

## Repetitive Excel tasks, VBA helps you out

Weiping Zhang, BeiGene (Beijing) Co., Ltd.

### ABSTRACT

Visual Basic for Applications (VBA) is a programming language developed by Microsoft. It has been widely applied in different areas in the past decades. Now more and more powerful programming languages like R, Python, JavaScript have been used in pharmaceutical development. But VBA still has its own advantages. This paper will explore the potential and possibility of VBA application in pharmaceutical development by giving two examples.

VBA is built in Microsoft Excel and can use Excel as UI directly as long as Excel has been installed on the computer. Since almost everyone has already installed and used Microsoft Excel on their own PC. So, the application developed by Excel is very easy to use and being distributed. No learning curve and no additional software to be installed. Based on this, we have developed a VBA tool for DM and CRA who don't have any programming knowledge. This tool will be distributed to each DM and CRA to help them check local lab reference range (LLRR) files. It will check logic/typo issues and generate an issue summary automatically to save manual work.

VBA also allows programmatic control for almost all functions of Excel. Therefore, VBA can be a powerful tool to automate repetitive tasks in Excel. We also have developed a VBA tool to standardize our programming specification which is written in Excel. This tool will help identify the variable label from specification and generate the corresponding SAS dataset. So, we don't need to adjust our SAS program for different programming specifications.

In this paper, Microsoft Excel for Microsoft 365 MSO (16.0.14326.20936) and SAS Enterprise Guide 7.15 are being used.

### INTRODUCTION

Our daily routines often involve many repetitive Excel tasks. Repetitive Excel tasks are boring, time-consuming and error prone. So, automating these works can greatly save time, improve accuracy, and free us to work on more creative tasks. When it comes to automation, you may think of R, Python, JavaScript and so on. But as a programming language built in Microsoft Excel, VBA has its own unique advantages.

Many, perhaps most, non-programmers in the pharmaceutical industry such as DM and CRA are not good at using programming software. But repetitive Excel tasks often make up a large part of their daily work. The question then becomes, how can we find a programming software which interacts with Excel well and has a low entry barrier? The answer lies in VBA. VBA can automate nearly every operation that you can perform with a mouse, keyboard, or a dialog box in Excel. Perhaps other programming software can do these things as well, but VBA is built in Microsoft Excel. Almost all staffs are familiar with Microsoft Excel and have installed Microsoft Excel on their PC. Therefore, VBA tools are obviously more user-friendly for all kinds of employees. Based on these advantages of VBA, we have developed the local lab reference range (LLRR) tool for DM and CRA. This tool is used to check logic/typo issues, compare the finalized LLRR files with data entered in EDC, and generate the issue summary automatically. For VBA tool end users, knowledge of the VBA language or other programming software is not necessary, they can do all operations via this LLRR Tool in Microsoft Excel, which they are familiar with.

As mentioned above, VBA tools can do a lot for non-programmers. How about SAS programmers? VBA is not only a tool to automate repetitive Microsoft Excel tasks, it can also run SAS codes. When a task requires both of Microsoft Excel and SAS, you can automatically accomplish all the steps in Microsoft Excel with VBA, without switching between Microsoft Excel and SAS. Due to the flexibility of VBA to interact with Microsoft Excel and SAS, SAS programmers can also benefit from VBA. Based on these advantages of VBA, we have developed the standard spec tool to extract the variable label from specification which is written in Microsoft Excel, and then generate the corresponding SAS dataset. Therefore, we don't need to manually assign labels to each variable in SAS programs.

This paper will explore the potential and possibility of VBA application in pharmaceutical development by giving these two examples.

## LLRR TOOL

The LLRR tool can help DM and CRA to check logic/typo issues in the initial LLRR files, compare the finalized LLRR files with data entered in EDC, and generate the issue summary automatically. It has a user-friendly interface and the checking can be completed within seconds. The key VBA knowledge used in this tool is described below.

### PART 1 USER INTERFACE DESIGN

The LLRR tool has two portals for DM and CRA because of the different requirements of the two departments (Figure 1 and Figure 3). In addition, the DM portal has two functions: checking issues and comparing the LLRR files with EDC data (Figure 1 and Figure 2). You can use VBA statements to give the interface the appropriate appearance at different portals and different functions. When the "DM" or "CRA" option button is clicked, the interface of the corresponding portal is displayed. Similarly, when the "New" or "QC" option button is clicked, the interface of the corresponding function is displayed.

	A	B
1	Template Excel Name	
2	User Excel name	
3	LLN&ULN Checklist name	
4	Function Role	<input checked="" type="radio"/> DM <input type="radio"/> CRA
5	Check Type	<input checked="" type="radio"/> New <input type="radio"/> QC
6		
7		
8		
9		
10		<div style="display: flex; justify-content: space-around;"> <div style="border: 1px solid gray; padding: 5px; width: 30%;">Select Template Excel</div> <div style="border: 1px solid gray; padding: 5px; width: 30%;">Select LLN&amp;ULN checklist</div> <div style="border: 1px solid gray; padding: 5px; width: 30%;">Select User Excel</div> </div>
11		
12		
13		
14		<div style="border: 1px solid gray; padding: 10px; width: 60%; margin: 0 auto;">Check</div>
15		
16		
17		
18		<div style="border: 1px solid gray; padding: 10px; width: 60%; margin: 0 auto;">Generate Auto-message</div>
19		
20		
21		
22		
23		

< >
tool
Auto-message
Unit Library
Site Name Library
Usage
Preparations
Version
+

**Figure 1. DM Portal of the LLRR Tool (New Function)**

	A	B
1	EDC LLRR Excel Name	
2	User Excel name	
3		
4	Function Role	<input checked="" type="radio"/> DM <input type="radio"/> CRA
5	Check Type	<input type="radio"/> New <input checked="" type="radio"/> QC
6		
7		
8		
9		
10		Select EDC LLRR Excel
11		Select User Excel
12		
13		Check
14		
15		
16		
17		Generate Auto-message
18		
19		
20		
21		
22		
23		

tool | Auto-message | Unit Library | Site Name Library | Usage | Preparations | Version | (+)

**Figure 2. DM Portal of the LLRR Tool (QC Function)**

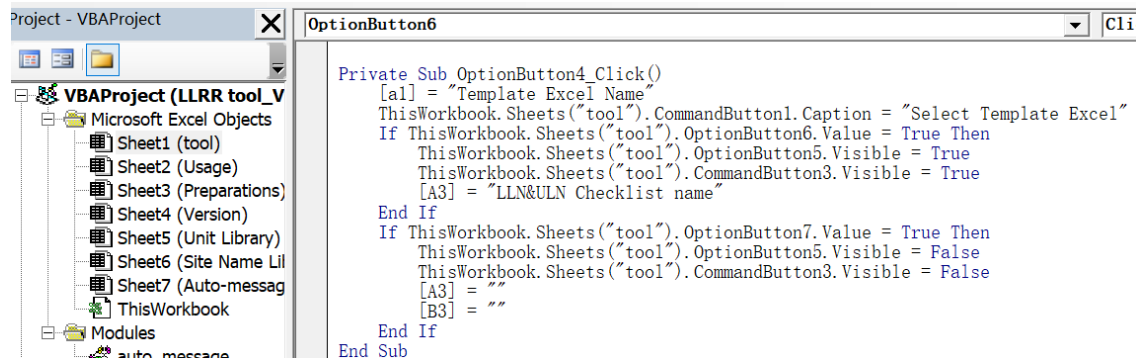
	A	B
1	Template Excel Name	
2	User Excel name	
3		
4	Function Role	<input type="radio"/> DM <input checked="" type="radio"/> CRA
5	Check Type	<input checked="" type="radio"/> New
6		
7		
8		
9		
10		Select Template Excel
11		Select User Excel
12		
13		Check
14		
15		
16		
17		
18		
19		
20		
21		
22		
23		

tool | Auto-message | Unit Library | Site Name Library | Usage | Preparations | Version | (+)

**Figure 3. CRA Portal of the LLRR Tool**

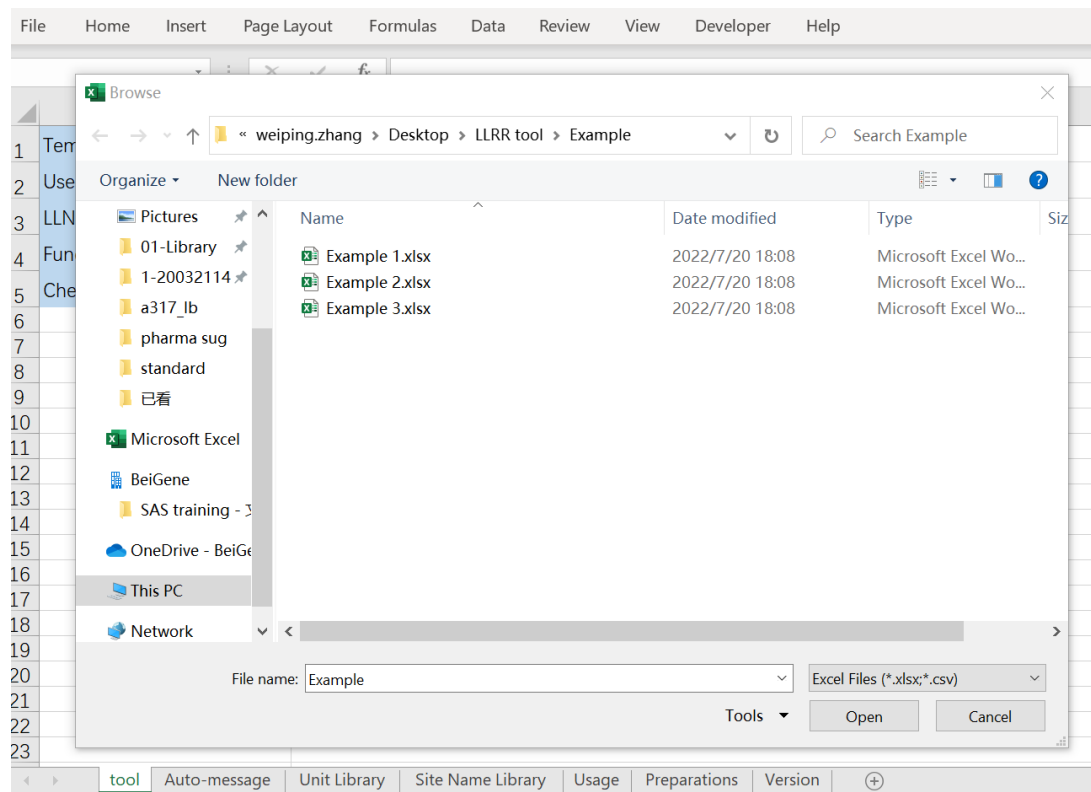
Figure 4 gives an example code for changing the text in cells and buttons and controlling whether the buttons are display or not by choosing option buttons. When the "New" option button (OptionButton4) is clicked, "Template Excel Name" will be written in cell "A1" and the caption of the CommandButton1 will become "Select Template Excel". If the "DM" option button (OptionButton6) is selected, the "QC" option button (OptionButton5) and the "Select LLN&ULN checklist" command button (CommandButton3) will be visible. If the "CRA" option button (OptionButton7) is selected, the "QC" option button (OptionButton5) and the "Select LLN&ULN checklist" command button (CommandButton3) will be invisible and the

contents of "A3" and "B3" cells will be cleared.



**Figure 4. Code of the Interface Design**

In addition, you can open a file dialog to select files with FileDialog object. For example, in this tool, when the "Select Template Excel", "Select EDC LLRR Excel", or "Select User Excel" command button is clicked, a file dialog appears and files can be selected. Then Excel displays the path for each file in the corresponding cells from cells "B1" to "B3" (Figure 6). The following code (Figure 7) gives an example of how to open the file dialog via VBA.



**Figure 5. Window of the File Dialog**

	A	B
1	Template Excel Name	C:\Users\weiping.zhang\Desktop\LLRR tool\Example\Example 1.xlsx
2	User Excel name	C:\Users\weiping.zhang\Desktop\LLRR tool\Example\Example 2.xlsx
3	LLN&ULN Checklist name	C:\Users\weiping.zhang\Desktop\LLRR tool\Example\Example 3.xlsx
4	Function Role	<input checked="" type="radio"/> DM <input type="radio"/> CRA
5	Check Type	<input checked="" type="radio"/> New <input type="radio"/> QC
6		
7		
8		
9		
10		<input type="button" value="Select Template Excel"/> <input type="button" value="Select LLN&amp;ULN checklist"/> <input type="button" value="Select User Excel"/>
11		
12		
13		
14		<input type="button" value="Check"/>
15		
16		
17		
18		<input type="button" value="Generate Auto-message"/>
19		
20		

**Figure 6. Display the Path for Each File in the Corresponding Cells**

```

Sub open_template_files()
Dim arr, excelname As String
With Application.FileDialog(msoFileDialogFilePicker)
.AllowMultiSelect = False
.InitialFileName = ThisWorkbook.Path
.Filters.Clear
.Filters.Add "Excel Files", "*.xlsx; *.csv"
.Filters.Add "All Files", "*.*"
If .Show = -1 Then
Range("B1").Value = .SelectedItems(1)
End If
End With
End Sub

```

**Figure 7. Code of the File Dialog**

## PART 2 LOGIC CHECK

After selecting the portal, function and files, users only need to click the "Check" command button and all the required check will be done automatically. You can use If-End statements to select the corresponding logic based on the value of the portal and function option buttons (Figure 8).

```

If ThisWorkbook.Sheets("tool").OptionButton6.Value = True Then
...Logic Statements...
End

```

**Figure 8. Use If-End Statements to Select the Corresponding Logic**

Figure 9 shows an example of a part of the check results. Cells with issues are marked in yellow, and corresponding issue flags are filled in the last column of the LLRR file.

Analyte	EDC Analyte	Category	Other Category Only enter if 'Category' column is selected as 'Other'	Sex	Min Age Range	Min Age Range Unit	Max Age Range	Max Age Range Unit	Issue flag (Please delete the column when finalized)
24 hr Total Protein	24 Hour Total Protein (Urine)	Urine		Both	18	Years	99	Years	
Alanine Aminotransferase (ALT)	Alanine Aminotransferase (ALT)	Chemistry		Female	0	Years	99	Years	Please enter [输入了非] of Comments column in English method
Alanine Aminotransferase (ALT)	Alanine Aminotransferase (ALT)	Chemistry		Male	18	Years	99	Years	
Albumin	Albumin	Chemistry		Both	18	Months	99	Months	Please check the Min and Max age unit
Alkaline Phosphatase	Alkaline Phosphatase	Chemistry		Both	50	Years	49	Years	Max age isn't greater than min age
Alkaline Phosphatase	Alkaline Phosphatase	Chemistry		Female	50.2	Years	99.1	Years	The Min Age and Max Age are not integer
Alkaline Phosphatase	Alkaline Phosphatase	Chemistry		Male	18	Years	99.2	Years	The Max Age is not integer
aPTT	aPTT	Coagulation		Both	18.1	Years	99	Years	The Min Age is not integer
Aspartate Aminotransferase (AST)	Aspartate Aminotransferase (AST)	Chemistry		Female	18	Years	99	Months	Please check the Max age unit Sex [Male] is missing, please confirm
Aspartate Aminotransferase (AST)	Aspartate Aminotransferase (AST)	Chemistry							[Sex] is missing [Min Age Range] is missing [Min Age Range Unit] is missing [Max Age Range] is missing [Max Age Range Unit] is missing [ULN] is missing Sex [Male] is missing, please confirm
Blood Urea Nitrogen	Blood Urea Nitrogen	Chemistry		Both	18	Years	99	Years	
Total calcium	Calcium	Chemistry		Both	18	Years	99	Years	
Creatine kinase	CK	Chemistry		Male	18	Years	99	Years	Sex [Female] is missing, please confirm
Creatine kinase	CK	Chemistry			18	Years	99	Years	[Sex] is missing
CK-MB (Activity)	CK-MB (Activity)	Chemistry		Both	18	Years	99	Years	Sex [Female] is missing, please confirm

Figure 9. Example of Check Results

### PART 3 GENERATE ISSUE SUMMARY

After checking LLRR files, the issue summary can be generated automatically by clicking "Generate Auto-message" command button. The issue summary is generated based on the issue flags in the check results and is displayed in the text box of the Auto-message sheet. Figure 10 shows such an example of parts of the issue summary. Figure 11 gives an example code of the text box.

I have highlighted the cells, where need your further action. Please find the issue flag in the attached LLRR.

For site [123456]

- On "Site & Lab Info" sheet
  1. Don't modify any project information [Lead Data Manager].
  2. Please confirm the site name, it's inconsistent with the EDC site name [The First Hospital of Lanzhou University].
  3. Please keep the file name is in "Study#\_LLRR\_Site#\_LabName\_YYYY-mm-dd" format and ensure the the Lab Name in "Site
  4. Please keep the file name is in "Study#\_LLRR\_Site#\_LabName\_YYYY-mm-dd" format and ensure the the collection date in
  5. Please enter [Clinical Research Associate] in English method.
  6. [Lab Address] is missing and Lab Range Type should be ticked, please complete.
- On "Analyte Info" sheet
  1. Don't modify (delete/update/new added) any analyte information on column A-D compared with template file. Please up
  2. [aPTT, HBV DNA, HCV RNA] analytes are missing in the LLRR form. Please check if they are available, if yes, please record tl
  - Column E-L rather than delete the entire row directly.
  - ▲ For analyte [24 Hour Total Protein (Urine)]
    1. Sex [Male] is missing in the specific effective period, please confirm the incomplete sex.
    2. The max age is not greater than the min age. Please check and update.
    3. The age range are overlapped, please check and update.
    4. Please confirm the unit as it can't be applied in EDC now. If confirmed, please provide the source document for further clai
    5. The information is duplicate for the rows of the analyte, please confirm if them could be combined into 1 row.
    6. [Min Age Range Unit, Max Age Range Unit] are missing, please complete.
  - ▲ For analyte [Alanine Aminotransferase (ALT)]
    1. Please confirm the min age unit and the max age unit of the analyte, if it should be selected as "Years".
    2. [LLN, ULN] are missing, please complete.
  - ▲ For analyte [Albumin]
    1. Please confirm the min age unit and the max age unit of the analyte, if it should be selected as "Years".
    2. [LLN, ULN] are missing, please complete.

Figure 10. Issue Summary

```
Sheet3 (Preparations) | ThisWorkbook.Sheets("Auto-message").TextBox1.Text = mes_def
```

Figure 11. Code of the Text Box

### STANDARD SPEC TOOL

The standard spec tool is used to extract the variable label from the specification which is written in Microsoft Excel and generate the corresponding SAS dataset. With the standard spec SAS dataset, we don't need to manually assign labels to each variable in SAS programs.

## PART 1 INTERFACE

Figure 12 shows the interface of the standard spec tool. Users can select original specification by clicking the "Select spec excel" command button. The code is similar to what's shown in Figure 7. After the specification path fills cell "A1", users can select sheets from the drop-down box in cell "B2" (Figure 13). Figure 14 shows the example code of the drop-down box. Then users need to select columns of the original specification that they want to keep in the standard spec from the drop-down box in cell "B3", fill cell "B10" with the rows which they want to delete from the standard spec, and select the folders ("Prod" or "Dev") where the SAS dataset will be generated. After completing all the above steps, the standard spec excel and the corresponding SAS dataset will be generated by clicking the "OK" command button.

	A	B	C	D	E	F
1	Workbook name					
2	Sheet name			Select spec excel		
3						
4	Title Row (Defaults to 1)	1				
5	Listing /Tab Name (Sheet Name)	Listing/Tab Name (if the listings has multiple tabs, use different tab names)				
6	Column Header (Label)	Column Header				
7	From Listing Header (Description)	From Listing Header				
8	Note for CP (Optional, just for standard refer	Note for CP				
9						
10	Delete row (separate by @, case insensitive)	comments@comment@dm comments@dm comment@flag@newflag@new flag@flag(old/new)				
11						
12	SAS dataset generate in	<input checked="" type="checkbox"/> Prod <input checked="" type="checkbox"/> Dev				
13						
14		OK				
15						
16						

Figure 12. Interface of the Standard Spec Tool

	A	B	C	D
1	Workbook name	C:\Users\weiping.zhang\Desktop\Standard spec tool\Example\Example1.xlsx		
2	Sheet name	<div style="border: 1px solid black; padding: 2px;">           Specs. Version Control            Instructions  <b>Specifications</b>            Design&amp;Logic_REC            Sample Output            Output Delivery Details            Signature Page            Vault Data            Note for CP         </div>		
3				
4	Title Row (Defaults to 1)			
5	Listing /Tab Name (Sheet Name)			
6	Column Header (Label)			
7	From Listing Header (Description)			
8	Note for CP (Optional, just for standard refer			
9				
10	Delete row (separate by @, case insensitive)	comments@comment@dm comments@dm comment@flag@newflag@new flag@flag(old/new)		
11				
12	SAS dataset generate in	<input checked="" type="checkbox"/> Prod <input checked="" type="checkbox"/> Dev		
13				
14		OK		
15				
16				

Figure 13. Drop-down Box

```

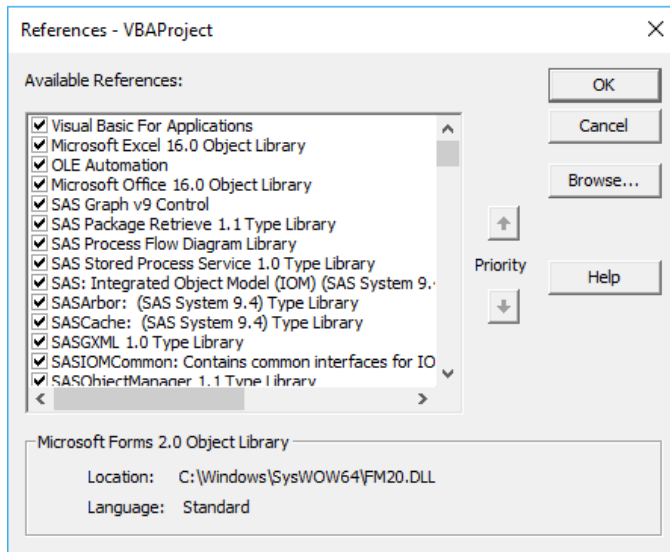
For i = 1 To fb.Sheets.Count
    Sheets("version").Range("S" & i) = fb.Sheets(i).Name
Next i
Sheets("tool").Range("B2").Select
With Selection.Validation
    .Delete
    .Add Type:=xlValidateList, AlertStyle:=xlValidAlertStop, Operator:=xlBetween, Formulas:="=version!$$1:$S" & i - 1
End With

```

Figure 14. Code of the Drop-down box

## PART 2 RUN SAS CODE VIA VBA

One of the very important parts of the standard spec tool is to generate the corresponding SAS dataset. In order to establish a connection to the SAS EG server, certain references will need to be installed and enabled on PC (Trace: Developer > Visual Basic > Tools > References). See figure 15 for more details.



**Figure 15. Add References**

With the execution of the VBA code in figure 16, the SAS code stored in the SAS\_code variable is running. And the log is saved in ms\_transas.log file in the same path as the standard spec tool. By making the username and password parameterized, this code can easily be used by multiple users.

```

Dim obObjectFactory As New SASObjectManager.ObjectFactory
Dim obObjectKeeper As New SASObjectManager.ObjectKeeper
Dim obSAS As SAS.Workspace
Dim observer As New SASObjectManager.ServerDef
observer.MachineDNSName = "sas94p"
observer.Port = 8592
observer.Protocol = ProtocolBridge
observer.Name = "SASApp_en - Workspace Server"
On Error GoTo ExitSub
Set obSAS = obObjectFactory.CreateObjectByServer("SASApp_en", True, observer, Username, Password)
On Error GoTo 0
DatafileDir = Replace(Replace(Replace(ThisWorkbook.Path, "\", "/"), "Z:/", "/usrfiles/"), "///ip-10-8-54-115", "")
toolsDir = Mid(DatafileDir, 1, InStrRev(DatafileDir, "/docs")) & "tools/"
obSAS.LanguageService.Submit (sas_code)
Dim sLog As String
sLog = obSAS.LanguageService.FlushLog(2000000)
Dim spath As String
spath = ThisWorkbook.Path & "\ms_transas.log"
Open spath For Output As #1
Write #1, sLog
Close #1
obSAS.Close

```

**Figure 16. Code of Running SAS Code via VBA**

## CONCLUSION

With the help of VBA, repetitive Excel tasks can be easily simplified and automated. This paper presents ideas of how to develop VBA tools for non-programmer and how VBA tools help our SAS programmers. Of course, there are a lot of other approaches which is available to achieve the same goal, such as R, Python, JavaScript. But the accessibility and the good interaction with Excel are VBA's own important advantages.

## REFERENCES

- Pratter, Frederick. 2005. "Access to SAS® Data Using the Integrated Object Model (IOM) in Version 9.1" <https://www.lexjansen.com/pharmasug/2005/ApplicationsDevelopment/ad05.pdf>
- Xiaoyang Yu. 2021. "AUTOMATING SAS EG INTEGRATION TECHNOLOGIES VIA EXCEL VBA" <https://www.pharmasug.org/proceedings/china2021/AD/Pharmasug-China-2021-AD045.pdf>



Microsoft. 2022. "Getting started with VBA in Office"

<https://docs.microsoft.com/en-us/office/vba/library-reference/concepts/getting-started-with-vba-in-office>

## CONTACT INFORMATION

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

Weiping Zhang  
BeiGene (Beijing) Co.,Ltd.  
weiping.zhang@beigene.com