PharmaSUG 2020 - Paper QT-127

Remove Strikethrough Texts from Excel Documents by VBA Macro

Li Liu, Merck & Co., Inc.

ABSTRACT

This paper focuses on removing strikethrough text embedded in Excel Spreadsheet documents programmatically using VBA Macro. Data Definition document (i.e., define.xml) is a critical part of FDA Submissions. ADaM specifications in Excel Spreadsheet format is the source document to create the Analysis Define.xml document. Before doing so, programmers must clean up the ADaM specifications by removing irrelevant text (e.g., strikethrough text embedded in Excel Spreadsheet documents for tracking). Manually removing strikethrough text from an Excel Spreadsheet is a cumbersome task to do & prone to errors. We must apply extra caution to make sure the entire strikethrough text removal when done manually. Also, the user must make sure that normal text remains and is not deleted by mistake.

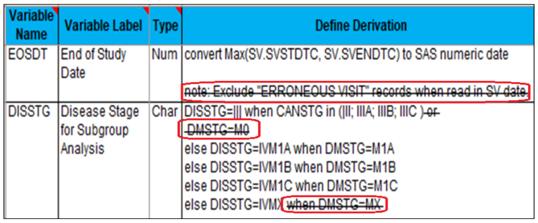
VBA Macro tool presented here removes strikethrough text by scanning the text in each cell of every sheet in an Excel spreadsheet and performing the cleanup of strikethrough text throughout the entire workbook automatically. It is a handy tool for programmers to generate high-quality data specifications for creating the Data Definition document.

INTRODUCTION

Within the Pharmaceutical industry, SAS programmers often encounter tasks such as removing strikethrough texts in Excel document. This task cannot be performed with SAS programming but can be achieved via VBA Macros.

VBA is the abbreviation of Visual Basic Application. It is an event-driven programming language developed by Microsoft Office Application. VBA Macro must be run within host environment applications, e.g., Microsoft Office Word, Excel, Access, Power Point, and Outlook. Now it is more and more common for SAS programmers to utilize VBA Macro to perform variety Excel actions. Removing strikethrough texts in Excel is a perfect example of such application.

When we develop ADaM (Analysis) Datasets, we first create ADaM specs in an Excel workbook. In order to keep track of changes in Excel document, we change un-wanted texts to strikethrough font in order to locate the changes made within the document (figure 1). Ultimately the strikethrough texts must be removed to make the Excel spreadsheet clean. However, we found that it is not so easy to accurately remove all those strikethrough texts manually and it is also prone to error. Normal texts instead of strikethrough texts may have been removed or some strikethrough texts could have been missed or hidden. With helps from Excel's VBA Macro the job of removing strikethrough text becomes easier, faster, and accurate.



Variable Name	Variable Label	Type	Define Derivation
EOSDT	End of Study Date	Num	convert Max(SV.SVSTDTC, SV.SVENDTC) to SAS numeric date ~~note: Exclude "ERRONEOUS VISIT" records when read in SV date.~~

DISSTG	Disease Stage for Subgroup Analysis	Char	DISSTG=III when CANSTG in (III; IIIA; IIIB; IIIC) ~~or DMSTG=M0~~
			else DISSTG=IVM1A when DMSTG=M1A
			else DISSTG=IVM1B when DMSTG=M1B
			else DISSTG=IVM1C when DMSTG=M1C
			else DISSTG=IVMX ~~when DMSTG=MX~~

Figure 1: Excel worksheet with strikethrough texts embedded

PROCESS

Step 1: Evoke VBA Macro Window.

As described above, VBA Macro should be run within Excel application as it provides the host environment. Thus, you open Excel document first and then evoke VBA Macro window to initialize VBA Macro built. Follow below to do so:

Select ***View*** in Excel Menu bar

Click on ***Macros***, select ***View Macros***, Macro window pops up

Type the Macro name "DelStrikethrough" in ***Macro*** window (figure 2)

Click on ***Create***, VBA Editor window pops up

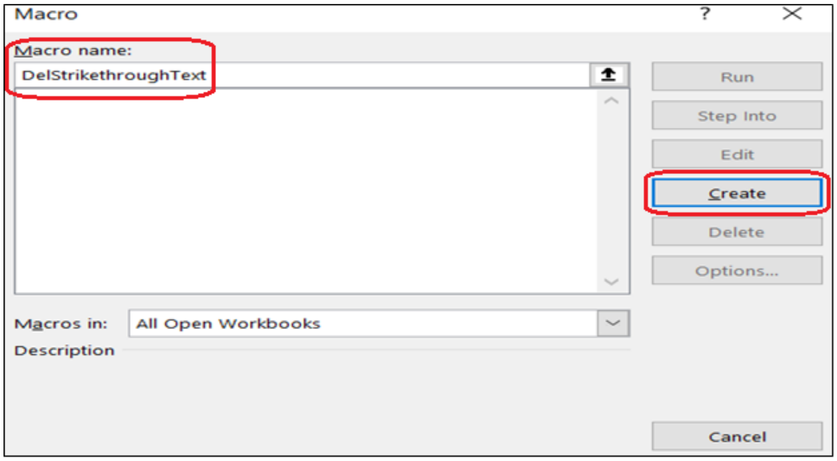


Figure 2: VBA Macro window

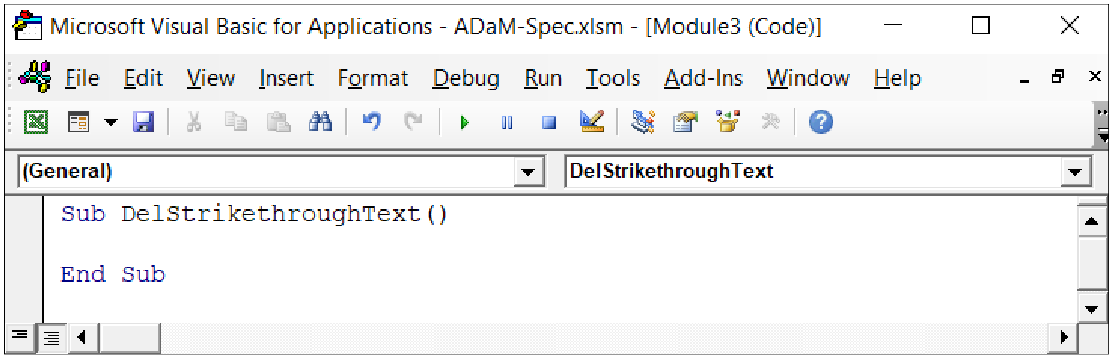
Step 2: Write VBA Macro in VBA Editor.

VBA Editor displayed as figure 3

VBA Editor always has two pre-written lines

- Macro starts with "***Sub***" followed by Macro Name in the first line
- Macro ends with "***End Sub***" (figure 3)

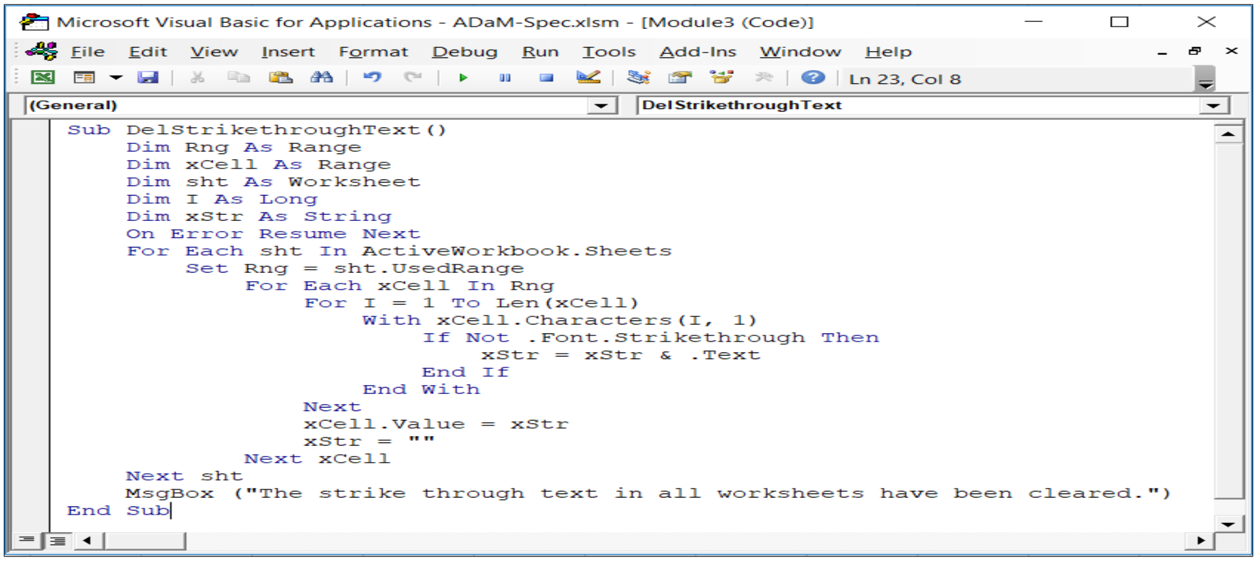
Write Macro in VBA Editor (figure 4)



```
Sub DelStrikethroughText()

End Sub
```

Figure 3: VBA Editor window


Microsoft Visual Basic for Applications - ADaM-Spec.xlsm - [Module3 (Code)]
File Edit View Insert Format Debug Run Tools Add-Ins Window Help
Ln 23, Col 8
(General)
DelStrikethroughText

```
Sub DelStrikethroughText()
    Dim Rng As Range
    Dim xCell As Range
    Dim sht As Worksheet
    Dim I As Long
    Dim xStr As String
    On Error Resume Next
    For Each sht In ActiveWorkbook.Sheets
        Set Rng = sht.UsedRange
            For Each xCell In Rng
                For I = 1 To Len(xCell)
                    With xCell.Characters(I, 1)
                        If Not .Font.Strikethrough Then
                            xStr = xStr & .Text
                        End If
                    End With
                Next
                xCell.Value = xStr
                xStr = ""
            Next xCell
    Next sht
    MsgBox ("The strike through text in all worksheets have been cleared.")
End Sub
```

Figure 4: VBA Macro on “DelStrikethroughText”

Step 3: Run VBA Macro.

Click on ***Run button*** (green triangle shape, figure 5) OR

Click on ***Run*** in Menu bar, then select ***Run button*** (green triangle shape, figure 6)

VBA Macro running starts, Excel application freeze, “***Not Responding***” displays (figure 7)

VBA Macro running completes successfully with ***notification*** window pops up (figure 8)

Check Excel document to make sure VBA Macro works as expected (figure 9)

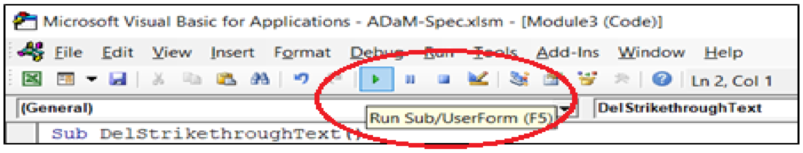

Microsoft Visual Basic for Applications - ADaM-Spec.xlsm - [Module3 (Code)]
File Edit View Insert Format Debug Run Tools Add-Ins Window Help
Ln 2, Col 1
(General)
Run Sub/UserForm (F5)
DelStrikethroughText
Sub DelStrikethroughText()

Figure 5: VBA Macro Run Button

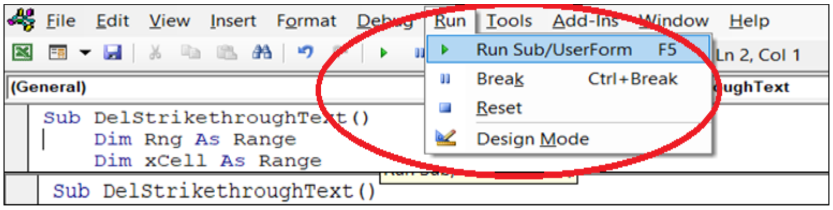

File Edit View Insert Format Debug Run Tools Add-Ins Window Help
Run Sub/UserForm F5
Break Ctrl+Break
Reset
Design Mode
Ln 2, Col 1
(General)
ughText
Sub DelStrikethroughText()
Dim Rng As Range
Dim xCell As Range
Sub DelStrikethroughText()

Figure 6: VBA Macro Run Menu

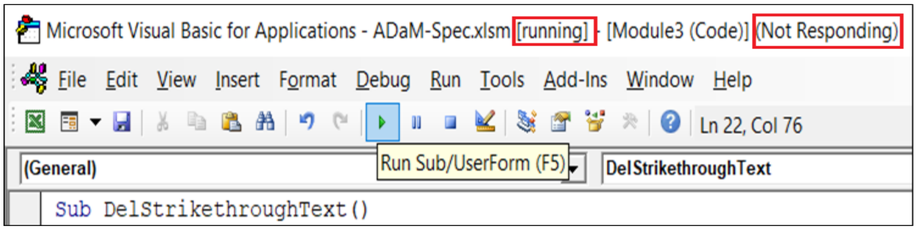
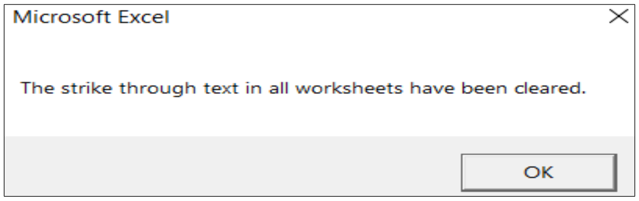

Microsoft Visual Basic for Applications - ADaM-Spec.xlsm [running] - [Module3 (Code)] (Not Responding)
File Edit View Insert Format Debug Run Tools Add-Ins Window Help
Ln 22, Col 76
(General)
Run Sub/UserForm (F5)
DelStrikethroughText
Sub DelStrikethroughText()

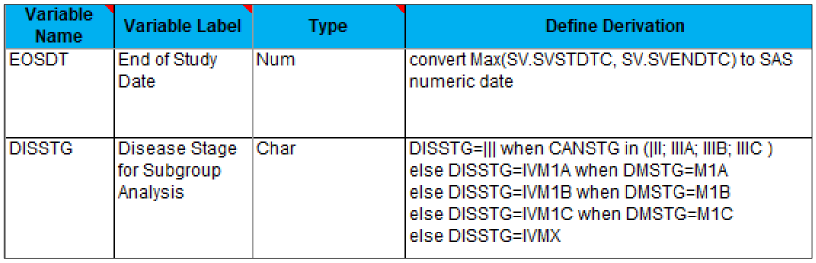
Figure 7: VBA Macro Running Status



Microsoft Excel

The strike through text in all worksheets have been cleared.

Figure 8: VBA Macro Running Completes Successfully



Variable Name	Variable Label	Type	Define Derivation
EOSDT	End of Study Date	Num	convert Max(SV.SVSTDTC, SV.SVENDTC) to SAS numeric date

DISSTG	Disease Stage for Subgroup Analysis	Char	DISSTG=||| when CANSTG in (|II; IIIA; IIIB; IIIC)
			else DISSTG=IVM1A when DMSTG=M1A
			else DISSTG=IVM1B when DMSTG=M1B
			else DISSTG=IVM1C when DMSTG=M1C
			else DISSTG=IVMX

Figure 9: Strikethrough texts removed without changes on normal texts

CONCLUSION

This paper introduces VBA Macro in Excel document for the task that SAS is unable to handle at this present time. VBA Macro assisted in removing strikethrough texts in Excel document in a much easier, faster, and reliable manner. This convenient and reliable solution also helps SAS programmers and statisticians save significant time while cleaning ADaM specs. Thus, it results in better control over the quality on the final version of ADaM specs

We have tested this VBA tool to all types of Excel documents, e.g., XLSM, XLS and XLSX formatted document. All has worked smoothly and efficiently, as anticipated.

ACKNOWLEDMENT

Special thanks to Sapan Patel and Jeff Xia for their review and comments of this paper. Sincere thanks to Jeff Xia and Jim Cawley for their testing of the VBA tool.

APPENDIX - VBA Code:

```
Sub DelStrikethroughText()
  Dim Rng As Range
  Dim xCell As Range
  Dim sht As Worksheet
  Dim I As Long
  Dim xStr As String
  Dim HasStrikethrough As Boolean
  On Error Resume Next
  For Each sht In ActiveWorkbook.Sheets
    Set Rng = sht.UsedRange
      For Each xCell In Rng
```

```
        HasStrikethrough = False
        For I = 1 To Len(xCell)
          With xCell.Characters(I, 1)
            If Not .Font.Strikethrough Then
              xStr = xStr & .Text
            Else
              If HasStrikethrough = False Then
                xStr = Trim(xStr)
                xStr = xStr & " "
              Else
                xStr = xStr & ""
              End If
              HasStrikethrough = True
            End If
          End With
        Next
        If HasStrikethrough = True Then
          xStr = Replace(xStr, ChrW(&HA0), " ")
          xCell.Value = WorksheetFunction.Trim(xStr)
        End If
        xStr = ""
      Next xCell
  Next sht
  MsgBox ("All the strikethroughed text in the current Workbook has been cleared.")
End Sub
```

CONTACT INFORMATION

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

Name: Li Liu, Merck Co., & Inc.
Address: 351 North Sumneytown Pike, North Wales, PA 19454
Work Phone, Email: 267-305-7815, li_liu2@merck.com