**PharmaSUG China 2021 - Paper AD-045**

# AUTOMATING SAS EG INTEGRATION TECHNOLOGIES VIA EXCEL VBA

Xiaoyang Yu, Anheart Therapeutics(Hangzhou) Co.,Ltd.

## ABSTRACT

SAS Enterprise Guide provides an API that allows users to automate almost every aspect of running Enterprise Guide projects or simply SAS programs. Visual Basic for Applications (VBA) under Microsoft Excel provides a rich environment for debugging and running VBA applications. This paper shows how you  can use VBA to access the automation API of Enterprise Guide to do sophisticated tasks, read SAS datasets through ActiveX Data Object Databse or build your own  applications that run SAS. Through VBA, you can create SAS programs on the fly, debug and run programs, save SAS log to files, and examine SAS ODS. You can accomplish many tasks automatically, which is not feasible through Enterprise Guide’s main interface. You can access and transform SAS datasets efficiently, which can not be retrive directly without SAS.

## INTRODUCTION

SAS Enterprise Guide offers point-and-click access to all of its feature and greatly improve programming efficiency, especially for non-programmer. In a typical SAS Enterprise Guide environment, we connect to remote Metadata Server, and we do not install local SAS library or PC SAS. Due to the restriction, SAS EG does not recognize local machines (local and network drives). To use these functions, one needs to go through guided wizard and point-and-click through SAS EG main interface. This makes us wonder whether there is a better way to accomplish this task in a programmable fashion (i.e, you can specific the format, filename etc in code and run in batch mode).

We explore many approaches and SAS stored processes. The SAS Enterprise Guide API provides a powerful mechanism to link SAS remote server to local. The concepts are familiar to us under windows environment, and this motivates us to do more research and we eventually find VBA a great tool to run almost every SAS jobs we needed. We can create SAS programs on the fly, run the programs, analyze or save SAS Listing Output, write complete SAS logs to files, examine SAS Output Delivery System, email the results or send out notifications. This approach addresses the programming issues of automation of SAS procedures between remote server and local, and it does not require installation of local SAS library or PC SAS inside SAS EG.

We successfully create a self-contained SAS running environment in Excel without running SAS EG main user interface. For SAS end users, knowledge of VB language is not necessary, one only work in Excel to write SAS programs. VBA not only provides an easy-to-use tool to run SAS programs similar to SAS command-line tool, but it seamlessly connects other objects or modules for us to build our own applications that run SAS as well. VBA under Excel is available for almost every windows user. We find VBA is useful to run SAS jobs through its step-by-step debug mode: watch intermediate variables, step into, run to cursors. Besides that, we can also access Excel formula, cells, named range, macros, and more. We also use VBA as toolkit to debug SAS codes.

## WRITE VBA PROGRAM TO RUN SAS PROGRAMS INTERACTIVELY

### STEP 1 CREATE EXCEL WORKBOOK WITH XLSM FORMAT

Create an Excel workbook, and save it as macro-enabled file xlsm format. Then click "Alt-F11", it brings you to VBA macro-editing mode. Right-click on the "project" on the left-hand panel, and choose "Insert" ->"Module", you will see a new module "Module1" created under "Modules" folder.

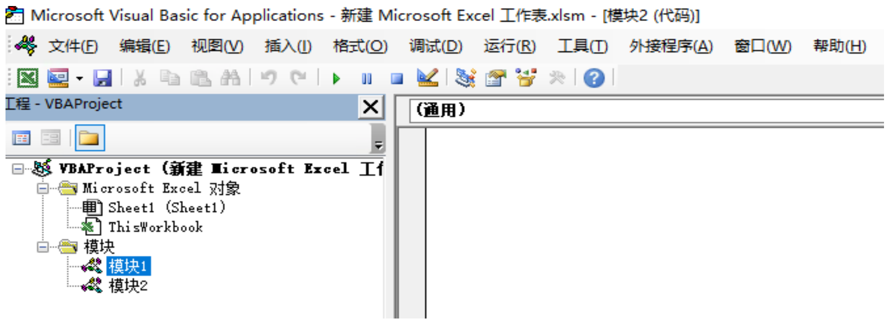

Microsoft Visual Basic for Applications - 新建 Microsoft Excel 工作表.xlsm - [模块2 (代码)]
文件(F) 编辑(E) 视图(V) 插入(I) 格式(O) 调试(D) 运行(R) 工具(T) 外接程序(A) 窗口(W) 帮助(H)
工程 - VBAProject
(通用)
VBAProject (新建 Microsoft Excel 工作
Microsoft Excel 对象
Sheet1 (Sheet1)
ThisWorkbook
模块
模块1
模块2

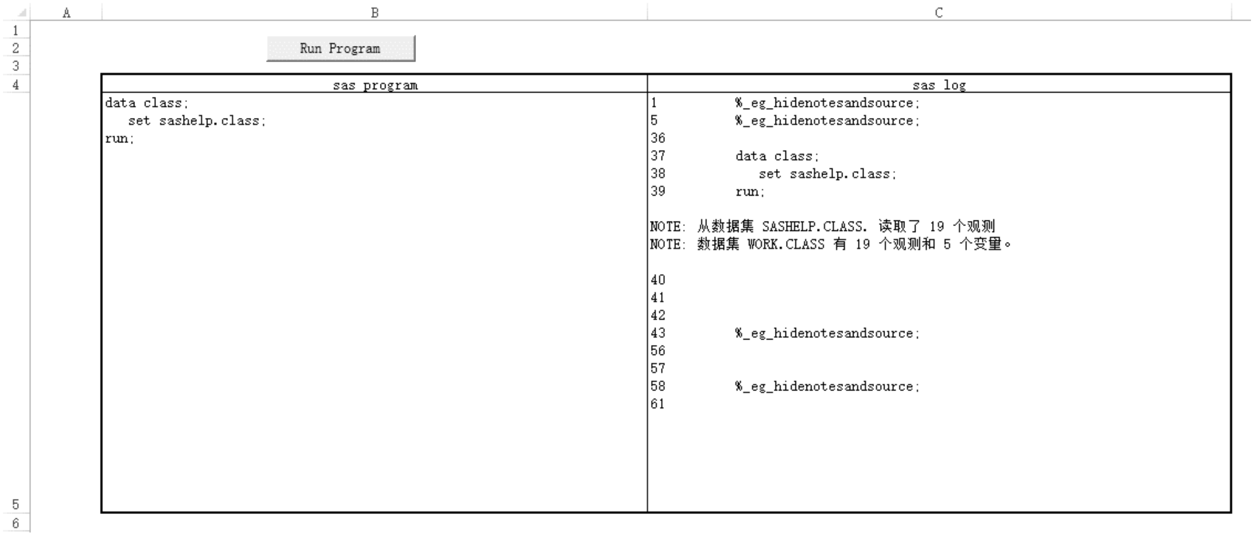
**Figure 1. Create XLSM File**

## STEP 2 WRITE VBA MACRO

### EXAMPLE 1: WRITE SAS CODES IN EXCEL WORKSHEET

The following example makes a connection to a SAS server, runs a SAS program through clicking the “Run Program” button, which is written in EXCEL “Sheet1" worksheet, in this example Cell “B4", it also brings back SAS Log to Cell "C4".(Figure 2).

Notice that SAS code in example is exactly same as traditional SAS programs in PC SAS or SAS EG. One can copy SAS codes (one step or multiple steps) from existing SAS programs and run. Users can interface through Excel Worksheet only to write and debug SAS programs. SAS Logs including error messages will be brought back to Excel worksheet right after running, no need to open another log file. Meanwhile, users can append every SAS Logs you are debugging to a complete LOG file without concerns about LOGs being overwritten by new run.

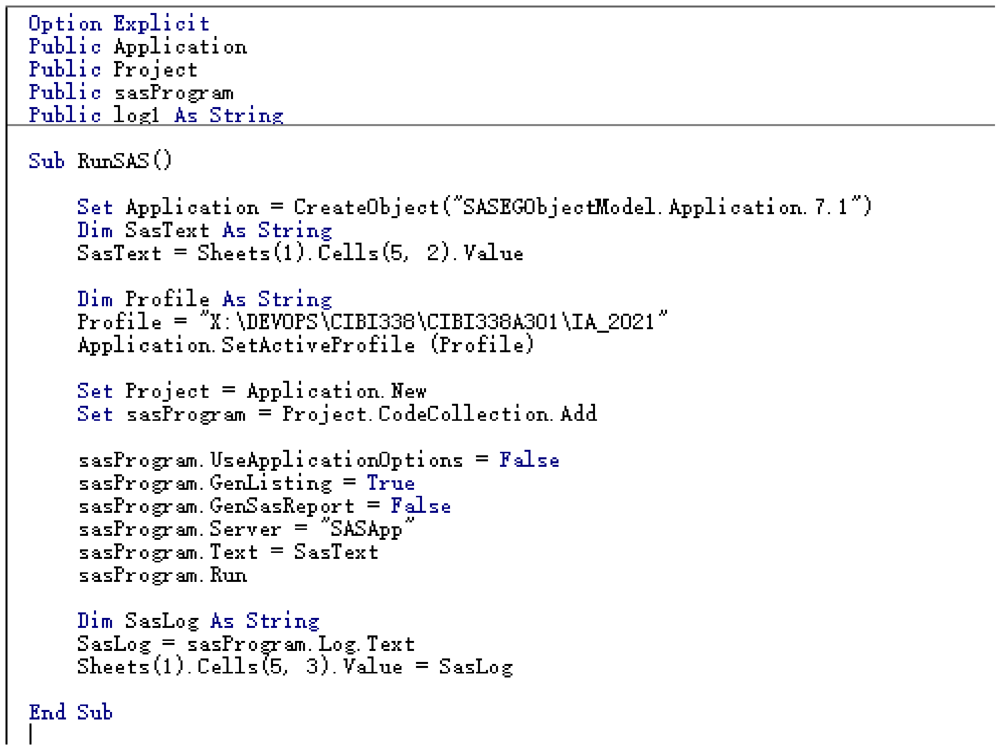


```
data class;
   set sashelp.class;
run;
```

```
1          %_eg_hidenotesandsource;
5          %_eg_hidenotesandsource;
36
37         data class;
38            set sashelp.class;
39         run;

NOTE: 从数据集 SASHELP.CLASS. 读取了 19 个观测
NOTE: 数据集 WORK.CLASS 有 19 个观测和 5 个变量。

40
41
42
43         %_eg_hidenotesandsource;
56
57
58         %_eg_hidenotesandsource;
61
```

**Figure 2. Run SAS Programs in Worksheet**

Through running VBA macro is written in one VBE (Figure 3) to connect SAS Enterprise Guide to complete the whole process.



```
Option Explicit
Public Application
Public Project
Public sasProgram
Public log1 As String

Sub RunSAS()

    Set Application = CreateObject("SASEGObjectModel.Application.7.1")
    Dim SasText As String
    SasText = Sheets(1).Cells(5, 2).Value

    Dim Profile As String
    Profile = "X:\DEVOPS\CIBI338\CIBI338A301\IA_2021"
    Application.SetActiveProfile (Profile)

    Set Project = Application.New
    Set sasProgram = Project.CodeCollection.Add

    sasProgram.UseApplicationOptions = False
    sasProgram.GenListing = True
    sasProgram.GenSasReport = False
    sasProgram.Server = "SASApp"
    sasProgram.Text = SasText
    sasProgram.Run

    Dim SasLog As String
    SasLog = sasProgram.Log.Text
    Sheets(1).Cells(5, 3).Value = SasLog

End Sub
```

**Figure 3. VBA Macro of Connecting SAS EG**

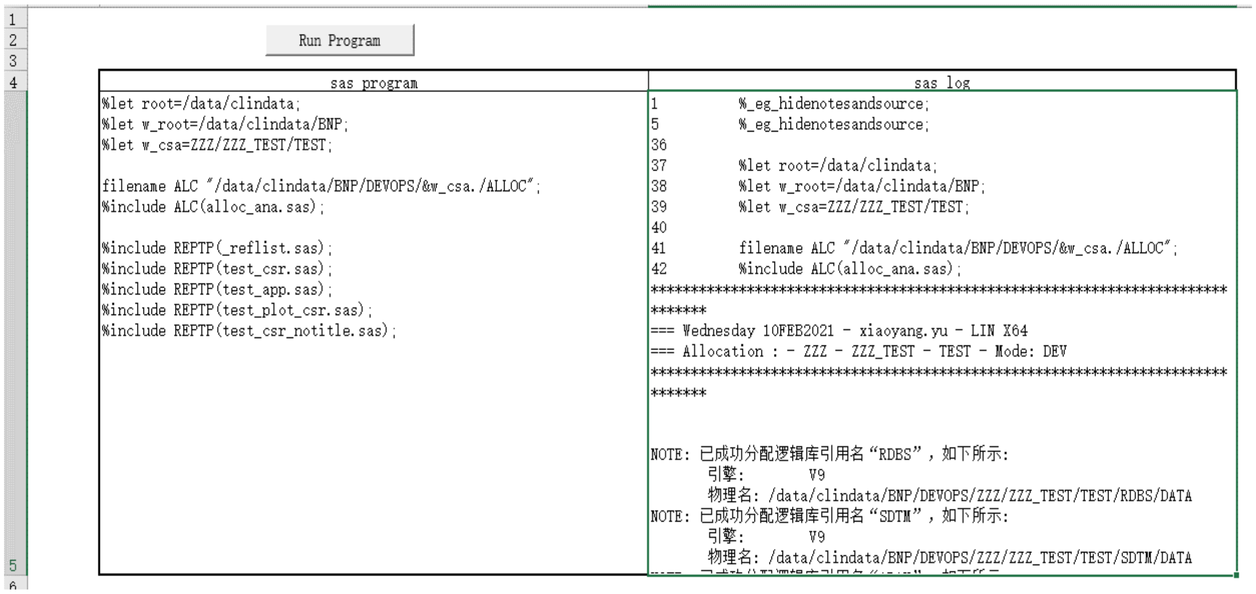
The purpose of “CreateObject” is to create the corresponding version of SAS EG application.

The "SetActiveProfile" uses the profile names obtained in the metadata when connecting to SAS EG. The "Applicatioin.New" and "CodeCollection.Add" create a project and add a new program code to it.

The SAS program is created by assigning plain text to “sasProgram.Text", the code is running on one of the server “SASApp" from servers list. The LOG is saved in “Sheet1" worksheet and also displayed in the "C4" cell as shown in Figure 2. Figure 3 shows many useful information in the VBA macro, the SAS codes are ran on the remote server and results are pushed back to COM-object and available in Excel. One can explore it and decide what to do with the output.

## EXAMPLE 2: Batch Run SAS Codes

The following code (Figure 4) demonstrates how to run SAS programs in a batch. It is not required to create new shell / bat file or to consider encoding since we can invoke SAS Enterprise Guide API. Example 2 shows how to batch run the programs to create sas dateasets, log, output etc..



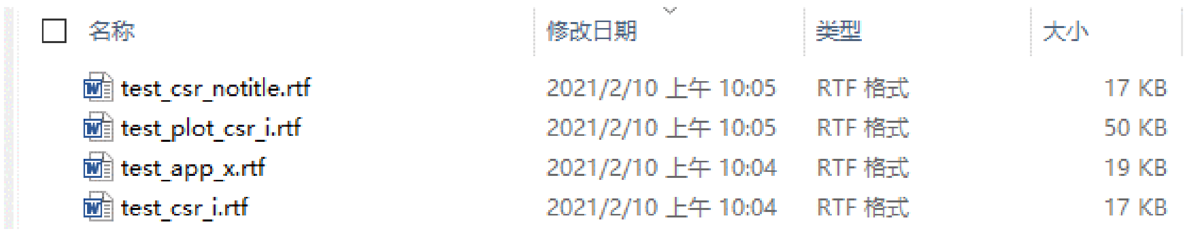
```
%let root=/data/clindata;
%let w_root=/data/clindata/BNP;
%let w_csa=ZZZ/ZZZ_TEST/TEST;

filename ALC "/data/clindata/BNP/DEVOPS/&w_csa./ALLOC";
%include ALC(alloc_ana.sas);

%include REPTP(_reflist.sas);
%include REPTP(test_csr.sas);
%include REPTP(test_app.sas);
%include REPTP(test_plot_csr.sas);
%include REPTP(test_csr_notitle.sas);
```

```
1          %_eg_hidenotesandsource;
5          %_eg_hidenotesandsource;
36
37         %let root=/data/clindata;
38         %let w_root=/data/clindata/BNP;
39         %let w_csa=ZZZ/ZZZ_TEST/TEST;
40
41         filename ALC "/data/clindata/BNP/DEVOPS/&w_csa./ALLOC";
42         %include ALC(alloc_ana.sas);
***********************************************************************
*******
=== Wednesday 10FEB2021 - xiaoyang.yu - LIN X64
=== Allocation : - ZZZ - ZZZ_TEST - TEST - Mode: DEV
***********************************************************************
*******


NOTE: 已成功分配逻辑库引用名“RDBS”，如下所示:
      引擎:        V9
      物理名: /data/clindata/BNP/DEVOPS/ZZZ/ZZZ_TEST/TEST/RDBS/DATA
NOTE: 已成功分配逻辑库引用名“SDTM”，如下所示:
      引擎:        V9
      物理名: /data/clindata/BNP/DEVOPS/ZZZ/ZZZ_TEST/TEST/SDTM/DATA
```

**Figure 4. Batch Run SAS Programs**

After initializing the SAS configuration, I called SAS programs in batches through the %include statement to submit batch run assignments.

According to the way of invoking SAS EG API by using VBA, The sas programs in the figure above have created outputs we need (Figure 5).



| 名称 | 修改日期 | 类型 | 大小 |
|---|---|---|---|
| test_csr_notitle.rtf | 2021/2/10 上午 10:05 | RTF 格式 | 17 KB |
| test_plot_csr_i.rtf | 2021/2/10 上午 10:05 | RTF 格式 | 50 KB |
| test_app_x.rtf | 2021/2/10 上午 10:04 | RTF 格式 | 19 KB |
| test_csr_i.rtf | 2021/2/10 上午 10:04 | RTF 格式 | 17 KB |

**Figure 5. Outputs of Batch Run**

Assume we have requirements of batch run many SAS programs at server locations. This technique is to provides a solution to connect SAS EG API to implementing our requirements. Otherwise there is no need to consider about the differences between UNIX and Windows operating systems.

## ACCESSING AND RETRIVING SAS DATASETS FROM EXCEL VBA

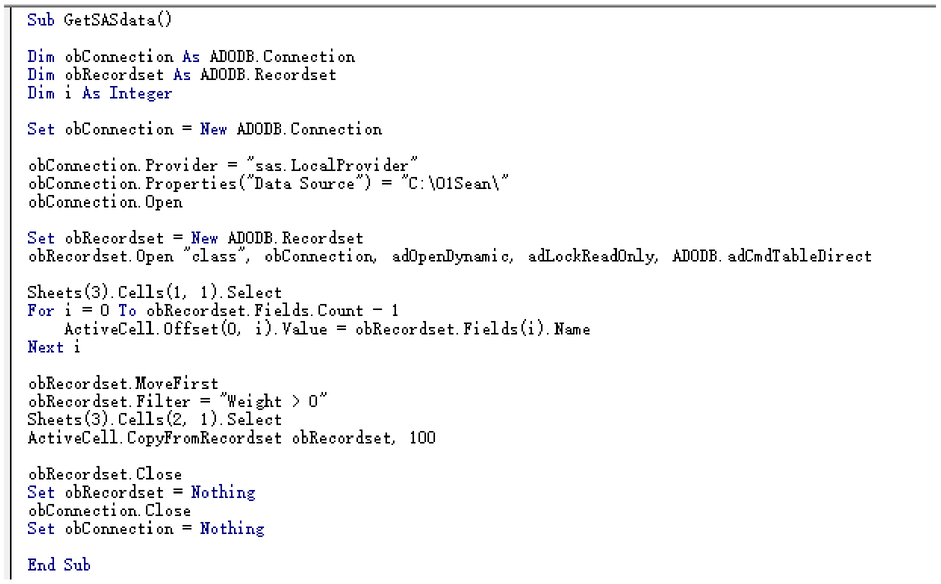
### MICROSOFT ADODB

In order to share data between SAS and Excel we will need to understand a bit about ADODB. For the purposes of this section there are two objects of interest – the Connection object and the Recordset object.

The Connection object provides properties to define the source of the data, and methods to manage the link between the client to the datasource. The Recordset object uses the Connection object to return data to the client. The Fields collection of the Recordset object provides data about the contents of the recordset.

## EXAMPLE 3: Read SAS Data to Excel

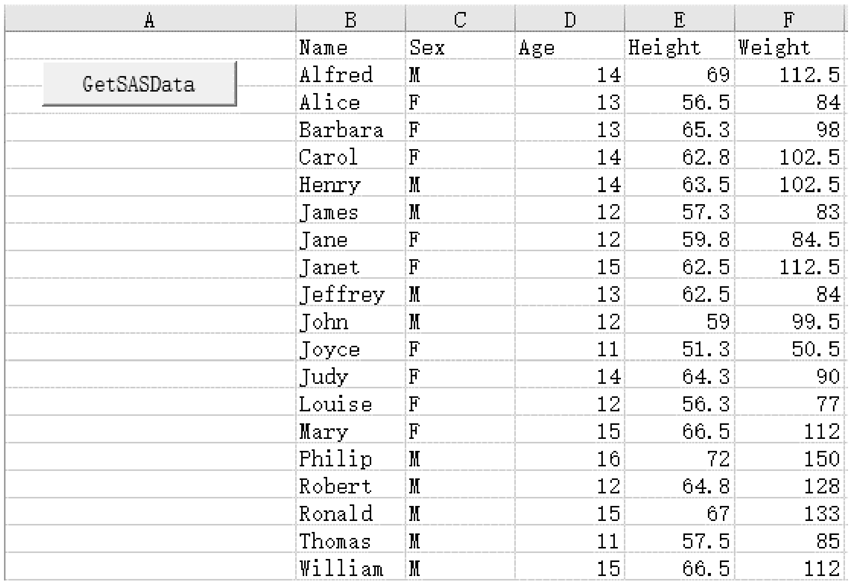
Let’s start with a simple connection and retrieval of data; in this case we will retrieve the contents of the table sashelp.class. To keep the demonstration as simple as possible and to dispaly the data Intuitively the Excel worksheet will use common spreadsheet components such as forms and buttons. It will return the contents of sashelp.class and display the contents in the Excel spreadsheet. Let us examine each of the lines of code here (see Figure 6).



```
Sub GetSASdata()

Dim obConnection As ADODB.Connection
Dim obRecordset As ADODB.Recordset
Dim i As Integer

Set obConnection = New ADODB.Connection

obConnection.Provider = "sas.LocalProvider"
obConnection.Properties("Data Source") = "C:\01Sean\"
obConnection.Open

Set obRecordset = New ADODB.Recordset
obRecordset.Open "class", obConnection, adOpenDynamic, adLockReadOnly, ADODB.adCmdTableDirect

Sheets(3).Cells(1, 1).Select
For i = 0 To obRecordset.Fields.Count - 1
    ActiveCell.Offset(0, i).Value = obRecordset.Fields(i).Name
Next i

obRecordset.MoveFirst
obRecordset.Filter = "Weight > 0"
Sheets(3).Cells(2, 1).Select
ActiveCell.CopyFromRecordset obRecordset, 100

obRecordset.Close
Set obRecordset = Nothing
obConnection.Close
Set obConnection = Nothing

End Sub
```

**Figure 6. Code of Reading SAS Dataset**

Through clicking GetSASData button to run the GetSASData macro in the VBE. The results of class.sas7bdat will display in the worksheet three. Please refer to the following figure(Figure 7).



| A | B | C | D | E | F |
|---|---|---|---|---|---|
| | Name | Sex | Age | Height | Weight |
| | Alfred | M | 14 | 69 | 112.5 |
| | Alice | F | 13 | 56.5 | 84 |
| | Barbara | F | 13 | 65.3 | 98 |
| | Carol | F | 14 | 62.8 | 102.5 |
| | Henry | M | 14 | 63.5 | 102.5 |
| | James | M | 12 | 57.3 | 83 |
| | Jane | F | 12 | 59.8 | 84.5 |
| | Janet | F | 15 | 62.5 | 112.5 |
| | Jeffrey | M | 13 | 62.5 | 84 |
| | John | M | 12 | 59 | 99.5 |
| | Joyce | F | 11 | 51.3 | 50.5 |
| | Judy | F | 14 | 64.3 | 90 |
| | Louise | F | 12 | 56.3 | 77 |
| | Mary | F | 15 | 66.5 | 112 |
| | Philip | M | 16 | 72 | 150 |
| | Robert | M | 12 | 64.8 | 128 |
| | Ronald | M | 15 | 67 | 133 |
| | Thomas | M | 11 | 57.5 | 85 |
| | William | M | 15 | 66.5 | 112 |

**Figure 7. Result of Class SAS Dataset**

## BUILD VBA APPLICATIONS THAT RUNS SAS

With all the bells and whistles, one can create sophisticated VBA applications that runs SAS. Figure 4 shows a SAS running application, type SAS programs or copy and paste your own programs (including

SAS macros), click “Run Program", the SAS program runs on remote server and generates logs, listing, output and datasets at local machine. It is self-contained and no need to open SAS EG main application. Meanwhile once we do not install SAS, VBA applications is also great tools of reading, querying and analysing SAS datasets(Figure 7). Advanced SAS users can run batch jobs, schedule jobs in parallel, or use SAS output as input to other applications.

## CONCLUSION

This paper presents ideas of using VBA to debug and run SAS codes through SAS Enterprise Guide API. VBA provides a powerful running environment and debug tool which can run SAS programs smoothly and efficiently as a SAS client does, but also seamlessly link remote server to local so that we could automate SAS batch jobs without SAS local mode. We could also access local applications and objects.

## REFERENCES

Getting Started With VBA in Excel 2010,

https://msdn.microsoft.com/en-us/library/office/ee814737(v=office.14).aspx

MS Support, Developer Tab Support,

https://support.office.com/en-us/article/show-the-developer-tab-e1192344-5e56-4d45-931b-e5fd9bea2d45

MS Support, Assign a macro to a Form or a Control Button

https://support.office.com/en-us/article/assign-a-macro-to-a-form-or-a-control-button-d58edd7d-cb04-4964-bead-9c72c843a283

MS Support, Automate tasks with the Macro Recorder

https://support.office.com/en-us/article/Automate-tasks-with-the-Macro-Recorder-974ef220-f716-4e01-b015-3ea70e64937b

Accessing SAS Code via Visual Basic for Applications

https://www.lexjansen.com/nesug/nesug12/cc/cc09.pdf

Using SAS Enterprise Guide the Same Way as Base SAS and More

https://support.sas.com/resources/papers/proceedings12/300-2012.pdf

Boost Your Programming Productivity with SAS Enterprise Guide

https://support.sas.com/resources/papers/proceedings/proceedings/sugi30/019-30.pdf

SAS Enterprise Guide Scripts

http://support.sas.com/documentation/onlinedoc/guide/

## CONTACT INFORMATION

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

Xiaoyang Yu
Anheart Therapeutics(Hangzhou) Co.,Ltd.
13521122930@163.com