PharmaSUG 2020 - Paper AD-055

A Set of VBA Macros to Compare RTF Files in a Batch

Jeff Xia, Merck & Co., Inc., Rahway, NJ, USA

Shunbing Zhao, Merck & Co., Inc., Rahway, NJ, USA

ABSTRACT

Post database lock changes in a clinical trial are impactful and can result in significant rework. Statistical programmers must access the updated data and regenerate the TLFs for CSR to maintain data and result traceability.

This paper briefly discusses the challenges in comparing two sets of RTF files before and after the post database lock changes, each set might contain tens or hundreds of files, and provides an efficient solution based on VBA technology. Three standalone VBA macros were developed to perform this essential task: 1) Compare: programmatically compare each RTF file in two different versions and record the track change(s); 2) Find_Change: scan every RTF file and see whether there are one or more changes between these two versions and produce a report to show whether a file has been changed or remains the same; 3) Change_Details: scan every RTF file and provides a report with all the details in the track changes for all RTFs with at least an update, i.e., what text was inserted, or deleted, etc. Each VBA macro will be described in this paper and reviewed with examples.

INTRODUCTION

If an error or inconsistency is observed in a locked database by clinical trial team and that error or inconsistency is in a critical data field, the database is required to be unlocked so that issue(s) can be addressed accordingly. This process is often called post database lock change(s).

Post database lock changes in clinical trials are impactful in many dimensions and can result in significant rework to produce tables, listings and figures (TLFs). To ensure the post database lock changes are correctly carried over to the TLF in CSR, it is very helpful to programmatically compare the TLFs that were generated using the original database against the one generated using the updated database, and find out what the difference is.

There are often tens or hundreds of TLFs in a single CSR, which requires significant effort and attention to compare them if done manually. This paper introduces three standalone VBA macros to relief the burden of manual comparison of the two versions of TLF: Compare, Find_Change and Change_Details.

VB MACRO 1: COMPARE

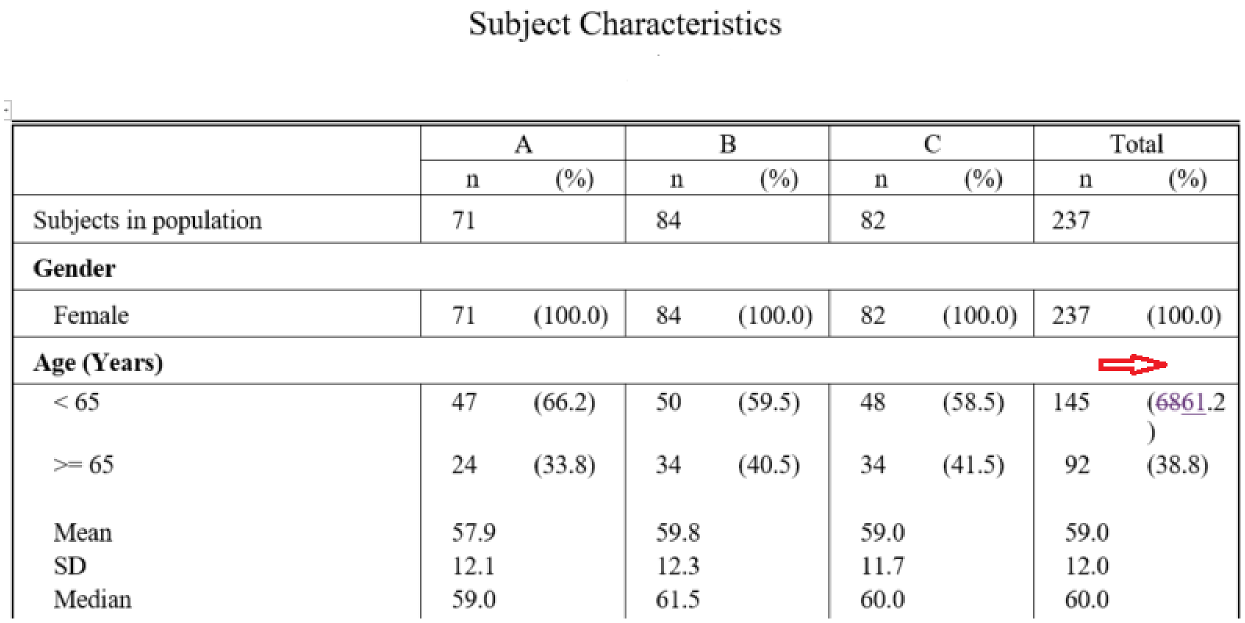
First, the VBA macro Compare asks the user to provide some key information about the comparison: 1) The path of the original RTF files, which were generated by using the original database; 2) The path of the revised RTF files, which were generated by using the updated database; 3) The path to save the comparison results. The macro then performs some basic error checking to validate the user inputs and ensures these folders exist before further operation. Once the validation is done, the macro compares each file in RTF format in the first folder against the one in the second folder, and saves the comparison result in the third folder. The VBA function to achieve the comparison is "Application.CompareDocuments". See below for the code snippet of the VBA macro Compare.

```
Sub Compare()
  Dim wd As Word.Application
  Dim odoc As Word.Document
  Dim rdoc As Word.Document
  Dim strOPath As String
  Dim strRPath As String
Dim strCPath As String
Dim strORTFfile As String
Dim ofiles() As String
Dim i As Integer

strOPath = InputBox("Please enter the folder of original documents:")
strRPath = InputBox("Please enter the folder of revised documents:")
strCPath = InputBox("Please enter the folder to save the comparison: ")

Set wd = GetObject(, "Word.Application")
If wd Is Nothing Then
  Set wd = CreateObject("Word.Application")
End If

ReDim Preserve ofiles(0)

strORTFfile = Dir(strOPath & "\" & ("*.rtf"), vbNormal)

Do While strORTFfile <> Empty
  ReDim Preserve ofiles(UBound(ofiles) + 1)
  ofiles(UBound(ofiles)) = strORTFfile
  strORTFfile = Dir
Loop

For i = 1 To UBound(ofiles)
  If Dir(strRPath & "\" & ofiles(i)) <> Empty Then
    Set odoc = wd.Documents.Open(strOPath & "\" & ofiles(i))
    Set rdoc = wd.Documents.Open(strRPath & "\" & ofiles(i))
    Set ndoc = Application.CompareDocuments(OriginalDocument:=odoc, _
               RevisedDocument:=rdoc, _
               Destination:=wdCompareDestinationNew, _
               Granularity:=wdGranularityWordLevel, _
               CompareFormatting:=True, _
               CompareCaseChanges:=True, _
               CompareWhitespace:=True, _
               CompareTables:=True, _
               CompareHeaders:=True, _
               CompareFootnotes:=True, _
               CompareTextboxes:=True, _
               CompareFields:=True, _
               CompareComments:=True, _
               CompareMoves:=True, _
               RevisedAuthor:="Merck & Co., Inc.", _
               IgnoreAllComparisonWarnings:=False)

    ActiveWindow.ShowSourceDocuments = wdShowSourceDocumentsNone
               ActiveWindow.Visible = False
               ofiles(i) = Replace(ofiles(i), Chr(13), "")
               ndoc.SaveAs2 FileName:=strCPath & "\" & ofiles(i),_
               FileFormat:=wdFormatRTF, LockComments:=False, _
               Password:="", AddToRecentFiles:=True, WritePassword:="",_
               ReadOnlyRecommended:=False, EmbedTrueTypeFonts:=False, _
               SaveNativePictureFormat:=False, SaveFormsData:=False, _
               SaveAsAOCELetter:=False, CompatibilityMode:=0

    odoc.Close SaveChanges = False
    rdoc.Close SaveChanges = False
```

```
      ndoc.Close SaveChanges = False
    End If
  Next
End Sub
```

Display 1: Code Snippet of the VBA macro Compare.

The macro compares each RTF from the first folder against the one in the second folder with the same file name, and saves the comparison result in the third folder. See below for an example of comparison result with track change: the original number is 68.2, a new number is 61.2.



Subject Characteristics

	A		B		C		Total	
	n	(%)	n	(%)	n	(%)	n	(%)
Subjects in population	71		84		82		237	
Gender								
Female	71	(100.0)	84	(100.0)	82	(100.0)	237	(100.0)
Age (Years)								
< 65	47	(66.2)	50	(59.5)	48	(58.5)	145	(~~68~~61.2)
>= 65	24	(33.8)	34	(40.5)	34	(41.5)	92	(38.8)
Mean	57.9		59.8		59.0		59.0	
SD	12.1		12.3		11.7		12.0	
Median	59.0		61.5		60.0		60.0	

Display 2 above: Output of VB macro Compare: Track Changes Are Automatically Generated

VB MACRO 2: FIND_CHANGE

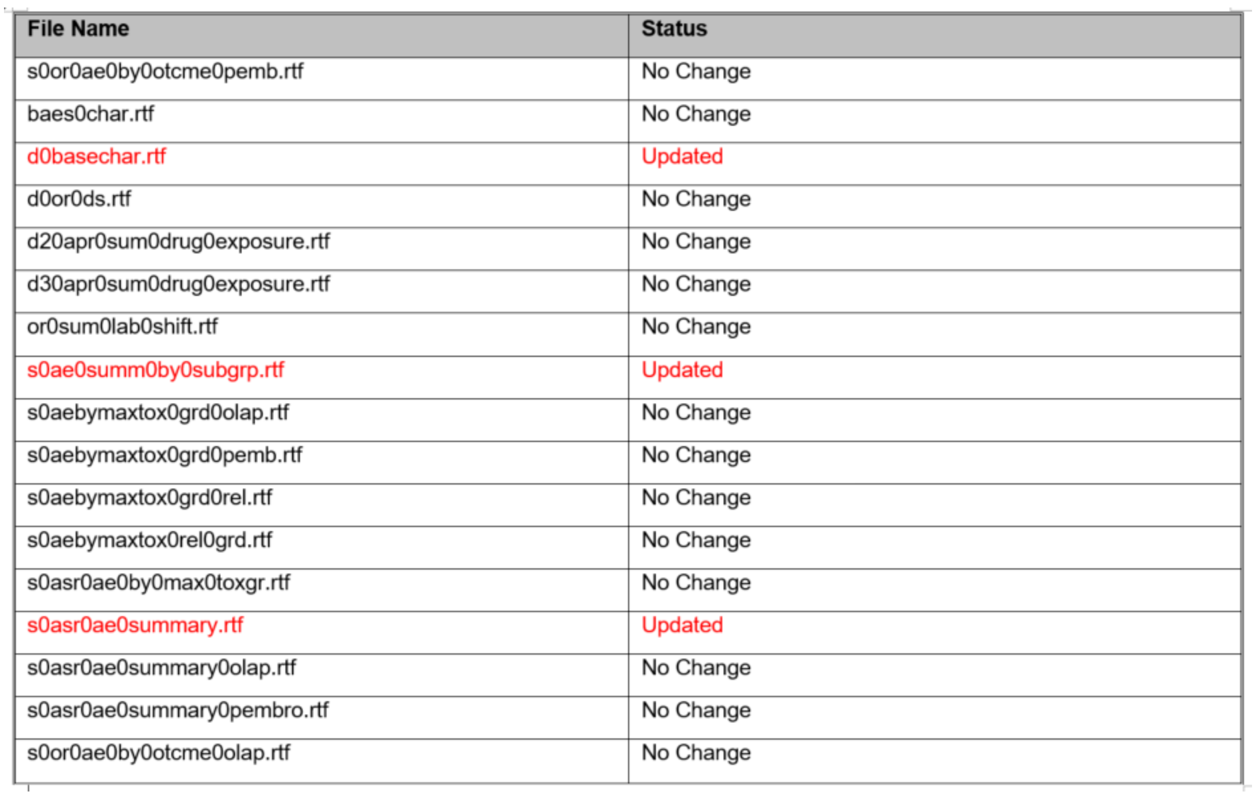
It will be very tedious and time consuming to manually open each RTF in the third folder and find out whether it has been updated or not. The VBA macro Find_Change performs this check programmatically.

First, the macro Find_Change asks the user to specify the file extension of the documents in the third folder, the default value is ".rtf"; the path name where the comparison result is saved, and the file name that can be used to record the findings of the macro Find_Change. Then the macro initiates a new MS Word document, and inserts a table with two columns, "File Name" and "Status". Lastly the macro opens each RTF file in the third folder, and determines whether there is any update (track changes) in the document: if the value of Revisions.Count is not 0, then the document has track changes, which means the document has been updated in the post database lock process. Accordingly, the macro will insert a new line in the table in the newly initialized MS Word document based on the value of Revisions.Count. If the value is 0, then it populates the Status column with the text of "No Change", else if the value is more than 0, the macro populates the Status column with the text "Updated", and the macro highlights the entire row in red to draw user's attention as well. See below for the code snippet of the VBA macro Find_Change.

```
Documents.Open FileName:=MyPath & MyFile, ConfirmConversions:= _
            False, ReadOnly:=False, _
            AddToRecentFiles:=False, PasswordDocument:="", _
            PasswordTemplate:="", Revert:=False, WritePasswordDocument:="", _
            WritePasswordTemplate:="", Format:=wdOpenFormatAuto,_
            XMLTransform:=""
```

```
Set MyDoc = ActiveDocument
If MyDoc.Revisions.Count = 0 Then
    Set MyRow = MyTable.Rows.Add
    With MyRow
        .Cells(1).Range.Text = MyFile
        .Cells(1).Range.Font.TextColor = wdColorBlack
        .Cells(2).Range.Text = "No Change"
        .Cells(2).Range.Font.TextColor = wdColorBlack
    End With
Else
    Set MyRow = MyTable.Rows.Add
    With MyRow
        .Cells(1).Range.Text = MyFile
        .Cells(1).Range.Font.TextColor = wdColorRed
        .Cells(2).Range.Text = "Updated"
        .Cells(2).Range.Font.TextColor = wdColorRed
    End With
End If
ActiveDocument.Close
```

Display 3: Code Snippet of the VBA macro Find_Change.



File Name	Status
s0or0ae0by0otcme0pemb.rtf	No Change
baes0char.rtf	No Change
d0basechar.rtf	Updated
d0or0ds.rtf	No Change
d20apr0sum0drug0exposure.rtf	No Change
d30apr0sum0drug0exposure.rtf	No Change
or0sum0lab0shift.rtf	No Change
s0ae0summ0by0subgrp.rtf	Updated
s0aebymaxtox0grd0olap.rtf	No Change
s0aebymaxtox0grd0pemb.rtf	No Change
s0aebymaxtox0grd0rel.rtf	No Change
s0aebymaxtox0rel0grd.rtf	No Change
s0asr0ae0by0max0toxgr.rtf	No Change
s0asr0ae0summary.rtf	Updated
s0asr0ae0summary0olap.rtf	No Change
s0asr0ae0summary0pembro.rtf	No Change
s0or0ae0by0otcme0olap.rtf	No Change

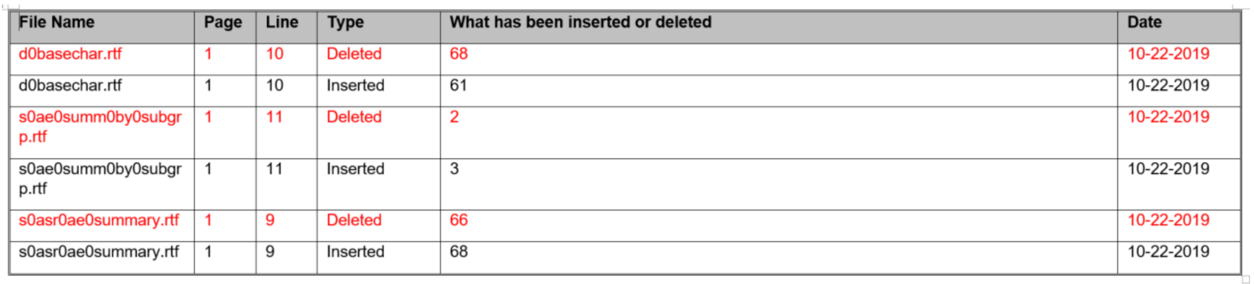
Display 4: Output of VB macro Find_Change: Report of Each RTF File with Color-Coded Status

VB MACRO 3: CHANGE_DETAIL

The third VBA macro retrieves all track changes from each RTF file and puts them in a table. It provides an overview of all changes for the entire set of the updated TLFs. See below for the key code snippet as well as the sample output. If the text was deleted from the original TLF, then the macro highlights the entire row in red to draw user's attention.

```
Documents.Open FileName:=MyPath & MyFile, ConfirmConversions:= _
            False, ReadOnly:=False,_
            AddToRecentFiles:=False, PasswordDocument:="", _
            PasswordTemplate:="", Revert:=False, WritePasswordDocument:="", _
            WritePasswordTemplate:="", Format:=wdOpenFormatAuto,_
            XMLTransform:=""
Set MyDoc = ActiveDocument
   If MyDoc.Revisions.Count = 0 Then
       ' do nothing, if needed, msgbox can be inserted for debug purpose
   Else
      For Each MyRevision In MyDoc.Revisions
        Select Case MyRevision.Type
          Case wdRevisionInsert, wdRevisionDelete
            With MyRevision
               'Get the changed text
               strText = .Range.Text
                   Set MyRange = .Range
                      Do While InStr(1, MyRange.Text, Chr(2)) > 0
                        i = InStr(1, strText, Chr(2))
                          If MyRange.Footnotes.Count = 1 Then
                             strText = Replace(Expression:=strText, _
                             Find:=Chr(2), Replace:="[footnote reference]", _
                             Start:=1, Count:=1)
                             MyRange.Start = MyRange.Start + i
                          ElseIf MyRange.Endnotes.Count = 1 Then
                             strText = Replace(Expression:=strText, _
                             Find:=Chr(2), Replace:="[endnote reference]", _
                             Start:=1, Count:=1)
                             MyRange.Start = MyRange.Start + i
                          End If
                       Loop
              End With
              'Add 1 to counter
              n = n + 1
              'Add row to table
              Set MyRow = MyTable.Rows.Add
              'Insert data in cells in MyRow
              With MyRow
                 .Cells(1).Range.Text = MyFile
                 .Cells(2).Range.Text = _
                     MyRevision.Range.Information(wdActiveEndPageNumber)
                 .Cells(3).Range.Text = _
                     MyRevision.Range.Information(wdFirstCharacterLineNumber)
                 If MyRevision.Type = wdRevisionInsert Then
                     .Cells(4).Range.Text = "Inserted"
                     MyRow.Range.Font.Color = wdColorAutomatic
                 Else
                     .Cells(4).Range.Text = "Deleted"
                     'Apply red color
                     MyRow.Range.Font.Color = wdColorRed
                 End If
                 'The inserted/deleted text
                 .Cells(5).Range.Text = strText
                 'The revision date
                 .Cells(6).Range.Text = Format(MyRevision.Date, "mm-dd-yyyy")
             End With
          End Select
           Next MyRevision
     End If
ActiveDocument.Close
```

Display 5: Code Snippet of the VBA macro Change_Details.



File Name	Page	Line	Type	What has been inserted or deleted	Date
d0basechar.rtf	1	10	Deleted	68	10-22-2019
d0basechar.rtf	1	10	Inserted	61	10-22-2019
s0ae0summ0by0subgrp.rtf	1	11	Deleted	2	10-22-2019
s0ae0summ0by0subgrp.rtf	1	11	Inserted	3	10-22-2019
s0asr0ae0summary.rtf	1	9	Deleted	66	10-22-2019
s0asr0ae0summary.rtf	1	9	Inserted	68	10-22-2019

Display 6: Output of Macro Change_Details: Report of Each Updated RTF File with Details in Track Change

CONCLUSION

This paper presents three useful standalone VBA macros that can be used to aid the post database lock process. The first macro 'Compare' programmatically compares RTF files in a batch, and records updates using the function of MS Word Track Changes; the second macro 'Find_Change' can be used to find out whether a file was updated or not without opening each one manually. The third macro 'Change_Details' retrieves the details on the track changes in each RTF file and provides an overview of the updates in the entire CSR TLFs. These three VBA macros enable users to evaluate the impact of post database changes and avoid any unintentional changes in a more efficient and effective way.

ACKNOWLEDGMENTS

The authors would like to thank Mary Varughese and Cynthia He for their great support and valuable input into this paper.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Name: Jeff Xia
Enterprise: Merck Address: 126 E. Lincoln Avenue
City, State ZIP: Rahway, NJ 07065-4607
Work Phone: 732-594-6439
E-mail: jeff.xia@merck.com
Web: www.merck.com

Name: Shunbing Zhao
Enterprise: Merck Address: 126 E. Lincoln Avenue
City, State ZIP: Rahway, NJ 07065-4607
Work Phone: 732-594-3976
E-mail: shunbing.zhao@merck.com
Web: www.merck.com