

SCAPROC Impact Analysis - Providing QA and Compliance Insights

Sandeep Juneja, SAS Institute, Cary NC

Ben Bocchicchio, SAS Institute, Cary NC

Stijn Rogiers, SAS Institute, Belgium

ABSTRACT

One of the biggest challenges in submissions is to make sure that all the referenced inputs and outputs in SAS programs for data sets and TLFs are all consistent and none of them were refreshed or updated at a later date than the outputs generated by using them as input. To update a specific SAS program or data set, it is critical to understand what impact it will have downstream on the programs or data sets that use it as input.

SCAPROC is a new procedure that runs a *SAS Code Analyzer*. This SAS Code Analyzer captures metadata information about the contents of the SAS code that is run. It also collects information on files used in the code's input and output, as well as macro variables used while the code is running.

How does this procedure relate to quality assurance (QA) and compliance? Just imagine if you could collect all the metadata about all the SAS code that is run to generate output for a FDA submission. You could programmatically prove that:

- all the data referenced for input is used consistently
- all macros called in the programs are consistent (without worrying about calling a generic macro when a project-specific macro is required)
- all designated output is saved to the correct location

Would this make you feel more confident about your submission?

This presentation reveals these uses of SCAPROC and introduces new concepts such as Automated Impact Analysis using workflows in the SAS® Life Science Analytics Framework (LSAF) application and converting captured metadata to more machine-readable format (RDF) for graphical impact analysis.

INTRODUCTION

The SCAPROC procedure uses the SAS Code Analyzer, which captures information about input, output, and the use of macro symbols from a SAS program while it is running. The SAS Code Analyzer can write this information and the information in the original SAS file to a file that you specify. To create consistent output when developing SAS code for clinical trial analysis, it is paramount to use the correct data sets as input. The output can be either the submission data sets or the supporting figures, tables, and listings. SAS programmers have developed various processes to ensure that the correct data is chosen; however, issues can still arise. The PROC SCAPROC procedure is another tool to help verify that SAS programs are producing consistent results. The output can be used to perform impact analyses. The PROC SCAPROC procedure first appeared in Release 9.2 of Base SAS® Software.

PROC SCAPROC Setup

To produce the PROC SCAPROC output, wrap the **scaproc** statements around the executing SAS code as shown in Figure 1. **Figure 1: PROC SCAPROC Usage**

```
* Start SCAPROC;
proc scaproc; write; run;
proc scaproc; record "C:/AnalyzeIO/scafiles/sca.txt"
EXPANDMACROS; run;

* SAS PROGRAM CODE TO BE ANALYZED;
libname datalib "C:/AnalyzeIO";
data datalib.class;
    set sashelp.class;
run;

* End SCAPROC;
```

Figure 1: PROC SCAPROC Usage

```
* Start SCAPROC;
proc scaproc; write; run;
proc scaproc; record "C:/AnalyzeIO/scafiles/sca.txt"
EXPANDMACROS; run;

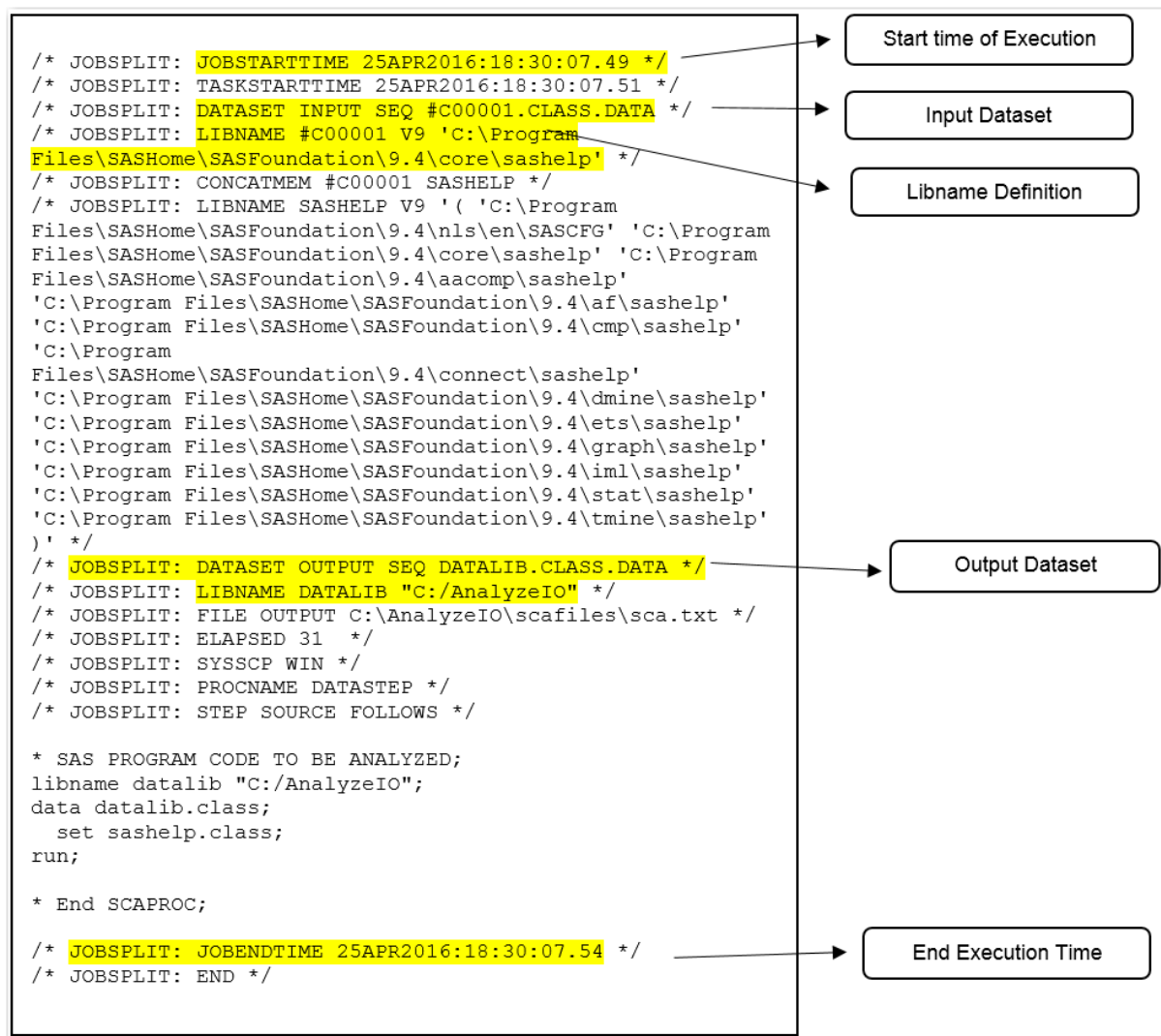
* SAS PROGRAM CODE TO BE ANALYZED;
libname datalib "C:/AnalyzeIO";
data datalib.class;
    set sashelp.class;
run;

* End SCAPROC;
```

PROC SCAPROC RESULTS

The results from running the PROC SCAPROC on a simple PROC PRINT SAS program generates the following information captured in Figure 2.

Figure 2: PROC SCAPROC Results



IMPACT ANALYSIS

AnalyzeIO.sas is a custom SAS macro that parses the **scaproc** file and creates a data set with all dependencies, including the type of use (input or output) with the file's path and name along with its last modified date. This is important for impact analysis. Examination of the results may reveal that, during the program execution, an input source data set could have been incorrectly selected or the output destination of the results incorrectly stored.

The parsing program is where the magic happens. It analyzes the SCAPROC output and generates a data set that contains details on the inputs to the code, macros used, format catalog used, and the outputs generated from the code. The PC SAS Macro **AnalyzeIO.sas** can be downloaded from the SAS Support site link <http://support.sas.com/kb/58878>.

Figure 3: PC Impact Analysis Output

	sas_task	order_id	type	obj_type	file_path	timeinfo	createdDate	modifiedDate	fileSize
1	C:\AnalyzeIO\Study\Programs\ae.sas	1	JOBSTARTTIME	JOBSTARTTIME	C:\AnalyzeIO\Study\Programs\ae.sas	16AUG2016 08:04:23.60	25Apr2016:18:25:28	15Aug2016:20:14:50	756 Bytes
2	C:\AnalyzeIO\Study\Programs\ae.sas	2	JOBENDTIME	JOBENDTIME	C:\AnalyzeIO\Study\Programs\ae.sas	16AUG2016 08:04:23.77	25Apr2016:18:25:28	15Aug2016:20:14:50	756 Bytes
3	C:\AnalyzeIO\Study\Programs\ae.sas	3	INPUT	FILE	C:\AnalyzeIO\Study\Data\ae.sas7bdat		25Apr2016:14:43:50	16Aug2016:08:04:23	128 KB
4	C:\AnalyzeIO\Study\Programs\ae.sas	3	INPUT	FILE	C:\AnalyzeIO\Study\Data\formats.sas7bcat		25Apr2016:14:40:45	25Apr2016:14:42:53	17 KB
5	C:\AnalyzeIO\Study\Programs\ae.sas	3	INPUT	FILE	C:\AnalyzeIO\Study\macros\setup.sas		25Apr2016:14:40:45	25Apr2016:14:49:19	349 Bytes
6	C:\AnalyzeIO\Study\Programs\ae.sas	3	INPUT	FILE	C:\Program Files\SASHome\SASFoundation\9.4\core\sashelp\class.sas7bdat		03Dec2015:21:07:24	25Jun2015:01:56:11	128 KB
7	C:\AnalyzeIO\Study\Programs\ae.sas	4	OUTPUT	FILE	C:\AnalyzeIO\Study\Data\ae.sas7bdat		25Apr2016:14:43:50	16Aug2016:08:04:23	128 KB

This process when executed within the SAS® Life Science Analytics Framework will capture more metadata for each INPUT and OUTPUT file definition (Figure 4).

Figure 4: SAS® Life Science Analytics Framework Impact Analysis Output

Obs	sas_task	ord...	type	obj_type	file_path	timeinfo	createdBy	createdDate	modifiedBy	modifiedDate	file Size
1	/SAS/Files/StudyPrograms/api_test.sas	1	JOBSTARTTIME	JOBSTARTTIME	/SAS/Files/StudyPrograms/api_test.sas	16AUG2016:12:07:04.14	sajune	Sat Aug 13 22:4	sajune	Sat Aug 13 22:4	1 KB
2	/SAS/Files/StudyPrograms/api_test.sas	2	JOBENDTIME	JOBENDTIME	/SAS/Files/StudyPrograms/api_test.sas	16AUG2016:12:07:04.73	sajune	Sat Aug 13 22:4	sajune	Sat Aug 13 22:4	1 KB
3	/SAS/Files/StudyPrograms/api_test.sas	3	INPUT	FILE	/SAS/Files/StudyPrograms/api_test.sas		sajune	Sat Aug 13 22:4	sajune	Sat Aug 13 22:4	1 KB
4	/SAS/Files/StudyPrograms/api_test.sas	3	INPUT	FILE	/SAS/Files/StudyData/ae2.sas7bdat		sajune	Sat Aug 13 22:4	sajune	Sat Aug 13 22:4	2 MB
5	/SAS/Files/StudyPrograms/api_test.sas	3	INPUT	FILE	/SAS/Files/StudyDocuments/CM_specs1.xlsxbak		sajune	Sat Aug 13 22:4	sajune	Sat Aug 13 22:4	4 KB
6	/SAS/Files/StudyPrograms/api_test.sas	3	INPUT	FILE	/SAS/Files/StudyDocuments/CM_specs.xlsx		sajune	Sat Aug 13 22:4	sajune	Sat Aug 13 22:4	8 KB
7	/SAS/Files/StudyPrograms/api_test.sas	4	OUTPUT	FILE	/SAS/Files/StudyData/ae4.sas7bdat		sajune	Sat Aug 13 22:4	sajune	Sat Aug 13 22:4	2 MB
8	/SAS/Files/StudyPrograms/api_test.sas	4	OUTPUT	FILE	/SAS/Files/StudyDocuments/test.rtf		sajune	Sat Aug 13 22:4	sajune	Sat Aug 13 22:4	25 KB
9	/SAS/Files/StudyPrograms/api_test.sas	4	OUTPUT	FILE	/SAS/Files/StudyDocuments/CM_specs1.xlsx		sajune	Sat Aug 13 22:4	sajune	Sat Aug 13 22:4	4 KB

Note: SAS Itemstores, SAS Indexes, SAS database views (Oracle), web streams (PROC HTTPS, PROC SOAP), and inputs pulled from Java JAR files are not captured in PROC SCAPROC output.

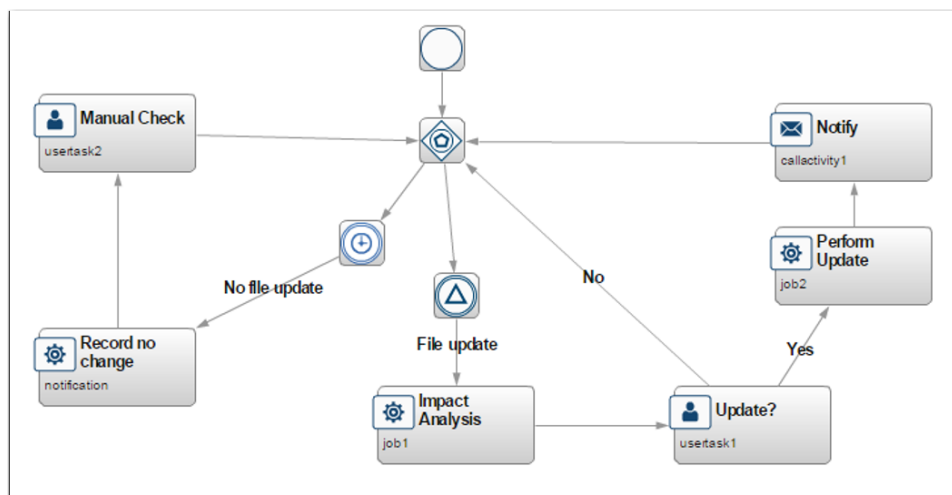
IMPACT ANALYSIS WORKFLOW

The SAS® Life Science Analytics Framework provides capabilities to execute workflows developed using Business Process Model and Notation (BPMN) standards. It enables you to develop a workflow process allowing teams to define automated processes, triggered by events.

For example, when any file is updated, a process can be triggered to automatically assess the impact on files that use the updated file as input and notify users of the update.

Furthermore, the programs that use the updated file as input can be executed automatically. This ensures that users are reviewing outputs generated based on refreshed data and not on an outdated copy of the data.

Figure 5: SAS® Life Science Analytics Framework Impact Analysis Workflow



GRAPHICAL IMPACT ANALYSIS

Ontologies are already available to represent statistical data in Resource Description Framework (RDF) format. However, rather than representing whole SAS data sets in RDF format, you can use open-source tools such as Protégé to develop required ontology to represent interested SAS data set values in RDF format and represent data set information as shown in Figure 6.

Figure 6: SAS Data Set RDF Turtle Representation

```

@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>.
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>.
@prefix xsd: <http://www.w3.org/2001/XMLSchema#>.

# definition
<aio:#input> rdfs:range <aio:#ae.sas>.
<aio:#sasprg> rdfs:domain <aio:#ae.sas>.
<aio:#ae.sas> rdfs:domain <aio:#output>.

# Program
<aio:#ae.sas> <aio:#Location> "C:/Study/Programs"^^<xs:string>;
    <aio:#createdby> "sajune"^^<xs:string>;
    <aio:#creationdate> "21JAN2016:21:10:51"^^<xs:dateTime>;
    <aio:#lastmodifiedby> "sajune"^^<xs:string>.

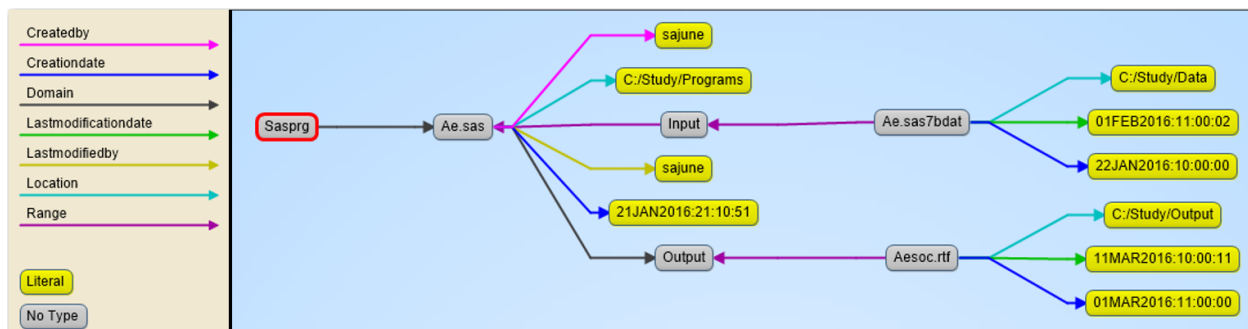
# Input
<aio:#ae.sas7bdatt> <aio:#Location> "C:/Study/Data"^^<xs:string>;
    <aio:#creationdate> "22JAN2016:10:00:00"^^<xs:dateTimeStamp>;
    <aio:#lastmodificationdate> "01FEB2016:11:00:02"^^<xs:dateTimeStamp>;
    rdfs:range <aio:#input>.

# Output
<aio:#aesoc.rtf> <aio:#Location> "C:/Study/Output"^^<xs:string>;
    <aio:#creationdate> "01MAR2016:11:00:00"^^<xs:dateTime>;
    <aio:#lastmodificationdate> "11MAR2016:10:00:11"^^<xs:dateTime>;
    rdfs:range <aio:#output>.

```

Open-source tools such as Gruff or RDF Gravity can provide a combined graphical view of the inputs and outputs for files in studies (Figure 7).

Figure 7: Graphical RDF N3 Representation



Several different impact analyses can be performed on the resulting data set. By sub-setting the information in the *type* variable (input), and creating new variables from the *file_path* variable, you can easily produce a frequency of all data sets used in the protocol. The question of whether the correct data sets have been chosen can easily be answered. The same is true for the output files generated; you can answer the question of whether all the output has been saved to the same, correct location.

You can perform other impact analyses to determine which programs must be re-run because of changes to individual data sets (due to an un-freeze of the data) or the updating of macros used in the code.

LIMITATIONS AND CONSIDERATIONS

PROC SCAPROC has some limitations in the information it can glean from *code analyzer* output. The procedure does not pull information about the following items that maybe used in SAS code:

- SAS Itemstores
- SAS Indexes
- SAS database views (for example an oracle view)
- Web streams (from PROC HTTPS or PROC SOAP)
- Inputs pulled from Java JAR files

To ensure all the output is collected correctly when using this procedure, consider running the SAS code in its own session. This is because any SAS macros used in your SAS code are copied into the Sasmacr catalog in your work library. If you use the same SAS session to run a program that uses the same macro, SAS pulls it from the work catalog and writes this information to the PROC SCAPROC output. The parsing code developed and tested for this paper drops all information from the work area, causing knowledge of using this macro to be lost. Therefore, it is advisable to run this procedure in Batch Submit.

CONCLUSION

In conclusion, the PROC SCAPROC is a very powerful procedure to conduct a real-time impact analysis that is easy to implement. Armed with this information, a wide array of analyses can be performed to ensure the correct data is used in the SAS code and the results are all stored in the correct location. Other insights can also be gleaned from this information, such as macro and format catalog usage.

REFERENCES

Thies, Eric and Langston, Rick. 2008. *Introducing the SAS® Code Analyzer. SAS® Global Forum 2008 Conference*. Cary, NC: SAS Institute Inc.

CONTACT INFORMATION

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

Name: Sandeep Juneja
Enterprise: SAS Institute
Address: SAS Campus Drive
City, State ZIP: Cary, NC 27513
Work Phone: (919) 531 0541
E-mail: Sandeep.Juneja@sas.com

Name: Ben Bocchicchio
Enterprise: SAS Institute
Address: SAS Campus Drive
City, State ZIP: Cary, NC 27513
Work Phone: (919) 531 3704
E-mail: ben.bocchicchio@sas.com

Name: Stijn Rogiers
Enterprise: SAS Institute
Address: Hertenbergstraat 6,
City, State ZIP: Tervuren, Belgium
Work Phone: +32 276 608 23
E-mail: Stijn.Rogiers@sas.com

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.