

A Practical Application of VTABLE

Dr. Kannan Deivasigamani, HSBC, Sumathi Subramaniam, Citi

ABSTRACT

This SESUG paper demonstrates a practical application of VTABLE in SAS® which is also referred as the Dictionary. While VTABLE is rich in meta-data, the example presented will just be one of the many possible ways VTABLE can be used. This paper is intended for audiences who wish to leverage the meta-data of the files used in their reporting process and are developers or programmers. This is commonly noticed in operational reporting and the industry that the example here applies is the financial operational reporting within mortgage banking business.

INTRODUCTION

You may have read about “Dictionary.Tables”, which is a source of information within a SAS session available for developers. Programmers and application developers can take advantage of meta-data based on different requirements and can make the application or program more robust due to the ability to process based on meta-data. Many articles and papers have been published in the past many years about the different types of information available within SAS VTABLEs (Lafler, 2005). This paper provides an idea of how it is used in some leading banks.

BUSINESS REQUIREMENT

We have some processes that look for transactions recorded in the past 30 days or past 10 days depending on the business requirement. They can be based on files that may have been on an irregular schedule. In this example, let us work with a requirement that necessitates the processing of most recent files received within the past hour, the business needs to process 5 files received within the past 2 days.

This calls for a way to look up the creation date of the file within SAS. Fortunately, SAS provides session tracking variables that are available through VTABLE which is available through SASHELP library. This not only stores information about SAS files that are part of the standard SAS libraries, but also stores information from a user's current SAS session and libraries defined within the program and file-names defined within the program. The columns those are available for reading provides useful information that allows a user to operate with much flexibility. In this paper, I show example code of how the files are created and the corresponding Meta data is stored in the SASHELP.VTABLE Meta data dataset and how it can be retrieved to dynamically build code that can further be used in subsequent processing.

A PRACTICAL EXAMPLE

For ease of testing, the example was coded and tested in SAS University edition. The SAS sample dataset “CLASS” was used in this example. The program creates 5 datasets with different number of observations just to have some distinction between each of them. There is a delay introduced to simulate multiple days or weeks of data interval. In this case a 60 second delay has been forced. The SLEEP function has been called to create this forced pause within the code in a macro. This macro is called after each SAS dataset is created. Doing this ensures that the create timestamp of each dataset differs by a minute or more. After the creation of the files, a step queries the DICTIONARY Table by invoking the VTABLE in the next step using the variable pointing to the timestamp of the dataset and by filtering those that were created in the past 3 minutes. This retrieves the files created in the past few minutes which is only a few of the files as shown in the results. The selected files can be further carried over to other processing based on the business need.

EXAMPLE TESTED USING SAS UNIVERSITY EDITION

The code below shows, how this logic works using SAS University edition:

```
options nocenter mlogic mprint symbolgen obs=max
errors=1;

data _null_;
    starttime = datetime();
    format starttime datetime.;
    put '==Start Timestamp==' starttime;
run;

libname lib1 '/folders/myfolders/data1';
filename file1 '/folders/myfolders/data1/file1.txt';

%macro wait_a_minute;
    data _null_;
        call sleep(1,60);
    run;
%mend wait_a_minute;

data lib1.dsn1;
    set sashelp.class (obs=1);
run; %wait_a_minute;

data lib1.dsn2;
    set sashelp.class (obs=2);
run; %wait_a_minute;

data lib1.dsn3;
    set sashelp.class (obs=3);
run; %wait_a_minute;

data lib1.dsn4;
    set sashelp.class (obs=4);
run; %wait_a_minute;

data lib1.dsn5;
    set sashelp.class (obs=5);
run; %wait_a_minute;
```

```

proc contents data=lib1._all_;run;

data in2;
  set sashelp.vtable;
  where libname = 'LIB1'
    and crdate ge datetime() - (60*60);

  format ts datetime.;
  ts = datetime()-(60*60);

  call symput ('ts',ts);
run;

proc print data=in2;
  format crdate datetime.;
  title "==in2== &ts";
run;

data new;
  set in2(keep=memname crdate);
  where crdate ge datetime() - (60*4);
run;

proc print data=new;
  title1
  '=====';
  title2 'Data from files created based on parm';
  title3 "Created on &sysdate, by &sysjobid";
run;

data _null_;
  endtime = datetime();
  format endtime datetime.;
  put '==End Timestamp==' endtime;
run;

```

Figure 1 is a Visio process flow showing the steps in the program.

PROCESS FLOW - AN APPLICATION OF VTABLE

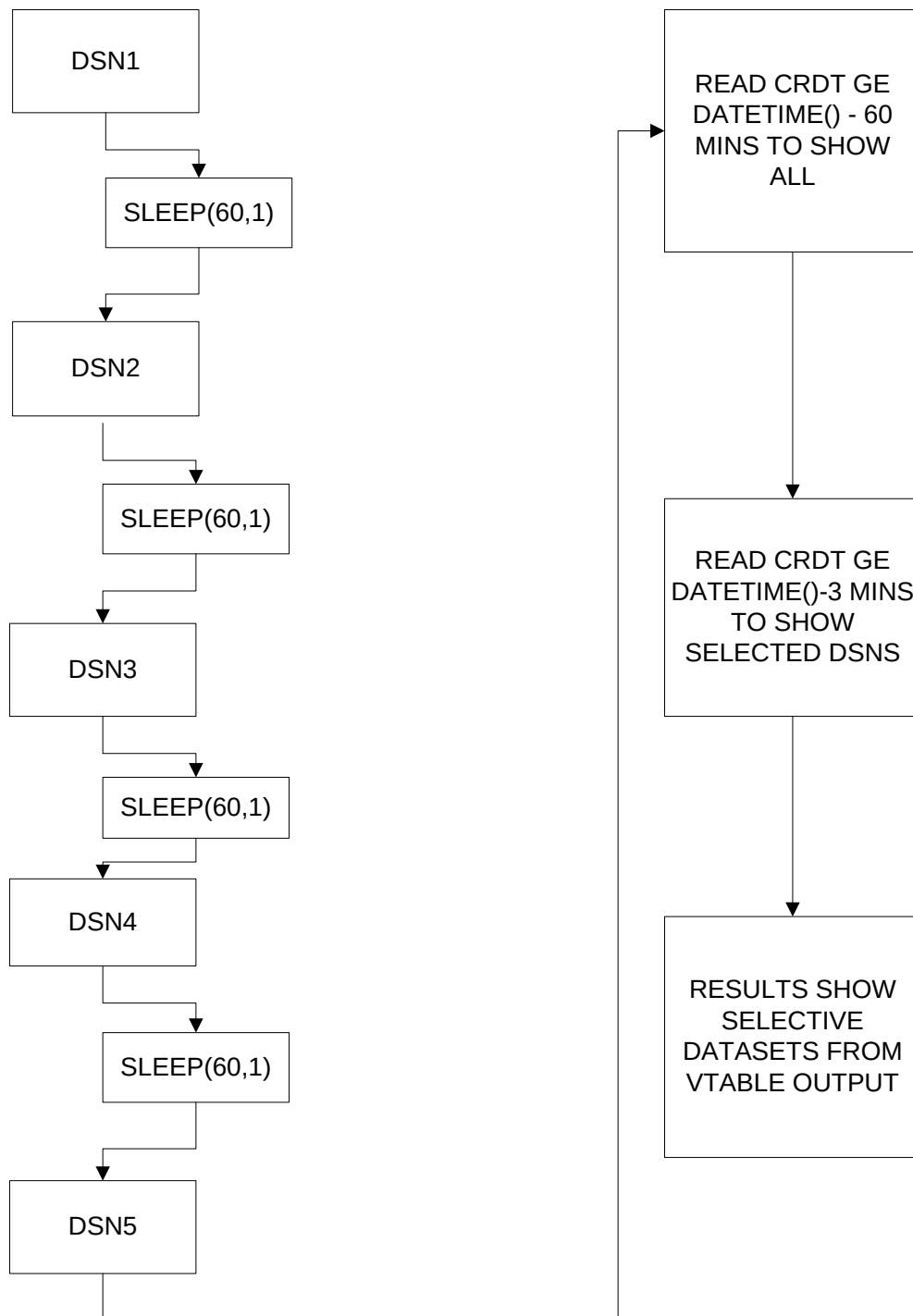


Figure 1. Process flow of the code example

Figure 2 displays the datasets created within the code and has the creation timestamp displayed in the second last column from the right labeled as “crdate”.

Obs	libname	memname	memtype	dbms_memtype	memlabel	typemem	crdate	modate
1	LIB1	DSN1	DATA			DATA	27SEP17:15:08:16	27SEP17:15:08:16
2	LIB1	DSN2	DATA			DATA	27SEP17:15:09:16	27SEP17:15:09:16
3	LIB1	DSN3	DATA			DATA	27SEP17:15:10:16	27SEP17:15:10:16
4	LIB1	DSN4	DATA			DATA	27SEP17:15:11:16	27SEP17:15:11:16
5	LIB1	DSN5	DATA			DATA	27SEP17:15:12:16	27SEP17:15:12:16

Figure 2

The results of the tested code are displayed in Figure 3.

=====		
<i>Data from files created based on parm</i>		
<i>Created on 27SEP17, by 9540</i>		
	=====	
Obs	memname	crdate
1	DSN3	27SEP17:15:10:16
2	DSN4	27SEP17:15:11:16
3	DSN5	27SEP17:15:12:16

Figure 3

CONCLUSION

The availability of meta-data through VTABLE has made our reporting more slick and hope it benefits your EUC and reports as well as it did for us. As it is clear, the data points available in the DICTIONARY TABLE are many and can be used in different ways based on user requirements. Customized solutions are possible with VTABLES and are up to the users to determine what is best for one's organizational needs.

REFERENCES

Lafler, Kirk Paul. April 10-13, 2005. Exploring Dictionary Tables and Views. *Proceedings of the SUGI 30*, Philadelphia, PA. Available at <http://www2.sas.com/proceedings/sugi30/070-30.pdf>

ACKNOWLEDGMENTS

I would like to thank Harry Droogendyk and Ron Fehd for accepting my paper and having me present it at SESUG 2017. Many thanks to my parents for proof reading my paper and my children for helping with logistic support and providing encouragement. Thanks to the senior management team from HSBC and the senior management from Citi for allowing us to participate and present as their employee and for providing the time to travel and do the same.

RECOMMENDED READING

- *Base SAS® Procedures Guide*
- *SAS® For Dummies®*

CONTACT INFORMATION

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

Dr. Kannan Deivasigamani

HSBC

TEXTdrk@gmail.com

<https://www.linkedin.com/in/kannandeivasigamani>

Sumathi Subramaniam

Citi

Sumisubra_us@yahoo.com

<https://www.linkedin.com/in/sumathi-subramaniam-910b54a2/>