

My Experiences in Adopting SAS® Cloud Analytic Services into Base SAS® Processes

Steven Sober, SAS Institute Inc., Cary, NC

ABSTRACT

The SAS® Platform is greatly enhanced by SAS® Cloud Analytic Services (CAS) and SAS® Viya®. As a Base SAS® programmer, I want to share my experiences in adopting CAS into Base SAS processes that prepare data for modeling, reporting, and visualizations that are enabled for CAS.

INTRODUCTION

This paper is based on SAS® Viya® 3.3 and SAS® 9.4 (TS1M5), which has enabled more Base SAS® programming components to leverage CAS. For the last 2 years, I have been helping SAS customers convert Base SAS code to adopt CAS. To assist in that task, I have worked with SAS Professional Services & Delivery (PSD) to create a utility that can produce reports based on SAS jobs. This paper reviews the findings from running this utility against multiple SAS jobs found in the CORE sample directory of a SAS 9.4 (TS1M5) environment (C:\... \SASHome\SASFoundation\9.4\core\sample).

Note that SAS PSD and Pre-Sales offers its customers the findings from the utility as a service. Contact Steven.Sober@sas.com if you are interested in this service.

BASE SAS PROCEDURES THAT USE CAS ACTIONS

From the [Base SAS 9.4 Procedures Guide](#), we learn that the procedures in Table 1 are enabled for CAS. For procedures that process in CAS, the golden rule is that all source and target tables for the procedure must be stored in a [caslib](#). For the Base SAS procedures MEANS, REPORT, SUMMARY, and TABULATE, the functionality mimics the functionality provided by the Base SAS in-database capabilities that execute in Hadoop or a supported relational database management system such as Oracle, Teradata, or DB2.

Column	Description
PROC APPEND	Adds rows from a CAS table to the end of a SAS data set, and adds rows from a SAS data set to the end of a CAS table.
PROC CONTENTS	Shows the contents of a CAS table and prints the directory of the caslib.
PROC COPY	Copies entire SAS libraries or specific members of the library.
PROC DATASETS	Manages CAS tables.
PROC DELETE	Deletes SAS data sets and CAS tables.
PROC DS2	Manipulates data with DS2 language statements.
PROC DSTODS2	Enables you to submit DS2 language statements from a Base SAS session.

Column	Description
<u>PROC FCMP</u>	Enables you to create, test, and store SAS functions, CALL routines, and subroutines before you use them in other SAS procedures or in DATA steps.
<u>PROC FEDSQL</u>	Manipulates data and performs reporting with FedSQL language statements.
<u>PROC FORMAT</u>	Creates user-defined informats to read data and user-defined formats to display data.
<u>PROC LUA</u>	Enables you to run statements from the Lua programming language within SAS code.
<u>PROC MEANS</u>	Computes descriptive statistics; can produce printed output and output data sets. By default, PROC MEANS produces printed output.
<u>PROC REPORT</u>	Combines features of the PRINT, MEANS, and TABULATE procedures with features of the DATA step in a single report-writing tool that can produce a variety of reports.
<u>PROC SCOREACCEL</u>	Provides an interface to the CAS server for DATA step and DS2 model publishing and scoring.
<u>PROC SUMMARY</u>	Computes descriptive statistics; can produce a printed report and create an output data set. By default, PROC SUMMARY creates an output data set.
<u>PROC TABULATE</u>	Displays descriptive statistics in tabular format, using some or all of the variables in a data set.
<u>PROC TRANSPOSE</u>	Transforms SAS data sets so that observations become variables and variables become observations.

Column	Description
<u>PROC FORMAT</u>	Creates user-defined informats to read data and user-defined formats to display data.
<u>PROC LUA</u>	Enables you to run statements from the Lua programming language within SAS code.
<u>PROC MEANS</u>	Computes descriptive statistics; can produce printed output and output data sets. By default, PROC MEANS produces printed output.
<u>PROC REPORT</u>	Combines features of the PRINT, MEANS, and TABULATE procedures with features of the DATA step in a single report-writing tool that can produce a variety of reports.
<u>PROC SCOREACCEL</u>	Provides an interface to the CAS server for DATA step and DS2 model publishing and scoring.

Table 1. Base SAS Procedures That Are CAS Enabled

THE DATA STEP IS CAS ENABLED

From [SAS Cloud Analytic Services 3.3: DATA Step Programming](#), we learn that big data running in a single thread is slow. The DATA step runs faster in CAS because it can run in multiple threads in parallel. In CAS, hundreds of threads are available across several machines. When you program your DATA step to run in CAS, all of those threads become available for you to use to improve the performance with massively parallel processing. Parallel processing speeds the execution of big data by starting the DATA step on multiple machines and dividing processing workload among threads on these machines.

SAS VIYA READINESS UTILITY

To gain an understanding of converting Base SAS processes to leverage CAS, we run the CORE sample jobs through the SAS Viya Readiness Utility, which generates multiple reports that we use to produce our findings. The reports are based on the syntax of the SAS code in one or more SAS jobs. For DATA Step code, the utility scores each word of the syntax based on whether that word is CAS enabled or not. For SAS procedures and DATA step, we generate a frequency report that gives us a high-level understanding of the steps used to accomplish the task.

Let's take a closer look at the reports generated by the utility.

FREQUENCY REPORT

This report, shown in Output 1, provides a count of all DATA steps and procedures included in programs processed by the utility. It indicates whether DATA steps and procedures are CAS-enabled candidates for conversion into a CAS action set, or programs that must run in a workspace server because they are not CAS enabled.

**A Service SAS Offers
SAS Viya Code Readiness
309 SAS Programs**

1

Step	Frequency	Percent	Cumulative Frequency	Cumulative Percent
CAS Enabled - PROC FORMAT	68	2.59	68	2.59
CAS Enabled - PROC MEANS	38	1.45	106	4.03
CAS Enabled - PROC REPORT	14	0.53	120	4.56
CAS Enabled - PROC SUMMARY	21	0.80	141	5.36
CAS Enabled - PROC TABULATE	50	1.90	191	7.27
CAS Enabled - PROC TRANSPOSE	20	0.76	211	8.03
Candidate for CAS - DATA STEP	667	25.37	878	33.40
Candidate for CAS - PROC ANOVA	2	0.08	880	33.47
Candidate for CAS - PROC CORR	21	0.80	901	34.27
Candidate for CAS - PROC GAM	1	0.04	902	34.31
Candidate for CAS - PROC GLM	22	0.84	924	35.15
Candidate for CAS - PROC PRINCOMP	1	0.04	925	35.18
Candidate for CAS - PROC REG	7	0.27	932	35.45
Candidate for CAS - PROC STANDARD	5	0.19	937	35.64
Candidate for CAS - PROC UNIVARIATE	67	2.55	1004	38.19

Output 1. Partial Frequency Report

PROGRAM SUMMARY REPORT

This report, shown in Output 2, lists each program processed by the utility and contains the following information:

Column	Description
Program Name	Name of the program processed by the utility.
LOC	Lines of code within the program processed. Maps to the Line column of the detail report.
Steps	Number of program steps. A DATA step, procedure, or a LIBNAME statement would be considered a step.
Viya	Number of keywords that are not supported in SAS Viya.
CAS	Number of keywords that are not supported in CAS.

Column	Description
Macro	Number of macro statements. Any keyword that begins with a % is counted, except for macro statements that can be used outside a DATA step or procedure. An example is the %PUT statement. This information is included because unresolved macro code could not be assessed by the utility and needs to be manually evaluated by the reviewer.
Include	Number of %INCLUDE statements. Any %INCLUDE statement is counted. This information is included because source code referenced by the %INCLUDE statement could not be assessed by the utility and needs to be manually evaluated by the reviewer.

Table 2. Program Summary Report Columns and Descriptions

*A Service SAS Offers
SAS Viya Code Readiness
309 SAS Programs*

Program Name	LOC	Steps	Viya	CAS	Macro	Include
append.sas	29	9	0	10	0	0
armsamp.sas	88	21	0	0	38	0
biorythm.sas	52	6	0	4	0	0
bpg01r01.sas	146	28	5	38	16	0
bpg02r01.sas	110	19	0	22	0	0
bpg05r01.sas	51	15	0	17	0	0
bpg07r01.sas	277	59	0	62	0	0
bpg09r01.sas	84	19	0	11	0	0
bpg11r01.sas	59	16	0	17	0	0
bpg15r01.sas	35	9	5	16	0	0

Output 2. Partial Program Summary Report

STEP SUMMARY REPORT

This report, shown in Output 3, lists each program processed by the utility and provides detail about the steps identified. This report contains the following information:

Description	Description
Program Name	Name of the program processed by the utility. Appears at the top of each page.
Line Number / Step	Line number each step begins on and the step type. Line numbers correspond to source code found in the \detail directory and not the original code processed. Maps to the Line column of the detail report.
Viya	Number of keywords that are not supported in SAS Viya.
CAS	Number of keywords that are not supported in CAS.
Macro	Number of macro statements. Any keyword that begins with a % is counted, except for macro statements that can be used outside a DATA step or procedure. An example is the %PUT statement. This information is included because unresolved macro code could not be assessed by the utility and needs to be manually evaluated by the reviewer.
Include	Number of %INCLUDE statements. Any %INCLUDE statement is counted. This information is included because source code referenced by the %INCLUDE statement could not be assessed by the utility and needs to be manually evaluated by the reviewer.

Table 3. Step Summary Report Columns and Descriptions

**A Service SAS Offers
SAS Viya Code Readiness
309 SAS Programs**

1

append.sas

Line Number / Step	Viya	CAS	Macro	Include
1 DATA STEP	0	2	0	0
7 PROC APPEND	0	0	0	0
8 PROC PRINT	0	1	0	0
10 DATA STEP	0	2	0	0
16 PROC APPEND	0	0	0	0
17 PROC PRINT	0	1	0	0
20 DATA STEP	0	2	0	0
25 PROC APPEND	0	0	0	0
26 PROC PRINT	0	2	0	0

Output 3. Partial Step Summary Report

CATEGORY SUMMARY REPORT

This report, shown in Output 4, lists each program processed by the utility and provides detail about the categories and keywords identified. This report contains the following information:

Column	Description
Program Name	Name of the program processed by the utility. Appears at the top of each page.
Category	The type of keyword identified.
Keyword	The specific keyword not supported in SAS Viya or CAS. These also include macro statements and %INCLUDE statements not supported by the utility.
Viya	Number of keywords that are not supported in SAS Viya.
CAS	Number of keywords that are not supported in CAS.

Table 4. Category Summary Report Columns and Descriptions

**A Service SAS Offers
SAS Viya Code Readiness
309 SAS Programs**

append.sas

Category	Keyword	Viya	CAS
statement	TITLE	0	4
statementDataStep	CARDS	0	3
	INPUT	0	3

Output 4. Partial Category Summary Report

DETAIL REPORTS

One detail report is produced for each SAS job. These reports contain the detail information used in all summary reports and can be used to determine specific instances where a keyword, macro statement, or %INCLUDE statement was identified. These reports contain the following information:

Column	Description
Category	Source code submitted to the utility is divided into four sections: OPENCODE (code appearing between a QUIT or RUN statement and a DATA step or procedure statement), DATASTEP, PROCEDURE, and LIBNAME.
Search Type	Keyword searches are context sensitive, because the same keyword could appear in a DATA step or procedure, and SAS Viya or CAS might not support this keyword in both instances.
Restriction	The utility is based on keywords that are not supported by either SAS Viya or CAS. This column indicates which environment restricts use of the keyword.
Keyword	The SAS Viya or CAS restricted keyword.

Column	Description
Flags	Three flags provide various information about the line of source code. M indicates that the macro keyword was identified. I indicates that the %INCLUDE statement was identified. D indicates that a data-related statement was identified, such as a DATA statement, a procedure statement, or a LIBNAME statement. These statements must be evaluated individually to determine whether code changes are necessary for the program to run in CAS.
Line	Line number of the program code assigned during source code preprocessing. Maps to the LOC column in the Program Summary report. Maps to the Line Number / Step column in the Step Summary report.
Code	The line of code evaluated.

Table 5. Detail Report Columns and Descriptions

```

append.txt
-----
Flags: M=%macro I=%include D=Data Related

Category Search Type Restriction Keyword Flags Line Code
-----
DATASTEP No Matching Keywords D 1 DATA TODAY;
DATASTEP statementDataStep CAS INPUT 2 INPUT ID AMOUNT TYPE;
DATASTEP No Matching Keywords 3 DATE='01DEC87'd;
DATASTEP No Matching Keywords 4 FORMAT DATE DATE7.;
DATASTEP statementDataStep CAS CARDS 5 CARDS;
DATASTEP No Matching Keywords 6 0001 450.27 1 0002 200.50 2 0024 10.00 1 0045 299.80 1 0700
PROCEDURE No Matching Keywords D 7 PROC APPEND BASE=JOURNAL;
PROCEDURE No Matching Keywords D 8 PROC PRINT DATA=JOURNAL;
PROCEDURE statement CAS TITLE 9 TITLE 'MASTER AFTER FIRST TRANSACTIONS ADDED';
DATASTEP No Matching Keywords D 10 DATA TODAY2;
DATASTEP statementDataStep CAS INPUT 11 INPUT ID AMOUNT TYPE;
DATASTEP No Matching Keywords 12 DATE='01DEC87'd;

```

Output 5. Partial Detail Report

UTILITY FINDINGS

Using the Frequency report, we can calculate a SAS Viya Readiness score that is based on what percentage of the SAS code can benefit from the distributed processing that CAS provides. The following findings are based on the 309 SAS jobs in the SAS 9.4 (T1M5) CORE sample directory (C:\Program Files\SASHome\SASFoundation\9.4\core\sample).

In the first finding, shown in Figure 1, we see that 8.03% of the 2,629 steps are CAS enabled, provided that the source and target tables for the procedures are CAS tables. For the in-database procedures (MEANS, REPORT, SUMMARY, TABULATE) the functionality supported in CAS is the same SAS has provided for years by running these procedures in-database.

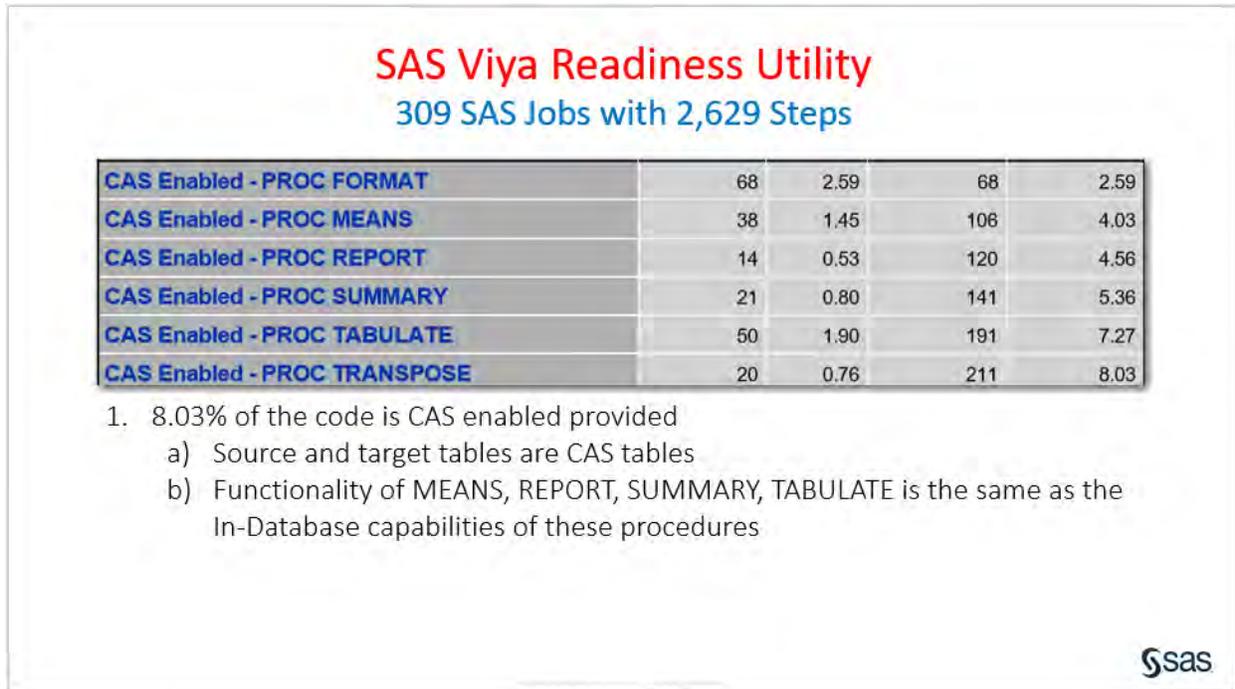


Figure 1. Base SAS Procedures

In all of the customer code reviews that I have performed over the last six years, it has become clear that the DATA step is a very popular coding technique that has been embraced by the SAS community. As we see in Figure 2, 25.37% of the steps are the DATA step. This value is representative, but it has also been much higher in multiple customer code reviews I have performed. For example, when I worked with a company in the Health-Care industry, the DATA step represented 70% of all of the code.

SAS Viya Readiness Utility

309 SAS Jobs with 2,629 Steps

Candidate for CAS - DATA STEP	667	25.37	878	33.40
--------------------------------------	-----	-------	-----	-------

1. 25.37% of the code is DATA Step which is CAS enabled provided
 - a) Source and Target tables are in CAS
 - b) Review category_summary.pdf to gain an understanding of which key words are not CAS enabled and which SAS job they appear in
 - a) WHERE clauses can push down to CAS provided the WHERE is on the Source(s) table(s) to the DATA Step
 - b) CARDS, EXECUTE, FILE, INFILE, INPUT, MODIFY, REPLACE, SYMGET, SYMPUT, SYMPUTX....
 - a) Leave these DATA Steps "as is" because these statements do not execute in CAS
 - c) %include statements
 - a) Can be in the body of a DATA Step and still run in CAS provided all the statements in the include file are CAS enabled



Figure 2. DATA Step

In Figure 3, we can review the analytical procedures used in the sample code. The value of 4.85% is low and not representative of actual customer reviews.

SAS Viya Readiness Utility

309 SAS Jobs with 2,629 Steps

Candidate for CAS - PROC ANOVA	2	0.08	880	33.47
Candidate for CAS - PROC CORR	21	0.80	901	34.27
Candidate for CAS - PROC GAM	1	0.04	902	34.31
Candidate for CAS - PROC GLM	22	0.84	924	35.15
Candidate for CAS - PROC PRINCOMP	1	0.04	925	35.18
Candidate for CAS - PROC REG	7	0.27	932	35.45
Candidate for CAS - PROC STANDARD	5	0.19	937	35.64
Candidate for CAS - PROC UNIVARIATE	67	2.55	1004	38.19
Workspace Server - PROC IML	1	0.04	2070	78.74

1. 4.85% of the code are analytical procedures
 - a) Have a SAS analytical coders review these and determine if they can simulate these procedures using CASL or other VDMML procedures.



Figure 3. Analytical Procedures

In Figure 4, we can identify which SAS jobs contain the analytical procedures.

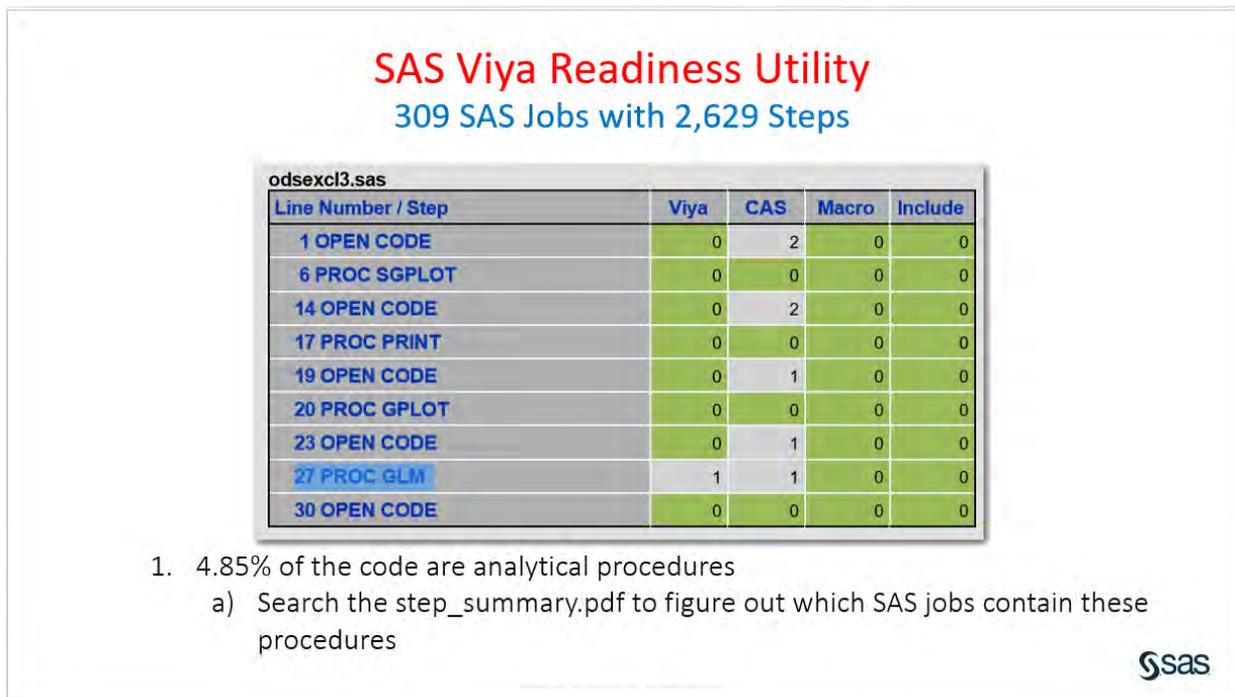


Figure 4. STEP_SUMMARY Report

In Figure 5, we see that 3.65% of the sample code is PROC SQL code. This value is extremely low and not representative of actual customer code reviews. Like the DATA step, PROC SQL is a very popular coding technique that typically is in the range of 15% to 25% of the code.

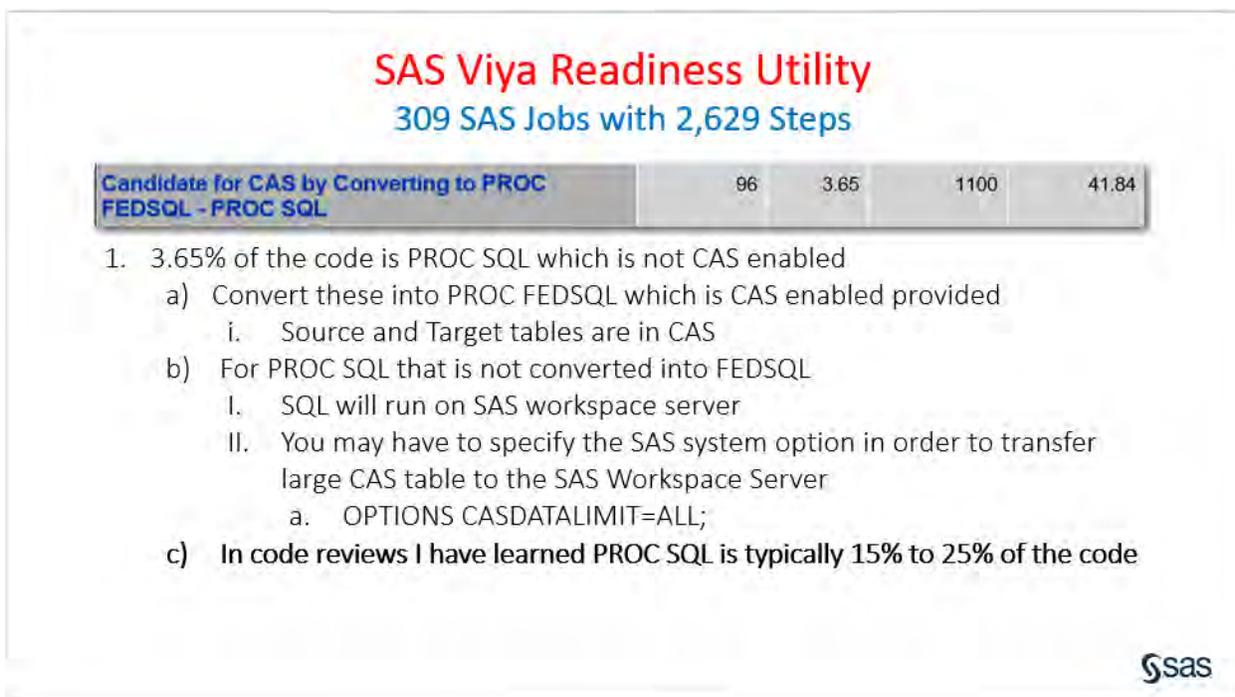


Figure 5. PROC SQL

In Figure 6, we see that 4.15% of the code is PROC SORT code. This value is also extremely low and not representative of actual customer reviews, where I typically see a value in the range of 15% to 25%.

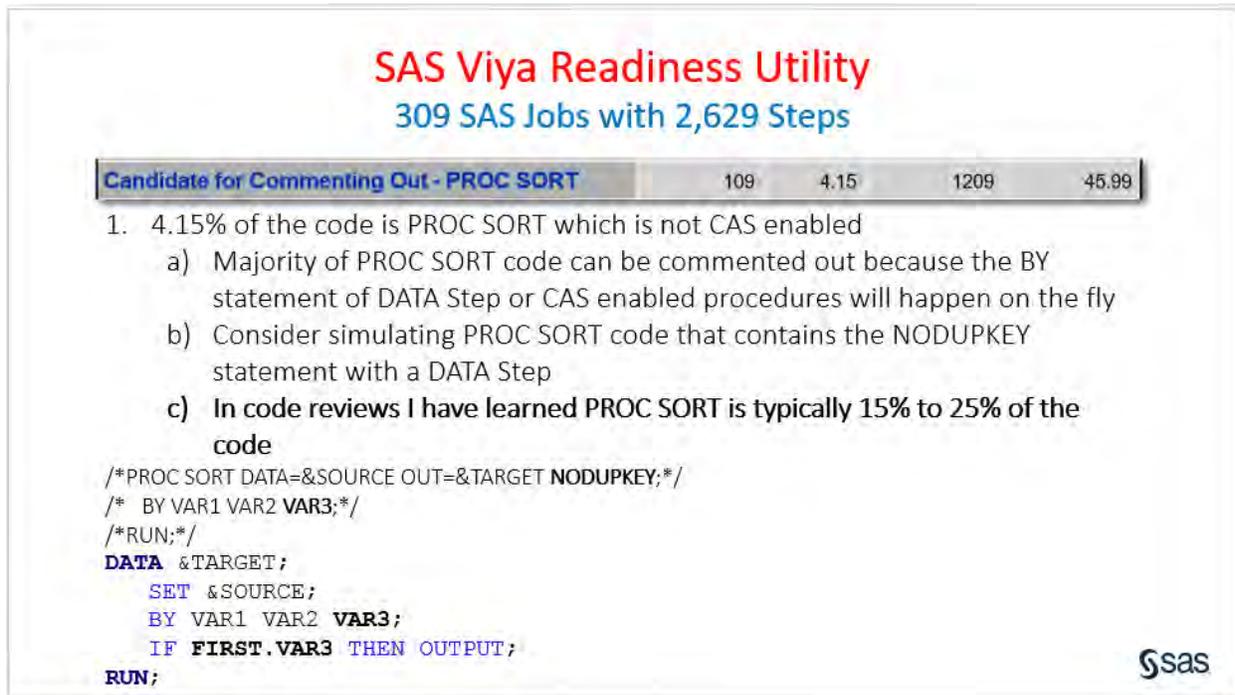


Figure 6. PROC SORT

In Figure 7, we see that OPEN CODE is 24.57% of the code, which is very representative of actual customer code reviews. OPEN CODE is any code between the DATA step or procedure boundaries (for example, LIBNAME statements, FILENAME statements, ODS statements, and macro declarations). What I have noticed in code reviews is that a lot of the DATA step code can be part of the macro declarations. Macros can be useful in generating code and submitting the code to the CAS server to run. However, the macro language itself does not run on the CAS server.

In calculating the SAS Viya Readiness score, we never include the OPEN SOURCE percentage. This implies we are being very conservative in our score calculation.

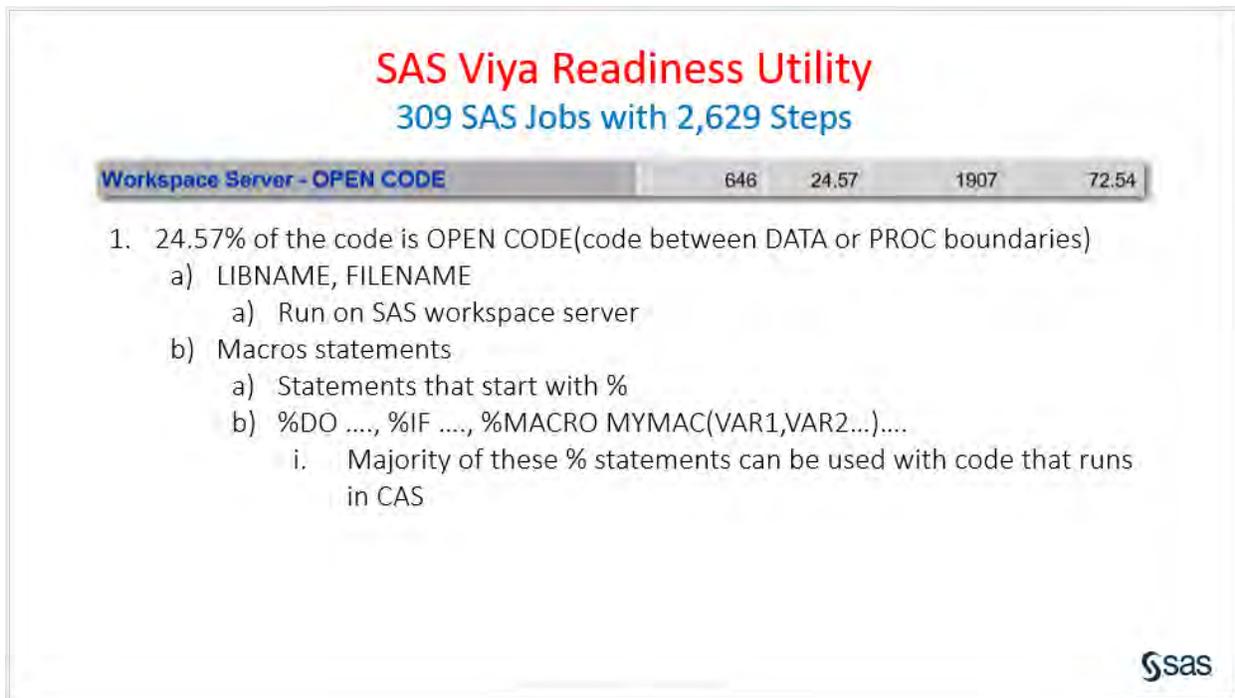


Figure 7. OPEN CODE

In Figure 8, we see that PROC APPEND code represents 0.30% of the code. This value is low and not representative of actual customer reviews. Typically, the value is in the range of 1% to 5%.

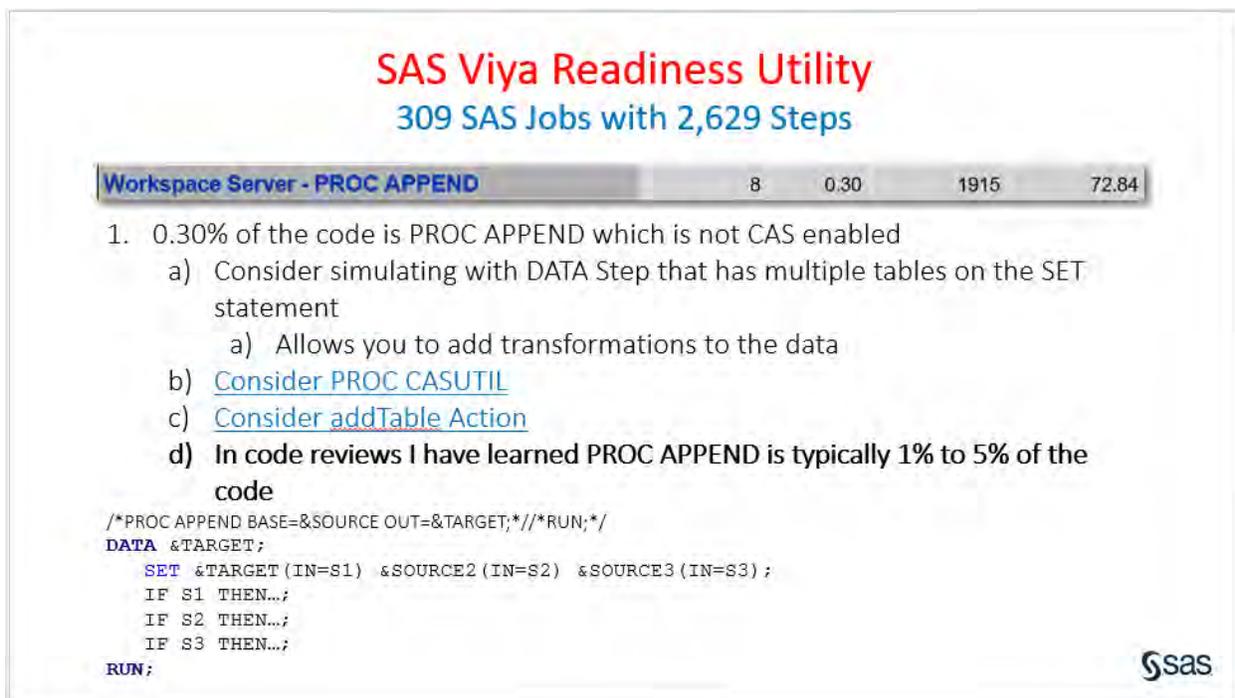


Figure 8. PROC APPEND

In Figures 9 and 10, we see all of the procedures that must run on a SAS 9.4 (T1M5) workspace server. The value of 27.91% of the code is representative of actual code reviews.

SAS Viya Readiness Utility

309 SAS Jobs with 2,629 Steps

In-Database / MapReduce - PROC FREQ	24	0.91	1233	46.90
In-Database / MapReduce - PROC RANK	24	0.91	1257	47.81
Workspace Server - LIBNAME	4	0.15	1261	47.97
Workspace Server - PROC CALENDAR	24	0.91	1939	73.75
Workspace Server - PROC CHART	52	1.98	1991	75.73
Workspace Server - PROC COMPARE	8	0.30	1999	76.04
Workspace Server - PROC CONTENTS	8	0.30	2007	76.34
Workspace Server - PROC DATASETS	17	0.65	2024	76.99
Workspace Server - PROC DISPLAY	2	0.08	2026	77.06
Workspace Server - PROC DOCUMENT	3	0.11	2029	77.18
Workspace Server - PROC FORMS	11	0.42	2040	77.60



Figure 9. Non-CAS Enabled Procedures

In Figure 10, we can review the scores based on our findings. It turns out that 41.50% of the 2,629 steps of the CORE sample code is compatible with SAS Viya, while 27.91% of the steps must run on a SAS workspace server.

The score of 41.50% is low and not representative of actual customer code reviews. Here are the actual findings from running the utility on 4,853 customer SAS jobs containing 130,036 steps (drum roll): 64.11% of the steps are compatible with SAS Viya, while 10.80% of the steps must run on the SAS 9.4 (T1M5) workspace server. Now you might be wondering about the remaining 25% of the code. This 25% represents the analytical procedures and OPEN CODE. We must manually review this code to determine what analytical procedures we can simulate using [PROC CAS](#) or the other [CAS-enabled procedures](#) delivered with SAS Viya 3.3. We also have to review the OPEN CODE to determine whether the [macros](#) being defined allow code to run in CAS.

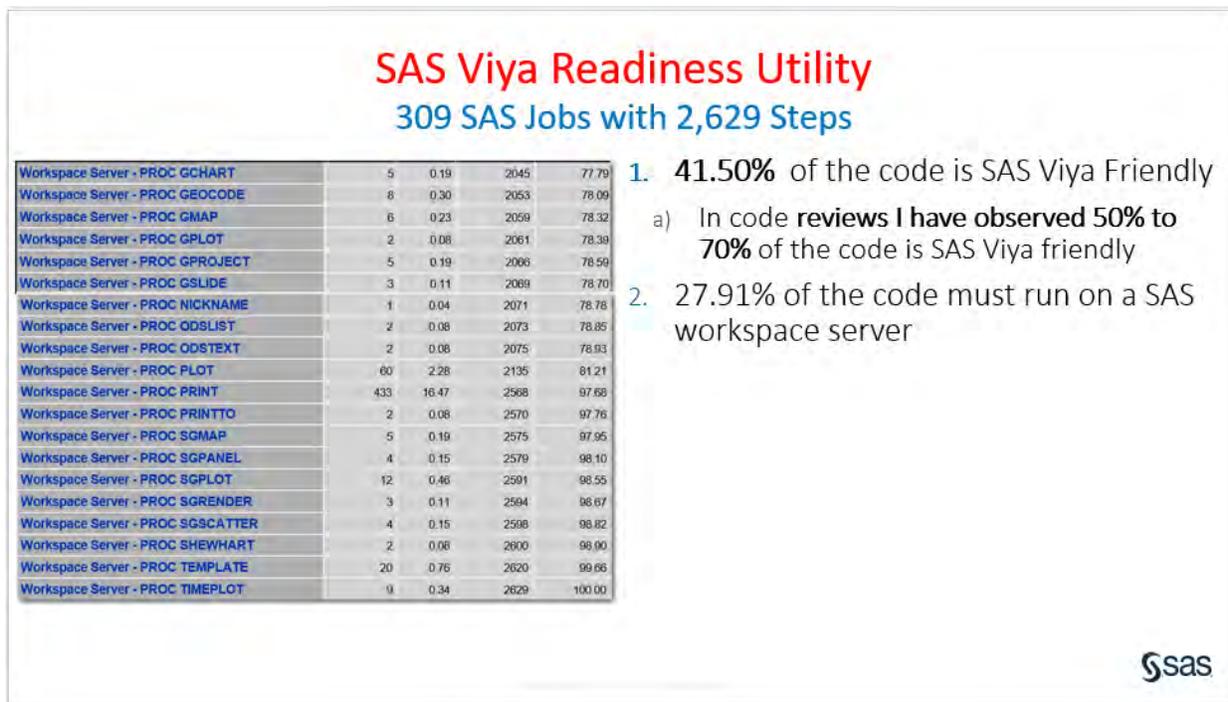


Figure 10. Non-CAS Enabled Procedures

In Figure 11, we can see the suggested next steps after determining the utility findings. If you are interested in having your SAS code run through the utility, contact Steven.Sober@sas.com.

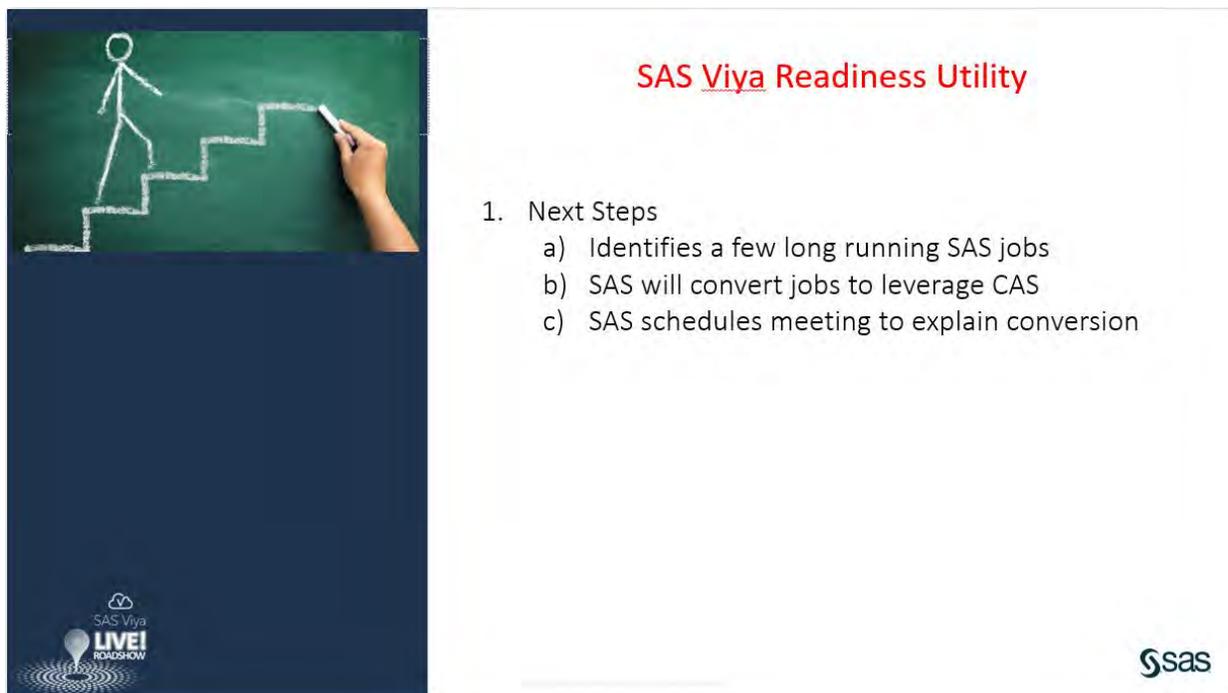


Figure 11. Next Steps

CONCLUSION

Base SAS coding techniques run faster when you leverage SAS Cloud Analytic Services.

ACKNOWLEDGMENTS

I would like to thank Bryan Simmons, Debbie Hartman, Brian Kinnebrew, Phil Weiss, and Clark Bradley for their insights and support.

RECOMMENDED READING

SAS Institute Inc. 2017. "Procedures That Use CAS Actions." In *Base SAS 9.4 Procedures Guide*. 7th ed. Cary, NC: SAS Institute Inc. Available: http://go.documentation.sas.com/?cdcId=pgmsascdc&cdcVersion=9.4_3.3&docsetId=proc&docsetTarget=titlepage.htm&locale=en (accessed March 1, 2018).

SAS Institute Inc. 2017. "The DATA Step and CAS." In *SAS Cloud Analytic Services 3.3: DATA Step Programming*. Cary, NC: SAS Institute Inc. Available: http://go.documentation.sas.com/?cdcId=pgmsascdc&cdcVersion=9.4_3.3&docsetId=casdspm&docsetTarget=n17qunh9lpqavtn1t8mv40r0s5tu.htm&locale=en (accessed March 1, 2018).

Secosky, Jason. 2017. "DATA Step in SAS Viya: Essential New Features." *Proceedings of the SAS Global Forum 2017 Conference*. Available: <https://www.sas.com/content/dam/SAS/support/en/technical-papers/SAS0118-2017.pdf>.

Weiss, Phil. "Getting Your SAS® Code to Run Multi-Threaded in SAS Viya 3.3." Available: https://blogs.sas.com/content/sgf/files/2018/01/TechnicalWhitePaper_SASViya_Jan2018_FINAL.pdf. Last modified January 2018. Accessed on March 1, 2018.

CONTACT INFORMATION

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

Steven Sober
SAS Institute Inc
919 531 9644
Steven.Sober@sas.com
<http://www.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.