BASUG 2025

The (ODS) Output of Your Desires:

Creating Designer Reports and Data Sets

Louise S. Hadden, Independent Consultant

ABSTRACT

SAS® procedures can convey an enormous amount of information – sometimes more information than is needed. Most SAS procedures generate ODS objects behind the scenes. SAS uses these objects with style templates that have custom buckets for certain types of output to produce the output that we see in all destinations (including the SAS listing). By tracing output objects and ODS templates using ODS TRACE (DOM) and by manipulating procedural output and ODS OUTPUT objects, we can pick and choose just the information that we want to see. We can then harness the power of SAS data management and reporting procedures to coalesce the information collected and present the information accurately and attractively.

INTRODUCTION

The Output Delivery System (ODS) delivers what used to be printed output in many convenient forms. What many of us don't realize is that "printed output" from procedures (whether the destination is PDF, RTF, or HTML) is the result of SAS® packaging a collection of items that come out of a procedure that most people want to see in a predefined order (aka template.) With tools such as ODS TRACE, ODS TRACE DOM, ODS OUTPUT, ODS SELECT, ODS EXCLUDE, ODS destinations, SAS metadata, SAS Graphics Editor, and reporting procedures, this paper explores the many buried treasures of procedural output and ODS output objects and demonstrates how to use these objects to get exactly the information that is needed, in exactly the format wanted.

Sample programs will be provided on the author's GitHub page. These programs use a modified copy of the SASHELP.HEART data set provided with SAS software and demonstrate basic concepts relating to ODS OUTPUT. Code snippets from the programs and screenshots of results and logs are provided in the text of the paper and in the presentation to illustrate key concepts.

This presentation is suitable for all levels of proficiency and will be useful for programmers working in all industries. Examples shown were run using SAS 9.4 Maintenance Release 8 on a Windows Server platform, using Display Manager, Enterprise Guide, and batch processing.

GETTING STARTED

There are four basic concepts involved in re-tooling ODS output objects to produce custom reports:

1. Identify / locate your ODS output object(s) using ODS TRACE or output data set(s) .

2. Analyze your ODS output object(s) or output data set(s) using basic SAS procedures and review.

3. Manipulate your ODS output object(s) or output data set(s) using SAS data steps, ODS options, and/or procedures.

4. Report on your final data set(s).

These four concepts are explored below.

CREATING THE TEST DATA SET

SAS provides numerous data sets in the SASHELP folder that are available with 9.4 installations and elsewhere. As noted above, the SASHELP.HEART data set, based on the Framingham Heart Study, started in 1948 in Framingham, MA, was selected to demonstrate the capabilities of ODS OUTPUT. The Framingham Heart Study (FHS) is one of the most influential epidemiological studies in medical history. Its primary goal was to identify common risk factors that contribute to cardiovascular disease (CVD). The

data set, based on the Framingham Heart Study and deidentified, contains demographic and medical data. Additional elements added include a group variable (HeartID), an ID variable (seqnum), Sample Weight, BMI, BMI Category, binaries for female and male, and Obesity Category. Missing variable labels were also added.

```
data out1.heart (label="Enhanced version of SASHELP.HEART");
        set sashelp.heart;

*** create missing variable labels;
        label status="Status"
        sex="Sex"
        height="Height"
        weight="Weight"
        Diastolic="Diastolic Blood Pressure"
        Systolic="Systolic Blood Pressure"
        Smoking="Smoking Status"
        Cholesterol="Cholesterol Status";

*** create a faux group variable;
        HeartID=modz(_n_,4);

*** create a faux id;
        SeqNum=_n_;

*** create a faux weight;
        SampleWeight=1+mod(_n_,10);

*** create BMI variables;
        bmi= weight / height**2 * 703;
        select;
                when (1 le bmi lt 18.5) bmi_cat = 1;
                when (18.5 le bmi lt 25) bmi_cat = 2;
                when (25 le bmi lt 30) bmi_cat = 3;
                when (30 le bmi) bmi_cat = 4;
                otherwise  bmi_cat = .;
        end;
        select;
                when (1 le bmi lt 30) obesity_cat = 1;
                when (30 le bmi lt 35) obesity_cat = 2;
                when (35 le bmi lt 40) obesity_cat = 3;
                when (40 le bmi) obesity_cat = 4;
                otherwise  obesity_cat = .;
        end;
select;
        when (sex='Female') female = 1;
            otherwise  female = 0;
    end;
    select;
        when (sex='Male') male = 1;
            otherwise male = 0;
    end;

label   HeartID="Heart Group Variable"
        SeqNum="Sequence Number"
        bmi="BMI"
        bmi_cat="BMI Category"
        obesity_cat="Obesity Category"
        SampleWeight="Sample Weight"
male="Binary: Male"
female="Binary: Female";
run;
```

Program 1: Creating Analysis File

DISCOVERING ODS OUTPUT OBJECTS

The next step is to identify ODS OUTPUT objects created via SAS procedures. Most SAS procedures generate ODS objects behind the scenes. SAS uses these objects in conjunction with style templates that have custom “buckets” for certain types of output to produce the output we see in all destinations (including the SAS listing).

Use the ODS TRACE command to identify ODS output objects. Note that objects can include graphs and plots as well as “print” data files. Choose the procedure that you are using, and “surround” the procedure (including any ODS GRAPHICS calls, ODS SELECT and ODS EXCLUDE statements, etc. you may be using) with ODS TRACE and ODS TRACE OFF commands. The optional “/ label” below includes the label path for the output object in the record. You can use a label path anywhere that you can use a path.

```
ods trace on / label;
ods rtf file=yourfilename.rtf' path=odsout style=styles.journal2;
ods graphics on;
. . . your procedures here . . .
ods rtf close;
ods graphics off;
ods trace off;
```

Program 2. ODS TRACE Syntax

If you are running interactively, you will see the ODS objects generated in the results window on the left, and can select, view, and save these objects. In some cases, you can manipulate the objects. ODS trace information will be in your SAS log. Optionally, you can write the trace information to the listing output with the “/ listing” command.

Although the ODS objects can be viewed in the results window, you will not get enough information about the objects to truly customize your output, so SAS logs and lists constitute an essential tool for both interactive and batch users.

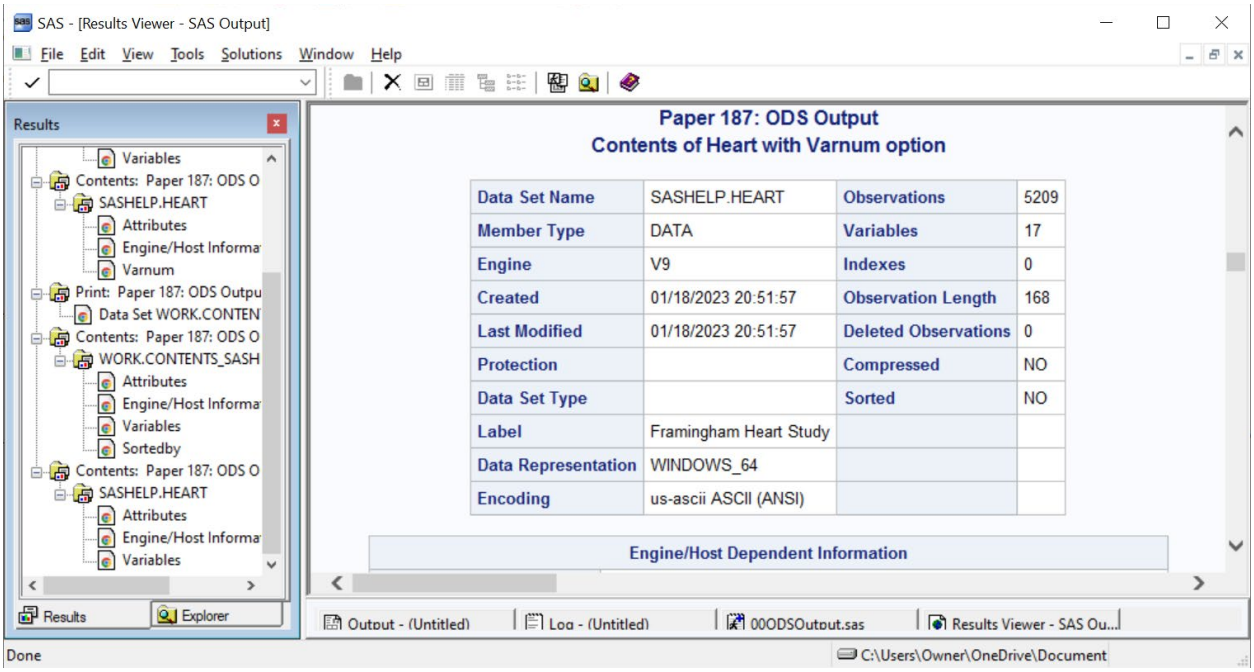


Figure 1. ODS Output Data Sets

If you run in batch mode, output from the ODS TRACE command will automatically be included in your log file, and optionally in your list.

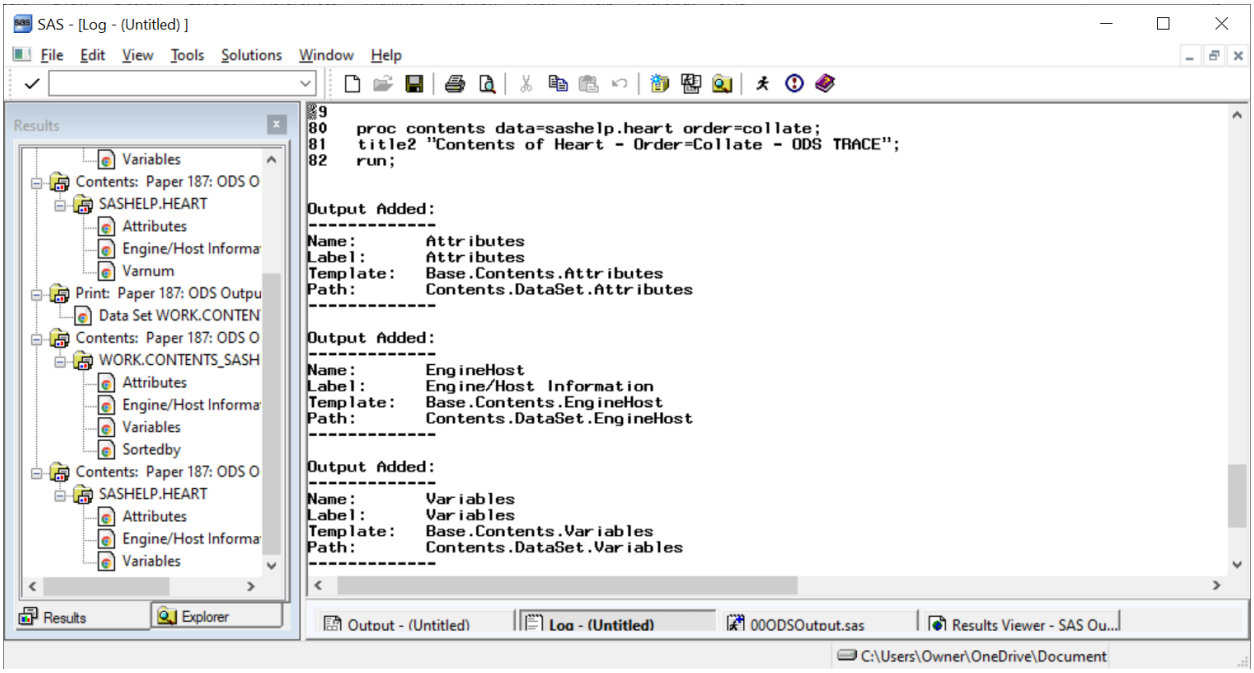


Figure 2: Log displaying ODS TRACE results

Paper 187: ODS Output
Contents of Heart - Order=Collate - ODS TRACE

Data Set Name	SASHELP.HEART	**Observations**	5209
Member Type	DATA	**Variables**	17
Engine	V9	**Indexes**	0
Created	01/18/2023 20:51:57	**Observation Length**	168
Last Modified	01/18/2023 20:51:57	**Deleted Observations**	0
Protection		**Compressed**	NO
Data Set Type		**Sorted**	NO
Label	Framingham Heart Study		
Data Representation	WINDOWS_64		
Encoding	us-ascii ASCII (ANSI)		

Engine/Host Dependent Information	
Data Set Page Size	65536
Number of Data Set Pages	14
First Data Page	1
Max Obs per Page	389
Obs in First Data Page	372
Number of Data Set Repairs	0
ExtendObsCounter	YES
Filename	C:\Program Files\SASHome\SASFoundation\9.4\core\sashelp\heart.sas7bdat
Release Created	9.0401M8
Host Created	X64_SR12R2
Owner Name	BUILTIN\Administrators
File Size	960KB
File Size (bytes)	983040

Figure 3: Listing Output from PROC CONTENTS (collate) – Attributes and Engine Host Information

Alphabetic List of Variables and Attributes				
#	Variable	Type	Len	Label
12	AgeAtDeath	Num	8	Age at Death
5	AgeAtStart	Num	8	Age at Start
3	AgeCHDdiag	Num	8	Age CHD Diagnosed
15	BP_Status	Char	7	Blood Pressure Status
14	Chol_Status	Char	10	Cholesterol Status
13	Cholesterol	Num	8	
2	DeathCause	Char	26	Cause of Death
8	Diastolic	Num	8	
6	Height	Num	8	
10	MRW	Num	8	Metropolitan Relative Weight
4	Sex	Char	6	
11	Smoking	Num	8	
17	Smoking_Status	Char	17	Smoking Status
1	Status	Char	5	
9	Systolic	Num	8	
7	Weight	Num	8	
16	Weight_Status	Char	11	Weight Status

Figure 4. Listing Output from PROC CONTENTS (collate) - Variables

ADDITIONAL NOTES ON THE ODS TRACE STATEMENT

Another option on the ODS TRACE statement is ODS TRACE DOM. DOM stands for the ODS Document Object Manager, which provides background information to ODS about document style selectors, etc. As with other ODS output objects, ODS TRACE DOM provides information in the log about ODS style elements which can be especially useful for controlling and enhancing ODS output. Options apart from DOM, LISTING, and LABEL include the EXCLUDED statement which writes information about objects excluded by the ODS EXCLUDE statement to the log or list.

PROCEDURAL OPTIONS EFFECTS

Most procedures, even the humble PROC CONTENTS, have options that direct the output that is created. PROC CONTENTS by default creates output sorted by variable name. The default is order by collate – i.e., alphabetical order, which can also vary by the collating sequence used by the system. We see from the figures above that PROC CONTENTS ORDER=COLLATE produces three output objects: Attributes, EngineHost, and Variables. What happens when we put ODS TRACE around PROC CONTENTS ORDER=VARNUM?

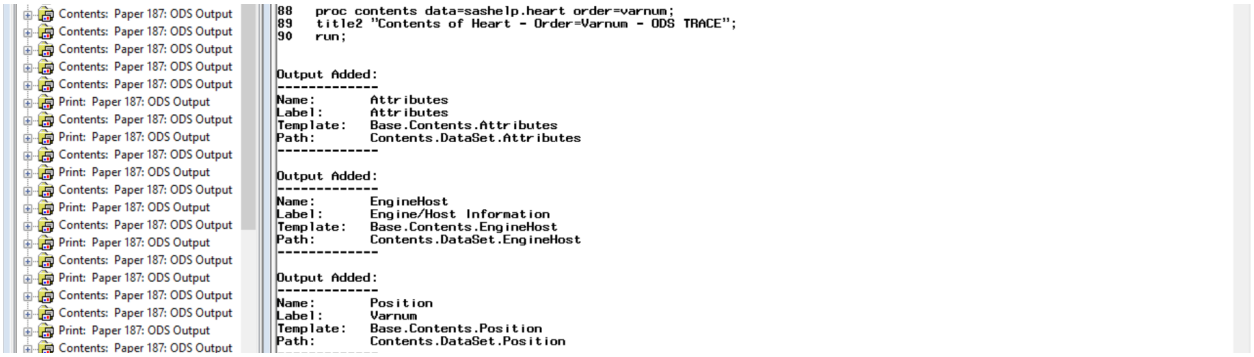


Figure 5. PROC CONTENTS (VARNUM) TRACE

The first two ODS OBJECTS are the same, but the third is different.

#	Variable	Type	Len	Label
1	Status	Char	5	
2	DeathCause	Char	26	Cause of Death
3	AgeCHDdiag	Num	8	Age CHD Diagnosed
4	Sex	Char	6	
5	AgeAtStart	Num	8	Age at Start
6	Height	Num	8	
7	Weight	Num	8	
8	Diastolic	Num	8	
9	Systolic	Num	8	
10	MRW	Num	8	Metropolitan Relative Weight
11	Smoking	Num	8	
12	AgeAtDeath	Num	8	Age at Death
13	Cholesterol	Num	8	
14	Chol_Status	Char	10	Cholesterol Status
15	BP_Status	Char	7	Blood Pressure Status
16	Weight_Status	Char	11	Weight Status
17	Smoking_Status	Char	17	Smoking Status

Figure 6. Contents Listing (VARNUM) – Position

Some procedures, particularly statistical procedures, have a multitude of ODS Output Objects for every variation from procedural options, including ODS graphics. Each object has a name, label, template, and path. These can be utilized in post-processing the ODS Output Objects.

ADDITIONAL EFFECTS

Sometimes other factors related to data sets have effects on ODS Output Objects. PROC CONTENTS (and its influential friends PROC DATASETS, Dictionary Tables, and Dictionary Views) document everything about a data set. For example, if a data set is sorted or indexed, that is either included in existing PROC CONTENTS ODS Output Objects or in a separate ODS Output Object. We sort the analysis file by seqnum and rerun our PROC CONTENTS inside an ODS TRACE sandwich.

```
Output Added:
-------------
Name:        Sortedby
Label:       Sortedby
Template:    Base.Contents.Sortedby
Path:        Contents.DataSet.Sortedby
-------------
```

Figure 7. Sortedby ODS Output Object in ODS TRACE log output

What is present in the PROC CONTENTS listing output is reflected in the ODS Output Objects. There is both a difference in the PROC CONTENTS listing (in the Attributes section and ODS Output Object) and the addition of printed Sort Information and the Sortedby ODS Output Object.

Paper 187: ODS Output
Contents on Corrected Data Set - Sorted

Data Set Name	OUT1.HEART	Observations	5209
Member Type	DATA	Variables	25
Engine	V9	Indexes	0
Created	05/22/2025 22:10:21	Observation Length	232
Last Modified	05/22/2025 22:10:21	Deleted Observations	0
Protection		Compressed	NO
Data Set Type		Sorted	YES
Label	Enhanced version of SASHELP.HEART		
Data Representation	WINDOWS_64		
Encoding	wlatin1 Western (Windows)		

Figure 8. PROC CONTENTS LISTING of Sorted Data Set – Attributes

Sort Information	
Sortedby	SeqNum
Validated	YES
Character Set	ANSI

Figure 9. PROC CONTENTS SORTEDBY ODS OUTPUT OBJECT

ODS GRAPHICS AS ODS OUTPUT OBJECTS

Graphic images created by SAS were the original output objects, sent to “devices” which were once actual printers. Just as PROC PRINT or PROC REPORT produce output objects, ODS GRAPHICS does as well. SAS provides the template a graphic was created in, and the path for the graphic object. This allows users to save, post process, and use a graphic image if they desire. To edit a file with the SAS Graphics Editor (SGE) use the following statement prior to running graphics producing procedures: ODS LISTING SGE=ON;. This creates an editable (using the ODS graphics editor) file which allows you to change titles, background colors, etc. in your graphs which you can then save.

```
ods graphics on;
ods select Plots SSPlots;
proc univariate data=heartplots plot;
   by sex;
   var bmi;
title2 "PROC UNIVARIATE PLOTS";
run;
ods graphics off;
ods select all;
```

Program 3: PROC UNIVARIATE with plots and ODS SELECT

```
Output Added:
-------------
Name:         Plots
Label:        Plots for bmi
Template:     base.univariate.Graphics.Plots
Path:         Univariate.ByGroup1.bmi.Plots
-------------
NOTE: The above message was for the following BY group:
      Sex=Female
```

Figure 10: ODS TRACE of PROC UNIVARIATE with plots and a by statement

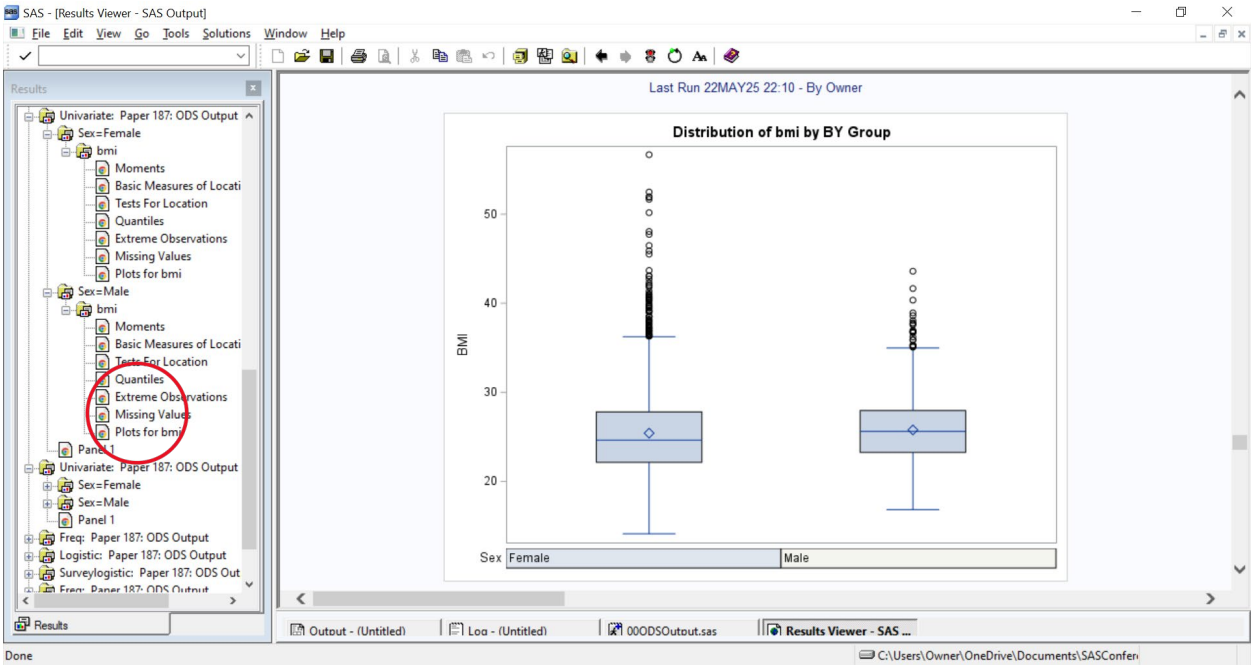


Figure 11. Graphic Output from PROC UNIVARIATE

EXPLORING AND OUTPUTTING ODS OUTPUT OBJECTS

Once we know what ODS OUTPUT OBJECTS (or OOOs) have been created by our programs, we can explore them. In Display Manager it is possible to click on objects in the results window to the left, which opens the object, and get some idea of what has been created, and even save those temporary ODS Output objects from the window, but in order to do some adaptive reuse on the OOOs you need more. That is where ODS OUTPUT comes in. Going back to PROC CONTENTS, we generate work or permanent data sets from the temporary ODS OUTPUT OBJECTS. Using the object names gleaned from ODS TRACE, we can write an ODS OUTPUT statement, outputting attributes1, enginehost1, and variables1.

```
ods output attributes=attributes1 enginehost=enginehost1 variables=variables1;

proc contents data=sashelp.heart order=collate;
title2 "Collate contents with ODS OUTPUT objects";
run;

ods output close;

proc contents data=attributes1;
title2 "Contents of Attributes ODS Output (Collate)";
run;

proc print data=attributes1 (obs=5) noobs;
title2 "Test Print Attributes ODS Output (Collate)";
run;
```

Program 4. ODS OUTPUT OBJECT exploration

Then, we can perform some discovery on those saved data sets: I typically code a variable listing including variable type, and a test print of a few observations.

Alphabetic List of Variables and Attributes

#	Variable	Type	Len	Format
2	Label1	Char	20	
5	Label2	Char	21	
1	Member	Char	256	
3	cValue1	Char	22	
6	cValue2	Char	4	
4	nValue1	Num	8	D12.3
7	nValue2	Num	8	D12.3

Figure 12. Contents of COLLATE Contents Attributes Data Set

Paper 187: ODS Output
Test Print Attributes ODS Output (Collate)

Member	Label1	cValue1	nValue1	Label2	cValue2	nValue2
SASHELP.HEART	Data Set Name	SASHELP.HEART	.	Observations	5209	5209.000000
SASHELP.HEART	Member Type	DATA	.	Variables	17	17.000000
SASHELP.HEART	Engine	V9	.	Indexes	0	0
SASHELP.HEART	Created	01/18/2023 20:51:57	1989694317	Observation Length	168	168.000000
SASHELP.HEART	Last Modified	01/18/2023 20:51:57	1989694317	Deleted Observations	0	0

Figure 13. Test Print of COLLATE Contents Attribute Data Set

The variable names in the contents are uninformative, and the test print barely resembles the actual listing output. SAS has templates it uses to put the stored output objects into. When we think of linking datasets, we look to match on a key and that is not possible here. However, with a little effort and mapping, you can translate the ODS output objects into useful tools.

Also note that in some cases SAS provides both a numeric and character version of variables – this can be handy if your client has specific formatting requirements that do not fit the norm.

As an example, I am frequently given an empty table shell to fill with the results from PROC LOGISTIC. Often the statistics required from multiple sources (ODS Output Objects). Odds ratios appear in a different OOO than parameter estimates. In this case, I used the parameter estimates and odds ratio output to construct a single data set with all the statistics the analyst wanted. Note that to merge the two data sets, I needed to remove the intercept line from the parameter estimates. The reasoning behind the test prints and PROC CONTENTS outputs becomes clear. The descriptive data set may include longer variable descriptions you may not want to carry in a large data set and variables used for formatting (for example, shading, italicizing, or bolding.) You can also create macro variables for printing from some ODS output objects using data steps and call symput or PROC SQL INTO: and use in titles or column headers.

ADDITIONAL NOTES ON THE ODS OUTPUT STATEMENT

Options for the ODS OUTPUT statement include ODS SELECT and ODS EXCLUDE which allow you to choose elements that are sent to ODS destinations. SAS has default behaviors for ODS Outputs (all items are EXCLUDED unless overwritten in a ODS OUTPUT statement) and other destinations (all items are SELECTED unless otherwise specified.) If you want to see the defaults, use the ODS SHOW

statement, which provides a selection list in the log. It is possible to customize all your ODS output by maintaining your own selection list(s).

MANIPULATING ODS OUTPUT OBJECTS

Reviewing all the output above allows us to choose statistics from various data sets and merge it with a descriptive data set for printing. The possibilities are endless. In this case, I used parameter estimates and odds ratio output to construct a single data set with all the statistics the analyst wanted. Note that to merge the two data sets, I needed to remove the intercept line from the parameter estimates. The reasoning behind the test prints and PROC CONTENTS outputs becomes clear. The descriptive data set may include longer variable descriptions you may not want to carry in a large data set and variables used for formatting (for example, shading, italicizing, or bolding.) You can also create macro variables for printing from some ODS output objects using data steps and call symput or PROC SQL INTO:.

ODS OUTPUT OBJECTS BY EXAMPLE

Below follow two examples of how ODS output objects can be manipulated to create custom reports.

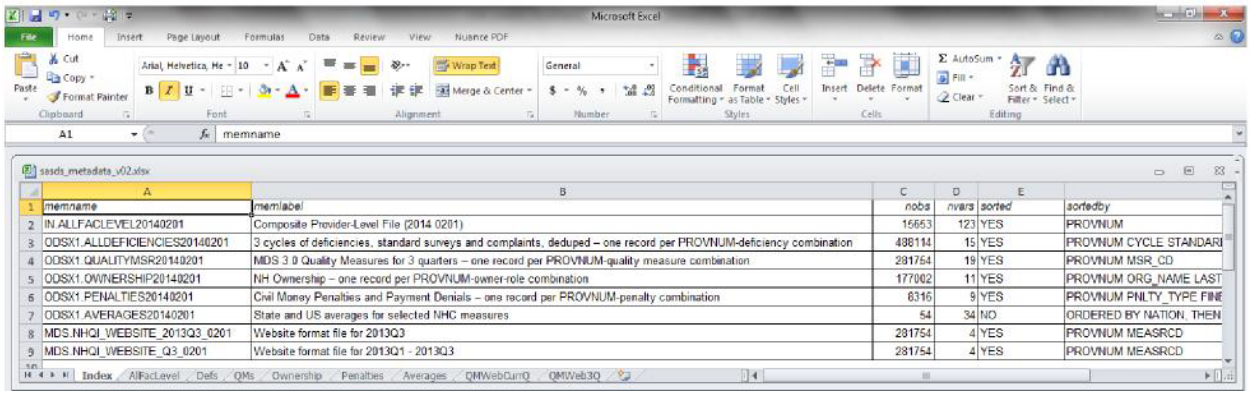
CREATE A MULTI-TABBED METADATA SPREADSHEET

We wished to create an Excel workbook with an index tab with basic information about all data sets being delivered and documented for a given month in each domain (files to be uploaded to DATA.MEDICARE.GOV), plus tabs for each data set with selected details on variables. In this case, we used the CONTENTS procedure in conjunction with ODS output objects. PROC CONTENTS does create an output data set (OUT=); however, it does not contain one desired piece of information, the variables that a data set are sorted by (if any). It is also rectangular, meaning that many variables on a system level are included on every record. ODS RTF or ODS HTML calls sandwiched around PROC CONTENTS also yields a lot of unnecessary information and would require a lot of post-editing. A macro was designed that produces PROC CONTENTS for each data set, outputs ODS output objects, manipulates the objects and outputs two temporary data sets (a header line for the index tab, and a variable listing) for each data set.

After running ODS TRACE on the PROC CONTENTS statement, we reviewed our log and identified all the output objects generated by the contents procedure and their system names. Note that the output objects may vary depending on the procedure AND procedural options – and in some cases, the data! For example, SAS/STAT®'s PROC SURVEYFREQ produces different output objects based on whether you are running a one-way table or crosstabs and may NOT generate chi-squared statistics if there are any 0 cells in a crosstab. In our case, if a data set is not sorted, PROC CONTENTS will not generate ODS output objects relating to sorting (SORTEDBY). If you want to report on sorting, you may need to have conditional processing that will indicate that a dataset is not sorted if there are no ODS output objects related to sorting. In the case of PROC CONTENTS, you will not get the variable "informat" in the VARIABLES output object if no variables in the data set have informats.

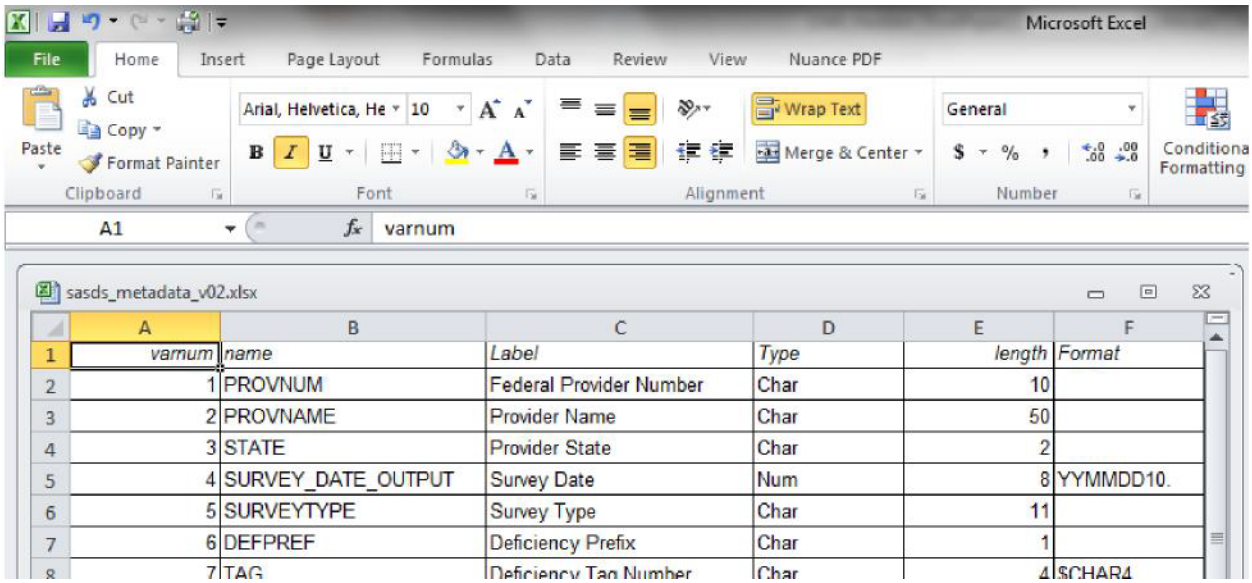
The ATTRIBUTES, VARIABLES, and SORTEDBY output objects were used to construct two designer data sets for each data set to be documented. Note that because one of the input data sets is not sorted, conditional macro processing was used to create SORTEDBY if it does not exist.

Once the header or index file and individual data set variable listings have been built, a multi-tabbed Excel workbook is created, via ODS.TAGSETS.EXCELXP pre-Version 9.4 M4 and the ODS EXCEL destination for M4 and later. ODS gives more control over formatting than a simple PROC EXPORT. We can name our tabs and set column widths within SAS for individual worksheets using options: sheet_name names each tab, and Absolute_Column_Width sets the width for each column. We set and unset the sheet interval as table to generate a tab for each report.



	memname	memlabel	nobs	nvars	sorted	sortedby
2	IN.ALLFACLEVEL20140201	Composite Provider-Level File (2014 0201)	15663	123	YES	PROVNUM
3	ODSX1.ALLDEFICIENCIES20140201	3 cycles of deficiencies, standard surveys and complaints, deduped – one record per PROVNUM-deficiency combination	498114	15	YES	PROVNUM CYCLE STANDARI
4	ODSX1.QUALITYMSR20140201	MDS 3.0 Quality Measures for 3 quarters – one record per PROVNUM-quality measure combination	281754	19	YES	PROVNUM MSR_CD
5	ODSX1.OWNERSHIP20140201	NH Ownership – one record per PROVNUM-owner-role combination	177002	11	YES	PROVNUM ORG_NAME LAST
6	ODSX1.PENALTIES20140201	Civil Money Penalties and Payment Denials – one record per PROVNUM-penalty combination	6316	9	YES	PROVNUM PNLTY_TYPE FINE
7	ODSX1.AVERAGES20140201	State and US averages for selected NHC measures	54	34	NO	ORDERED BY NATION, THEN
8	MDS.NHQI_WEBSITE_2013Q3_0201	Website format file for 2013Q3	281754	4	YES	PROVNUM MEASRCD
9	MDS.NHQI_WEBSITE_Q3_0201	Website format file for 2013Q1 - 2013Q3	281754	4	YES	PROVNUM MEASRCD

Figure 14. Generation of Excel Documentation – Index Tab



	varnum	name	Label	Type	length	Format
2	1	PROVNUM	Federal Provider Number	Char	10	
3	2	PROVNAME	Provider Name	Char	50	
4	3	STATE	Provider State	Char	2	
5	4	SURVEY_DATE_OUTPUT	Survey Date	Num	8	YYMMDD10.
6	5	SURVEYTYPE	Survey Type	Char	11	
7	6	DEFPREF	Deficiency Prefix	Char	1	
8	7	TAG	Deficiency Tag Number	Char	4	$CHAR4

Figure 15. Generation of Excel Documentation – File Documentation tab

CREATE A TABLE FOR A JOURNAL ARTICLE

The example shown below manipulates ODS output objects including Chi Squared statistics as well as frequency output from the PROC SURVEYFREQ procedure run on a large complex data set. No post-processing was required.

Table 1. Selected Characteristics by BMI Category

Characteristic	*Total 2005*	*Total 2005*	*Chi Squared P-Value*	*Underweight (BMI<18.5)*	*Normal Weight (BMI 18.5-24.9)*	*Overweight (BMI 25-29.9)*	*Obese (BMI 30+)*
Total Sample	**15,195 (946,008)**	**100.0±0.00**	**NA**	**1.2±0.14**	**38.3±0.82**	**47.6±0.68**	**12.9±0.55**
Sex							
Men	11,395 (804,888)	85.1±0.71	<.0001	0.9±0.16	35.2±0.94	50.2±0.79	13.7±0.61
Women	3,800 (141,120)	14.9±0.71	<.0001	2.8±0.30	56.1±1.24	32.9±1.01	8.2±0.63
Age							
17-20	1,187 (130,680)	13.8±0.99	<.0001	1.6±0.40	53.8±1.95	37.9±1.78	6.7±1.35
21-30	6,180 (481,590)	50.9±1.21	<.0001	1.5±0.21	42.4±0.98	44.7±0.96	11.4±0.58
31-39	4,610 (230,112)	24.3±1.05	<.0001	0.6±0.17	26.9±1.22	54.8±0.91	17.7±0.85
40+	3,218 (103,627)	11.0±0.75	<.0001	0.7±0.20	25.2±1.15	57.2±1.22	16.9±0.98
Educational Attainment							
High School or Less	4,409 (352,500)	37.3±1.56	<.0001	1.1±0.26	43.9±1.26	41.8±0.95	13.3±0.93
Some College	6,146 (380,144)	40.2±1.26	<.0001	1.6±0.20	34.5±0.80	49.7±0.79	14.2±0.67

Figure 16. Table 1 Generation

CONCLUSION

The examples shown above are simplistic, but the basic concepts remain the same whether you are doing a complex weighted multivariate analysis or a simple PROC FREQ. Using ODS output objects and output data sets to create highly customized output can be an enormous time saver. With ODS output objects / output data sets and SAS, the sky is the limit!

REFERENCES

Dawber, T. R., Meadors, G. F., & Moore, F. E. (1951). Epidemiological approaches to heart disease: The Framingham Study. *American Journal of Public Health and the Nation's Health*, 41(3), 279–286. https://doi.org/10.2105/AJPH.41.3.279

Hadden, Louise, 2014. "Build your Metadata with PROC CONTENTS and ODS OUTPUT", Proceedings of SAS Global Forum 2014.

Hadden, Louise. "With a Trace: Making Procedural Output and ODS Output Objects Work for You". Proceedings of SAS Global Forum 2013 Conference. May 2013.

Hadden, Louise. "Working with ODS TRACE Statement and ODS Output Objects", Proceedings of SAS Global Forum 2019.

Hughes, Troy Martin and Hadden, Louise. "DOMinate your ODS Output with PROC TEMPLATE, ODS Cascading Style Sheets (CCS), and the ODS Document Object Model (DOM) by Mastering ODS TRACE DOM, Parsing CSS with a PROC TEMPLATE Front End, and Overriding ODS Style Inheritance with Customized Styles". Proceedings of PharmaSUG 2019. June 2019.

Kalt, Mike and Zender, Cynthia. "Introduction to ODS Graphics for the Non-Statistician". Proceedings of SAS Global Forum 2011 Conference. April 2011.

Raithel, Michael A., 2011. "PROC DATASETS: the Swiss Army Knife of SAS® Procedures." Proceedings of SAS Global Forum 2011.

Williams, Christianna. "Any WAY you Want it: Getting the Right TYPEs of Observations out of PROC SUMMARY or MEANS." Proceedings of SAS Global Forum 2008 Conference. April 2008.

Zender, Cynthia. "The Greatest Hits: ODS Essentials Every User Should Know". Proceedings of SAS Global Forum 2011 Conference. April 2011.

CONTACT INFORMATION

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

Louise S. Hadden saslouisehadden@gmail.com

Full code examples are available upon request.