

## Project Automation? Scripting helps you out

Sean Yan, Shaozun Zhang, Jason Wang, WuXi CDS, Shanghai, China

### ABSTRACT

Before the delivery of the SDTM/ADaM the lead programmer will need to do a series of repeated process for the deliverable package. This could be labor intensive and time consuming when some intermediate part failed and leads to a start over. This paper introduces the automation method through scripting on how to execute certain series of SAS programs, produce the Pinnacle 21 validation report using desired parameter, processing the validation report using an existing commented validation report. This allows a full automation of the project from program executing, domain validation, validation report processing under a simple double click.

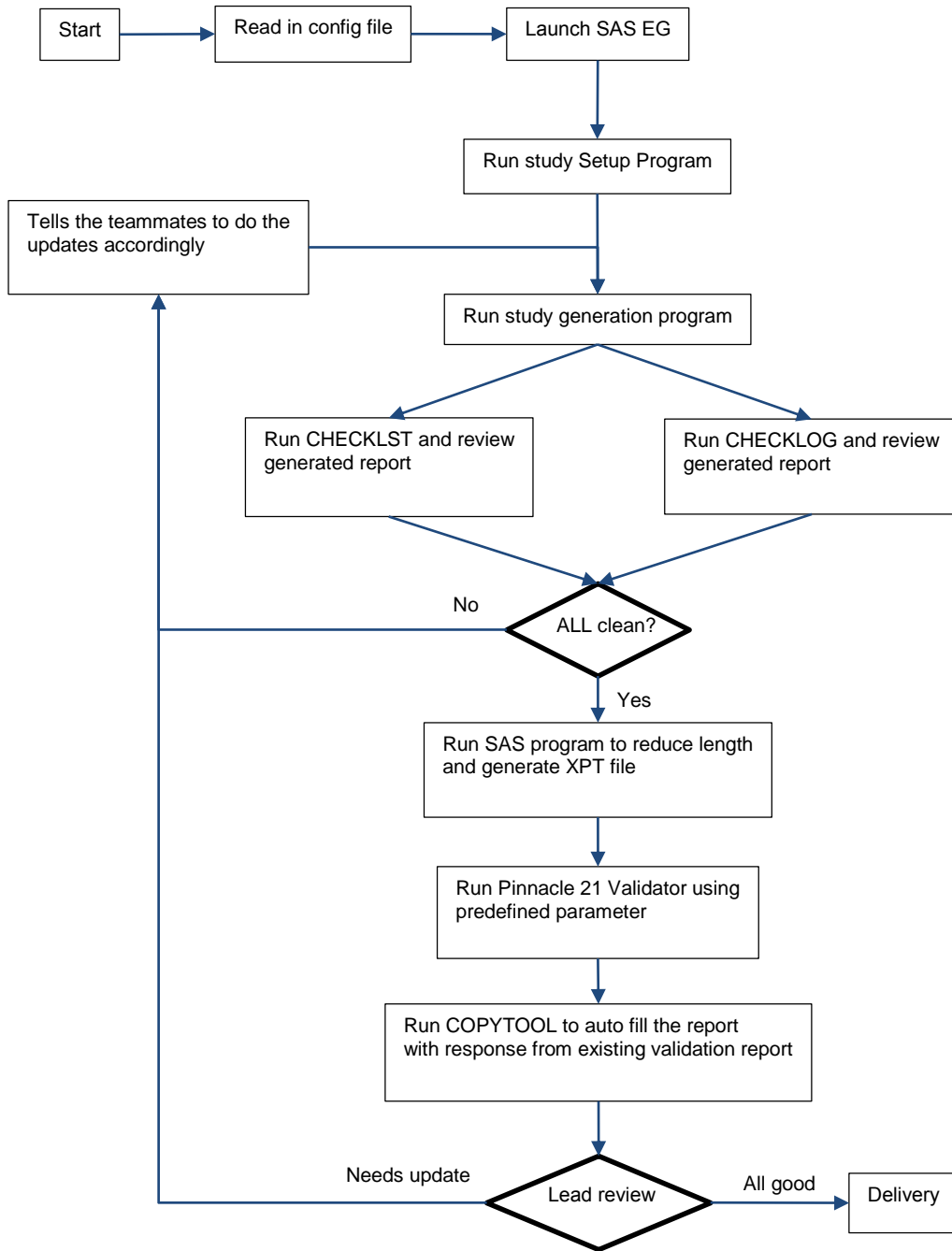
### INTRODUCTION

Many times, for the SDTM/ADaM deliverable package preparation the Lead programmers will be drowned into the procedures of program execution, log/compare result review, domain validation and validation report review. This could be labor costing, time consuming and even frustrating when there is some unwanted log issue or a variable has minor mismatch or the CDISC report needs to be responded manually. In order to make leader's life easier we created OneKeyCDISCReport (hereinafter referred to as OKCR) using VBScript to automatically execute

1. Program execution
2. Log & Compare result review
3. Domain validation
4. Validation report processing

### GENERAL LOGIC OF OKCR

The general logic of our OKCR is almost same when the lead need to prepare the SDTM/ADaM deliverable. It consist of four main steps as program execution, log & compare result review, domain validation and validation report processing. Below is the Logic diagram of the whole OKCR.



Logic diagram of OKCR

## PROGRAM EXECUTION

### Read in Config file

At the very beginning, the OKCR reads in a configuration file named automation.config. The configuration file contains the information the OKCR needs. Eg. the location of the program to be executed, the path of the log&lst file to be checked and the parameters the Pinnacle validator to be used.

Below is an example of how the automation.config looks like.

```
Dim studysetup, sdtmrunall, xptprm, logfilespath, lstpath, xptpath, valtype, scriptdt, valrep, refrep, ctversion, meddraversion
```

```
'study setup sasprogram
studysetup = "\\Server\study\program\sastest.sas"

'full path of the previous pinnacle21 report to refer
refrep = "\\ Server\study\document\opencdisc-report-2017-05-26T04-09.xlsx"

'Version of CDISC CT
ctversion = "2016-06-24"

'Meddra version
meddraversion="19.0"
```

The “Dim” statement defines the parameters that the program is going to use and then specifies the parameters with detailed information. Mostly like the Macro variables we used in the SAS program.

### Launch SAS EG and Run Study Program

After the configuration the OKCR executes the study setup program and the study program. In order to do this the OKCR needs to launch the SAS EG program and then read in the studysetup program and runs it. This all happens behind the desk which makes the whole process much faster than manually operated.

```
' Create a new Project
Set Project = sasapp.New()

' open the file and put the text into the fileString variable
Dim GfileSys, GfReadStream, GfileString
Set GfileSys = CreateObject("Scripting.FileSystemObject")
Set GfReadStream = GfileSys.OpenTextFile(studysetup, 1, False)
GfileString = GfReadStream.ReadAll()

' add a new code object to the Project
Set sasProgram = Project.CodeCollection.Add

' Set the server (by Name) and text for the code
sasProgram.Server = "SASApp_PRD"

' Create the SAS program to run
sasProgram.text = GfileString

' Run the code
sasProgram.Run
```

Similarly the OKCR reads in and runs the study program which contains our SDTM/ADaM programs. The program itself has the whole project combined by using %include statement.

```
filename runall "\\server\study\sas\ " ;
%include runall(DM);
%include runall(SE);
%include runall(QC_DM);
%include runall(QC_SE);
```

Each program will output its log to a defined location. And the QC program will also output a compare result to an .lst file for further use.

## LOG/COMPARE RESULT REVIEW

After the OKCR executed the study program it calls the CHECKLOG and CHECKLST program to review the log file and compare results generated by the study program to see if the logs are clean and if there is any outstanding issues in the compare result. If there is anything outstanding in the log or compare the program will generate a txt file condensing the part that needs to be updated and automatically opens the file for lead's convenience.

### Compare Result Review

In the compare results there are certain patterns for a clean compare results and if the compare result is not clean we can also target the mismatch pattern to see where the compare went wrong. Here is an example

```
Set mypat1 = new regexp
With mypat1
  .Global = False
  .IgnoreCase = True
  .Pattern = "Number of Variables with Differing Attributes: (\d+)"
End With

Set mypat2 = new regexp
With mypat2
  .Global = False
  .IgnoreCase = True
  .Pattern = "number of variables compared with some observations unequal: (\d+)"
End With

Set mypat3 = new regexp
With mypat3
  .Global = False
  .IgnoreCase = True
  .Pattern = "Number of Variables in [^\s]+ but not in [^\s]+ (\d+)"
End With

Set mypat4 = new regexp
With mypat4
  .Global = False
  .IgnoreCase = True
  .Pattern = "Number of observations in [^\s]+ but not in [^\s]+ (\d+)"
End With
```

The above statements targets at the patterns that typically appears in a mismatch compare result. And the mismatch results are outputted to a text file as below

```
AE:
Variables with Differing Attributes.
EX:
Variables compared with some observations unequal.
MH:
Observations not in common.
SV:
variables not in common.
```

### Log Check Review

For the logs as we all know that the SAS log has not only the error and warning that needs our attention, but some notes are also not permitted in the log output, eg. "Converted", "not valid". So the CHECKLOG program searches for this kind of key words and output to the text file shown as below.

```
DM:
NOTE: Character values have been converted to numeric values at the places given
by: (Line):(Column).
NOTE: Variable sexn is uninitialized.
```

```
ERROR: Variable raceoth is not on file WORK.TEST.
WARNING: The data set WORK.TEST may be incomplete.  When this step was stopped
there were 0 observations and 28 variables.
WARNING: Data set WORK.TEST was not replaced because this step was stopped.
SE:
WARNING: Multiple lengths were specified for the BY variable usubjid by input data
sets. This might cause unexpected results.
NOTE: MERGE statement has more than one data set with repeats of BY values.
NOTE: At least one W.D format was too small for the number to be printed. The
decimal may be shifted by the "BEST" format.
```

With this output the lead programmer will be easy to see if there is anything outstanding without scrolling down the hundreds of pages of log file or compare results. And just need simple copy and paste to deliver the study issue to the teammates for update. Once the OKCR identifies any log issue or compare mismatch it will pop up prompt to the user to decide whether to continue with the following procedures. And if everything goes well in the log and compare result the OKCR automatically moves forward.

## DOMAIN VALIDATION&REPORT PROCESSING

Once the datasets are ready the OKCR the moves to the domain validation and report processing part. In this part the OKCR calls several programs in a row.

### 1. Call SAS macro ReduceLength.sas

This macro reads in the max length of a variable and then assigns this length to that variable. This is to avoid a validation warning says that the length is too long for the variable compare to its actual length.

### 2. Call SAS macro GenerateXPT.sas

This macro simply uses Proc Copy procedure to transform the data from SAS7BDAT to XPT format for later validation use.

### 3. Call function RunValidator

This program reads the parameters from the config file and transfer the parameter to the Pinnacle 21 validator and generate related SDTM/ADaM validation report. Here is an example of the RunValidator.

```
sub runvalidator

ws.Run "%Comspec% /c java -jar
""\kcifile\docs\GlobalDataStandards\Applications\Validator\pinnacle21-community
2.2\components\lib\Validator-cli-2.1.5.jar"" _
    & " -type=" & valtype & " -source=" & xptpath & "*.xpt" _
    & " -report=" & valrep & " -report:overwrite=Yes" _
    & " -
config=""\kcifile\docs\GlobalDataStandards\Applications\Validator\pinnacle21-
community 2.2\components\config\sdtm 3.2.xml"" " _
    & " -config:cdisc=" & ctversion & " -config:meddra=" & meddraversion, 0,
True

end sub
```

In the program the parameter valtype, xptpath, valrep, ctversion and meddraversion were already defined in the configuration file.

### 4. Call function CopyResponse

For the SDTM/ADaM deliverable we do not only provide the CDISC validation report, we also provide the response to the issues in the report (in column G) to pre-answer the questions from the client. And it could be painful if the lead has to manually respond to the issue one by one even this was done in previous delivery for same issue. The CopyResponse function reads in a previous annotated CDISC report and creates a dictionary, the domain name & issue detail will be combined as key and the previous response as value. Then the program reads the newly generated CDISC report and applies the dictionary to the new report. So if the new report has

the same domain plus issue detail combination the previous response will be automatically copied over to the new report. Below is an example of how this works.

*The previously annotated CDISC report*

DM			
<a href="#">SD0087</a>	RFSTDTC is not provided for a randomized subject	Warning	2 Test response1
<a href="#">SD0088</a>	RFENDTC is not provided for a randomized subject	Warning	21 Test response 2

*The dictionary from the annotated CDSIC report*

ID Key	Value
DM + RFSTDTC is not provided for a randomized subject	Test response1
DM + RFENDTC is not provided for a randomized subject	Test response 2

*The newly generated report*

DM			
<a href="#">SD0087</a>	RFSTDTC is not provided for a randomized subject	Warning	2
<a href="#">SD0088</a>	RFENDTC is not provided for a randomized subject	Warning	21
<a href="#">SD1210</a>	Missing value for RFICDTC	Warning	13

If there appears new issues that was not in the previous CDISC report the lead then needs to review the issue to see if this needs to be updated in the domain or could be explained in the report.

## CONCLUSION

In this paper we introduced the tool OneKeyCDISCReport and its logic with detailed algorithm. The tool will save the lead program a substantial amount of time during preparing the SDTM/ADaM deliverable package. The OKCR mainly saves time in several points.

- Study log review. Even if we have the log outputted under a certain location it could be time consuming to review hundreds of pages of log files. A condensed log output directly point which domain has log issue and what kind of the issue is.
- Study compares review. When manually go over the compare result the lead will need to scroll down dozens of pages of results in the hope of nothing is going wrong.
- CDISC report processing. The lead will need to manually copy and paste dozens of lines of response to the new report while reviewing if the new issue needs to be updated.

Due to the possibility that the study may need to be run 4 to 5 times with potential updates due to log issue/compare mismatch/CDISC update. The above repeated process may devour several hours from the lead programmer. And OKCR is here to help.

## REFERENCES

Chris Hemedinger. 2012. "Not Just for Scheduling: Doing More with SAS® Enterprise Guide® Automation." *Proceedings of the SAS Global Forum 2012 Conference*. Available at [http://support.sas.com/documentation/onlinedoc/guide/examples/SASGF2012/Hemedinger\\_298-2012.pdf](http://support.sas.com/documentation/onlinedoc/guide/examples/SASGF2012/Hemedinger_298-2012.pdf)

Pinnacle 21. "Using OpenCDISC Validator CLI." 2017-06-17. Available at <https://www.pinnacle21.com/using-opencdisc-validator-cli>.

Microsoft. "VBScript Language Reference." 2017-06-17. Available at [https://msdn.microsoft.com/en-us/library/d1wf56tt\(v=vs.84\).aspx](https://msdn.microsoft.com/en-us/library/d1wf56tt(v=vs.84).aspx).

## CONTACT INFORMATION

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

Name: Sean Yan  
Enterprise: WuXi CDS

Address: 4th floor, Tower 18th (WuXiAppTec in Wuhan), D4 district in Biolake, 666 Gaoxin Avenue, Eastlake Hi-Tech Development Zone  
City, State ZIP: Wuhan, Hubei 430075  
Work Phone: 18672757841  
Fax: 027-6539 5393  
E-mail: yan\_xiang@wuxiapptec.com  
Web: <https://www.wuxiapptec.com.cn/>

Name: Shaozun Zhang  
Enterprise: WuXi CDS  
Address: 4th floor, Tower 18th (WuXiAppTec in Wuhan), D4 district in Biolake, 666 Gaoxin Avenue, Eastlake Hi-Tech Development Zone  
City, State ZIP: Wuhan, Hubei 430075  
Work Phone: 15972127335  
Fax: 027-6539 5393  
E-mail: Zhang\_shaozun@wuxiapptec.com  
Web: <https://www.wuxiapptec.com.cn/>

Name: Jason Wang  
Enterprise: WuXi CDS  
Address: 4th floor, Tower 18th (WuXiAppTec in Wuhan), D4 district in Biolake, 666 Gaoxin Avenue, Eastlake Hi-Tech Development Zone  
City, State ZIP: Wuhan, Hubei 430075  
Work Phone: 13986183664  
Fax: 027-6539 5393  
E-mail: Jason\_wang@wuxiapptec.com  
Web: <https://www.wuxiapptec.com.cn/>

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are trademarks of their respective companies.