

ProjectTrackIt: Automate your project using SAS®

Abhishek Bakshi, Cytel, Pune, India

ABSTRACT

Entering information in a project programming tracker is one of the menial tasks taking up time which causes hindrance for project leads to get accurate and up to date information of the project at any given time. So how about getting it done within a click? This paper discusses a tool that solves this problem by using `%ReadPDF` & `%QCCheck` macro utilities. These when used in conjunction with ODS creates a fully automated project tracking spreadsheet which gives a single shot overview of any project in real time. This utility also creates some of the specification documents/template and other project related documents which are required for regulatory submission. Thus, it reduces the dependency on manual entry of information and saves programmer's time. Furthermore functionalities of this utility will be discussed in the paper in detail

1. INTRODUCTION

In a fast paced production environment of a clinical study, where timelines are short and last minute change requests are frequent, quick setup of the project, timely and accurate tracking of project status and quick wind-up of the project are vital to the timely delivery of the outputs for regulatory submissions.

This paper introduces a SAS® utility that extracts annotation from SDTM annotated CRF, creates a template for project tracking, performs the project startup related activities to eliminate manual entry of information, keep track of validation comparison status and re-validation requirements. This utility also ensures that a project team always has an up-to-date picture of project status at any given time.

2. OVERVIEW

The utility (Fig 1) is implemented for PC SAS® 9.2 and uses `%ReadPDF`, `%ProjectTracking` & `%QCCheck` macro utilities in conjunction with the ODS TAGSETS EXCEL.XP¹. Below are the features of the utility.

`%ReadPDF` features as follows:

- ❖ Extract the annotation data from SDTM annotated-CRF to SAS® dataset
- ❖ Annotation validation as per SDTM guidelines
- ❖ Create SDTM specifications from extracted data

`%ProjectTracking` features as follows:

- ❖ Create project allocation plan for entering the project allocations,
- ❖ Create blank SAS® program files for both (source & validator) with header information
- ❖ Create blank Validation finding & Issues log documents
- ❖ Create runall.bat batch file for all the codes in the project

`%QCCheck` features as follows:

- ❖ Extract the last modified time stamp for the SAS® codes & Datasets
- ❖ Check log (source & validator) is error & warning free
- ❖ Display PROC COMPARE results & compare run time for all codes in one common dataset
- ❖ Displays whether validation code is batch submitted after the developers code

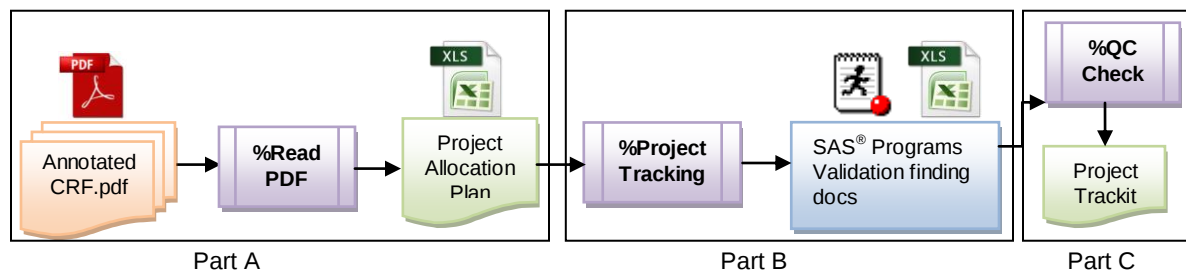
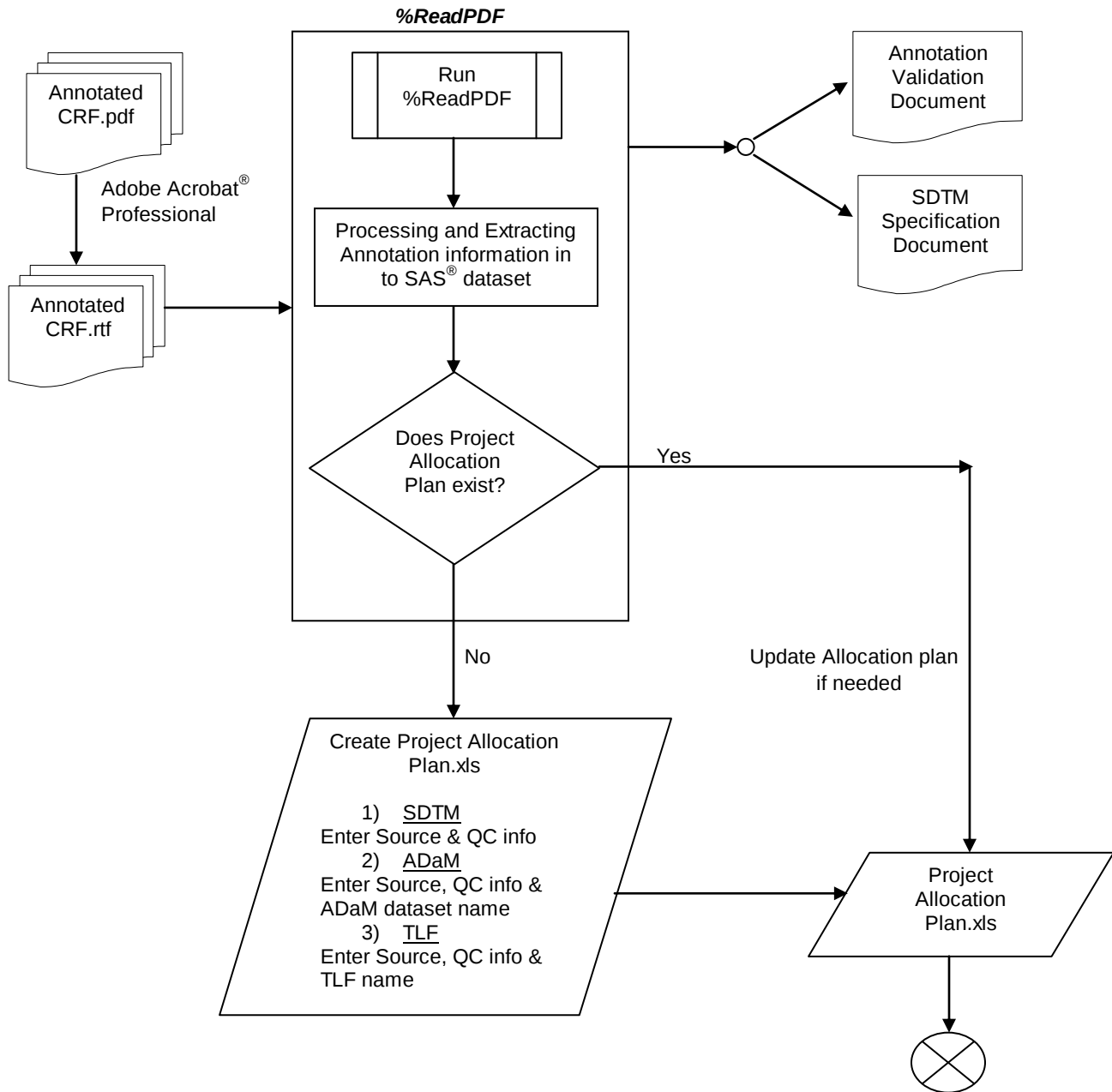


Fig 1: `%ProjectTrackit` in nutshell

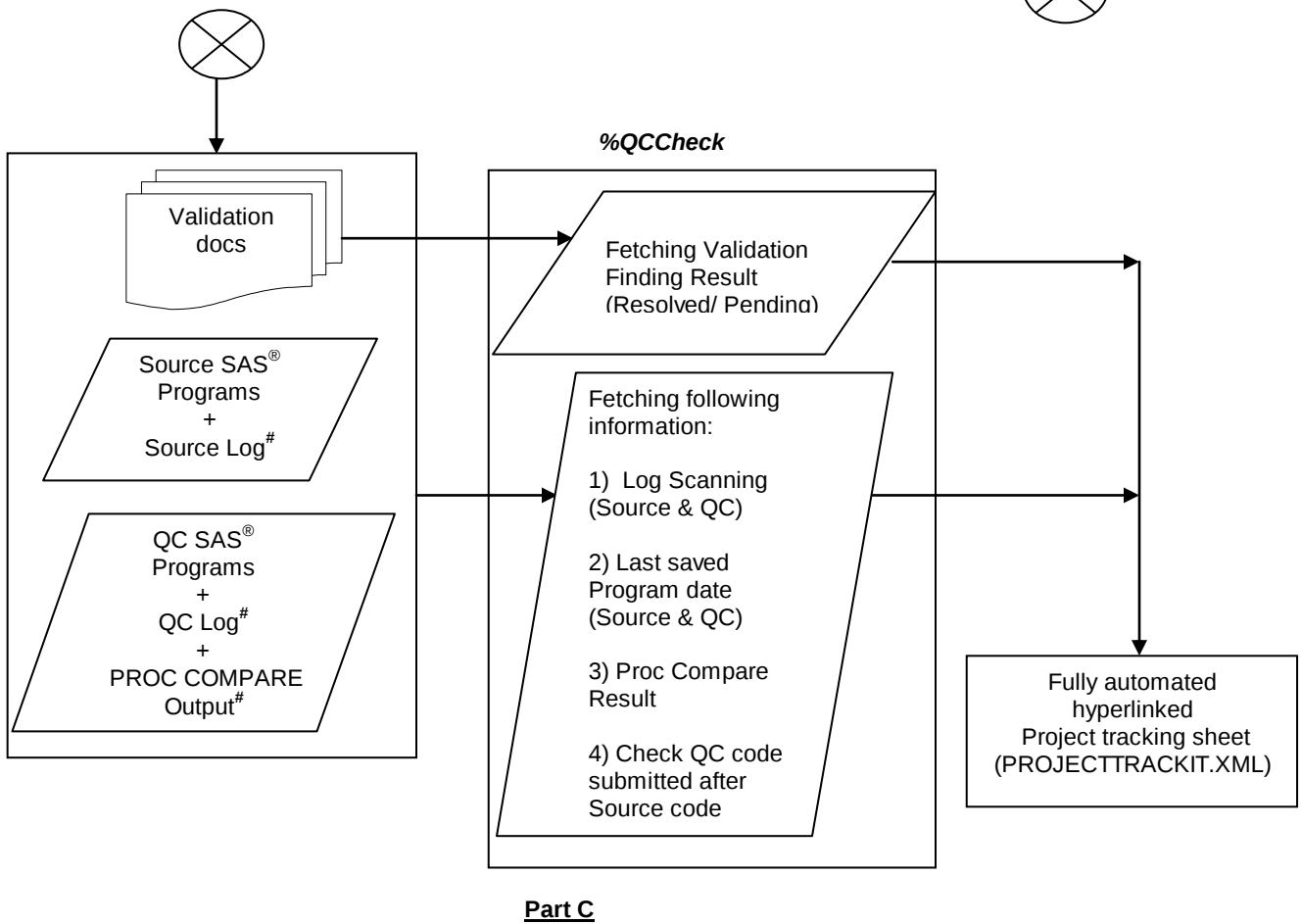
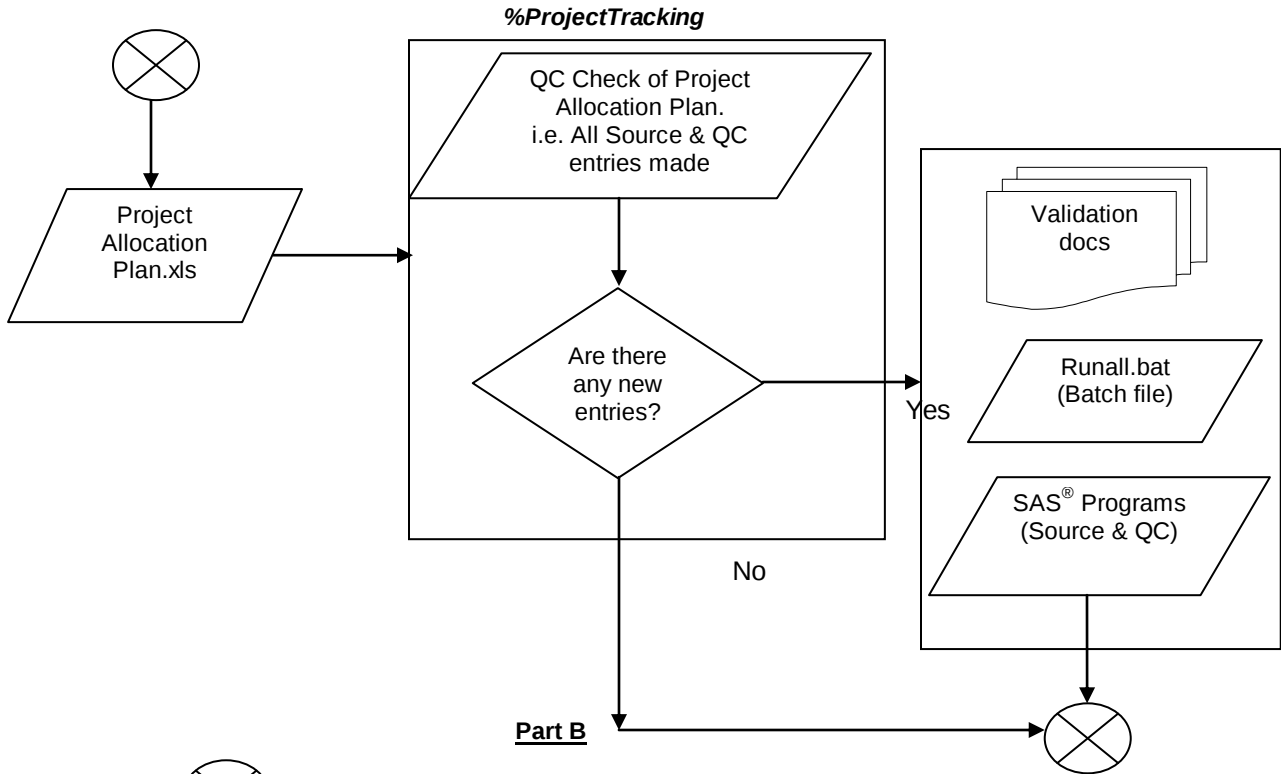
3. DESIGN

Below is the *ProjectTrackit* utility flow chart(Part A, Part B & Part C)



Part A

PhUSE 2013



Logs & Outputs are generated from the SAS® codes, executed outside the utility

PHUSE 2013

4. PROJECTTRACKIT FILE

The output of the utility is an XML based file (*ProjectTrackit.xml*). Once run, it gives a single snapshot of complete project status at any given time (Any reporting event for a trial like interim or final analysis can be defined as project). It contains the list of all domains/dataset/tables present in the project, source & validation programmer's user id, hyperlinked program name, log, output, program last modified date, validation finding status, PROC COMPARE result and run time all in one central location. The following is the sample *ProjectTrackit.xml*, the columns of this file have been split into two for the purpose of understanding (Table 1).

Domain/ Dataset/ TLF	Source Code	Source Log	Source Output Dataset	Source Code Last Update Time(EST)	Source Initial	QC Code	QC Log
AE	c-ae.sas	c-ae.log	ae.sas7bdat	27AUG2013:12:27:08	ABC	qc-ae.sas	qc-ae.log
CM	c-cm.sas	c-cm.log	cm.sas7bdat	28AUG2013:12:13:09	DEF	qc-cm.sas	qc-cm.log
CO	c-co.sas	c-co.log	co.sas7bdat	28AUG2013:10:12:10	XYZ	qc-co.sas	qc-co.log
DF	c-df.sas	c-df.log	df.sas7bdat	29AUG2013:11:09:10	PQR	qc-df.sas	qc-df.log
DM	c-dm.sas	c-dm.log	dm.sas7bdat	26AUG2013:09:20:12	ABC	qc-dm.sas	qc-dm.log
DS	c-ds.sas	c-ds.log	ds.sas7bdat	27AUG2013:18:34:07	DEF	qc-ds.sas	qc-ds.log
EX	c-ex.sas	c-ex.log	ex.sas7bdat	27AUG2013:05:47:05	XYZ	qc-ex.sas	qc-ex.log
HE	c-he.sas	c-he.log	he.sas7bdat	28AUG2013:16:17:01	PQR	qc-he.sas	qc-he.log
IE	c-ie.sas	c-ie.log	ie.sas7bdat	26AUG2013:20:30:03	ABC	qc-ie.sas	qc-ie.log
OI	c-oi.sas	c-oi.log	oi.sas7bdat	29AUG2013:10:56:07	DEF	qc-oi.sas	qc-oi.log
SX	c-sx.sas	c-sx.log	sx.sas7bdat	28AUG2013:19:24:18	XYZ	qc-sx.sas	qc-sx.log
VS	c-vs.sas	c-vs.log	vs.sas7bdat	29AUG2013:05:27:19	PQR	qc-vs.sas	qc-vs.log
SUPPAE	c-ae.sas	c-ae.log	suppae.sas7bdat	27AUG2013:12:27:08	ABC	qc-ae.sas	qc-ae.log

Domain/ Dataset/ TLF	QC Output Dataset	QC Code Last Update Time(EST)	QC Initial	Validation Finding	Re-Test (Pass/QC Re-run)	PROC COMPARE (Pass/Fail)
AE	qc_ae.sas7bdat	29AUG2013:11:27:08	PQR	Pending	Pass	Fail
CM	qc_cm.sas7bdat	28AUG2013:22:10:29	XYZ	No Finding	Pass	Pass
CO	qc_co.sas7bdat	29AUG2013:05:02:13	DEF	Pending	QC Re-run	Pass
DF	qc_df.sas7bdat	29AUG2013:11:55:34	ABC	No Finding	Pass	Pass
DM	qc_dm.sas7bdat	26AUG2013:10:36:20	PQR	No Finding	Pass	Fail
DS	qc_ds.sas7bdat	28AUG2013:22:45:48	XYZ	No Finding	Pass	Pass
EX	qc_ex.sas7bdat	27AUG2013:05:47:05	DEF	Resolved	Pass	Pass
HE	qc_he.sas7bdat	29AUG2013:10:14:59	ABC	Resolved	QC Re-run	Pass
IE	qc_ie.sas7bdat	26AUG2013:21:03:39	PQR	No Finding	Pass	Pass
OI	qc_oi.sas7bdat	29AUG2013:22:20:10	XYZ	Resolved	Pass	Pass
SX	qc_sx.sas7bdat	29AUG2013:14:05:25	DEF	Resolved	Pass	Pass
VS	qc_vs.sas7bdat	29AUG2013:15:19:41	ABC	No Finding	Pass	Pass
SUPPAE	qc_suppae.sas7bdat	29AUG2013:11:27:08	PQR	Pending	Pass	Fail

Table1. Sample *ProjectTrackit.xml*

5. IMPLEMENTATION OF THE UTILITY

5.1 ON THE FIRST RUN

- ❖ Copy the program (*ProjectTrackit.sas*) in the project directory and execute the code
- ❖ Allocate the resources in project allocation plan sheet (*ProjectAllocationPlan.xls*)
- ❖ Blank SAS® codes with headers are created as per the project allocation plan
- ❖ *ProjectTrackit.xml* file is created with only program(Source & QC) and initials column populated

5.2 DURING THE PROJECT

- ❖ Run the utility at any given time
- ❖ Following information is fetched from the project directory
 - Last update time stamp for SAS® programs(Source & QC)
 - Overall Project QC Status(pass/fail)
 - Verify whether validation finding are resolved or pending
- ❖ Creates *ProjectTrackit.xml* with all column(Table 1) populated

6. BENEFITS OF THE AUTOMATED PROJECT TRACKING

6.1 VALIDATION OF CRF ANNOTATION AS PER SDTM GUIDELINES

SDTM annotated CRF serves as one of the first specification document for programmers so it is vital for clinical teams to have CRFs annotated correctly. Manual review is usually the method used to ensure whether annotations are done as per SDTM guidelines. But in case of multiple studies with large number of analysis and having volumes of CRF pages it becomes difficult to review them manually. This limitation can be overcome by using this utility. The utility imports CRF Annotations to a SAS® dataset and generates a validation report that gives out the list of custom domains present in the CRF and the list of variables which are not as per SDTM IG 3.1.2.(Fig 2 & Table 2)

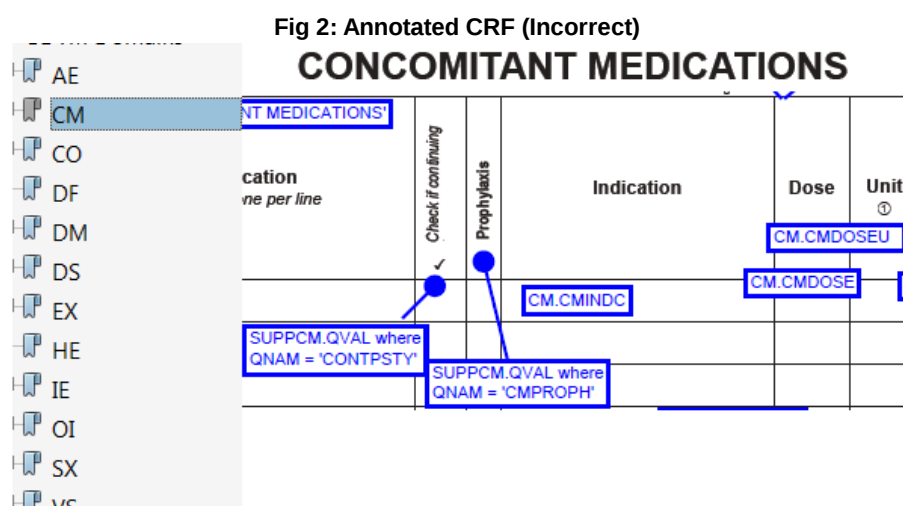


Table 2: Annotation Validation report

Test Condition	Domain	Variable name
List of custom domain variables	DF	DFCAT
		DFORRES
		DFTESTCD
	HE	HECAT
		HETERM
	OI	OICAT
		OIORRES
		OITESTCD
	SA	SACAT
		SADTC
		SAORRES
	SX	SXCAT
		SXOCCUR
		SXTERM
Variable not as per SDTM IG 3.1.2	CM	CMDOSEU

PhUSE 2013

6.1.1 IMPACT OF AUTOMATION

- ❖ Annotation discrepancies can be discovered before the programming team starts coding
- ❖ Reduces the rework if incorrectly annotated
- ❖ Gives the list of all custom domain at one central location

6.2 CREAT SDTM SPECIFICATION DOCUMENT

This utility creates a SDTM specification document from the annotated CRF. (Fig 3) It also stores the specification document in a SAS® dataset so If annotation gets updated it automatically adds it to specification document.

no	Domain Name (Prefix)	Domain.Variabl e	Variable Label	Type	Controlled Terms or Format	Role	Core	References
1	AE	AE.STUDYID	Study Identifier	Char		Identifier	Req	
2	AE	AE.DOMAIN	Domain Abbreviation	Char	**AE	Identifier	Req	
3	AE	AE.SUBJID	Unique Subject Identifier	Char		Identifier	Req	
4	AE	AE.AESEQ	Sequence Number	Num		Identifier	Req	
5	AE	AE.AEGRPID	Group ID	Char		Identifier	Perm	
6	AE	AE.AEREFID	Reference ID	Char		Identifier	Perm	
7	AE	AE.AESPID	Sponsor-Defined Identifier	Char		Identifier	Perm	
8	AE	AE.AETERM	Event	Char		Topic	Req	
9	AE	AE.AEMODIFY	Modified Reported Term	Char		Qualifier	Perm	
10	AE	AE.AEDECOD	Dictionary-Derived Term	Char	*	Qualifier	Req	
11	AE	AE.AECAT	Category for Adverse Event	Char	*	Grouping Qualifier	Perm	
12	AE	AE.AESCAT	Subcategory for Adverse Event	Char	*	Grouping Qualifier	Perm	
13	AE	AE.AEPRESP	Pre-specified Adverse Event	Char	NY	Record Qualifier	Perm	
14	AE	AE.AEBODSYS	Body System or Organ Class	Char	SOC	Record Qualifier	Exp	
15	AE	AE.AELOC	Location of the Reaction	Char	*	Record Qualifier	Perm	
16	AE	AE.AESEV	Severity/Intensity	Char	AESEV	Record Qualifier	Perm	

Fig 3: Sample SDTM Specification document

6.2.1 IMPACT OF AUTOMATION

- ❖ Specification document is created well before the programming team starts coding
- ❖ Automatically updates the specification document sheet
- ❖ Information about all domains can be found at one central location
- ❖ Specification document can be easily maintained

6.3 RESOURCE ALLOCATION PLAN

The utility helps in creation of the most important part of any project, the resource allocation plan. It creates separate sheets each for SDTM, ADaM & TLF where project leads enter the resource information. Once all the required columns (highlighted in yellow Table 4) of sheets are filled, program executes further.

SN	Domain	Dev Code	Dev Name	Val Code	Val Name	SN	Domain	Dev Code	Dev Name	Val Code	Val Name
1	AE	c-ae	ABC	qc-ae	PQR	1	ADSL	c-adsl	ABC	qc-adsl	PQR
2	CM	c-cm	DEF	qc-cm	XYZ	2	ASDAE	c-adae	DEF	qc-adae	XYZ
3	CO	c-co	XYZ	qc-co	DEF	3	ASLBASE	c-aslbase	XYZ	qc-aslbase	DEF
4	DF	c-df	PQR	qc-df	ABC	4	ASLEFF	c-asleff	PQR	qc-asleff	ABC
5	DM	c-dm	ABC	qc-dm	PQR	ADaM Sheet					
6	DS	c-ds	DEF	qc-ds	XYZ	Yellow Highlighted is entered by Study Lead					
7	EX	c-ex	XYZ	qc-ex	DEF	Purple Highlighted is created by utility after execution					
8	HE	c-he	PQR	qc-he	ABC	SN	Domain	Dev Code	Dev Name	Val Code	Val Name
9	IE	c-ie	ABC	qc-ie	PQR	1	t-ae-sf.rtf	c-t-ae-sf	ABC	qc-t-ae-sf	PQR
10	OI	c-oi	DEF	qc-oi	XYZ	2	t-ae-ae.rtf	c-t-ae-ae	DEF	qc-t-ae-ae	XYZ
11	SX	c-sx	XYZ	qc-sx	DEF	3	t-aw-we.rtf	c-t-aw-we	XYZ	qc-t-aw-we	DEF
12	VS	c-vs	PQR	qc-vs	ABC	4	t-pr-we.rtf	c-t-pr-we	PQR	qc-t-pr-we	ABC
SDTM Sheet						TLF Sheet					
Yellow Highlighted section is entered by Study Lead						Yellow Highlighted is entered by Study Lead					
						Purple Highlighted is created by utility after execution					

Table 4: ProjectAllocationPlan.xls Snapshot

PhUSE 2013

6.3.1 IMPACT OF AUTOMATION

- ❖ Automatically creates list of SDTM domains as it is directly linked to annotated CRF
- ❖ Least manual interference is needed as most of the columns in allocation plan sheets are created automatically
- ❖ Facilitates in the creation of Source & QC codes of the projects
- ❖ Reduces efforts in project management

6.4 CREATE BLANK SAS PROGRAMS

The utility creates blank standard header with name of the code, path of the code, description, author and code creation date from the project allocation plan.

```

/*-----
PROGRAM          : c-ae.sas
PATH             : Z:\ClientX\ProductX\Analysis\Production\SDTM\
SAS VERSION      : 9.2
CLIENT           : ClientX
DESCRIPTION       : To create AE SDTM dataset
AUTHOR           : ABC
DATE CREATED      : 27AUG2013:15:11:05
PROGRAM ALGORITHM :
-----*/;

%getdir;

```

Fig 4: Blank SAS® code header

6.4.1 IMPACT OF AUTOMATION

- ❖ GPP(Good Programming Practice) for program header is checked
- ❖ Program headers can be customized as per project requirement
- ❖ Programs are incorporated automatically with global directory setup macro calls (%getdir in Fig 4)

6.5 CREATE VALIDATION FINDING DOCUMENTS

The utility also creates a blank validation finding & issue log document (Fig 5) when run for the first time. Validation finding documents are also stored as SAS® datasets for further automation.

A	B	C	D	E	F	G	H	I
Issue No	Validator	Dataset Name/ TLF Number/Artifact Id	Issue	Date	Programmer	Programmers Response	Date	Status (Resolved/ Pending)

Issue No	Dataset Name/ TLF Number/Artifact Id	Issue Category	Description of Issue	Date	Raised by	Response	Date	Response by	Status (Resolved/ Pending)

Fig 5: Validation Finding & Issues log

Once any validation finding is entered in the document and utility is run then it sends an email to the source programmer automatically from project allocation plan regarding the addition of new issue. And after the issue is resolved and utility is run, it sends an email is sent back to the validator confirming the resolution of the issue.

6.5.1 IMPACT OF AUTOMATION

- ❖ Reduces the effort in project documentation
- ❖ Helps to raise issue in a timely manner
- ❖ Tracks issues and creates a issue repository in a SAS® dataset

PhUSE 2013

6.6 CREATE BATCH FILE

The utility creates a runall.bat script for batch submission of the code for each of the sheets present in the project allocation plan. (Fig 6)

```
start/w C:\PROGRA~1\SAS\SASFOU~1\9.2(32-bit)\SAS.EXE -icon -nologo -xmin -sysin C-AE.sas
start/w C:\PROGRA~1\SAS\SASFOU~1\9.2(32-bit)\SAS.EXE -icon -nologo -xmin -sysin C-CM.sas
start/w C:\PROGRA~1\SAS\SASFOU~1\9.2(32-bit)\SAS.EXE -icon -nologo -xmin -sysin C-CO.sas
start/w C:\PROGRA~1\SAS\SASFOU~1\9.2(32-bit)\SAS.EXE -icon -nologo -xmin -sysin C-DF.sas
start/w C:\PROGRA~1\SAS\SASFOU~1\9.2(32-bit)\SAS.EXE -icon -nologo -xmin -sysin C-DM.sas
start/w C:\PROGRA~1\SAS\SASFOU~1\9.2(32-bit)\SAS.EXE -icon -nologo -xmin -sysin C-DS.sas
start/w C:\PROGRA~1\SAS\SASFOU~1\9.2(32-bit)\SAS.EXE -icon -nologo -xmin -sysin C-EX.sas
start/w C:\PROGRA~1\SAS\SASFOU~1\9.2(32-bit)\SAS.EXE -icon -nologo -xmin -sysin C-HE.sas
start/w C:\PROGRA~1\SAS\SASFOU~1\9.2(32-bit)\SAS.EXE -icon -nologo -xmin -sysin C-IE.sas
start/w C:\PROGRA~1\SAS\SASFOU~1\9.2(32-bit)\SAS.EXE -icon -nologo -xmin -sysin C-OI.sas
start/w C:\PROGRA~1\SAS\SASFOU~1\9.2(32-bit)\SAS.EXE -icon -nologo -xmin -sysin C-SX.sas
start/w C:\PROGRA~1\SAS\SASFOU~1\9.2(32-bit)\SAS.EXE -icon -nologo -xmin -sysin C-VS.sas
```

Fig 6: Batch file for SDTM files

6.6.1 IMPACT OF AUTOMATION

- ❖ Reduces programming effort of creating batch file manually
- ❖ Utility contains a macro parameter (with default value set as 'No') if changed to 'Yes', it runs the batch script automatically during the execution and fetches the latest project information
- ❖ Reduces project monitoring effort

6.7 CAPTURE LAST MODIFIED TIME STAMP FOR SAS® CODES /DATASETS & LOG SCANNING

The utility captures the last modified system date and time for all the codes & datasets and places them in the final project tracking sheet. It also checks whether the log is error and warning free (Table 5)

Domain/ Dataset/ TLF	Source Code	Source Log	Source Output Dataset	Source Code Last Update Time(EST)	Source Initials
ADSL	c-adsl.sas	c-adsl.log	c-adsl.sas7bdat	29AUG2013:12:27:08	ABC
ADAE	c-adae.sas	c-adae.log	c-adae.sas7bdat	29AUG2013:12:13:09	DEF
ASLBASE	c-aslbase.sas	c-aslbase.log	c-aslbase.sas7bdat	30AUG2013:10:12:10	XYZ
ASLEFF	c-asleff.sas	c-asleff.log	c-asleff.sas7bdat	31AUG2013:11:09:10	PQR

Table 5: Sample ADaM sheet in *ProjectTrackit.xml*

6.7.1 IMPACT OF AUTOMATION

- ❖ Removes manual entry of code completion date
- ❖ Gives the most updated project status at any given time
- ❖ Unclean logs can be easily identified by the color codes(Yellow contains warning and Red contains error)
- ❖ Codes/dataset/logs/outputs are hyperlinked and can be directly open from this sheet

6.8 DISPLAY PROC COMPARE RESULT

The utility fetches information about QC status from a SAS® dataset (QCStatus.sas7bdat) that is created by using %QCCheck utility². This dataset contains PROC COMPARE result, timestamp of execution and source dataset Pass/ Fail status. (Fig 7)

Dataset/Table Name	Pass/Fail	Time Run	Compare Result
qc-mh	Pass	31MAY2012:15:33	1. The two data sets compared are exactly equal
qc-pc	Fail	29AUG2012:00:10	1. Data set labels differ 2. Comparison data set has variable not in base
qc-pc-toc	Fail	27AUG2012:18:09	1. Comparison data set has variable not in base 2. A value comparison was unequal
qc-pc_vd	Fail	29AUG2012:00:05	1. Data set labels differ 2. Comparison data set has variable not in base
qc-qs	Pass	31MAY2012:15:34	1. The two data sets compared are exactly equal
qc-relrec	Fail	31MAY2012:15:34	1. Base data set has observation not in comparison 2. A value comparison was unequal

Fig 7: Sample QCStatus.sas7bdat

PhUSE 2013

For TLF sheet the code outputs can be directly open from the TLF column in the sheet. (Table 6)

Domain/ Dataset/ TLF	QC Log	QC Output	QC Code Last Update Time(EST)	QC Initial	Validation Finding	Retest (Pass/QC Re-run)	Proc Compare Pass/Fail
t-ae-saf.rtf	qc-t-ae-saf.log	qc_t_ae_saf.sas7bdat	1SEP2013:11:27:08	PQR	Pending	Pass	Pass
t-ae-aes.rtf	qc-t-ae-aes.log	qc_t_ae_aes.sas7bdat	28AUG2013:22:10:29	XYZ	No Finding	Pass	Pass
t-aw-wer.rtf	qc-t-aw-wer.log	qc_t_aw_wer.sas7bdat	29AUG2013:05:02:13	DEF	Pending	QC Re-run	Fail
t-pr-wer.rtf	qc-t-pr-wer.log	qc_t_pr_wer.sas7bdat	29AUG2013:11:55:34	ABC	No Finding	Pass	Pass

Table 6: Sample TLF sheet in *ProjectTrackit.xml*

6.8.1 IMPACT OF AUTOMATION

- ❖ Gives QC status at any given time
- ❖ Fetches information about validation finding results
- ❖ Retest column checks whether QC code is batch submitted after source
- ❖ PROC COMPARE output (.lst) can be directly accessed from project tracking sheet

7. FUTURE ENHANCEMENTS

There are many possibilities of attaching other “plug-in” utilities to this utility because project tracking sheet is a central location of storing vital project related information. The following are some on the plug-in utilities

- ❖ To check the quality of the SAS[®] code (whether code is written as per GPP).
- ❖ Creation of SAS[®] Dashboard from the *ProjectTrackit* sheet.
- ❖ To track the actual efforts spent on the code (Effort tracking tool).
- ❖ To create Define.xml for regulatory submissions.

8. CONCLUSIONS

Automation of various project management activities can greatly enhance a team's ability to timely and effectively submit the deliverables. This utility aims to achieve this goal and creates a tracking document at a central place where a manager can automate some routine tasks effectively and accurately track the project status.

9. REFERENCES

1. Creating That Perfect Data Grid Using the SAS[®] Output Delivery System by Chevell Parker, SAS Institute Inc., Cary, NC: <http://www2.sas.com/proceedings/forum2008/258-2008.pdf>
2. A Few Quick and Efficient Ways to Compare Data by Abraham Pulavarti, ICON Clinical Research, North Wales, PA: <http://www.nesug.org/Proceedings/nesug10/po/po27.pdf>

ACKNOWLEDGMENTS

I would like to thank Cytel colleagues for providing their valuable input for this paper and the utility.

CONTACT INFORMATION

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

Author Name : Abhishek Bakshi
 Company : Cytel Statistical Software & Services Pvt. Ltd
 Address : Pune, India
 Email : Abhishek.Bakshi@cytel.com
 Web : <http://www.cytel.com>

Brand and product names are trademarks of their respective companies.