How Did This Get Made?

Matt Becker, SAS Institute, Cary, North Carolina

ABSTRACT

What data was used for my result? What macros or SAS® code did I pull in? What was created? Our time is sometimes spent reviewing code for <libname>.<dataset> references, %include statements and other key code to document or describe how a result was created. Within our metaphorical bowl, we are defining the ingredients that went in to create our happy meal.

With the SAS SCAPROC and some text mining, we can easily automate the documentation of our "bowl." It implements the SAS code analyzer which captures information about the input, output and use of macro symbols from a SAS job. By writing out the resulting information, we can use this output for documentation AND/OR parse the information to cleanly list the key ingredients we are interested in.

INTRODUCTION

In my 20+ years of working in the CRO industry, I wrote thousands of programs creating data sets, tables, listings and figures. Each of those programs could potentially read in data sets, macros, input files and then create some form of output. During QC and review, there were always questions such as:

1)   What input data set(s) did you use?

2)   What macros were called?

3)   Did you use a format catalog?

4)   What was created (data, output file, etc.)?

When these questions were posed, I had to scan my programs and macros for <libname>.<dataset>, <dataset>, %include statements, %<macro> and many other potential string searches.

What if there was an easier way? What if there was a way to parse my code and provide that information to me in a nice manner? With the SAS SCAPROC and some text mining, the future is now! We can now get a nice list of input files/data, %include files, created dataset(s) and more for every execution of a program.

INVOCATION OF SCAPROC

To invoke the SAS code analyzer procedure, the simplest way is to add the following code to the beginning and end of your program or code you would like analyzed:

PROC SCAPROC;

  RECORD 'record.txt';

RUN;

…… <program code> …….

PROC SCAPROC;

  WRITE;

RUN;

The code above first turns on SCAPROC and specifies where to write the SCAPROC results. In this case, it will write to the file record.txt. After the code you want SCAPROC to execute on, you add the last PROC SCAPROC call with the statement 'WRITE;'. This tells SCAPROC that you are done and the code between the initial SCAPROC and the "WRITE" SCAPROC is analyzed and the details written to the file record.txt.

Figures 1 and 2 shows the SCAPROC syntax and arguments.

Syntax

PROC SCAPROC;

      RECORD *filespec* <ATTR> <OPENTIMES> <INTCON> <EXPANDMACROS>
      <GRID *filespec* <RESOURCE *"resource name"*> <INHERITLIB> <NOOPTIMIZE> >;
      WRITE;

Table of Procedure Tasks and Examples

Statement	Task
PROC SCAPROC	Implement the SAS Code Analyzer
RECORD	Specify a filename or a fileref to contain the output of the SAS Code Analyzer
WRITE	Output information to the record file

Figure 1. SCAPROC Syntax

Required Argument

filespec
    specifies a physical filename in quotation marks, or a fileref, that indicates a file to contain the output of the SAS Code Analyzer. The output is the original SAS source and comments that contain information about the job. For more information about the output comments, see Results.

Optional Arguments

ATTR
    outputs additional information about the variables in the input data sets and views.

OPENTIMES
    outputs the open time, size, and physical filename of the input data sets.

INTCON
    outputs information about table integrity constraints.

EXPANDMACROS
    expands macro invocations into separate tasks.

GRID
    filespec
        specifies a physical filename in quotation marks, or a fileref, that points to a file that will contain the output of the Grid Job Generator.

    RESOURCE *"resource name"*
        specifies the resource to use in the `grdsvs_enable` function call. The default is SASApp.

    INHERITLIB
        adds INHERITLIB=(USER) to the SIGNON statements in the grid-enabled job. By default, the user library (if one is specified) is assigned in each remote session.

        Note: INHERITLIB should be used only when the USER library is not globally accessible to all remote sessions.
    NOOPTIMIZE
        disables the combining and reordering of tasks for grid-enabled jobs.

Figure 2. SCAPROC Arguments

SCAPROC RESULTS

The SCAPROC results are in a format that is readable, but definitely takes some work to pull the information of interest. Figure 3 is a subset example of the output from a SCAPROC on an AE Listing Program (RLAE.SAS):

Figure 3. RECORD.TXT from RLAE.SAS Program

As represented in Figure 3, there is a lot of information to be read – much of which we are not interested in keeping. Thus, we need a text mining program to read in the RECORD.TXT file and parse it for the information we are interested in.

For example, let's assume we are interested in data set input/outputs and file input/outputs. In the next section, we will write text mining code using Perl regular expressions to pick out that information.

MINING THE SCAPROC OUTPUT

Examples of the types of JOBSPLIT: <types> that are created from SCAPROC are numerous. You can find a list of the types of results created at support.sas.com in the documentation for your specific version. Please note that SCAPROC first appeared in SAS V9.2. Below is a link to a list of the types of JOBSPLIT from v9.4:

http://support.sas.com/documentation/cdl/en/proc/68954/HTML/default/viewer.htm#p0k5uaxpaz2uzin1qvbqmmafnqtl.htm

For out example, we are interested in pulling out 4 distinct lines from the SCAPROC output:

1) JOBSPLIT: DATASET INPUT

2) JOBSPLIT: DATASET OUTPUT

3) JOBSPLIT: FILE INPUT

4) JOBSPLIT: FILE OUTPUT

We will use the Perl Expressions: PRXPARSE, PRXMATCH and PRXPOSN to mine and pull the information desired from the SCAPROC output file. A nice tip sheet for Perl Regular Expressions can be found at:

https://support.sas.com/rnd/base/datastep/perl_regexp/regexp-tip-sheet.pdf

Figure 4 contains the SAS code used to read in the 4 types of information using Perl Regular Expressions we are interested in.

```sas
filename sca_rec 'C:\biometrics\XMB111\XMB111\analyses_mjb\CSR\programs\prod\record.txt';

data myinfo;
   infile sca_rec truncover;
   input @1 scaline $256.;
   length type myinfo $100;
   keep type myinfo;
   retain prxifile prxofile prxidata prxodata;
   if _n_=1 then do;
     prxifile = prxparse("!\bJOBSPLIT: FILE INPUT (\b.*\b) \*/!");
     prxofile = prxparse("!\bJOBSPLIT: FILE OUTPUT (\b.*\b) \*/!");
     prxidata = prxparse("!\bJOBSPLIT: DATASET INPUT (\b.*\b) \*/!");
     prxodata = prxparse("!\bJOBSPLIT: DATASET OUTPUT (\b.*\b) \*/!");
   end;
   if prxmatch(prxifile,scaline)>0 then do;
     myinfo=prxposn(prxifile,1,scaline);
     type='Input File';
     output myinfo;
   end;
   else if prxmatch(prxofile,scaline)>0 then do;
     myinfo=prxposn(prxofile,1,scaline);
     type='Output File';
     output myinfo;
   end;
   else if prxmatch(prxidata,scaline)>0 then do;
     myinfo=prxposn(prxidata,1,scaline);
     type='Input Dataset';
     output myinfo;
   end;
   else if prxmatch(prxodata,scaline)>0 then do;
     myinfo=prxposn(prxodata,1,scaline);
     type='Output Dataset';
     output myinfo;
   end;
 run;
```

Figure 4. Code to Mine SCAPROC Log

The output from the above code after sorting and removing duplicates is shown in Figure 5.

Obs	type	myinfo
1	Input Dataset	MULTI WORK.AE.DATA
2	Input Dataset	SEQ DERIVED.TOC_PROG.DATA
3	Input Dataset	SEQ SDTM.AE.DATA
4	Input Dataset	SEQ SDTM.DM.DATA
5	Input Dataset	SEQ WORK.DM.DATA
6	Input Dataset	SEQ WORK.TOC_PROG.DATA
7	Input File	C:\Program Files\SASHome\ReportFontsforClients\9.4\saswcur.ttf
8	Input File	C:\Windows\Fonts\cour.ttf
9	Input File	C:\Windows\Fonts\times.ttf
10	Input File	C:\Windows\Fonts\timesbd.ttf
11	Input File	C:\biometrics\XMB111\XMB111\analyses_mjb\CSR\programs\macro\prod\mcase.sas
12	Input File	C:\biometrics\XMB111\XMB111\analyses_mjb\CSR\programs\macro\prod\mnumobs.sas
13	Input File	C:\biometrics\XMB111\XMB111\analyses_mjb\CSR\programs\macro\prod\mpageof.sas
14	Input File	C:\biometrics\XMB111\XMB111\analyses_mjb\CSR\programs\macro\prod\mtitle.sas
15	Input File	C:\biometrics\XMB111\XMB111\analyses_mjb\CSR\programs\prod\output\listings\a16_02_07_01_lae1.rtf
16	Output Dataset	SEQ WORK.AE.DATA
17	Output Dataset	SEQ WORK.DM.DATA
18	Output Dataset	SEQ WORK.TOC_PROG.DATA
19	Output File	C:\Users\matbec\AppData\Local\Temp\SAS Temporary Files_TD11888_L7C139_\sashtml.htm
20	Output File	C:\biometrics\XMB111\XMB111\analyses_mjb\CSR\programs\prod\output\listings\a16_02_07_01_lae1.rtf
21	Output File	C:\biometrics\XMB111\XMB111\analyses_mjb\CSR\programs\prod\record.txt

Figure 5. SCAPROC Mining Output

CONCLUSION

PROC SCAPROC can be used to create and/or analyze the information used and created from SAS programs. Information such as dataset(s) used, files created, data created and files used that are commonly used in life science table, figure, listing and dataset creation can easily be collected. Impact analyses such as frequency of all datasets used in the protocol can be reviewed. Questions such as "What dataset(s) were used in the ADAE creation?" or "Have all output been saved to the correct location?" can be answered.

We have touched briefly on the SCAPROC and the information that can be gleaned from the resulting log. There are many more options and results that can be both added and mined from the SCAPROC procedure output.

RECOMMENDED READING

- Bocchicchio, Ben and Juneja, Sandeep. "QA and Compliance Insights Using the SCAPROC Procedure," PharmaSUG 2016. http://www.pharmasug.org/proceedings/2016/MS/PharmaSUG-2016-MS04.pdf

- SCAPROC Procedure http://support.sas.com/documentation/cdl/en/proc/68954/HTML/default/viewer.htm#n05aazp6jtoup0n1qjee3h7jto24.htm

- Beggs, Andrew. "Proc SCAPROC – SAS Logging Unleashed." Amadeus Software Limited. http://www.amadeus.co.uk/sas-training/papers/1/2/13/proc-scaproc---sas-logging-unleashed.pdf

- SAS9 Perl Regular Expressions Tip Sheet https://support.sas.com/rnd/base/datastep/perl_regexp/regexp-tip-sheet.pdf

CONTACT INFORMATION

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

Matt Becker
SAS Institute
100 SAS Campus Drive
Cary, NC 27513
Matt.Becker@sas.com

THANK YOU

Thank you to Eric Thies, Rick Langston, Ben Bocchicchio, Sandeep Juneja and Andrew Beggs for the excellent SCAPROC presentations provided in the past.