Pharma SUG 2024 - Paper MM- 226

Methodology for Automating TOC Extraction from Word Documents to Excel

Jeetender Chauhan, Madhusudhan Ginnaram, Sarad Nepal, Jaime Yan Merck & Co., Inc., Rahway, NJ, USA

ABSTRACT

In the complex realm of large-scale clinical study management, efficient data handling and traceability are paramount. This paper introduces a beneficial tool designed within Excel, which utilizes Visual Basic Applications (VBA) to revolutionize this process. The tool's primary functionality is automatically extracting table titles from mock document templates and seamlessly populating them into an Excel file. This process is streamlined for convenience: a simple click on the 'Prepare Extract' button enables the tool to prompt the user to select a target mock document, autonomously transferring the relevant content into Excel. Furthermore, the tool incorporates a comprehensive trace system, crucial for monitoring clinical studies' progression and validation levels, thereby significantly reducing manual data entry and potential errors. This paper will focus on this innovative tool's technicalities, applications, and benefits in managing complex clinical data. Automating title extraction and significantly enhancing traceability advances clinical study management, offering a more structured, efficient, and error-reducing approach for handling extensive reports in large studies.

INTRODUCTION

In clinical research, the way data is managed, transformed, and analyzed is crucial to the success of large-scale studies. As clinical trials become more complex, data handling, organization, and traceability become increasingly challenging. Although some aspects of data handling are automated, traditional manual data entry methods are still used in certain areas, which can be time-consuming and lead to inefficient and error-prone data. Moreover, maintaining traceability and ensuring data integrity is difficult using traditional methods. Therefore, there is a growing need for innovative tools and technologies to streamline the data management process and increase efficiency. Researchers, programmers, and study coordinators can revolutionize clinical studies and achieve robust and reliable study outcomes by leveraging software development and automation advancements. Automation is also helpful in monitoring and documenting the changes and updates in the study process.

Programmers spend a significant amount of their time creating and updating reports and mock shells. All clinical trials must be analyzed according to statistical analysis plans (SAP) created in collaboration with statisticians, clinical specialists, and other stakeholders. These mock shells provide the planned output and deliverables for programmers to work on and create. The entire plan is presented in a Microsoft Word format, including the table of contents (TOC), which details the number of deliverables and the titles for the output.

Whenever the programmer works on the reports, they need to copy the table of contents from the Word document to the Excel file where they document and keep track of all the reports generated and the work done. This manual copy-paste task is tedious and prone to human error. However, there is a way to automate this system, which is discussed in detail in this paper. This paper focuses on the automated extraction of table titles from the mock document template. The proposed macro in this paper identifies the text from the table of contents, extracts it from the Word document, and pastes it onto the Excel sheet with the help of an extract button. The macro was developed using Visual Basic for Application (VBA), and Microsoft Excel was chosen as the platform. Although most users are familiar with Excel, many may have yet to explore the possibility of embedding VBA macro systems within the application to perform routine tasks.

METHODOLOGY OVERVIEW

This procedure's main objective is to automate the extraction of TOC data from Microsoft Word documents into a structured format within Excel spreadsheets, which is particularly useful in pharmaceutical statistical programming. The methodology involves programmatically controlling Word and Excel applications to read TOC entries from Word documents and write them into Excel sheets without manual intervention. Notably, the approach adheres to the company's policy against using external tools and upholds the principles of making the solution simple, flexible, and easy to use.

DETAILED STEPS WITH EMBEDDED CODE

1. INITIALIZATION

The subroutine begins by declaring variables for interfacing with Word and Excel applications, document paths, TOC fields, and text manipulation.

```
Dim wordApp As Object
Dim wordDoc As Object
Dim docPath As String
Dim tocField As Object
Dim tocRange As Object
Dim paragraph As Object
Dim excelRow As Integer
On Error GoTo ErrorHandler
```

2. CREATING WORD APPLICATION INSTANCE

A new instance of the Word application is created and set to operate in the background. This instance is used to open the target Word document.

```
Set wordApp = CreateObject("Word.Application")
wordApp.Visible = False
```

3. OPENING THE DOCUMENT

The document is opened in read-only mode to prevent modifications during the extraction process.

```
Set wordDoc = wordApp.Documents.Open(docPath, ReadOnly:=True)
```

4. IDENTIFYING THE TOC FIELD

The subroutine iterates through all fields in the document to locate the TOC by checking against the predefined identifier for TOC fields (wdFieldTOC).

```
For Each tocField In wordDoc.Fields
    If tocField.Type = 13 Then ' wdFieldTOC
        Set tocRange = tocField.Result
        Exit For
    End If
Next tocField
```

5. EXTRACTING TOC CONTENT

Once the TOC field is found, its content is accessed. The subroutine iterates through each paragraph within the TOC, performing actions for each paragraph.

```
If Not tocRange Is Nothing Then
    excelRow = 1
    For Each paragraph In tocRange.Paragraphs
        Dim textLine As String
        textLine = Trim(paragraph.Range.Text)
        textLine = Trim(RemovePageNumber(textLine))
        If textLine <> "" And textLine <> Chr(13) Then
            ThisWorkbook.Sheets("Sheet1").Cells(excelRow, 1).Value = textLine
            excelRow = excelRow + 1
        End If
    Next paragraph
Else
    MsgBox "Table of Contents not found."
End If
```

6. ERROR HANDLING

The error handling mechanism ensures that any issues are addressed by displaying an error message and properly closing resources.

```
ErrorHandler:
MsgBox "An error has occurred: " & Err.Description
If Not wordDoc Is Nothing Then
    wordDoc.Close False
    Set wordDoc = Nothing
End If
If Not wordApp Is Nothing Then
    wordApp.Quit
    Set wordApp = Nothing
End If
```

7. UTILITY FUNCTION: REMOVE PAGE NUMBER

This helper function uses regular expressions to identify and remove page numbers from TOC entries.

```
Function RemovePageNumber(line As String) As String
    Dim regex As Object, matches As Object
    Set regex = CreateObject("VBScript.RegExp")
    regex.Pattern = "\s*\d+\s*$"
    regex.Global = True
    RemovePageNumber = regex.Replace(line, "")
End Function
```

CONCLUSION

This automated process of extracting TOC information and loading it into Excel enhances documentation efficiency, reduces the risk of error, and saves time, allowing statistical programmers to focus on analytical tasks rather than manual data entry. The method's adaptability to various project requirements makes it a valuable tool in the pharmaceutical industry's regulatory and reporting landscape, following simplicity, flexibility, and ease of use.

In clinical research, the efficient management, transformation, and analysis of data are vital for the success of studies, yet traditional manual data entry methods introduce inefficiencies and errors. This paper underscores the need for innovative automation tools to streamline data management processes and improve data integrity and traceability. Demonstrating the automation of Table of Contents extraction from Word to Excel using Visual Basic for Applications (VBA), we highlight a practical solution to reduce manual errors and save time, emphasizing simplicity, flexibility, and regulatory compliance. Adopting such auto-mated processes is crucial for enhancing operational efficiency and study outcomes in clinical research, displaying the significant impact of technological advancements on the field.

REFERENCES

P. Burmenko and T. Cardozo. "Come Out of Your Shell: A Dynamic Approach to Shell Implementation in Table and Listing Programs". In: Proceedings of PharmaSUG 2014. 2014.

I. Goldfarb and E. Zelichonok. "Macro To Produce SAS®-Readable Table of Content From TLF Shells". In: Proceedings of the PharmaSUG 2020. 2020.

Karen Walker and Jeff Cao. "Macro to Automate Creation and Sync of Shell Document and TOC". In:

ACKNOWLEDGMENTS

The authors are grateful for reviewing the paper and providing feedback from Erica Davis, Shi Changhong, Su Chao, Akers tad and our department head, Amy Gillespie.

CONTACT INFORMATION

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

Jeetender Chauhan
Merck &Co., Inc., Rahway, NJ, USA
jeetender.chauhan@merck.com

Sarad Nepal
Merck &Co., Inc., Rahway, NJ, USA
sarad.nepal@merck.com

Madhusudhan Ginnaram
Merck &Co., Inc., Rahway, NJ, USA
madhusudhan.ginnaram.reddy@merck.com

Jaime Yan
Merck &Co., Inc., Rahway, NJ, USA
mingyu.yan1@merck.com

TRADEMARK