof "CancelBttn"; 2: vClose
166 Caseof "TipsBttn": DialogShow("vTips";"vType")
167 EndSwitch Return
168 Label(vClose) //closes the callback dialog
169 DialogDismiss("vType";cbType[3]) x=MacroDialogResult vLoop=True
170 Return
171 Label(WaitMsg) DialogShow("vWait";"WordPerfect";cbWait) Return
172 Label(cbWait) If(cbWait[3]="CancelBttn") Assert(CancelCondition!) EndIf Return
173 Label(KillMsg) DialogDismiss("vWait";2) DialogDestroy("vWait") Return
174 // end of macro

```

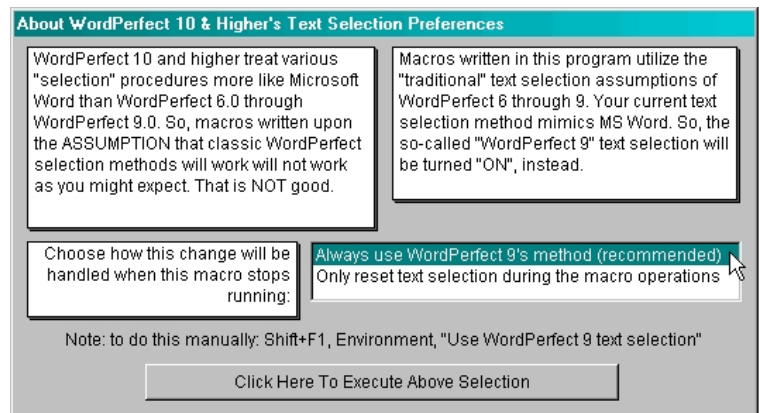
[Top of Chapter 9](#)

[Top of Macro](#)

[Chapter 10](#)

Wp9select.wcm. It exists for a single purpose – to deal with the text selection default used in WordPerfect 10 and higher. See the discussion in [Chapter 3](#) and [BMT](#) for other means of handling the problem. Download this macro at <http://www.dougloudenback.com/wp/wp9select.exe>.

When the 1<sup>st</sup> part of the macro code is run in WordPerfect 10 or higher and the "Environment" Preference for "Use WordPerfect 9 text selection" is not checked, this dialog will appear. The user can "leave" WordPerfect 9 text selection turned "On" or can turn it off when the macro(s) are done by using the 2<sup>nd</sup> section of the code in this or some other related macro.



```

1 //WordPerfect Macro: wp9select.wcm
2 //This identifies and sets Preferences in Wp10 through 12 for Wp9 text selection; gives model text
3 //7/2/2004: I've just learned from JDan Broadhead that platform ID's can be combined in the
4 //IfPlatform statement, so that instead of a separate statement for each (_VersionX!), a single
5 //statement could read like this:
6 //IfPlatform(_Version10!; _Version11!; _Version12!)
7 // x=?Wp9Selection
8 // If(x=0)
9 // selChange=1 ChangeSel
10 // PrefEnvironment (; WP9Selection: On!) PrefSave
11 // EndIf
12 //EndIfPlatform
13 //I'll leave the code "as is", but it would be simpler to combine the Platform ID's, as shown at Line 6, above
14
15 Application (WordPerfect; "WordPerfect"; Default!; "EN") VarErrChk(Off!)
16 vSelText=0 selChange=0 ChangeSel // this avoids a compilation warning if run under Wp 6.1, 7, 8 or 9
17
18 IfPlatform(_Version10!) // will compile only in WordPerfect 10
19 x=?Wp9Selection
20 If(x=0)
21 selChange=1 ChangeSel
22 PrefEnvironment (; WP9Selection: On!) PrefSave
23 EndIf
24 EndIfPlatform
25
26 IfPlatform(_Version11!) // will compile only in WordPerfect 11
27 x=?Wp9Selection

```

```
28     If(x=0)
29         selChange=1 ChangeSel
30         PrefEnvironment (; WP9Selection: On!) PrefSave
31     EndIf
32 EndIfPlatform
33
34 IfPlatform(_Version12!) // will compile only in WordPerfect 12
35     x=?Wp9Selection
36     If(x=0)
37         selChange=1 ChangeSel
38         PrefEnvironment (; WP9Selection: On!) PrefSave
39     EndIf
40 EndIfPlatform
41
42 // in a "real" macro, the following code would be used to reset Wp9 text selection in Wp 10, 11 & 12, IF
43 variable vSelText exists and =1; in this macro, by itself & under the stated conditions, Wp9 text selection
44 will remain "Off" unless the following code is present
45 If(Exists(vSelText) and vSelText=1)
46
47     IfPlatform(_Version10!) // will compile only in WordPerfect 10
48         PrefEnvironment (; WP9Selection: Off!) PrefSave
49     EndIfPlatform
50
51     IfPlatform(_Version11!) // will compile only in WordPerfect 11
52         PrefEnvironment (; WP9Selection: Off!) PrefSave
53     EndIfPlatform
54
55     IfPlatform(_Version12!) // will compile only in WordPerfect 12
56         PrefEnvironment (; WP9Selection: Off!) PrefSave
57     EndIfPlatform
58 EndIf
59 Return
60 Label(ChangeSel)
61 If(selChange=0) // this avoids a compilation warning if run under Wp 6.1, 7, 8 or 9
62     Else
63         DialogShow("vSelect"; "WordPerfect")
64         vSelText=Substr(vSelText; 1; 1)
65         If(vSelText="0") // in this or a related macro, this will be used to turn off Wp9 text selection
66             Discard(vSelText) Global vSelText=1 // this discards Local variable vSelText & sets Global
67         Else
68             Discard(vSelText) // no variable vSelText will exist
69         EndIf
70 EndIf
71 Return
```

NOTES



NOTES

NOTES