

Paper 086-2008

ODS and Output Data Sets: What You Need to Know

Myra A. Oltsik

Mediamark Research Inc., New York, NY

ABSTRACT

Most programmers use ODS to make pretty reports. I don't. I use ODS to collect data from a PROC using the ODS OUTPUT statement. The SAS Help documentation for using ODS OUTPUT is quite light, and information about table names — needed to create data sets — and their descriptions are spread out among the documentation. And while the documentation may say that an ODS OUTPUT table is available, that doesn't mean it's necessarily "usable", or that it looks like you might expect it to look¹. This paper attempts to put most everything you need to know in one place.

There are over 90 SAS® procedures for which ODS OUTPUT tables are available, over 75% of which are either SAS/STAT or SAS/ETS procedures. I am not a statistician, but there are enough Base SAS® PROCs with ODS OUTPUT tables to make this paper of interest to all.

INTRODUCTION

There is some confusion in the nomenclature of ODS: the Output *Object* is not the same as the Output *Destination*. ODS produces the Output object no matter what type of destination you designate: Listing, PDF, HTML, RTF, CSV or Output. The Output destination is a SAS® data set with the same structure you'd get by running a DATA step. For the remainder of this paper, when I refer to ODS OUTPUT, I will be referring to the ODS OUTPUT *Destination*.

Producing an ODS OUTPUT data set (or table, the terms are used interchangeably) requires a simple statement before running a procedure:

```
ods output <ODS table name>=<designated data set name>;
```

The simplest way to find out the output ODS table name for a PROC is to use the [ODS TRACE ON](#) statement, for example:

```
A) ods trace on;
   proc freq data=test.zip_code_data;
       tables FLAG;
   run;
   ods trace off;
```

This produces the following in the Log:

```
Output Added:
-----
Name:          OneWayFreqs
Label:         One-Way Frequencies
Template:      Base.Freq.OneWayFreqs
Path:          Freq.Table1.OneWayFreqs
-----
```

The "Name" is the ODS table name needed to produce a data set. The following code shows how to create a data set this way:

¹ The SAS® Help documentation states, "Because ODS already knows the logical structure of the data and its native form, ODS can output a SAS data set that represents exactly the same resulting data set that the procedure worked with internally."

```

B) ods listing close;
   ods output OneWayFreqs=OneWay;
   proc freq data=test.zip_code_data;
       tables FLAG;
   run;
   ods listing;

```

VIEWTABLE: One-Way Frequencies							
	Table	F_FLAG	FLAG	Frequency	Percent	CumFrequency	CumPercent
1	Table FLAG	0	0	2461	76.64	2461	76.64
2	Table FLAG	1	1	750	23.36	3211	100.00

Note that this table has more information in it than the output table created from the / OUT= statement. It includes the cumulative frequencies and percents that you'd see in the listing output. In addition, OneWayFreqs labels the name of the table and gives a formatted value of the displayed value. In this case, since FLAG isn't formatted, F_FLAG is a character copy of FLAG. This field is called FVARIABLE and can be deleted from the OneWayFreqs template associated with the OneWayFreqs table name.² Templates will be addressed later in this paper.

The [ODS TRACE ON](#) statement only gives the ODS table names associated with the *specific* statements and options used in this PROC FREQ. In fact, there are over 50 ODS table names associated with PROC FREQ alone, each depending upon the information the procedure statements request.

TABLES OF TABLE NAMES

If a procedure has ODS tables available, the information will be in a table in the "Details" section of SAS® Help. These tables give descriptions for each name along with any statement or option associated with it. While there are a few tables which list the ODS table names of groups of procedures, SAS® Help does not have all the ODS table names in one place. After searching through SAS® Help, I created a list of all the SAS® procedures with ODS table names (that I could find). That list appears in the Appendix to this paper.³ The documentation does not, however, list the names of the variables in the data vector for each table (at least not that I could find). The field names can only be obtained after an ODS OUTPUT dataset is created.

THAT'S WHAT I EXPECTED

After running a regression model, the programmer wants to use the parameter estimates from that model in a data step to create scores for several other data sets. Instead of retyping the estimates from the output listing, or even cutting and pasting the estimates, the programmer can create a data set which holds all the parameter estimates. But will the data set look like what is found in the listing? The following compares the listing output with the ODS table output:

² This information is not found anywhere in "SAS Help and Documentation". I finally found it described in [SAS Notes: SN-V8+-005025](#).

³ I have created a spreadsheet of all the procedures which can produce ODS output data sets, along with all the ODS table names associated with each procedure. It's not possible to attach the list to this paper. I will show the list at the presentation of this paper, and will make it available to those who request it.

```

C) ods output ParameterEstimates=Parameter_Estimates;
   proc reg data=data_chi_zip_ref;
       score: model
           RSPIND = VR1P21DC VR1P21NC VR3H23EC VR3H30IC VR3P85BC VR3P47AC
           / sle=.01 tol pcorr2 vif;
   run; quit;
   ods output close;

```

The listing output shows:

The REG Procedure
 Model: score
 Dependent Variable: RSPIND

Parameter Estimates

Variable	Label	DF	Parameter Estimate	Standard Error	t Value
Intercept	Intercept	1	0.20328	0.03899	5.21
VR1P21DC	SF1-P21:_%_Households:_Family_45_To_54	1	0.00030220	0.00049727	0.61
VR1P21NC	SF1-P21:_%_Households:_Non-Family_65_To_74	1	0.00111	0.00052170	2.13
VR3H23EC	SF3-H23:_%_Rooms_in_Housing_Units:_5_rooms	1	0.00141	0.00042819	3.30

Parameter Estimates

Variable	Label	DF	Pr > t	Squared Partial Corr Type II
Intercept	Intercept	1	<.0001	.
VR1P21DC	SF1-P21:_%_Households:_Family_45_To_54	1	0.0165	0.00142
VR1P21NC	SF1-P21:_%_Households:_Non-Family_65_To_74	1	0.0018	0.00239
VR3H23EC	SF3-H23:_%_Rooms_in_Housing_Units:_5_rooms	1	<.0001	0.00392

Parameter Estimates

Variable	Label	DF	Tolerance	Variance Inflation
Intercept	Intercept	1	.	0
VR1P21DC	SF1-P21:_%_Households:_Family_45_To_54	1	0.71786	1.39303
VR1P21NC	SF1-P21:_%_Households:_Non-Family_65_To_74	1	0.87735	1.13980
VR3H23EC	SF3-H23:_%_Rooms_in_Housing_Units:_5_rooms	1	0.75725	1.32058

A look at the ODS output table with the ViewTable shows:

	Model	Dependent	Variable	DF	Estimate	StdErr	tValue	Probt	SqPartCorrTypell	Toleranc	VarianceInf	Label
1	score	RSFIND	Intercept	1	0.20328	0.03899	5.21	<.0001	.	.	0	Intercept
2	score	RSFIND	VR1P21DC	1	0.00030220	0.00049727	0.61	0.5435	0.00048445	0.50143	1.99431	SF1-P21:_%_Households:
3	score	RSFIND	VR1P21NC	1	0.00111	0.00052170	2.13	0.0333	0.00593	0.67286	1.48619	SF1-P21:_%_Households:
4	score	RSFIND	VR3H23EC	1	0.00141	0.00042819	3.30	0.0010	0.01406	0.54514	1.83438	SF3-H23:_%_Rooms_in_

The OUTPUT table contains all the information shown in the listing, though in a different order, plus fields indicating the model and dependent variable.

The ODS OUTPUT statement lets you use dataset options (drop=, rename=, where=, etc.). If the only fields you want in your output table are the variable name, the estimate associated with it and its probability, modify the statement as follows:

```
ods output ParameterEstimates=Parameter_Estimates(keep=Variable Estimate
Probt);
```

You can then use the data in the Parameter_Estimates data set to capture the Estimate values for scoring other data sets. A data step to delete those variables whose probabilities aren't significant can be added. In the View Table above, the variable VR1P21DC has a Probt value of 0.5435. The following step can be included:

```
data par_est(drop=Probt);
  set Parameter_Estimates;
  if Probt le 0.05;
run;
```

THAT'S NOT WHAT I EXPECTED

After these experiences with PROC FREQ and PROC REG, I expected a similar outcome with PROC MEANS but didn't get it. The listing output looks like a table:

Variable	N	Mean	Std Dev	Minimum	Maximum
Age	19	13.3157895	1.4926722	11.0000000	16.0000000
Height	19	62.3368421	5.1270752	51.3000000	72.0000000
Weight	19	100.0263158	22.7739335	50.5000000	150.0000000

While I'd expect the ODS data set to have 3 records and 6 variables, it actually has 1 record and 18 variables, one variable for each combination of row and column seen in the listing. A work around was created, and presented at SUGI 31.⁴

A work around was also needed when using PROC COMPARE with ODS output. I wanted the list of variables in two data sets that were not equal. What I expected was a data set similar in structure to this listing:

⁴ See paper 059-31, "A Better Means-ODS Data Trap", by Myra A Oltsik and Peter Crawford.

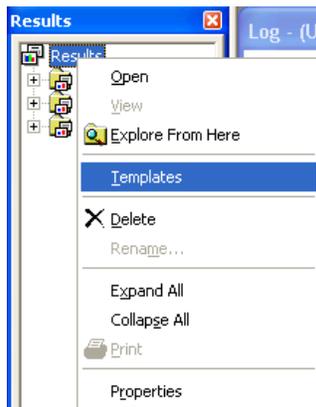
Variables with Unequal Values

Variable	Type	Len	Label	Ndif	MaxDif	MissDif
SUPER_CATEGORY	CHAR	23		33797		0
NAME	CHAR	13	Variable Name	80108		0
LENGTH	NUM	3	Variable Length	52007	5.000	0
VARNUM	NUM	4	Variable Number	79155	13251	0
LABEL	CHAR	123	LABEL OF FORMER VARIABLE	80115		0
MIN	NUM	6		530	5.1E9	0
MAX	NUM	6		11500	5.454E9	0
SUM	NUM	4		73487	50830	3213

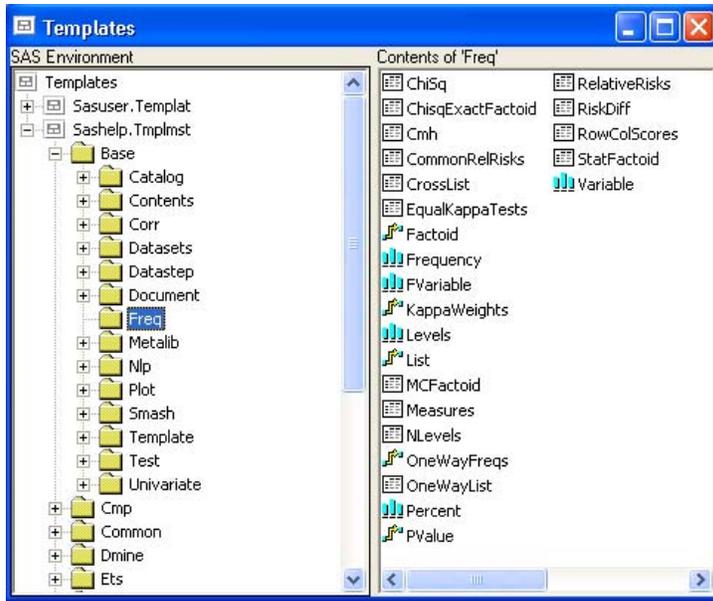
What I got, however, was a dataset with just 2 variables, TYPE and BATCH. TYPE is unexplained, and BATCH is a text capture of each line of the listing output, all in one variable. (See Appendix B) To get the names of the variables with unequal values, I used a data step which searched for the row containing Variable, Type, Len, etc., and then used the SCAN() function to extract the first word of each subsequent line. (The macro for this procedure is also in Appendix B.)

TEMPLATES

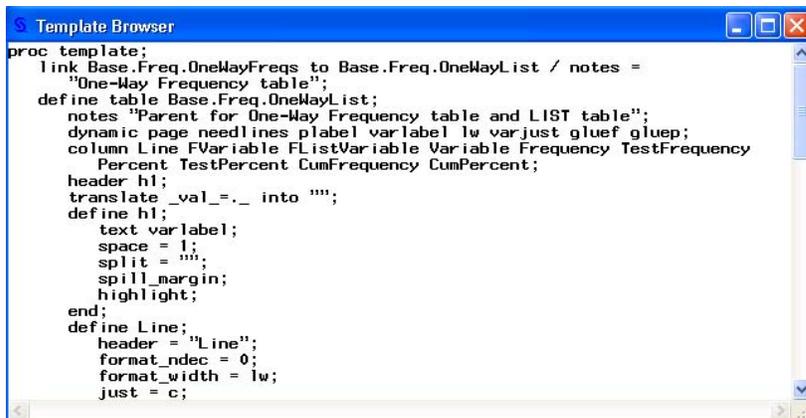
The [ODS TRACE ON](#) output shows the name of the template associated with the table name when one is available. You can look at the template itself. The output of example **A** above for PROC FREQ names the template "Base.Freq.OneWayFreqs". In the Results Window, click on the word "Results", right-click for the menu, and click on "Templates":



This opens up the Template Window. Click on the plus sign (+) next to "Sashelp.Tmplmst" to see one folder for each SAS® module. Since the template name indicates "Base.Freq", click on the plus sign next to "Base", and then click on "Freq". The right side of the window will show all templates associated with PROC FREQ.



In this case, double-clicking on “OneWayFreqs” will open the Template Browser for this specific template:



Though beyond the scope of this paper, you can copy the code and edit it for your own purposes.⁵

Not all output tables have templates. In the next example, the **ODS TRACE ON** output will be put in the listing just before the PROC FREQ results:

```

D) ods trace on / listing;
   proc freq data=test.zip_code_data;
       tables RECORDS * FLAG;
   run;
   ods trace off;

```

⁵ See “SAS Help and Documentation” Contents: SAS Products – Base SAS – Output Delivery System (ODS) – The Template Procedure. There are 7 standard types of templates: Column, Footer, Header, Link, Style, Template Table, and Tree.

```

Output Added:
-----
Name:          CrossTabFreqs
Label:         Cross-Tabular Freq Table
Data Name:
Path:          Freq.Table1.CrossTabFreqs
-----

```

Notice that there is no template name given. In addition, CrossTabFreqs is not included in the Templates Window above.

PERSIST AND MATCH_ALL

PERSIST allows you to gather data from more than one procedure or run of a procedure, when the procedures share an ODS table name. There are two options for PERSIST: PERSIST=PROC or PERSIST=RUN. The former can be used with different procedures, while the latter is cleared at the end of a particular procedure. MATCH_ALL is used to name a macro variable when different output data sets are created for each procedure or run.

The output for both of the models in the following PROC REG will be put into one table, RUN_PAR_EST:

```

E) ods output ParameterEstimates(PERSIST=RUN)=run_par_est;
   proc reg data= data_chi_zip_ref;
       score: model
           RSPIND = VR1P21DC VR1P21NC VR3H23EC VR3H30IC VR3P85BC VR3P47AC;
   run;
       score: model
           RSPIND = VR1P21DC VR1P21NC VR3H23EC VR3H30IC VR3P85BC;
   run;
quit;

```

The data set will be nearly identical to the one produced in example **C** above. An additional field is added to the data set, `_RUN_`, to show with which model the results are associated.

The differences of a model run with PROC REG and PROC GLM can be seen with the following:

```

F) ods output ParameterEstimates(PERSIST=PROC MATCH_ALL=par)=par_est1;
   proc reg data= data_chi_zip_ref;
       score: model
           RSPIND = VR1P21DC VR1P21NC VR3H23EC VR3H30IC VR3P85BC;
   run;

   proc glm data= data_chi_zip_ref;
       score: model
           RSPIND = VR1P21DC VR1P21NC VR3H23EC VR3H30IC VR3P85BC;
   run;
quit;
ods output close;

```

With PERSIST=PROC, you need the last ODS statement to make sure the data set is closed. The output data set for the PROC REG model will get the name PAR_EST1, and the output for the PROC GLM model will get the name PAR_EST2. The data set names are iterated automatically.⁶ Each data set contains an additional variable, _PROC_, noting the associated procedure. If after reviewing each, one wanted to concatenate the two data sets, this simple data step will do so:

```
data par_est;
  set &par;
run;
```

as &par contains both output data set names. Alternatively, if you never wanted to look at the models' tables separately, use the following ODS OUTPUT statement instead, removing the MATCH_ALL option:

```
ods output ParameterEstimates(PERSIST=PROC)=par_est;
```

NAMELEN OPTION

When I first used ODS OUTPUT with PROC LOGISTIC, my variable names were all 8 characters long, so when I went to run PROC LOGISTIC with variable names as long as the SAS allowed 32 characters, I was surprised to find the names truncated to 20 characters. A call to Tech Support pointed me to the NAMELEN option for use in the PROC statement. (The PROCs which use this option are noted in Appendix A with a + sign.) The variable name is not technically outputted, but rather an "effect name". For all but one, the length can be anywhere from 20 to 200. For PROC CATMOD, the length can be between 24 and 200.⁷ Here is a final example:

```
G) ods output ParameterEstimates;
proc glm data= data_chi_zip_ref namelen=32;
  score: model
  RSPIND = p1_vacant_hu_rented_sold_unoccup med3_earn_1999_pop_over_16;
run;
quit;
ods output close;
```

CONCLUSION

The ODS OUTPUT statement allows programmers to capture data output from procedures, but what you get depends upon the procedure being used. The documentation for ODS OUTPUT are limited at best, and scattered around the SAS® Help Documentation. This paper attempts to put in one location all the information you need to start working with ODS OUTPUT tables.

⁶ Had the statement read "ods output ParameterEstimates(PERSIST=PROC MATCH_ALL=par)=par_est;", the output for the PROC REG model would have been PAR_EST, and the output for the PROC GLM model would have been PAR_EST1.

⁷ The explanation in SAS® Help Documentation of the NAMELEN option varies slightly from PROC to PROC, but is essentially the same: "NAMELEN=n specifies the length of effect names in tables and output data sets to be n characters long, where n is a value between 20 and 200 characters. The default length is 20 characters." The definition in the PROC GLMMOD documentation differs the most from the others: "NAMELEN=n specifies the maximum length for an effect name. Effect names are listed in the table of parameter definitions and stored in the EFFNAME variable in the OUTPARM= data set. By default, n=20. You can specify if 20 characters are not enough to distinguish between effects, which may be the case if the model includes a high-order interaction between variables with relatively long, similar names."

CONTACT INFORMATION



Myra A. Oltsik

myra.oltsik@mediamark.com

VP, Analytic Programming and Databases
Market Solutions

Mediamark Research and Intelligence LLC

75 Ninth Ave, FL 5R

New York, NY 10011

www.mediemark.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.

Appendix A

LIST OF PROCs PRODUCING ODS TABLES

PROCEDURE (Module)	PROCEDURE (Module)
ACECLUS (SAS/STAT)	NPAR1WAY (SAS/STAT)
ANOVA + (SAS/STAT)	ORTHOREG (SAS/STAT)
ARIMA (SAS/ETS)	PDLREG (SAS/ETS)
AUTOREG (SAS/ETS)	PHREG (SAS/STAT)
CALENDAR (Base SAS)	PLAN (SAS/STAT)
CALIS (SAS/STAT)	PLOT (Base SAS)
CANCORR (SAS/STAT)	PLS (SAS/STAT)
CANDISC (SAS/STAT)	POWER (SAS/STAT)
CATALOG (Base SAS)	PRINCOMP (SAS/STAT)
CATMOD + (SAS/STAT)	PRINQUAL (SAS/STAT)
CHART (Base SAS)	PROBIT + (SAS/STAT)
CLUSTER (SAS/STAT)	QLIM (SAS/ETS)
COMPARE (Base SAS)	REG (SAS/STAT)
CORR (Base SAS)	REPORT (Base SAS)
CORRESP (SAS/STAT)	ROBUSTREG + (SAS/STAT)
DATASETS and CONTENTS (Base SAS)	RSREG (SAS/STAT)
DISCRIM (SAS/STAT)	SIMLIN (SAS/ETS)
ENTROPY -- Experimental (SAS/ETS)	SPECTRA (SAS/ETS)
FACTOR (SAS/STAT)	SQL (Base SAS)
FASTCLUS (SAS/STAT)	STANDARD (Base SAS)
FREQ (Base SAS and SAS/STAT)	STATSPACE (SAS/ETS)
GAM -- Experimental (SAS/STAT)	STDIZE (SAS/STAT)
GENMOD + (SAS/STAT)	STEPDISC (SAS/STAT)
GLM + (SAS/STAT)	SUMMARY (Base SAS)
GLMMOD + (SAS/STAT)	SURVEYFREQ (SAS/STAT)
GLMPower (SAS/STAT)	SURVEYLOGISTIC + (SAS/STAT)
HPF (SAS High-Performance Forecasting)	SURVEYMEANS (SAS/STAT)
HPFDIAGNOSE (SAS High-Performance Forecasting)	SURVEYREG (SAS/STAT)
INBREED (SAS/Genetics and SAS/STAT)	SURVEYSELECT (SAS/STAT)
KDE (SAS/STAT)	SYSLIN (SAS/ETS)
LATTICE (SAS/STAT)	TABULATE (Base SAS)
LIFEREG + (SAS/STAT)	TIMEPLOT (Base SAS)
LIFETEST (SAS/STAT)	TIMESERIES (SAS/ETS)
LOAN (SAS/ETS)	TPHREG -- Experimental + (SAS/STAT)
LOESS (SAS/STAT)	TPSPLINE (SAS/STAT)
LOGISTIC + (SAS/STAT)	TRANSREG (SAS/STAT)
MDC (SAS/ETS)	TREE (SAS/STAT)
MDS (SAS/STAT)	TSCSREG (SAS/ETS)
MEANS (Base SAS)	TTEST (SAS/STAT)
MI (SAS/STAT)	UCM (SAS/ETS)
MIANALYZE (SAS/STAT)	UNIVARIATE (Base SAS)
MIXED + (SAS/STAT)	VARCLUS (SAS/STAT)
MODECLUS (SAS/STAT)	VARCOMP (SAS/STAT)
MODEL (SAS/ETS)	VARMAX (SAS/ETS)
MULTTEST (SAS/STAT)	X11 (SAS/ETS)
NESTED (SAS/STAT)	X12 (SAS/ETS)
NLIN (SAS/STAT)	
NLMIXED (SAS/STAT)	+ -- NAMELEN= option available in the PROC statement
Nonlinear Optimization Methods (SAS/ETS)	Copyright © 2003 by SAS® Institute Inc., Cary, NC, USA. All rights reserved.


```

/* SEARCH FOR ROW WITH THE NEEDED INFORMATION */
data find_&sasfile(keep = ROW);
  set vars_&sasfile;
  if index(BATCH,"Variable") gt 0 and
     index(BATCH,"Type")      gt 0 and
     index(BATCH,"Len")       gt 0 and
     index(BATCH,"Ndif")      gt 0 and
     index(BATCH,"MaxDif")    gt 0
  ;
  ROW = _n_ + 1;
run;

/* CHECK IF ROW WITH INFORMATION EXISTS */
proc sql noprint;
  select NOBS into : obs
  from sashelp.vtable
  where
    LIBNAME = "WORK" and
    MEMNAME = upcase("find_&sasfile")
  ;
quit;

%if &obs gt 0 %then %do;

  /* NOTE ROW RECORD NUMBER */
  proc sql noprint;
    select ROW into : row
    from find_&sasfile
    ;
  quit;

  /* CREATE A DATASET STARTING AT ROW, DELETING UNNECESSARY ROWS */
  data diff_&sasfile(keep=VAR);
    set vars_&sasfile;
    if _n_ ge &row;
    if BATCH eq "" then delete;
    if index(BATCH,"Comparison") gt 0 and
       index(BATCH,"Method=EXACT") gt 0 and
       index(BATCH,"Variable") gt 0 and
       index(BATCH,"Variables") then delete
    ;

  /* THE FIRST WORD OF EACH RECORD IS THE NAME OF THE VARIABLE COMPARED UNEQUALLY */
  VAR = scan(BATCH,1);
run;

/* OBTAIN INFORMATION ABOUT VARIABLES COMPARED UNEQUALLY */
proc sql noprint;
  select ""||trim(VAR)||"" into : varsdiff
  separated by ", "
  from diff_&sasfile
  ;
  create table dvar_&sasfile as
  select
    MEMNAME,
    NAME,
    TYPE,
    VARNUM,

```

```
        LABEL,  
        FORMAT,  
        LENGTH  
    from sashelp.vcolumn  
    where  
        LIBNAME = "WORK" and  
        MEMNAME = upcase("comp_&sasfile") and  
        NAME in (&varsdiff)  
    ;  
quit;  
  
proc append base=dvar data=dvar_&sasfile force;  
run;  
  
proc datasets lib=work nolist;  
    delete  
        vars_&sasfile  
        find_&sasfile  
    ;  
run; quit;  
  
%end;  
  
%else %do;  
  
    proc datasets lib=work nolist;  
        delete  
            vars_&sasfile  
            comp_&sasfile  
            find_&sasfile  
            diff_&sasfile  
        ;  
    run; quit;  
  
%end;  
  
%end;  
  
%mend;  
  
options mprint;  
%mcompare;
```