

Prof. Seth Graebner

Instructions for programming and using macros for editing student papers

Code included at the end of this document

Instructions:

I find useful a suite of 16 individual macros, each responding to a particular frequent error in student papers, and each triggered by holding down Shift and Alt and typing a single letter code. Some of these you may not wish to use, or even to encode, since a useless macro takes up a letter code on your keyboard, at least until you reassign that letter to another macro.

The document "editing.docx" gives a list of the errors to which the different macros respond, the key-code for triggering the note-adding action, and the actual marginal note that the macro inserts. To edit these (if for instance you wanted your complaint about excessive use of the verb "to be" to say something slightly different from mine) it would not suffice simply to open the marginal comment in the document I'm sending you; you would need to get into the macro code itself, which I'll tell you about in a minute. "Editing.docx" is the document that will serve as your reminder of what key-code goes with what macro, until you begin to remember the ones you use. Once you get used to them, they're second nature, and they will save you time almost from the beginning.

This present document contains the actual code of the macros. My expertise is very limited on this, since I don't ordinarily use Microsoft's rubbishy products for my own work. (A plug to all academic writers: consider Nota Bene, the software made for us from the beginning. It could change your life. I don't own stock in the company, nor make any money from them.) So here's what my limited competence suggests. Most users will probably want to be able to use these macros on any document they open (not that they would on their own composition). For this reason, the user must copy the macros into his or her own "normal.dotm" file, the file containing default settings for page layout in Word, as well as user-created macros.

This is how you do it. In MS Word, open the "Developer" menu, and then click on the left-most icon, "Visual Basic." This will spawn the Visual Basic code editor (a separate window), with an open "document" into which you should paste the code from the end of this document. If your Visual Basic opens with other stuff already in it, just append my code to the end of it. Formatting should not matter at all.

Visual Basic is kind of a mess, but with a little attention, you can figure some things out. Individual macros have titles; all of mine begin "complain" in order to keep them easily separated from any others you may have or acquire. Each macro begins by the designation "Sub," for "subroutine," I think, and ends "End Sub." Any "if" must have a corresponding "end if." For the most part, people don't write macros in code, but rather record a series of actions in MS Word that the program then encodes as a macro. Nonetheless, it is possible to edit macros in Visual Basic, and it's here that users could change my wording, if they wanted, and have the change actually take effect next time they invoked the macro. The bulk of the code in my macros is needed to open and close the "commenting" function and return to the same "view" of the file you had before, with minimal jumpiness or need for user intervention. That's what all

Prof. Seth Graebner

Instructions for programming and using macros for editing student papers

Code included at the end of this document

Selection.TypeText Text:="You use too many forms of ""to be"" here, deadening your prose.
Rephrase with ""active"" verbs."

```
Selection.EscapeKey
ActiveWindow.View.ReadingLayout = Not ActiveWindow.View.ReadingLayout
ActiveWindow.View.ReadingLayout = Not ActiveWindow.View.ReadingLayout
If ActiveWindow.View.SplitSpecial = wdPaneNone Then
    ActiveWindow.ActivePane.View.Type = wdPrintView
Else
    ActiveWindow.View.Type = wdPrintView
End If
End Sub
```

```
Sub complain_contractions()
```

```
,
' complain_contractions Macro
' Add comment: "Formal English uses no contractions."
```

```
,
    Selection.Comments.Add Range:=Selection.Range
    Selection.TypeText Text:="Formal English uses no contractions."
    Selection.EscapeKey
    ActiveWindow.View.ReadingLayout = Not ActiveWindow.View.ReadingLayout
    ActiveWindow.View.ReadingLayout = Not ActiveWindow.View.ReadingLayout
    If ActiveWindow.View.SplitSpecial = wdPaneNone Then
        ActiveWindow.ActivePane.View.Type = wdPrintView
    Else
        ActiveWindow.View.Type = wdPrintView
    End If
End Sub
```

```
Sub complain_wrong_word()
```

```
,
' complain_diction Macro
' Add comment: "Wrong word."
```

```
,
    Selection.Comments.Add Range:=Selection.Range
    Selection.TypeText Text:="Wrong word."
    Selection.EscapeKey
    ActiveWindow.View.ReadingLayout = Not ActiveWindow.View.ReadingLayout
    ActiveWindow.View.ReadingLayout = Not ActiveWindow.View.ReadingLayout
    If ActiveWindow.View.SplitSpecial = wdPaneNone Then
        ActiveWindow.ActivePane.View.Type = wdPrintView
    Else
        ActiveWindow.View.Type = wdPrintView
    End If
End Sub
```

```
Sub complain_end_with_quote()
```

```
,
```

Prof. Seth Graebner

Instructions for programming and using macros for editing student papers

Code included at the end of this document

```
'complain_end_with_quote Macro
'Add comment: "Do not end paragraphs with quotes [...]"
,
  Selection.Comments.Add Range:=Selection.Range
  Selection.TypeText Text:="Do not end paragraphs with quotes. Instead, end with your own analysis of
the quoted material, so as to ensure its integration into your argument, and to prevent the quote from
usurping your own voice."
  Selection.EscapeKey
  ActiveWindow.View.ReadingLayout = Not ActiveWindow.View.ReadingLayout
  ActiveWindow.View.ReadingLayout = Not ActiveWindow.View.ReadingLayout
  If ActiveWindow.View.SplitSpecial = wdPaneNone Then
    ActiveWindow.ActivePane.View.Type = wdPrintView
  Else
    ActiveWindow.View.Type = wdPrintView
  End If
End Sub
```

```
Sub complain_full_title()
,
```

```
'complain_full_title Macro
'Add comment: "Give the full title on first citation."
,
  Selection.Comments.Add Range:=Selection.Range
  Selection.TypeText Text:="Give the full title on first citation."
  Selection.EscapeKey
  ActiveWindow.View.ReadingLayout = Not ActiveWindow.View.ReadingLayout
  ActiveWindow.View.ReadingLayout = Not ActiveWindow.View.ReadingLayout
  If ActiveWindow.View.SplitSpecial = wdPaneNone Then
    ActiveWindow.ActivePane.View.Type = wdPrintView
  Else
    ActiveWindow.View.Type = wdPrintView
  End If
End Sub
```

```
Sub complain_however()
,
```

```
'complain_however Macro
'Add comment: "'However' is not a conjunction, so this is a run-on sentence."
,
  Selection.Comments.Add Range:=Selection.Range
  Selection.TypeText Text:=""'"However"' is not a conjunction, so this is a run-on sentence.'"
  Selection.EscapeKey
  ActiveWindow.View.ReadingLayout = Not ActiveWindow.View.ReadingLayout
  ActiveWindow.View.ReadingLayout = Not ActiveWindow.View.ReadingLayout
  If ActiveWindow.View.SplitSpecial = wdPaneNone Then
    ActiveWindow.ActivePane.View.Type = wdPrintView
  Else
    ActiveWindow.View.Type = wdPrintView
```

Prof. Seth Graebner

Instructions for programming and using macros for editing student papers

Code included at the end of this document

```
End If
End Sub

Sub complain_impact()
'
' complain_impact Macro
' Add comment: "Impact" is a verb only in commercial (and similarly barbarous) forms of English.
'
    Selection.Comments.Add Range:=Selection.Range
    Selection.TypeText Text:=""Impact"" is a verb only in commercial (and similarly barbarous) forms of
English."
    Selection.EscapeKey
    ActiveWindow.View.ReadingLayout = Not ActiveWindow.View.ReadingLayout
    ActiveWindow.View.ReadingLayout = Not ActiveWindow.View.ReadingLayout
    If ActiveWindow.View.SplitSpecial = wdPaneNone Then
        ActiveWindow.ActivePane.View.Type = wdPrintView
    Else
        ActiveWindow.View.Type = wdPrintView
    End If
End Sub

Sub complain_missing_title()
'
' complain_missing_title Macro
' Add comment: "You need a title."
'
    Selection.Comments.Add Range:=Selection.Range
    Selection.TypeText Text:"You need a title."
    Selection.EscapeKey
    ActiveWindow.View.ReadingLayout = Not ActiveWindow.View.ReadingLayout
    ActiveWindow.View.ReadingLayout = Not ActiveWindow.View.ReadingLayout
    If ActiveWindow.View.SplitSpecial = wdPaneNone Then
        ActiveWindow.ActivePane.View.Type = wdPrintView
    Else
        ActiveWindow.View.Type = wdPrintView
    End If
End Sub

Sub complain_note_placement()
'
' complain_note_placement Macro
' Add comment: "Place all notes and other references at the ends of their sentences. [...]"
'
    Selection.Comments.Add Range:=Selection.Range
    Selection.TypeText Text:"Place all notes and other references at the ends of their sentences. In the
case of multiple notes in the same sentence, combine them into one."
    Selection.EscapeKey
    ActiveWindow.View.ReadingLayout = Not ActiveWindow.View.ReadingLayout
```

Prof. Seth Graebner

Instructions for programming and using macros for editing student papers

Code included at the end of this document

```
ActiveWindow.View.ReadingLayout = Not ActiveWindow.View.ReadingLayout
If ActiveWindow.View.SplitSpecial = wdPaneNone Then
    ActiveWindow.ActivePane.View.Type = wdPrintView
Else
    ActiveWindow.View.Type = wdPrintView
End If
End Sub
```

```
Sub complain_passive()
```

```
'
' complain_passive Macro
' Add comment: "Eliminate the passive voice"
```

```
'
    Selection.Comments.Add Range:=Selection.Range
    Selection.TypeText Text:="Eliminate the passive voice."
    Selection.EscapeKey
    ActiveWindow.View.ReadingLayout = Not ActiveWindow.View.ReadingLayout
    ActiveWindow.View.ReadingLayout = Not ActiveWindow.View.ReadingLayout
    If ActiveWindow.View.SplitSpecial = wdPaneNone Then
        ActiveWindow.ActivePane.View.Type = wdPrintView
    Else
        ActiveWindow.View.Type = wdPrintView
    End If
End Sub
```

```
Sub complain_floating_quote()
```

```
'
' complain_floating_quote Macro
' Add comment: "Always attach quotes to a sentence of your own."
```

```
'
    Selection.Comments.Add Range:=Selection.Range
    Selection.TypeText Text:="Always attach a quote to a sentence of your own."
    Selection.EscapeKey
    ActiveWindow.View.ReadingLayout = Not ActiveWindow.View.ReadingLayout
    ActiveWindow.View.ReadingLayout = Not ActiveWindow.View.ReadingLayout
    If ActiveWindow.View.SplitSpecial = wdPaneNone Then
        ActiveWindow.ActivePane.View.Type = wdPrintView
    Else
        ActiveWindow.View.Type = wdPrintView
    End If
End Sub
```

```
Sub complain_needs_reference()
```

```
'
' complain_needs_reference Macro
' Add comment: "Needs a reference."
```

```
'
    Selection.Comments.Add Range:=Selection.Range
```

Prof. Seth Graebner

Instructions for programming and using macros for editing student papers

Code included at the end of this document

```
Selection.TypeText Text:="Needs a reference."
Selection.EscapeKey
ActiveWindow.View.ReadingLayout = Not ActiveWindow.View.ReadingLayout
ActiveWindow.View.ReadingLayout = Not ActiveWindow.View.ReadingLayout
If ActiveWindow.View.SplitSpecial = wdPaneNone Then
    ActiveWindow.ActivePane.View.Type = wdPrintView
Else
    ActiveWindow.View.Type = wdPrintView
End If
End Sub

Sub complain_stylesheet()
'
'complain_stylesheet Macro
' Add comment: "See my stylistic advice sheets about this."
'
    Selection.Comments.Add Range:=Selection.Range
    Selection.TypeText Text:="See my stylistic advice sheets about this."
    Selection.EscapeKey
        ActiveWindow.View.ReadingLayout = Not ActiveWindow.View.ReadingLayout
    ActiveWindow.View.ReadingLayout = Not ActiveWindow.View.ReadingLayout
    If ActiveWindow.View.SplitSpecial = wdPaneNone Then
        ActiveWindow.ActivePane.View.Type = wdPrintView
    Else
        ActiveWindow.View.Type = wdPrintView
    End If
End Sub

Sub complain_title_format()
'
'complain_title_format Macro
' Add comment: "Italicize or underline book titles."
'
    Selection.Comments.Add Range:=Selection.Range
    Selection.TypeText Text:="Italicize or underline titles of works published with their own book covers.
Use quotation marks for all others."
    Selection.EscapeKey
        ActiveWindow.View.ReadingLayout = Not ActiveWindow.View.ReadingLayout
    ActiveWindow.View.ReadingLayout = Not ActiveWindow.View.ReadingLayout
    If ActiveWindow.View.SplitSpecial = wdPaneNone Then
        ActiveWindow.ActivePane.View.Type = wdPrintView
    Else
        ActiveWindow.View.Type = wdPrintView
    End If
End Sub

Sub complain_which_that()
```

Prof. Seth Graebner

Instructions for programming and using macros for editing student papers

Code included at the end of this document

```
,  
'complain_which_that Macro  
'Add comment: "Learn and respect the difference between 'which' and 'that.'" [S-A] W  
,  
  
Selection.Comments.Add Range:=Selection.Range  
Selection.TypeText Text:="Learn and respect the difference between ""which"" and ""that.""  
Selection.EscapeKey  
ActiveWindow.View.ReadingLayout = Not ActiveWindow.View.ReadingLayout  
ActiveWindow.View.ReadingLayout = Not ActiveWindow.View.ReadingLayout  
If ActiveWindow.View.SplitSpecial = wdPaneNone Then  
    ActiveWindow.ActivePane.View.Type = wdPrintView  
Else  
    ActiveWindow.View.Type = wdPrintView  
End If  
End Sub
```