**MACROS 201: The Shell Game**

Amy J. Schneider ~ Featherschneider Editorial Services ~ www.featherschneider.com

Editors Canada 2019 ~ Halifax, NS ~ June 8, 2019



The macros demonstrated/discussed in the handout and session are provided as plain text below; copy them as a single block of text and paste them into a module in your chosen template. (Remember the Second Rule of Macro Club: I recommend storing your macros in a separate template from Normal.dot. Open a new Word document and save as a template file type (suggested name: Macros201.dotm) to create a new template.)



Place this template in your STARTUP folder in your designated Templates folder to access the macros.

Templates folder location (you can change to a custom location using “Modify”):

File > Options > Advanced > General > File Locations > User templates

Startup folder location (you can change to a custom location using “Modify”):

File > Options > Advanced > General > File Locations > Startup (should be inside Templates folder)



Open the VBA Editor (VBE) to edit macros (see handout)

Macros are in the SampleMacros101 module (double-click to see code)



Review the macros named “A\_201\_[something]Shell” before the others; these are the simplified samples.



The ConverterShell macro (incomplete) is provided; you can practice editing macros by filling it in.



Remember, if you rename/copy/move macros, their keyboard shortcuts disappear and you will have to redefine them (Customize Keyboard > Categories > Macros).



**THE MACROS:**

Sub AutoTracking\_HyphenToEnDashUntracked()

'

' I have assigned this to Ctrl + Shift + Num-

'

' Macro orig. recorded 9/26/2004 by Hilary Powers

' Determine state of tracking (per Rich Adin, 8/14/2006)

Dim blnTR As Boolean

blnTR = ActiveDocument.TrackRevisions

' Turn tracking off if on (per RA)

ActiveDocument.TrackRevisions = False

Selection.Find.ClearFormatting

With Selection.Find

.Text = "-"

' Find the next hyphen

.Replacement.Text = ""

.Forward = True

.Wrap = wdFindContinue

.Format = True

.MatchCase = False

.MatchWholeWord = False

.MatchWildcards = False

.MatchSoundsLike = False

.MatchAllWordForms = False

End With

Selection.Find.Execute

' Hyphen is now selected

Selection.TypeText Text:="–"

' Replace with en dash

' Sets Track Changes to same state as before macro run (per RA)

ActiveDocument.TrackRevisions = blnTR

End Sub

Sub A\_201\_AutoTrackingShell()

'

' Remembers the current state of tracking (on or off)

' Turns tracking on or off (whichever you specify)

' Runs your macro

' Puts tracking back the way it was

'

' Macro orig. recorded 9/26/2004 by Hilary Powers

' Determine state of tracking (per Rich Adin, 8/14/2006)

Dim blnTR As Boolean

blnTR = ActiveDocument.TrackRevisions

' Turn tracking off if on; leave it off if it already was (per RA)

ActiveDocument.TrackRevisions = False

' PUT YOUR MACRO GUTS HERE

' Set Track Changes to same state as before macro run (per RA)

ActiveDocument.TrackRevisions = blnTR

ActiveDocument.Save

' ^^^^^^^ PUT THIS IN AS MANY MACROS AS YOU CAN

' (unless it slows Word down, as in frequently used "spot" macros")

End Sub

Sub A\_201\_LoopShell()

' Finds something and then does something based on what it finds

' Repeats until it doesn't find the thing anymore

ActiveDocument.TrackRevisions = False

Selection.Find.ClearFormatting

Selection.Find.Replacement.ClearFormatting

With Selection.Find

.Text = "PUT WHAT YOU WANT TO FIND HERE"

' Insert search input within quotes above

' This is the contents of the Find box

.Replacement.Text = ""

.Forward = True

.Wrap = wdFindContinue

.Format = False

.MatchCase = False

.MatchWholeWord = False

.MatchAllWordForms = False

.MatchSoundsLike = False

.MatchWildcards = True

'Everything above here is all the options you choose for your Find operation

End With

Selection.Find.Execute

'This line does the first find operation

Do While Selection.Find.Found

' PUT THE GUTS OF YOUR MACRO HERE

' The macro is executed here

Selection.Find.Execute

'This line performs the Find operation that was loaded into the Find box above

'And finds the next instance of what you want to find

Loop

'Loops back up to "Do While ..."

'You can string a bunch of loops one after the other in a single macro to repeat several operations

'As in the LoopApplyOutlineLevels macro

End Sub

Sub Loop\_ApplyOutlineLevels()

'Makes several passes through the document looking for inline codes to apply outline levels

'This creates an outline in the Document Map/Navigation pane

'Codes (<pn>, etc.) have already been manually inserted

'You decide what level to apply to each code

'I don't assign a keyboard shortcut to this because I run it only once per document

'First loop: part number (<pn>)

Selection.Find.ClearFormatting

Selection.Find.Replacement.ClearFormatting

With Selection.Find

.Text = "<pn>"

.Replacement.Text = ""

.Forward = True

.Wrap = wdFindContinue

.Format = False

.MatchCase = False

.MatchWholeWord = False

.MatchWildcards = False

.MatchSoundsLike = False

.MatchAllWordForms = False

End With

Selection.Find.Execute

' First find operation; <pn> is selected

Do While Selection.Find.Found

' Begin first loop

Selection.HomeKey Unit:=wdLine

' Move cursor to beginning of line

With Selection.ParagraphFormat

.OutlineLevel = wdOutlineLevel1

' Change outline level to Level 1

End With

Selection.MoveDown Unit:=wdParagraph, Count:=1

' Move cursor down to beginning of next paragraph

Selection.Find.Execute

Loop

'End of first loop cycle.

'Macro continues to pind the remaining <pn>s in the doc and applies Level 1 to each

'Until it reaches the end of the doc

'Moving on ...

Selection.HomeKey Unit:=wdStory

' Back to top of file to start again

'Next loop: part title (<pt>)

Selection.Find.ClearFormatting

Selection.Find.Replacement.ClearFormatting

With Selection.Find

.Text = "<pt>"

.Replacement.Text = ""

.Forward = True

.Wrap = wdFindContinue

.Format = False

.MatchCase = False

.MatchWholeWord = False

.MatchWildcards = False

.MatchSoundsLike = False

.MatchAllWordForms = False

End With

Selection.Find.Execute

' Next find operation; <pt> is selected

Do While Selection.Find.Found

' Begin first loop

Selection.HomeKey Unit:=wdLine

' Move cursor to beginning of line

With Selection.ParagraphFormat

.OutlineLevel = wdOutlineLevel1

' Change outline level to Level 1

End With

Selection.MoveDown Unit:=wdParagraph, Count:=1

' Move cursor down to beginning of next paragraph

Selection.Find.Execute

Loop

'Loop continues to end of doc

Selection.HomeKey Unit:=wdStory

'Back to top of file

'Next loop: chapter number (<cn>)

'I'll stop annotating here; follow along on your own!

Selection.Find.ClearFormatting

Selection.Find.Replacement.ClearFormatting

With Selection.Find

.Text = "<cn>"

.Replacement.Text = ""

.Forward = True

.Wrap = wdFindContinue

.Format = False

.MatchCase = False

.MatchWholeWord = False

.MatchWildcards = False

.MatchSoundsLike = False

.MatchAllWordForms = False

End With

Selection.Find.Execute

Do While Selection.Find.Found

Selection.HomeKey Unit:=wdLine

With Selection.ParagraphFormat

.OutlineLevel = wdOutlineLevel1

End With

Selection.MoveDown Unit:=wdParagraph, Count:=1

Selection.Find.Execute

Loop

Selection.HomeKey Unit:=wdStory

'Back to top of file

'Next loop: chapter title

Selection.Find.ClearFormatting

Selection.Find.Replacement.ClearFormatting

With Selection.Find

.Text = "<ct>"

.Replacement.Text = ""

.Forward = True

.Wrap = wdFindContinue

.Format = False

.MatchCase = False

.MatchWholeWord = False

.MatchWildcards = False

.MatchSoundsLike = False

.MatchAllWordForms = False

End With

Selection.Find.Execute

Do While Selection.Find.Found

Selection.HomeKey Unit:=wdLine

With Selection.ParagraphFormat

.OutlineLevel = wdOutlineLevel1

End With

Selection.MoveDown Unit:=wdParagraph, Count:=1

Selection.Find.Execute

Loop

Selection.HomeKey Unit:=wdStory

'Back to top of file

'Next loop: head level 1

Selection.Find.ClearFormatting

Selection.Find.Replacement.ClearFormatting

With Selection.Find

.Text = "<h1>"

.Replacement.Text = ""

.Forward = True

.Wrap = wdFindContinue

.Format = False

.MatchCase = False

.MatchWholeWord = False

.MatchWildcards = False

.MatchSoundsLike = False

.MatchAllWordForms = False

End With

Selection.Find.Execute

Do While Selection.Find.Found

Selection.HomeKey Unit:=wdLine

With Selection.ParagraphFormat

.OutlineLevel = wdOutlineLevel1

End With

Selection.MoveDown Unit:=wdParagraph, Count:=1

Selection.Find.Execute

Loop

Selection.HomeKey Unit:=wdStory

'Back to top of file

'Next loop: head level 2

Selection.Find.ClearFormatting

Selection.Find.Replacement.ClearFormatting

With Selection.Find

.Text = "<h2>"

.Replacement.Text = ""

.Forward = True

.Wrap = wdFindContinue

.Format = False

.MatchCase = False

.MatchWholeWord = False

.MatchWildcards = False

.MatchSoundsLike = False

.MatchAllWordForms = False

End With

Selection.Find.Execute

Do While Selection.Find.Found

Selection.HomeKey Unit:=wdLine

With Selection.ParagraphFormat

.OutlineLevel = wdOutlineLevel2

End With

Selection.MoveDown Unit:=wdParagraph, Count:=1

Selection.Find.Execute

Loop

Selection.HomeKey Unit:=wdStory

'Back to top of file

'Next loop: head level 3

Selection.Find.ClearFormatting

Selection.Find.Replacement.ClearFormatting

With Selection.Find

.Text = "<h3>"

.Replacement.Text = ""

.Forward = True

.Wrap = wdFindContinue

.Format = False

.MatchCase = False

.MatchWholeWord = False

.MatchWildcards = False

.MatchSoundsLike = False

.MatchAllWordForms = False

End With

Selection.Find.Execute

Do While Selection.Find.Found

Selection.HomeKey Unit:=wdLine

With Selection.ParagraphFormat

.OutlineLevel = wdOutlineLevel3

End With

Selection.MoveDown Unit:=wdParagraph, Count:=1

Selection.Find.Execute

Loop

Selection.HomeKey Unit:=wdStory

'Back to top of file

'Next loop: head level 4

Selection.Find.ClearFormatting

Selection.Find.Replacement.ClearFormatting

With Selection.Find

.Text = "<h4>"

.Replacement.Text = ""

.Forward = True

.Wrap = wdFindContinue

.Format = False

.MatchCase = False

.MatchWholeWord = False

.MatchWildcards = False

.MatchSoundsLike = False

.MatchAllWordForms = False

End With

Selection.Find.Execute

Do While Selection.Find.Found

Selection.HomeKey Unit:=wdLine

With Selection.ParagraphFormat

.OutlineLevel = wdOutlineLevel4

End With

Selection.MoveDown Unit:=wdParagraph, Count:=1

Selection.Find.Execute

Loop

Selection.HomeKey Unit:=wdStory

'Back to top of file

'Next loop: sidebar box title

Selection.Find.ClearFormatting

Selection.Find.Replacement.ClearFormatting

With Selection.Find

.Text = "<sbt>"

.Replacement.Text = ""

.Forward = True

.Wrap = wdFindContinue

.Format = False

.MatchCase = False

.MatchWholeWord = False

.MatchWildcards = False

.MatchSoundsLike = False

.MatchAllWordForms = False

End With

Selection.Find.Execute

Do While Selection.Find.Found

Selection.HomeKey Unit:=wdLine

With Selection.ParagraphFormat

.OutlineLevel = wdOutlineLevel2

End With

Selection.MoveDown Unit:=wdParagraph, Count:=1

Selection.Find.Execute

Loop

Selection.HomeKey Unit:=wdStory

'Back to top of file

'Next loop: unnumbered list head

Selection.Find.ClearFormatting

Selection.Find.Replacement.ClearFormatting

With Selection.Find

.Text = "<ulh>"

.Replacement.Text = ""

.Forward = True

.Wrap = wdFindContinue

.Format = False

.MatchCase = False

.MatchWholeWord = False

.MatchWildcards = False

.MatchSoundsLike = False

.MatchAllWordForms = False

End With

Selection.Find.Execute

Do While Selection.Find.Found

Selection.HomeKey Unit:=wdLine

With Selection.ParagraphFormat

.OutlineLevel = wdOutlineLevel2

End With

Selection.MoveDown Unit:=wdParagraph, Count:=1

Selection.Find.Execute

Loop

Selection.HomeKey Unit:=wdStory

'Back to top of file

'Next loop: box title

Selection.Find.ClearFormatting

Selection.Find.Replacement.ClearFormatting

With Selection.Find

.Text = "<bxt>"

.Replacement.Text = ""

.Forward = True

.Wrap = wdFindContinue

.Format = False

.MatchCase = False

.MatchWholeWord = False

.MatchWildcards = False

.MatchSoundsLike = False

.MatchAllWordForms = False

End With

Selection.Find.Execute

Do While Selection.Find.Found

Selection.HomeKey Unit:=wdLine

With Selection.ParagraphFormat

.OutlineLevel = wdOutlineLevel2

End With

Selection.MoveDown Unit:=wdParagraph, Count:=1

Selection.Find.Execute

Loop

Selection.HomeKey Unit:=wdStory

'Back to top of file

'Next loop: recipe subhead

Selection.Find.ClearFormatting

Selection.Find.Replacement.ClearFormatting

With Selection.Find

.Text = "<rsh>"

.Replacement.Text = ""

.Forward = True

.Wrap = wdFindContinue

.Format = False

.MatchCase = False

.MatchWholeWord = False

.MatchWildcards = False

.MatchSoundsLike = False

.MatchAllWordForms = False

End With

Selection.Find.Execute

Do While Selection.Find.Found

Selection.HomeKey Unit:=wdLine

With Selection.ParagraphFormat

.OutlineLevel = wdOutlineLevel2

End With

Selection.MoveDown Unit:=wdParagraph, Count:=1

Selection.Find.Execute

Loop

Selection.HomeKey Unit:=wdStory

'Back to top of file

'Next loop: recipe title

Selection.Find.ClearFormatting

Selection.Find.Replacement.ClearFormatting

With Selection.Find

.Text = "<rt>"

.Replacement.Text = ""

.Forward = True

.Wrap = wdFindContinue

.Format = False

.MatchCase = False

.MatchWholeWord = False

.MatchWildcards = False

.MatchSoundsLike = False

.MatchAllWordForms = False

End With

Selection.Find.Execute

Do While Selection.Find.Found

Selection.HomeKey Unit:=wdLine

With Selection.ParagraphFormat

.OutlineLevel = wdOutlineLevel3

End With

Selection.MoveDown Unit:=wdParagraph, Count:=1

Selection.Find.Execute

Loop

Selection.HomeKey Unit:=wdStory

'Final trip back to the top of the file

End Sub

Sub DialogBox\_Bookmarks()

'Creates a dialog box with choices for going to bookmarks in a document

'(in this case, a style sheet)

'The bookmarks are already placed in the document

'Dialog box enables you to go to any section in a long doc with only a few keystrokes

'Assign one keyboard shortcut to the dialog box macro (I use Alt + /)

'The symbol in each prompt is what you then type (plus Enter) to go to the bookmark

'Note that the first choice's key is /, same as the dialog box shortcut

'So I can hit Alt + / + /, or just Alt + / + Enter, for the first choice

'Alphabetical (the word list) tends to be the section I go to the most

'This code sets up the dialog box

Dim DialogTitle

Dim Prompt As String

Dim UserResp As String

DialogTitle = "AJS Style Sheet Bookmarks"

'This sets up the text that appears in the box

'You edit the text within the quotes

'I put parens around the prompts simply as a visual aid; you can leave them out if you want

'Make sure the characters to be typed match the Case choices below

'"+ vbCrLf" just means end of line (Visual Basic Carriage Return Line Feed)

Prompt = "( / ) Alphabetical" + vbCrLf

Prompt = Prompt & "( . ) Abbreviations" + vbCrLf

Prompt = Prompt & "( [ ) Publishers" + vbCrLf

Prompt = Prompt & "( ] ) Periodicals" + vbCrLf

Prompt = Prompt & "( , ) Names" + vbCrLf

Prompt = Prompt & "( ; ) Places" + vbCrLf

Prompt = Prompt & "( o ) Organizations" + vbCrLf

Prompt = Prompt & vbCrLf

'^^^This is a blank line, again, just as a visual aid

Prompt = Prompt & "( c ) Capitalization" + vbCrLf

Prompt = Prompt & "( g ) Grammar" + vbCrLf

Prompt = Prompt & "( n ) Numbers" + vbCrLf

Prompt = Prompt & "( p ) Punctuation" + vbCrLf

Prompt = Prompt & "( s ) Spelling" + vbCrLf

Prompt = Prompt & "( m ) Miscellaneous" + vbCrLf

Prompt = Prompt & "( d ) Address"

'Again, you can edit the text within quotes in the following line

UserResp = InputBox(Prompt$, "Go to bookmark:")

Select Case Left(UserResp, 1)

'And here are the choices:

'Again, first choice includes an option of "", which mean you can just hit enter

Case "/", ""

BKMKAlpha

'BKMKAlpha is a macro \*in this module\* that goes to the "Alphabetical" bookmark

'This is a case of a macro calling another macro!

'You can add/remove choices as you see fit

'Just make sure that the choices shown in the dialog box and the Case options match

'They don't have to be in the same order, but it makes things easier to follow if they are

Case "."

BKMKAbbrev

Case "["

BKMKPublishers

Case "]"

BKMKPeriodicals

Case ","

BKMKNames

Case ";"

BKMKPlaces

Case "o"

BKMKOrganizations

Case "c"

BKMKCapitalization

Case "g"

BKMKGrammar

Case "n"

BKMKNumbers

Case "p"

BKMKPunctuation

Case "s"

BKMKSpelling

Case "m"

BKMKMiscellaneous

Case "d"

BKMKAddress

End Select

End Sub

Sub A\_201\_DialogBoxShell()

'Based on code by Hilary Powers; additions by Howard Silcock

Dim DialogTitle

Dim Prompt As String

Dim UserResp As String

' Defining variables

DialogTitle = "NYCPub Codez"

' Title of dialog box

Prompt = "Press R for Red" + vbCrLf

Prompt = Prompt & "Press B for Blue" + vbCrLf

Prompt = Prompt & "Press G for Green"

' Put what you want the dialog to ask you here

' See more detailed notes in the DialogBoxBookmarks macro

UserResp = InputBox(Prompt$, "What's your favorite color?")

Select Case Left(UserResp, 1)

'Put the macro guts after each "Case" selection

'The "" means you can just hit Enter for the first choice

'So make the thing you choose most often the first option; one less keystroke

'You must hit Enter if you enter a value, to activate it

Case "r", ""

Selection.TypeText Text:="I like red!"

Selection.TypeParagraph

Case "b"

Selection.TypeText Text:="I like blue!"

Selection.TypeParagraph

Case "g"

Selection.TypeText Text:="I like green!"

Selection.TypeParagraph

End Select

End Sub

Sub DialogBox\_HighlightColors()

'This dialog box gives you a choice of highlight colors

'The text to be higlighted must be selected before you activate the dialog box

'I've assigned this dialog box macro to Ctrl + Alt + H

'Second keystroke is then y (or Enter), p, g, or 0

Dim DialogTitle

Dim Prompt As String

Dim UserResp As String

DialogTitle = "Choose highlight color:"

Prompt = "y. Yellow" + vbCrLf

Prompt = Prompt & "p. Pink" + vbCrLf

Prompt = Prompt & "g. Green" + vbCrLf

Prompt = Prompt & "0. None"

UserResp = InputBox(Prompt$, "Choose highlight color:")

Select Case Left(UserResp, 1)

Case "y", ""

HLTYellow

'Again, these are macros in this module that highlight the text in the selected color

'Note that the highlighting macros use the "turn tracking off" shell

Case "p"

HLTPink

Case "g"

HLTGreen

Case "0"

HLTNone

End Select

End Sub

Sub BKMKAbbrev()

'In style sheet, goes to "Abbreviations" bookmark

Selection.GoTo What:=wdGoToBookmark, Name:="Abbreviations"

Selection.Find.ClearFormatting

With Selection.Find

.Text = ""

.Replacement.Text = ""

.Forward = True

.Wrap = wdFindContinue

.Format = False

.MatchCase = False

.MatchWholeWord = False

.MatchWildcards = False

.MatchSoundsLike = False

.MatchAllWordForms = False

End With

End Sub

Sub BKMKPlaces()

'In style sheet, goes to "Places" bookmark

Selection.GoTo What:=wdGoToBookmark, Name:="Places"

Selection.Find.ClearFormatting

With Selection.Find

.Text = ""

.Replacement.Text = ""

.Forward = True

.Wrap = wdFindContinue

.Format = False

.MatchCase = False

.MatchWholeWord = False

.MatchWildcards = False

.MatchSoundsLike = False

.MatchAllWordForms = False

End With

End Sub

Sub BKMKPublishers()

'In style sheet, goes to "Publishers" bookmark

Selection.GoTo What:=wdGoToBookmark, Name:="Publishers"

Selection.Find.ClearFormatting

With Selection.Find

.Text = ""

.Replacement.Text = ""

.Forward = True

.Wrap = wdFindContinue

.Format = False

.MatchCase = False

.MatchWholeWord = False

.MatchWildcards = False

.MatchSoundsLike = False

.MatchAllWordForms = False

End With

End Sub

Sub BKMKPeriodicals()

'In style sheet, goes to "Publishers" bookmark

Selection.GoTo What:=wdGoToBookmark, Name:="Periodicals"

Selection.Find.ClearFormatting

With Selection.Find

.Text = ""

.Replacement.Text = ""

.Forward = True

.Wrap = wdFindContinue

.Format = False

.MatchCase = False

.MatchWholeWord = False

.MatchWildcards = False

.MatchSoundsLike = False

.MatchAllWordForms = False

End With

End Sub

Sub BKMKAlpha()

'In style sheet, goes to "Alphabetical" bookmark

Selection.GoTo What:=wdGoToBookmark, Name:="Alphabetical"

Selection.Find.ClearFormatting

With Selection.Find

.Text = ""

.Replacement.Text = ""

.Forward = True

.Wrap = wdFindContinue

.Format = False

.MatchCase = False

.MatchWholeWord = False

.MatchWildcards = False

.MatchSoundsLike = False

.MatchAllWordForms = False

End With

End Sub

Sub BKMKNames()

'In style sheet, goes to "Names" bookmark

Selection.GoTo What:=wdGoToBookmark, Name:="Names"

Selection.Find.ClearFormatting

With Selection.Find

.Text = ""

.Replacement.Text = ""

.Forward = True

.Wrap = wdFindContinue

.Format = False

.MatchCase = False

.MatchWholeWord = False

.MatchWildcards = False

.MatchSoundsLike = False

.MatchAllWordForms = False

End With

End Sub

Sub BKMKNumbers()

'In style sheet, goes to "Numbers" bookmark

Selection.GoTo What:=wdGoToBookmark, Name:="Numbers"

Selection.Find.ClearFormatting

With Selection.Find

.Text = ""

.Replacement.Text = ""

.Forward = True

.Wrap = wdFindContinue

.Format = False

.MatchCase = False

.MatchWholeWord = False

.MatchWildcards = False

.MatchSoundsLike = False

.MatchAllWordForms = False

End With

End Sub

Sub BKMKPunctuation()

'In style sheet, goes to "Punctuation" bookmark

Selection.GoTo What:=wdGoToBookmark, Name:="Punctuation"

Selection.Find.ClearFormatting

With Selection.Find

.Text = ""

.Replacement.Text = ""

.Forward = True

.Wrap = wdFindContinue

.Format = False

.MatchCase = False

.MatchWholeWord = False

.MatchWildcards = False

.MatchSoundsLike = False

.MatchAllWordForms = False

End With

End Sub

Sub BKMKOrganizations()

'In style sheet, goes to "Organizations" bookmark

Selection.GoTo What:=wdGoToBookmark, Name:="Organizations"

Selection.Find.ClearFormatting

With Selection.Find

.Text = ""

.Replacement.Text = ""

.Forward = True

.Wrap = wdFindContinue

.Format = False

.MatchCase = False

.MatchWholeWord = False

.MatchWildcards = False

.MatchSoundsLike = False

.MatchAllWordForms = False

End With

End Sub

Sub BKMKSpelling()

'In style sheet, goes to "Spelling" bookmark

Selection.GoTo What:=wdGoToBookmark, Name:="Spelling"

Selection.Find.ClearFormatting

With Selection.Find

.Text = ""

.Replacement.Text = ""

.Forward = True

.Wrap = wdFindContinue

.Format = False

.MatchCase = False

.MatchWholeWord = False

.MatchWildcards = False

.MatchSoundsLike = False

.MatchAllWordForms = False

End With

End Sub

Sub BKMKCapitalization()

'In style sheet, goes to "Capitalization" bookmark

Selection.GoTo What:=wdGoToBookmark, Name:="Capitalization"

Selection.Find.ClearFormatting

With Selection.Find

.Text = ""

.Replacement.Text = ""

.Forward = True

.Wrap = wdFindContinue

.Format = False

.MatchCase = False

.MatchWholeWord = False

.MatchWildcards = False

.MatchSoundsLike = False

.MatchAllWordForms = False

End With

End Sub

Sub BKMKGrammar()

'In style sheet, goes to "Grammar" bookmark

Selection.GoTo What:=wdGoToBookmark, Name:="Grammar"

Selection.Find.ClearFormatting

With Selection.Find

.Text = ""

.Replacement.Text = ""

.Forward = True

.Wrap = wdFindContinue

.Format = False

.MatchCase = False

.MatchWholeWord = False

.MatchWildcards = False

.MatchSoundsLike = False

.MatchAllWordForms = False

End With

End Sub

Sub BKMKMiscellaneous()

'In style sheet, goes to "Miscellaneous" bookmark

Selection.GoTo What:=wdGoToBookmark, Name:="Miscellaneous"

Selection.Find.ClearFormatting

With Selection.Find

.Text = ""

.Replacement.Text = ""

.Forward = True

.Wrap = wdFindContinue

.Format = False

.MatchCase = False

.MatchWholeWord = False

.MatchWildcards = False

.MatchSoundsLike = False

.MatchAllWordForms = False

End With

End Sub

Sub BKMKAddress()

'In style sheet, goes to "Address" bookmark

Selection.GoTo What:=wdGoToBookmark, Name:="Address"

Selection.Find.ClearFormatting

With Selection.Find

.Text = ""

.Replacement.Text = ""

.Forward = True

.Wrap = wdFindContinue

.Format = False

.MatchCase = False

.MatchWholeWord = False

.MatchWildcards = False

.MatchSoundsLike = False

.MatchAllWordForms = False

End With

End Sub

Sub HLTYellow()

' Determine state of tracking (per Rich Adin, 8/14/2006)

Dim blnTR As Boolean

blnTR = ActiveDocument.TrackRevisions

' Turn tracking off if on (per RA)

ActiveDocument.TrackRevisions = False

Options.DefaultHighlightColorIndex = wdYellow

Selection.Range.HighlightColorIndex = wdYellow

Selection.MoveRight Unit:=wdCharacter, Count:=1

' Sets Track Changes to same state as before macro run (per RA)

ActiveDocument.TrackRevisions = blnTR

End Sub

Sub HLTGreen()

' Determine state of tracking (per Rich Adin, 8/14/2006)

Dim blnTR As Boolean

blnTR = ActiveDocument.TrackRevisions

' Turn tracking off if on (per RA)

ActiveDocument.TrackRevisions = False

Options.DefaultHighlightColorIndex = wdBrightGreen

Selection.Range.HighlightColorIndex = wdBrightGreen

Selection.MoveRight Unit:=wdCharacter, Count:=1

ActiveDocument.TrackRevisions = True

' Sets Track Changes to same state as before macro run (per RA)

ActiveDocument.TrackRevisions = blnTR

End Sub

Sub HLTPink()

' Determine state of tracking (per Rich Adin, 8/14/2006)

Dim blnTR As Boolean

blnTR = ActiveDocument.TrackRevisions

' Turn tracking off if on (per RA)

ActiveDocument.TrackRevisions = False

Options.DefaultHighlightColorIndex = wdPink

Selection.Range.HighlightColorIndex = wdPink

Selection.MoveRight Unit:=wdCharacter, Count:=1

ActiveDocument.TrackRevisions = True

' Sets Track Changes to same state as before macro run (per RA)

ActiveDocument.TrackRevisions = blnTR

End Sub

Sub HLTNone()

' Determine state of tracking (per Rich Adin, 8/14/2006)

Dim blnTR As Boolean

blnTR = ActiveDocument.TrackRevisions

' Turn tracking off if on (per RA)

ActiveDocument.TrackRevisions = False

Options.DefaultHighlightColorIndex = wdNoHighlight

Selection.Range.HighlightColorIndex = wdNoHighlight

Selection.MoveRight Unit:=wdCharacter, Count:=1

' Sets Track Changes to same state as before macro run (per RA)

ActiveDocument.TrackRevisions = blnTR

End Sub

Sub A\_201\_ConverterShell()

' Based on a macro written by Hilary Powers, June 2000

' You must select text first; it must be in this list after "Case"

' Case sensitive (e.g., separate entries for "while" and "While")

' Replaces text only, with original formatting

' Use copy/paste to expand/insert sections

' No quotes needed in "Case" line if finding digits only

'Ctrl + Num+

'The following code sets up the dialog box and macro function; leave as is

Dim vNumerals As String

If Selection.Type = wdSelectionIP Then

MsgBox "Select something to convert first."

GoTo Last

Else

vNumerals = Selection.Text

End If

Select Case vNumerals

' You can add code after this using copy and paste

Case 1

Selection.TypeText Text:="one"

' The two lines of code above are what you must insert/edit for each item you add

Case 2

Selection.TypeText Text:="two"

Case 10

Selection.TypeText Text:="ten"

Case 20

Selection.TypeText Text:="twenty"

Case "1st"

Selection.TypeText Text:="first"

Case "30s"

Selection.TypeText Text:="thirties"

Case 100

Selection.TypeText Text:="a hundred"

Case 0

Selection.TypeText Text:=" hundred"

Case 0

Selection.TypeText Text:=" thousand"

Case ",000"

Selection.TypeText Text:=" thousand"

Case "00,000"

Selection.TypeText Text:=" hundred thousand"

Case "10,000"

Selection.TypeText Text:="ten thousand"

Case "100,000"

Selection.TypeText Text:="one hundred thousand"

Case "U.S."

Selection.TypeText Text:="United States"

Case "United States"

Selection.TypeText Text:="U.S."

Case "which"

Selection.TypeText Text:="that"

Case "that"

Selection.TypeText Text:="who"

Case "which"

Selection.TypeText Text:="although"

Case "Which"

Selection.TypeText Text:="Although"

'What other things are you always changing from one thing to another?

'State abbreviations, recipe measurements

'Times, symbols, abbreviations

'Words you never type correctly the first time

'Editable code ends here

End Select

Last:

End Sub

Sub DialogBox\_ApplyNYCPubCodez()

'Based on code by Hilary Powers; additions by Howard Silcock

'Dialog box offers choices for applying client styles

'Styles are already defined in attached template

'See DialogBoxBookmarks macro for general annotations

Dim DialogTitle

Dim Prompt As String

Dim UserResp As String

DialogTitle = "NYCPub Codez"

Prompt = "i. Italic" + vbCrLf

Prompt = Prompt & "b. Bold" + vbCrLf

Prompt = Prompt & "m. Small caps" + vbCrLf

Prompt = Prompt & "u. Subscript" + vbCrLf

Prompt = Prompt & "p. Superscript" + vbCrLf

Prompt = Prompt + vbCrLf

Prompt = Prompt & "c. CO Body Text" + vbCrLf

Prompt = Prompt & "s. Space Break FL" + vbCrLf

Prompt = Prompt & "t. Body Text"

UserResp = InputBox(Prompt$, "Apply style:")

Select Case Left(UserResp, 1)

Case "i", ""

Selection.Style = ActiveDocument.Styles("ITAL")

'ITAL is the client's defined character style for italic

'Text must be selected before activating the dialog box for character styles

Case "b"

Selection.Style = ActiveDocument.Styles("BOLD")

Case "m"

Selection.Style = ActiveDocument.Styles("SCAP")

Case "u"

Selection.Style = ActiveDocument.Styles("SUB")

Case "p"

Selection.Style = ActiveDocument.Styles("SUP")

'The following styles are paragraph styles

'Cursor must be in the paragraph (anywhere) before dialog box is activated

Case "c"

Selection.Style = ActiveDocument.Styles("03 CO Body Text")

Case "s"

Selection.Style = ActiveDocument.Styles("04 Space Break FL")

Case "t"

Selection.Style = ActiveDocument.Styles("04 Body Text")

End Select

End Sub

Sub AutoTracking\_ITALuntracked()

'This is one of three macros that apply a character style UNTRACKED

'Use it for cleaning up stray italic/bold/roman that does not require author attention

'Such as spaces, punctuation, partial words

'Because it's untracked, you do not have to select precisely,

'as long as everything you select is what you're changing to

'So for a stray roman comma and space within an italic passage,

'just select from a few characters before the roman to a few characters after

'The change won't show and the text will be clean

'and you won't have spent time selecting those exact characters.

'Also good for cleaning up multiple strays in a row;

'just select the whole paragraph or whatever and BAM it's all italic

' Determine state of tracking (per Rich Adin)

Dim blnTR As Boolean

blnTR = ActiveDocument.TrackRevisions

' Turn tracking off if on (per Rich Adin)

ActiveDocument.TrackRevisions = False

Selection.Style = ActiveDocument.Styles("ITAL")

' Macro guts is just one line!

' Sets Track Changes to same state as before macro run (per RA)

ActiveDocument.TrackRevisions = blnTR

End Sub

Sub AutoTracking\_DPFuntracked()

'See AutoTracking\_ITALuntracked for annotations

' Determine state of tracking (per Rich Adin)

Dim blnTR As Boolean

blnTR = ActiveDocument.TrackRevisions

' Turn tracking off if on (per Rich Adin)

ActiveDocument.TrackRevisions = False

Selection.Style = ActiveDocument.Styles("Default Paragraph Font")

' Sets Track Changes to same state as before macro run (per RA)

ActiveDocument.TrackRevisions = blnTR

End Sub

Sub AutoTracking\_BOLDuntracked()

'See AutoTracking\_ITALuntracked for annotations

' Determine state of tracking (per Rich Adin)

Dim blnTR As Boolean

blnTR = ActiveDocument.TrackRevisions

' Turn tracking off if on (per Rich Adin)

ActiveDocument.TrackRevisions = False

Selection.Style = ActiveDocument.Styles("BOLD")

' Sets Track Changes to same state as before macro run (per RA)

ActiveDocument.TrackRevisions = blnTR

End Sub