

Notes on using Sky with WordEmbed

by John Bealle

Fellow indexers - I have just completed a Word embedded index using WordEmbed with Sky. James Lamb was very helpful to me, and I would like to share some of the things I learned. Much of this is in the manual and may be common knowledge among seasoned pros. But I was caught off guard by the way embedding works. - John

Sorting

- Sorting is done when word embeds the index. It follows Word sorting rules and will not abide by any Sky settings.
- Using Sky ignore and hidden formatting has no effect on the Word embedded sort order. The only possible reason to use these is for convenience while working with the index.
- Word has equivalent codes for forced sorting: hidden = tilde; ignore = curly brackets. So, "~000~overview" will force this heading to the top. "{The }Road Not Taken" will file under "Road"—note that the space must be ignored or it will cause the entry to be sorted with special characters at the beginning.

Special characters

- There are several important characters that are used in Word indexing tags and cannot appear within index headings: colon, semicolon, quotation mark. Also, the comma, colon, and semicolon are interpreted by WordEmbed and used to construct Word index tags. This is discussed on p18 of the WordEmbed manual.
- It seems that WordEmbed distinguishes commas within heading text from those which precede locators. Thus a heading like "*Maltese Falcon, The*, 79.26, 113.88" is handled correctly. But the meaning of colons and semicolons cannot be determined without special coding. In the example given on p19, "Airplane II: The Sequel" would trigger an unintended main:sub interpretation.
- Remember that the Sky "Adjust Punctuation" setting moves commas inside quotes. In the generated index, the first locator should be preceded by a comma-space without any intervening characters.
- To use any of these characters in index heading text, you must use special coding. Colons and semicolons can be used literally with curly braces: "Airplane II{:} The Sequel" is the example in the manual. Quotation marks are preceded with a backslash. So \" (backslash-quote) would insert a literal straight quote in an entry and will prevent the quote from being interpreted by the Word index generator.
- From testing, going from Sky to a generated index to WordEmbed, it appears that using two backslash characters in Sky will produce four backslashes in the generated rtf, which is the proper number for escaping a character in WordEmbed.
- It is possible to create problems by editing the index in an rtf editor (Word, Wordpad, etc.); also, the number of backslash characters may be more than the editor displays. A text editor like

Wordpad will display the raw characters without processing them as rtf codes. If you have trouble with characters, you may want to Generate (not Edit/Print) and process the rtf in WordEmbed without editing.

- Remember that Word does not ignore leading quotation marks, so you also need to use brackets: `{\\"}{The }Road Not Taken{\\"}`

Embedding

- When you finish indexing and are ready to embed, save the raw indexed document in a safe place. Make a copy for embedding so that if things go wrong you have the original raw document to go back to.
- The embedding process is very resource intensive. A large index can be slow and complex to process. Program errors can be caused by a large file.
- If you are having trouble embedding, break your index into smaller parts.
 - - (1) Divide the index into smaller files, e.g., Index-A-F.rtf, Index-G-L.rtf, and so on.
 - - (2) You may separate the index between any two main headings (do not separate subheadings from their parent).
 - - (3) Embed the first part, e.g., Index-A-F.rtf. When it finishes successfully, save the file.
 - - (4) Embed the second part, e.g., Index-G-L.rtf.
 - - (5) The new entries won't appear until you update the index field codes. Right click anywhere in the index (not in the white space), and select "Update Field." You can also use F9 to update all codes in the document, but this process may take a long time to complete.
 - - (6) Continue adding segments until you are done.

Checklist

CODING (things to do during the indexing phase):

- - bracket leading articles in subheadings, "{as }Marxist critique"
- - straight quotes, not curly
- - tilde for forced sorting, e.g., "~nineteeneightyfour~1984"
- - no coding in cross references—Word will not process tilde or bracket coding in cross references
- - there are two Sky bugs that affect use of tilde. (1) search and replace Pattern Matching doesn't work with formatting, so you can't use it to replace hidden text with the WordEmbed sort character (Workaround: include the sort character inside the hidden text, then add that character to the Ignore Character list in Index Options so it doesn't sort, then use Wildmatch to remove hidden formatting at the end. This will be fixed in Sky 7.0.12. (2) tilde corrupts any formatting that follows (Workaround: a substitute character for the tilde)
- - tilde summary: (1) substitute character to avoid corrupting other formatting; (2) add substitute character to Ignore list in Index Options so it sorts correctly in Sky; (3) use hidden formatting on the whole string so it doesn't appear in the Preview Pane; (4) before generating use Wildmatch to remove hidden formatting; (5) then generate and change character to tilde in rtf

editor

DOCUMENT PREP (things to do before embedding):

- - bracket all commas (not sure if this affects sorting stops in Word)
- - bracket all colons and semicolons
- - only quotes need to be escaped: enter in Sky as {"} which may become {\}, then *just* before embedding open in text editor (not rtf editor) and change to {\\"}

EMBEDDING:

- - after generating the index in Sky, open with rtf editor (Wordpad)
- - do any format editing (such as tilde change if using Sky)
- - break the document into 800-lines-or-smaller chunks. On my computer 800 index lines took around 10 min—I felt this was the maximum time I wanted to spend risking something going wrong
- - before generating each chunk open the rtf with a text editor (Notepad) and change escaped quotes to {\\"}. Do not open with an rtf editor (Wordpad/Word) after you do this.

EDITING (if you need to edit the embedded index):

- - in the embedded Word doc, use Show Paragraphs to see field codes
- - edit text in the codes (not in the generated index)
- - use colons for subheadings
- - text is duplicated, separated by forward slash, in this manner: {XE: display text / sort text}

SICI Subroutines

I have developed several data-entry subroutines using Sky's SICI programming interface. These are extremely useful, mainly in filling in the Page field automatically. They run via Microsoft Word VBA within the indexed document, which happens also to be where WordEmbed runs also. So they are able to interact with both WordEmbed and Sky.

As a brief example, let's say you encounter the passage, "In his analysis, Jones follows the structural approach of Mary Douglas." If you wanted to index the name "Mary Douglas" you might select the name and then create the index tag using that text span. Then you'd paste the WordEmbed tag into the page field and type the name in Main. The SICI subroutine will create a new Sky record, fill in the Page field automatically, invert the phrase "Mary Douglas" to "Douglas, Mary", and fill it in the Main field—all without ever leaving the document.

To set up the macros in Microsoft Word, use the following steps:

- In the Word document, open the VBA editor with Tools > Macro > Visual Basic Editor (Alt-F11).
- Then use View > Project Explorer (Ctrl-R) to make sure the Project Explorer is displayed.
- Right-click "Modules" and select New > Module. Name is Sky-SICI or something similar.
- In the web browser below this text, select the VBA code with the mouse and copy it to the

clipboard. Go back to VBA and paste it in the new module.

- Click in the module and move the cursor down until it is inside the "Sub AssignMacroKeys()" subroutine—which is the first one after the introductory comments.
- Use Run > Run Sub/User Form (F5) to run AssignMacroKeys(). This will bind the key assignments so you'll be able to run the other macros from the keyboard rather than going into VBA.
- Now you're set—you can see at the top of the Module what they keys are programmed to do.

```
Option Compare Text
'Sub NewRecBlankMain()
'F1=LOCATOR ONLY
'Adds a new record with Main=empty, Page=clipboard
'Key Binding: Sft-Ctl-Alt-F1
'-
'Sub NewRec()
'F2=MAIN
'Adds a new record with Main=selected text, Page=clipboard
'Key Binding: Sft-Ctl-Alt-F2
'-
'Sub NewRecInvert()
'F3=INVERTED
'Adds a new record with Main=inverted name from selected text, Page=clipboard
'Key Binding: Sft-Ctl-Alt-F3
'-
'Sub NewRecWithSub()
'F4=INPUT SUB
'Adds a new record with Main=selected text, Sub=Input Box text, Page=clipboard
'Key Binding: Sft-Ctl-Alt-F4
'-
'Sub NewRecItalics()
'F6=MAIN ITALICS
'Adds a new record with Main=selected text in italics, Page=clipboard
'Key Binding: Sft-Ctl-Alt-F6
'-
Sub AssignMacroKeys()
CustomizationContext = NormalTemplate
KeyBindings.Add KeyCode:=BuildKeyCode(wdKeyAlt, wdKeyShift, wdKeyControl, wdKeyF1),
-
    KeyCategory:=wdKeyCategoryCommand, Command:="NewRecBlankMain"
KeyBindings.Add KeyCode:=BuildKeyCode(wdKeyAlt, wdKeyShift, wdKeyControl, wdKeyF2),
-
    KeyCategory:=wdKeyCategoryCommand, Command:="NewRec"
KeyBindings.Add KeyCode:=BuildKeyCode(wdKeyAlt, wdKeyShift, wdKeyControl, wdKeyF3),
-
    KeyCategory:=wdKeyCategoryCommand, Command:="NewRecInvert"
KeyBindings.Add KeyCode:=BuildKeyCode(wdKeyAlt, wdKeyShift, wdKeyControl, wdKeyF4),
-
    KeyCategory:=wdKeyCategoryCommand, Command:="NewRecWithSub"
KeyBindings.Add KeyCode:=BuildKeyCode(wdKeyAlt, wdKeyShift, wdKeyControl, wdKeyF6),
-
    KeyCategory:=wdKeyCategoryCommand, Command:="NewRecItalics"
MsgBox ("Sky Key Bindings Assigned")
End Sub
'=====
```

```

Sub SICI_Template()
'this is a template for Sky SICI VB/VBA programs
Dim CI As SICI10.Interface
Dim ST As New SICI10.StyledText 'only needed for styled text functiosn
On Error Resume Next
Set CI = New SICI10.Interface
If Not CI Is Nothing Then
    With CI
        If CI.RequestConnection Then

'YOUR PROGRAM GOES HERE

            Call CI.CloseConnection
        Else
            MsgBox "Could not create SICI Interface object."
        End If
    End With
End If
End Sub
'=====
Sub NewRecBlankMain()
'F1=LOCATOR ONLY
'Adds a new record with Main=empty, Page=clipboard
'Key Binding: Sft-Ctl-Alt-F1
Dim ClipHold As New DataObject
Dim CI As SICI10.Interface
Dim ST As New SICI10.StyledText 'only needed for styled text functiosn
On Error Resume Next
Set CI = New SICI10.Interface
If Not CI Is Nothing Then
    With CI
        If CI.RequestConnection Then
            Set ClipHold = New DataObject
            ClipHold.GetFromClipboard
            errx = CI.AddRow(" ", ClipHold.GetText)
            errx = CI.FirstCell
            errx = CI.LastRow
            Call CI.CloseConnection
            AppActivate "Sky"
        Else
            MsgBox "Could not create SICI Interface object."
        End If
    End With
End If
End Sub
'=====
Sub NewRec()
'Adds a new record with Main=selected text, Page=clipboard
'Key Binding: Sft-Ctl-Alt-F2
Dim ClipHold As New DataObject
Dim CI As SICI10.Interface
Dim ST As New SICI10.StyledText 'only needed for styled text functiosn
On Error Resume Next
Set CI = New SICI10.Interface
If Not CI Is Nothing Then
    With CI
        If CI.RequestConnection Then
            Set ClipHold = New DataObject
            ClipHold.GetFromClipboard

```

```

        errx = CI.AddRow(Selection.Text, ClipHold.GetText)
        Call CI.CloseConnection
    Else
        MsgBox "Could not create SICI Interface object."
    End If
End With
End If
End Sub
'=====
Sub NewRecInvert()
'Adds a new record with Main=inverted name from selected text, Page=clipboard
'Key Binding: Sft-Ctl-Alt-F3
Dim ClipHold As New DataObject
Dim CI As SICI10.Interface
Dim ST As New SICI10.StyledText 'only needed for styled text functiosn
Dim celltext As String
On Error Resume Next
Set CI = New SICI10.Interface
If Not CI Is Nothing Then
    With CI
        If CI.RequestConnection Then
            celltext = Selection.Text
            X = GetSetting("SkyNameInverter", "Forms", "Prefixes", "le|la|del|van|
de")
            prefixarr = Split(X, "|")
            X = GetSetting("SkyNameInverter", "Forms", "Suffixes", "Jr.|Sr.|III")
            suffixarr = Split(X, "|")
            'remove and save suffix
            regex.Pattern = "^(.+), (.*)$"
            suffixtext = regex.Replace(ST.PlainStr(celltext), "$2")
            For i = 0 To UBound(suffixarr)
                If suffixtext = suffixarr(i) Then
                    suffixtextsave = ST.RightStr(celltext, ST.LenStr(suffixtext))
                    celltext = ST.LeftStr(celltext, ST.LenStr(celltext) -
ST.LenStr(suffixtext))
                    celltext = ST.RTrimStr(celltext)
                    ST.SKYText = celltext
                    Call ST.RemoveTrailingPunctuation(",")
                    celltext = ST.SKYText
                    Exit For
                Else
                    If i = UBound(suffixarr) Then suffixtext = ""
                End If
            Next
            'bind prefix
            For i = 0 To UBound(prefixarr)
                If ST.InString(celltext, prefixarr(i) & " ") = 1 Then celltext =
ST.InsertStr(celltext, ST.InString(celltext, prefixarr(i) & " ") +
Len(prefixarr(i)), "|")
                If ST.InString(celltext, " " & prefixarr(i) & " ") > 0 Then
                    celltext = ST.InsertStr(celltext, ST.InString(celltext, " " & prefixarr(i) & " ") +
Len(prefixarr(i)), "|")
                If ST.InString(celltext, "|" & prefixarr(i) & " ") > 0 Then
                    celltext = ST.InsertStr(celltext, ST.InString(celltext, "|" & prefixarr(i) & " ") +
Len(prefixarr(i)), "|")
                Next 'now any prefix words are bound to what follows, e.g., "Manuel de|
la|Rosa"
                regex.Global = True
                regex.Pattern = "\s?\|\s?"
            End For
        End With
    End If
End Sub

```

```

        celltext = regex.Replace(celltext, "|")
        regex.Global = False
    'no commas (invert)
    If InStr(1, celltext, ",") = 0 Then
        CellTextArr = ST.SplitStr(celltext, " ")
        celltext = CellTextArr(UBound(CellTextArr))
        celltext = ST.InsertStr(celltext, 1000, ",")
        For i = 0 To UBound(CellTextArr) - 1
            celltext = ST.InsertStr(celltext, 1000, " ")
            celltext = ST.InsertStr(celltext & CellTextArr(i),
ST.LenStr(celltext), "")
        Next
    'comma (uninvert or shift)
    Else
        CellTextArr = ST.SplitStr(celltext, " ")
        celltext = CellTextArr(UBound(CellTextArr))
        For i = 0 To UBound(CellTextArr) - 1
            celltext = ST.InsertStr(celltext, 1000, " ")
            celltext = ST.InsertStr(celltext & CellTextArr(i),
ST.LenStr(celltext), "")
        Next
        ST.SKYText = celltext 'remove trailing comma
        Call ST.RemoveTrailingPunctuation(",")
        celltext = ST.SKYText
        regex.Pattern = "^([A-Z]\.) (.+,.+)$" 'now might have "M. Smith,
John"

        If regex.test(ST.PlainStr(celltext)) Then
            Do 'now might have "M. Smith, John"
                CellTextArr = ST.SplitStr(celltext, " ")
                celltext = CellTextArr(UBound(CellTextArr))
                For i = 0 To UBound(CellTextArr) - 1
                    celltext = ST.InsertStr(celltext, 1000, " ")
                    celltext = ST.InsertStr(celltext & CellTextArr(i),
ST.LenStr(celltext), " ")
                Next
                ST.SKYText = celltext 'remove trailing comma
                Call ST.RemoveTrailingPunctuation(",")
                celltext = ST.SKYText
            Loop While regex.test(ST.PlainStr(celltext)) 'test again, might
be "R. L. P. Stein"
        End If
    End If
    'clean up from prefixes and suffixes
    If suffixtextsave <> "" Then celltext =
ST.InsertStr(ST.InsertStr(celltext, 1000, " ") & suffixtextsave, 1000, "")
    'reattach suffix
    regex.Pattern = "\|" 'unbind prefix
    regex.Global = True
    celltext = regex.Replace(celltext, " ")
    regex.Global = False
    'write cell all done

    Set ClipHold = New DataObject
    ClipHold.GetFromClipboard
    errx = CI.AddRow(celltext, ClipHold.GetText)
    Call CI.CloseConnection
Else
    MsgBox "Could not create SICI Interface object."

```

```

        End If
    End With
End If
End Sub
'=====
Sub NewRecWithSub()
'Adds a new record with Main=selected text, Sub=Input Box text, Page=clipboard
'Key Binding: Sft-Ctl-Alt-F4
Dim ClipHold As New DataObject
Dim CI As SICI10.Interface
Dim EntryText As String
On Error Resume Next
Set CI = New SICI10.Interface
If Not CI Is Nothing Then
    With CI
        If CI.RequestConnection Then
            X = InputBox("Enter subheading text:")
            EntryText = Selection.Text & vbTab & X
            Set ClipHold = New DataObject
            ClipHold.GetFromClipboard
            errx = CI.AddRow(EntryText, ClipHold.GetText)
            Call CI.CloseConnection
        Else
            MsgBox "Could not create SICI Interface object."
        End If
    End With
End If
End Sub
'=====
Sub NewRecItalics()
'Adds a new record with Main=selected text, Page=clipboard
'Key Binding: Sft-Ctl-Alt-F6
Dim ClipHold As New DataObject
Dim SelHold As New DataObject
Dim CI As SICI10.Interface
Dim ST As New SICI10.StyledText 'only needed for styled text functiosn
On Error Resume Next
Set CI = New SICI10.Interface
If Not CI Is Nothing Then
    With CI
        If CI.RequestConnection Then
'YOUR PROGRAM GOES HERE

            Set ClipHold = New DataObject
            Set SelHold = New DataObject
            ClipHold.GetFromClipboard
            SelHold.SetText ("/i1" & Selection.Text & "/i0")
            errx = CI.AddRow(SelHold.GetText, ClipHold.GetText)
            Call CI.CloseConnection
        Else
            MsgBox "Could not create SICI Interface object."
        End If
    End With
End If
End Sub
'=====
Sub WordEmbedConvert()
'Prepares Sky index for WordEmbed
'Curly Brackets around commas, semicolons, colons

```



```

'Curly quotes converted to straight
'Quotes escaped and bracketed
Dim CI As SICI10.Interface
Dim ST As New SICI10.StyledText 'only needed for styled text functiosn
On Error Resume Next
Set CI = New SICI10.Interface
If Not CI Is Nothing Then
    With CI
        If CI.RequestConnection Then

'YOUR PROGRAM GOES HERE
Dim i As Integer, j As Integer
xerr = CI.LoadIndexRows()
For i = 1 To CI.IndexRows.Count
For j = 0 To 2
xrec = CI.IndexRows(i).Heading(j)
xrecnew = ""
    If xrec <> "" Then
        For k = 1 To Len(xrec)
            xchar = Mid(xrec, k, 1)
            Select Case xchar
                Case Chr(147)
                    xrecnew = xrecnew & "{\\\"}"
                Case Chr(148)
                    xrecnew = xrecnew & "{\\\"}"
                Case """"
                    xrecnew = xrecnew & "{\\\"}"
                Case ", "
                    xrecnew = xrecnew & "{,}"
                Case ":"
                    xrecnew = xrecnew & ":{:}"
                Case ";"
                    xrecnew = xrecnew & "{;}"
                Case Else
                    xrecnew = xrecnew & xchar
            End Select
        Next k
        CI.IndexRows(i).Heading(j) = xrecnew
    End If
Next j
Next i
CI.CommitRowChanges ("WordEmbed Setup")
'#####
        Call CI.CloseConnection
    Else
        MsgBox "Could not create SICI Interface object."
    End If
End With
End If
End Sub
'=====

```