

VIEWS

International SAS Programmer Community



News, Hints, Tips and Information for SAS® Users

Issue 66, 4th Quarter 2021

See www.sas.com/en_us/events/sas-global-forum.html for more details.

This newsletter issue will be a SAS macro special, as it contains an article from our new VIEWS Consultant Jay Iyengar about managing macro variables, and I've written a solution to a macro problem posed by Anders Skölleremo. As usual I have publicised some SAS-related events for the next few months, and I have also added some more SAS formats, options and functions that you may or may not have come across before, but you might find interesting, and even useful. First, though, is some news about SAS Global Forum 2022!

If you would like to contribute an article, to re-visit and improve an existing article, or just discuss the possibility of doing so, please feel free to send an email to me at views-uk@hollandnumerics.org.uk.

Philip R Holland (Newsletter Editor)

Did You Know?

If you have a useful hint or tip, please send it to the Editor, and share it with the VIEWS membership.

Managing Macro Variables

[SAS] In SAS DATA STEP programming, users can manage SAS variables on a data set through the DROP and KEEP statements, or respective data set options. What about macro variables? Does this capability extend to the SAS macro facility?

With %PUT, programmers can check and validate macro variables and values, and print macro variables with their current values in the SAS Log. In a %PUT statement, you can specify single macro variables, multiple macro variables, or categories of macro variables, such as automatic, user-defined, global and local. The example below uses %PUT _ALL_ to print all existing macro variables to the Log:

```
%PUT _ALL_ ;

GLOBAL CLIENTMACHINE 10.0.2.2
GLOBAL GRAPHINIT
GLOBAL GRAPHTERM
...
AUTOMATIC SYSTIME 19:04
AUTOMATIC SYSUSERID sasdmo
AUTOMATIC SYSVER 9.4
```

The SAS Macro facility features capabilities to help you manage macro variables during a SAS session. In the SAS community, much attention is given to the methods used to create and define macro variables; %LET, CALL SYMPUT, and PROC SQL INTO:. With %SYMDEL, macro variables can be deleted during a SAS session.

Deleting macro variables will be valuable in code that contains nested macro calls, and when executed will generate multiple local symbol tables. Also, deleting macro variables may be necessary to prevent conflicts between global and local macro variables which share the same name.

%SYMDEL can be used in open code, or inside a macro definition. It will only delete global macro variables. However, %SYMDEL can be used to delete single macro variables, as well as multiple macro variables. In contrast to %PUT, %SYMDEL cannot delete classes of macro variables through the use of keywords (_ALL_, _GLOBAL_, etc.).

This tip provides a technical solution to the issue of deleting groups of macro variables, using the **sashelp.vmacro** view. The **sashelp** view **vmacro** contains metadata on macro variables. As shown below, the view contains a list of defined and available macro variables in a current SAS session.

Membership of VIEWS is **free** and you can register for notification of all future VIEWS News issues by registering at blog.hollandnumerics.org.uk, and then emailing views-uk@hollandnumerics.org.uk, or using the blog site menu, to request access to the VIEWS UK forum. Members will receive an email notification whenever a new issue of the free quarterly VIEWS News is published on the web site, please inform us when you change your email address, otherwise your registration may be cancelled.

Stop Press!

Is this the End of the SAS Global Forum?

"We celebrate that many SAS users have 2022 plans to gather in person at the local and regional level, thanks to the efforts of dedicated leaders and volunteers. At SAS, we are collaborating with these event efforts to help bring the best users content to in-person experiences where it is safe to do so." - SAS web site.

- **Q1 - SAS Amplify:**
A virtual conference for SAS Partners.
- **Q2 - SAS Innovate:**
Face-to-face conferences for Business in the Americas, EMEA and Asia Pacific regions. So far only the conference in the Americas 2022 has been scheduled in San Diego, CA, where it was to hold SAS Global Forum 2022.
- **Q1/Q2 - SAS Hackathon:**
Virtual competition to convert curiosity into innovation.
- **Q3/Q4 - SAS Explore:**
A virtual conference for SAS Technologists.

Scope	Name	Offset	Value
GLOBAL	CLIENTMACHINE	0	10.0.2.2
GLOBAL	GRAPHINIT	0	
GLOBAL	GRAPHTERM	0	
GLOBAL	SASWORKLOCATION	0	"/tmp/SAS_workC27800.."
GLOBAL	SYSCASINIT	0	0
GLOBAL	SYSSTREAMINGLOG	0	True
GLOBAL	SYSUSERNAME	0	Sasdemo
GLOBAL	SYS_SQL_IP_ALL	0	-1
GLOBAL	SYS_SQL_IP_STMT	0	

Using the PROC SQL INTO clause with a SEPARATED BY option, developers can store a concatenated list of macro variable names in a macro variable. The example below demonstrates this task using the respective variable from the **vmacro** view in the **sashelp** library and inserting it into a macro variable named **macvname**.

```
PROC SQL NOPRINT;
  SELECT name
  INTO   :macvname SEPARATED BY ' '
  FROM   sashelp.vmacro
  WHERE  scope EQ 'GLOBAL'
        AND name NOT IN ('SYSCASINIT' 'GRAPHTERM'
                          'SYSUSERNAME' 'SYSSTREAMINGLOG'
                          'SYS_SQL_IP_STMT' 'SYS_SQL_IP_ALL')
  ;
QUIT;

%SYMDEL &macvname. macvname;

%PUT _GLOBAL_;
```

In the code, six macro variables have been excluded from the macro variable list which are necessary to run a SAS session. Using the scope variable, the query has been subsetting to include only the global macro variables. The **&macvname** macro variable has been placed in the %SYMDEL statement, which will resolve to the list of macro variables to delete. Finally the **macvname** macro variable itself has been deleted to prevent the list from cluttering up the Log.

```
85      %SYMDEL &macvname. macvname;
86
87      %PUT _GLOBAL_;
GLOBAL GRAPHTERM
GLOBAL SYSCASINIT 0
GLOBAL SYSSTREAMINGLOG true
GLOBAL SYSUSERNAME sasdemo

GLOBAL SYS_SQL_IP_ALL -1
GLOBAL SYS_SQL_IP_STMT
```

The excerpt from the SAS Log confirms the macro variables have been deleted from the global symbol table.

Jay Iyengar, Data Systems Consultants

Ask The Consultant

This part of VIEWS News is where you can get your technical questions answered. Send your questions to the Editor.

Saving resolved SAS code from macros

Q: How do you save the resolved SAS code from a macro, so it can be viewed for debugging and re-use?

I'm grateful for this question from Anders Skölleremo, VIEWS Consultant

A:[SAS] Anyone who knows me well will not be surprised that I would want to solve this puzzle, especially as Anders had stated: "It is NOT possible to do it in the way that I thought"!

First of all the resolved SAS code needed to be captured in some way. Fortunately there are a number of SAS options that could help here. MPRINT echoes the macro code, but not the resolved SAS code, to the Log. MLOGIC echoes the macro logic, i.e. the %IF and %DO statements, but again not the resolved SAS code. In fact MACROGEN does exactly what was needed, and also prefixes the resolved code with "MACROGEN(macro name)", so it is mostly self-documenting too.

Because the SAS code will be manipulating the SAS options to settings other than those currently in use, the redirection code saved the current settings, and the reset code restores them, so nothing will be permanently altered in the SAS session.

The following OPTIONS statement has been included purely to document the current settings:

```
OPTIONS NOTES NOMACROGEN NOMPRINT NOMLOGIC;
```

The following macro code is based on the macro discussed by Steve Bloom in *VIEWS News* 64:

```
%MACRO setnotrunc(
  dsout = /*the name of the output dataset*/
  ,dsin = /*the name of the input datasets
          separated by single white spaces*/
  ,indexvar = indexvar /*the name of the index
                        variable so the input dataset can
                        be identified in the output dataset*/
);

PROC SQL;
  CREATE TABLE &dsout. AS
  SELECT *
    ,1 AS &indexvar.
  FROM   %scan(&dsin., 1, %str( ))
  %do igc = 2 %to
    %sysfunc(countw(&dsin., %str( )));
  OUTER UNION CORRESPONDING
  SELECT *
    ,&igc. AS &indexvar
  FROM   %scan(&dsin., &igc., %str( ))
  %end;
  ;
QUIT;

%MEND;
```

```

%MACRO concat(
  dsout = /*the name of the output dataset*/
  ,dsin = /*the name of the input datasets
          separated by single white spaces*/
);

DATA &dsout.;
  SET %scan(&dsin.,1, %str( ))
%do igc = 2 %to
  %sysfunc(countw(&dsin., %str( )));
  %scan(&dsin., &igc., %str( ))
%end;
;
RUN;

%MEND;

* Test code for the macros *;
DATA a;
  x = "w";
RUN;

DATA b;
  x = "wc";
RUN;

```

The following code should be added before the macro calls you would like to view. First the current settings of the SAS options that will be changed are saved in macro variables. Secondly the SAS options are changed to settings suitable for viewing the resolved macro code. Finally a text file is created in the WORK library folder to store that code, and the Log text redirected to it.

```

*-----Redirect: start-----*;
* Save current options *;
DATA _NULL_;
  CALL SYMPUT('macrogen'
              ,getoption('macrogen'));
  CALL SYMPUT('mprint'
              ,getoption('mprint'));
  CALL SYMPUT('mlogic'
              ,getoption('mlogic'));
  CALL SYMPUT('_notes'
              ,getoption('notes'));
  CALL SYMPUT('_ls'
              ,getoption('linesize'
                          , 'keyword'));

  STOP;
RUN;

* Update options *;
OPTIONS MACROGEN NOMPRINT NOMLOGIC NONOTES
        LS = 256;

* Redirect log to a text file *;
FILENAME text
  "%sysfunc(PATHNAME(work))/resolved.txt";
PROC PRINTTO LOG = text NEW;
RUN;
*-----Redirect: end-----*;

```

Note that it is **not** possible to redirect the Log output to the Print output using PROC PRINTTO, which is why I've redirected the Log output to a temporary text file instead!

Now to call the macros of interest:

```

* Call macros *;
%setnotrunc(dsout=dsout, dsin=a b);
%concat(dsout=dsout2, dsin=a b);

```

After the macro calls have been run restore the writing of the records to the SAS Log, restore the SAS option settings to those saved in macro variables, except the LINESIZE (LS), then print the relevant text records from the redirected Log to the Output destination, and restore the LINESIZE setting.

```

*-----Reset+print: start-----*;
* Reset reditect of log *;
PROC PRINTTO;
RUN;

* Reset options *;
OPTIONS &macrogen. &mprint. &mlogic.
        &notes.;

* Copy resolved macro statements in text file to
  output window *;
DATA _NULL_;
  INFILE text TRUNCOVER LRECL = 256;
  LENGTH previous $256;
  INPUT;
  IF _INFILE_ EQ: 'MACROGEN' THEN DO;
    previous = LAG(_INFILE_);
    FILE PRINT;
    IF SCAN(previous, 1, ':') NE
      SCAN(_INFILE_, 1, ':')
      AND previous NE '' then PUT / '-----';
    PUT _INFILE_;
  END;
RUN;

* Reset options *;
OPTIONS &ls.;
*-----Reset+print: end-----*;

```

The output from this program is given below:

```

MACROGEN(SETNOTRUNC):  PROC SQL;
MACROGEN(SETNOTRUNC):  CREATE TABLE dsout AS SELECT *, 1
AS indexvar FROM a OUTER UNION CORRESPONDING SELECT *, 2
AS indexvar FROM b ;
MACROGEN(SETNOTRUNC):  QUIT;
-----
MACROGEN(CONCAT):      DATA dsout2;
MACROGEN(CONCAT):      SET a b ;
MACROGEN(CONCAT):      RUN;

```

Note that the LISTING Output destination line length is limited to 256 characters, which may be exceeded by the PROC SQL statement, so resolving such code should be limited to a text file, and not a LISTING Output destination.

Philip R Holland (Newsletter Editor)

Formats, Options, and Functions

This section is devoted to the description of useful, or unusual, SAS Formats, Options, and Functions. Please read the SAS documentation to find out more about them.

[SAS] The SAS formats \$OCTALw. and OCTALw. convert character and numeric values to their corresponding octal (base 8) representations, respectively. The SAS informat \$OCTALw. converts octal representations to character values.

[SAS 9.6M6] SAS system option ACCESSIBLECHECK detects and logs ODS output that is not accessible. Beginning in SAS 9.4M6, accessible tables and graphs are available for the ODS HTML5, ODS PDF and ODS EPUB destinations. Accessible tables and graphs are not available for the ODS LISTING and the ODS OUTPUT destinations. The level of accessibility for other ODS destinations has not been determined.

The default option setting though is **NOACCESSIBLECHECK**, which does not check for common violations of accessibility standards.

[SAS] The SAS function **TRANSLATE()** replaces individual characters in a text variable. The related **TRANWRD()** function, which has been mentioned here in previous issues, replaces sub-strings.

In Brief

- Back issues of *VIEWS News* are available from the VIEWS UK forum at **blog.hollandnumerics.org.uk**. Please register for free membership on the blog, and then request free access to the VIEWS UK forum.
- The SAS Professional Forum group on LinkedIn, which currently has over 36,900 members. Everyone is welcome to request to join, but discussions here are restricted to SAS-related topics: **www.linkedin.com/groups?gid=70702**.
- The SAS Opportunities group on LinkedIn, which currently has over 14,300 members. Everyone is welcome to request to join, but discussions here are also restricted to SAS-related topics: **www.linkedin.com/groups?gid=805217**.
- In the UK and Ireland the SUGUKI group had been holding regular meetings in London, and also occasional meetings in Bristol, Dublin, Edinburgh, Leeds and Manchester. Unfortunately lockdown has necessitated running webinars instead. For more details and to register for future events please go to **meetup.com/SUGUKI**.
- Fancy joining the SAS EMEA Analytics Explorers, then please send your **country, name, email address** and **company** with your permission to pass these details on to SAS to **views-uk@hollandnumerics.org.uk**. Other regions include Asia Pacific, Canada, Latin America and US, but to join these communities based on where you live, you will now need to provide this information to a current member of these communities, instead of me.
- WPS Community Edition provides users with free access to WPS Analytics software including a SAS language compiler with coding and workflow tools on unlimited data volumes. A Community Edition licence key will be issued that will activate the software for up to six months. At the end of this term, users may be invited to renew their licence and to upgrade to the latest available version of the software. No technical support service is offered with the Community Edition. Download from **www.worldprogramming.com/try-or-buy/editions**. Note that the software is made available without warranty or indemnity, and is not made available to US residents.

Diary

Are you organising an event that would be of interest to the VIEWS readership? Let us know as we are interested in all non-profit making events related to SAS.

Q1 2022

?? **SAS Amplify**, Virtual (dates to be confirmed)

A virtual conference for SAS Partners. For more details please go to the SAS Events web site at **www.sas.com/en_us/events/sas-global-forum.html**.

Q1/Q2 2022

?? **SAS Hackathon**, Virtual (dates to be confirmed)

A virtual competition to convert curiosity into innovation. For more details please go to the SAS Events web site at **www.sas.com/en_us/events/sas-global-forum.html**.

April 2022

10-13 **SAS Innovate Americas**, San Diego, CA, USA

A face-to-face conference for Business. For more details please go to the SAS Events web site at **www.sas.com/en_us/events/sas-global-forum.html**.

Q2 2022

?? **SAS Innovate EMEA** (dates/venue to be confirmed)

A face-to-face conference for Business. For more details please go to the SAS Events web site at **www.sas.com/en_us/events/sas-global-forum.html**.

?? **SAS Innovate Asia Pacific** (dates/venue to be confirmed)

A face-to-face conference for Business. For more details please go to the SAS Events web site at **www.sas.com/en_us/events/sas-global-forum.html**.

Q3/Q4 2022

?? **SAS Explore**, Virtual

A virtual conference for SAS Technologists. For more details please go to the SAS Events web site at **www.sas.com/en_us/events/sas-global-forum.html**.

Contacts

Jay Iyengar , Data Systems Consultants	Web: www.linkedin.com/in/datasysconsult
SAS UK	Tel: +44-(0)1628-486933 Web: www.sas.com/uk

VIEWS

Web Site	Web: blog.hollandnumerics.org.uk
Newsletter Editor	Email: views-uk@hollandnumerics.org.uk

VIEWS News is published quarterly for the VIEWS International SAS Programmer Community by Holland Numerics Limited.