

A Few Practical Uses for Data from the SASHELP Views

Presenter:

Victor Pontes

Inatec, Inc.

vpontes@inatecinc.com

Overview

- The SASHELP views are read-only datasets containing meta-data
- They can be read in like data in any other dataset in a library
- They contain session specific data
- Old School: Dictionary Tables (SQL only)
- This presentation will cover a very small fraction of their potential uses

In SAS v9.2:

- 29 views containing session data
- Assorted other useful tables (zip code lists, currency abbreviations, sample data)
- Browse the SASHELP library to take a look

In this presentation, we will look at some uses of only four of them:

- VCOLUMN – Information about variables
- VTABLE – Information about datasets
- VOPTIONS – Information about SAS system options settings
- VEXTFIL – Information on external files

Ex 1: Variables

- SASHELP.VCOLUMN contains data describing variables in datasets
- Roughly equivalent to an output dataset from proc contents
- Use to determine variable length, type, format

Contents of SASHELP.VCOLUMN

#	Variable	Type	Len	Flags	Label
1	libname	Char	8	P--	Library Name
2	memname	Char	32	P--	Member Name
3	memtype	Char	8	P--	Member Type
4	name	Char	32	P--	Column Name
5	type	Char	4	P--	Column Type
6	length	Num	8	P--	Column Length
7	npos	Num	8	P--	Column Position
8	varnum	Num	8	P--	Column Number in Table
9	label	Char	256	P--	Column Label
10	format	Char	49	P--	Column Format
11	informat	Char	49	P--	Column Informat
12	idxusage	Char	9	P--	Column Index Type
13	sortedby	Num	8	P--	Order in Key Sequence
14	xtype	Char	12	P--	Extended Type
15	notnull	Char	3	P--	Not NULL?
16	precision	Num	8	P--	Precision
17	scale	Num	8	P--	Scale
18	transcode	Char	3	P--	Transcoded?

Suppose we want to handle variable name overlap
between two datasets that we plan to merge

Two datasets with overlapping names

```
data pre;  
  input id bmi wt;  
  cards;  
1 20 150  
2 30 180  
;  
run;
```

```
data post;  
  input id bmi wt;  
  cards;  
1 17 140  
2 20 150  
;  
run;
```

SAS - [VIEWTABLE: Sashelp.Vcolumn]

File Edit View Tools Data Solutions Window Help

✓

Explorer

Contents of 'Sashelp'

Table Templt

Treport Vallopt

	Library Name	Member Name	Member Type	Column Name	Column Type	Column Length	Column Position	Column Number in Table	Column Label
742	WORK	POST	DATA	id	num	8	0	1	
743	WORK	POST	DATA	bmi	num	8	8	2	
744	WORK	POST	DATA	wt	num	8	16	3	
745	WORK	PRE	DATA	id	num	8	0	1	
746	WORK	PRE	DATA	bmi	num	8	8	2	
747	WORK	PRE	DATA	wt	num	8	16	3	

```

proc sort data = sashelp.vcolumn out = workvars (keep = memname name);
    by name;
    where libname = "WORK";
run;

data _null_;
    length renbuild $1000.;
    retain renbuild ' ' ;
    merge workvars (where = (memname = "PRE" and name ne 'id') in = inpre)
           workvars (where = (memname = "POST" and name ne 'id') in = inpost)
    ;
    by name;
    if (inpre and inpost);
    renbuild = trim(renbuild) ||" rename = "||compress("("||name||"="||name||"_2)");
    call symput("rens",renbuild);
run;

options mprint symbolgen;

data prepost;
    merge pre post (&rens);
    by id;
run;

```

Dataset
WORKVARS:

1 PRE BMI

2 PRE ID

3 PRE WT

4 POST BMI

5 POST ID

6 POST WT

NOTE: There were 2 observations read from the data set WORK.WORKVARS.

WHERE (memname='PRE') and (name not = 'id');

NOTE: There were 2 observations read from the data set WORK.WORKVARS.

WHERE (memname='POST') and (name not = 'id');

NOTE: DATA statement used (Total process time):

real time 0.01 seconds

cpu time 0.01 seconds

312 data prepost;

313 merge pre post (&rens);

**SYMBOLGEN: Macro variable RENS resolves to
rename = (bmi=bmi_2) rename = (wt=wt_2)**

314 by id;

315 run;

NOTE: There were 2 observations read from the data set WORK.PRE.

NOTE: There were 2 observations read from the data set WORK.POST.

NOTE: The data set WORK.PREPOST has 2 observations and 5 variables.

NOTE: DATA statement used (Total process time):

real time 0.01 seconds

cpu time 0.01 seconds

Use to Populate an Array:

Lets say we need to recode all numeric variables in a dataset

```
proc sql noprint;
  select name into :numvars separated by ' '
  from sashelp.vcolumn
  where libname = "Y2008"
  and memname = "QX"
  and type = "num"
  and name ^= 'ID'
  ;
quit;

data recode;
  set y2008.qx;

  array num_vars &numvars;

  do i = 1 to dim(numvars);
    ....;
  end;
run;
```

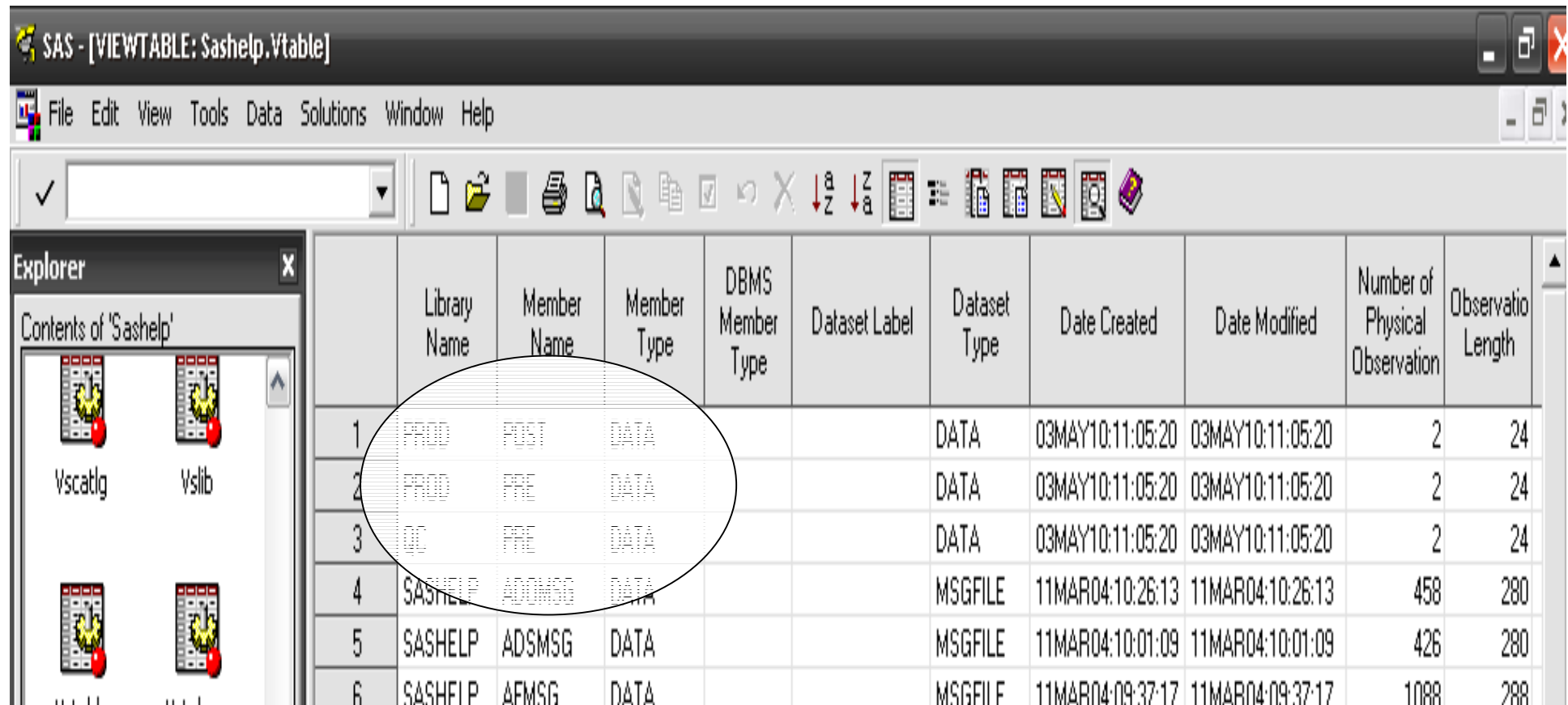
Ex 2: Datasets

- SASHELP.VTABLES
- Use it to compare datasets, libraries
- Roughly equivalent to output of proc datasets

Contents of SASHELP.VTABLE

#	Variable	Type	Len	Flags	Format	Informat	Label
1	libname	Char	8	P--			Library Name
2	memname	Char	32	P--			Member Name
3	memtype	Char	8	P--			Member Type
4	dbms_memtype	Char	32	P--			DBMS Member Type
5	memlabel	Char	256	P--			Dataset Label
6	typemem	Char	8	P--			Dataset Type
7	crdate	Num	8	P--	DATETIME.	DATETIME.	Date Created
8	modate	Num	8	P--	DATETIME.	DATETIME.	Date Modified
9	nobs	Num	8	P--			Number of Physical Observations
10	obslen	Num	8	P--			Observation Length
11	nvar	Num	8	P--			Number of Variables
12	protect	Char	3	P--			Type of Password Protection
13	compress	Char	8	P--			Compression Routine
14	encrypt	Char	8	P--			Encryption
15	npage	Num	8	P--			Number of Pages
16	filesize	Num	8	P--			Size of File
17	pcompress	Num	8	P--			Percent Compression
18	reuse	Char	3	P--			Reuse Space
19	bufsize	Num	8	P--			Bufsize
20	delobs	Num	8	P--			Number of Deleted Observations
21	nlobs	Num	8	P--			Number of Logical Observations
22	maxvar	Num	8	P--			Longest variable name
23	maxlabel	Num	8	P--			Longest label
24	maxgen	Num	8	P--			Maximum number of generations
25	gen	Num	8	P--			Generation number
26	attr	Char	3	P--			Dataset Attributes
27	indxtype	Char	9	P--			Type of Indexes
28	datarep	Char	32	P--			Data Representation
29	sortname	Char	8	P--			Name of Collating Sequence
30	sorttype	Char	4	P--			Sorting Type
31	sortchar	Char	8	P--			Charset Sorted By
32	reqvector	Char	24	P--	\$HEX48.	\$HEX48.	Requirements Vector
33	datarepname	Char	170	P--			Data Representation Name
34	encoding	Char	256	P--			Data Encoding
35	audit	Char	3	P--			Audit Trail Active?
36	audit_before	Char	3	P--			Audit Before Image?
37	audit_admin	Char	3	P--			Audit Admin Image?
38	audit_error	Char	3	P--			Audit Error Image?
39	audit_data	Char	3	P--			Audit Data Image?

Compare Libraries:
Suppose we want to throw a warning if PROD and QC
libraries don't have matching datasets



SAS - [VIEWTABLE: Sashelp.Vtable]

File Edit View Tools Data Solutions Window Help

✓

Explorer

Contents of 'Sashelp'

Vscatlg Vslib

	Library Name	Member Name	Member Type	DBMS Member Type	Dataset Label	Dataset Type	Date Created	Date Modified	Number of Physical Observation	Observation Length
1	PROD	POST	DATA			DATA	03MAY10:11:05:20	03MAY10:11:05:20	2	24
2	PROD	PRE	DATA			DATA	03MAY10:11:05:20	03MAY10:11:05:20	2	24
3	QC	PRE	DATA			DATA	03MAY10:11:05:20	03MAY10:11:05:20	2	24
4	SASHELP	ADSMMSG	DATA			MSGFILE	11MAR04:10:26:13	11MAR04:10:26:13	458	280
5	SASHELP	ADSMMSG	DATA			MSGFILE	11MAR04:10:01:09	11MAR04:10:01:09	426	280
6	SASHELP	AFMSG	DATA			MSGFILE	11MAR04:09:37:17	11MAR04:09:37:17	1088	288

```
proc sort data = sashelp.vtable out = prod_ds (keep = memname);  
  by memname;  
  where libname = "PROD";  
run;
```

PROD_DS:

1	POST
2	PRE

```
proc sort data = sashelp.vtable out = qc_ds (keep = memname);  
  by memname;  
  where libname = "QC";  
run;
```

QC_DS:

1	PRE
---	-----

```
data chkds;  
  merge prod_ds (in = p) qc_ds (in = q);  
  by memname;  
  if p and not(q) then put "WARNING: Dataset " memname " in PROD but not QC.";  
  else if q and not(p) then put "WARNING: Dataset " memname " in QC but not PROD.";  
run;
```

```
387
388 data chkds;
389     merge prod_ds (in = p) qc_ds (in = q);
390     by memname;
391     if p and not(q) then put "WARNING: Dataset " memname " in PROD but not QC.";
392     else if q and not(p) then put "WARNING: Dataset " memname " in QC but not PROD.";
393 run;
```

WARNING: Dataset POST in PROD but not QC.

NOTE: There were 2 observations read from the data set WORK.PROD_DS.

NOTE: There were 1 observations read from the data set WORK.QC_DS.

NOTE: The data set WORK.CHKDS has 2 observations and 39 variables.

NOTE: DATA statement used (Total process time):

real time	0.04 seconds
cpu time	0.00 seconds

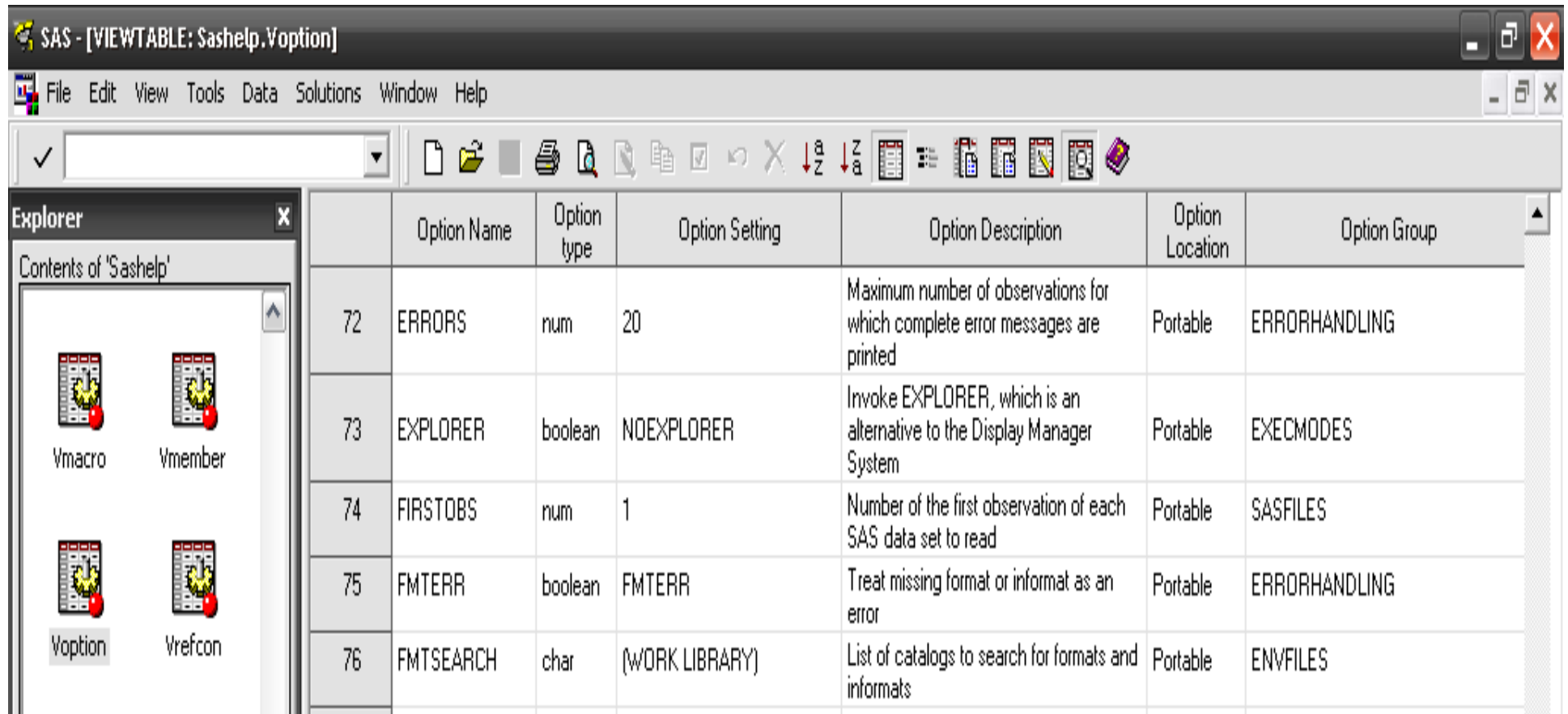
Ex 3: System Options

- SASHELP.VOPTION
- Use to get current state of any SAS option
- Roughly equivalent to GETOPTION()
function

Contents of SASHELP.VOPTION

#	Variable	Type	Len	Flags	Label
1	optname	Char	32	P--	Option Name
2	opttype	Char	8	P--	Option type
3	setting	Char	1024	P--	Option Setting
4	optdesc	Char	160	P--	Option Description
5	level	Char	8	P--	Option Location
6	group	Char	32	P--	Option Group

Suppose we want to temporarily set options, then restore the options to their original state



The screenshot shows the SAS VIEWTABLE: Sashelp.Voption window. The window has a menu bar (File, Edit, View, Tools, Data, Solutions, Window, Help) and a toolbar. On the left, there is an Explorer pane titled 'Contents of 'Sashelp'' showing icons for Vmacro, Vmember, Voption, and Vrefcon. The main area displays a table of SAS options.

	Option Name	Option type	Option Setting	Option Description	Option Location	Option Group
72	ERRORS	num	20	Maximum number of observations for which complete error messages are printed	Portable	ERRORHANDLING
73	EXPLORER	boolean	NOEXPLORER	Invoke EXPLORER, which is an alternative to the Display Manager System	Portable	EXECMODES
74	FIRSTOBS	num	1	Number of the first observation of each SAS data set to read	Portable	SASFILES
75	FMterr	boolean	FMterr	Treat missing format or informat as an error	Portable	ERRORHANDLING
76	FMTSEARCH	char	(WORK LIBRARY)	List of catalogs to search for formats and informats	Portable	ENVFILES

```
proc sql noprint;  
    select setting into :opts separated by ' '  
    from sashelp.voption  
    where group = 'MACRO' and opttype = 'boolean' and setting ne "MACRO"  
    ;  
quit;  
  
%put &opts;  
  
options nomprint nomlogic nomacrogen nosymbolgen /* etc. */;  
  
%secretmacro  
options symbolgen;  
options &opts;
```

```

429 proc sql noprint;
430   select setting into :opts separated by ' '
431   from sashelp.voption
432   where group = 'MACRO' and opttype = 'boolean' and setting ne 'MACRO'
433   ;
434 quit;

```

NOTE: PROCEDURE SQL used (Total process time):

real time	0.00 seconds
cpu time	0.00 seconds434

```

435 options nomprint nomlogic nomacrogen nosymbolgen /* etc. */;
436
437 %secretmacro

```

.....

```

438 options symbolgen;
439 options &opts;

```

**SYMBOLGEN: Macro variable OPTS resolves to NOCMDMAC
 NOIMPLMAC NOMAUTOLOCDISPLAY MAUTOSOURCE
 MERROR NOMEXECNOTE NOMFILE MLOGIC NOMLOGICNEST
 NOMPRINT NOMPRINTNEST NOMRECALL
 NOMSTORED SERROR SYMBOLGEN**

Ex 4: Program Name, Working Directory

- SASHELP.VEXTFIL
- Use to pull working directory, program name
- No direct equivalent, but similar to interactive mode
%SYSGET(SAS_EXECFILENAME),
%SYSGET(SAS_EXECPATH)
- Inconsistent, non-intuitive 'FILEREf' id

This SASHELP view looks different depending on how the program is submitted

Simple program TESTCODE.SAS:

```
proc print data = sashelp.vext;  
run;
```

This is how the output of SASHELP.VEXTFIL looks after
‘testcode.sas’ is submitted in a Windows interactive
session:

Obs	fileref	xpath	xengine
1	#LN00003	TERMINAL	
2	#LN00005	C:\Documents and Settings\Victor\My Documents	
3	#LN00007	C:\Documents and Settings\Victor\Desktop	
4	FL1	C:\Documents and Settings\Victor\Desktop\basug\testcode.sas	
5	#LN00011	C:\Documents and Settings\Victor\Desktop\basug\testcode.sas	
6	#LN00001	C:\Program Files\SAS\SAS 9.1\core\sasmsg	
7	#LN00001	C:\Program Files\SAS\SAS 9.1\access\sasmsg	
8	#LN00001	C:\Program Files\SAS\SAS 9.1\graph\sasmsg	
9	#LN00001	C:\Program Files\SAS\SAS 9.1\stat\sasmsg	
10	#LN00002	TERMINAL	

This is how the output looks when I run the exact same program in batch mode

Obs	fileref	xpath	xengine
1	#LN00003	C:\Documents and Settings\Victor\Desktop\basug\testcode.log	
2	#LN00004	C:\Documents and Settings\Victor\Desktop\basug\testcode.lst	
3	#LN00005	C:\Documents and Settings\Victor\Desktop\basug\testcode.sas	
4	#LN00001	C:\Program Files\SAS\SAS 9.1\core\sasmsg	
5	#LN00001	C:\Program Files\SAS\SAS 9.1\access\sasmsg	
6	#LN00001	C:\Program Files\SAS\SAS 9.1\graph\sasmsg	
7	#LN00001	C:\Program Files\SAS\SAS 9.1\stat\sasmsg	
8	#LN00002	TERMINAL	

We can't pull the program name based on the fileref if we don't know how the program will be submitted

- **Interactive Mode:**

```
4  FL1          C:\Documents and Settings\Victor\Desktop\basug\testcode.sas
5  #LN00011     C:\Documents and Settings\Victor\Desktop\basug\testcode.sas
```

- **Batch Mode:**

```
3  #LN00005     C:\Documents and Settings\Victor\Desktop\basug\testcode.sas
```

Suppose we want to name an output table based on the
program name

TESTCODE.SAS

```
data _null_;  
  set sashelp.vextfl;  
  where upcase(xpath) like '%.SAS';  
  call symput('outfile', tranwrd(upcase(xpath),'.SAS','.RTF'));  
run;  
  
%put &outfile;
```

It looks like it works correctly:

```
13 data _null_;  
14     set sashelp.vextfl;  
15     where upcase(xpath) like '%.SAS';  
16     call symput('outfile', tranwrd(upcase(xpath),'.SAS','.RTF'));  
17 run;
```

NOTE: There were 3 observations read from the data set SASHELP.VEXTFL.

WHERE UPCASE(xpath) like '%.SAS';

NOTE: DATA statement used (Total process time):

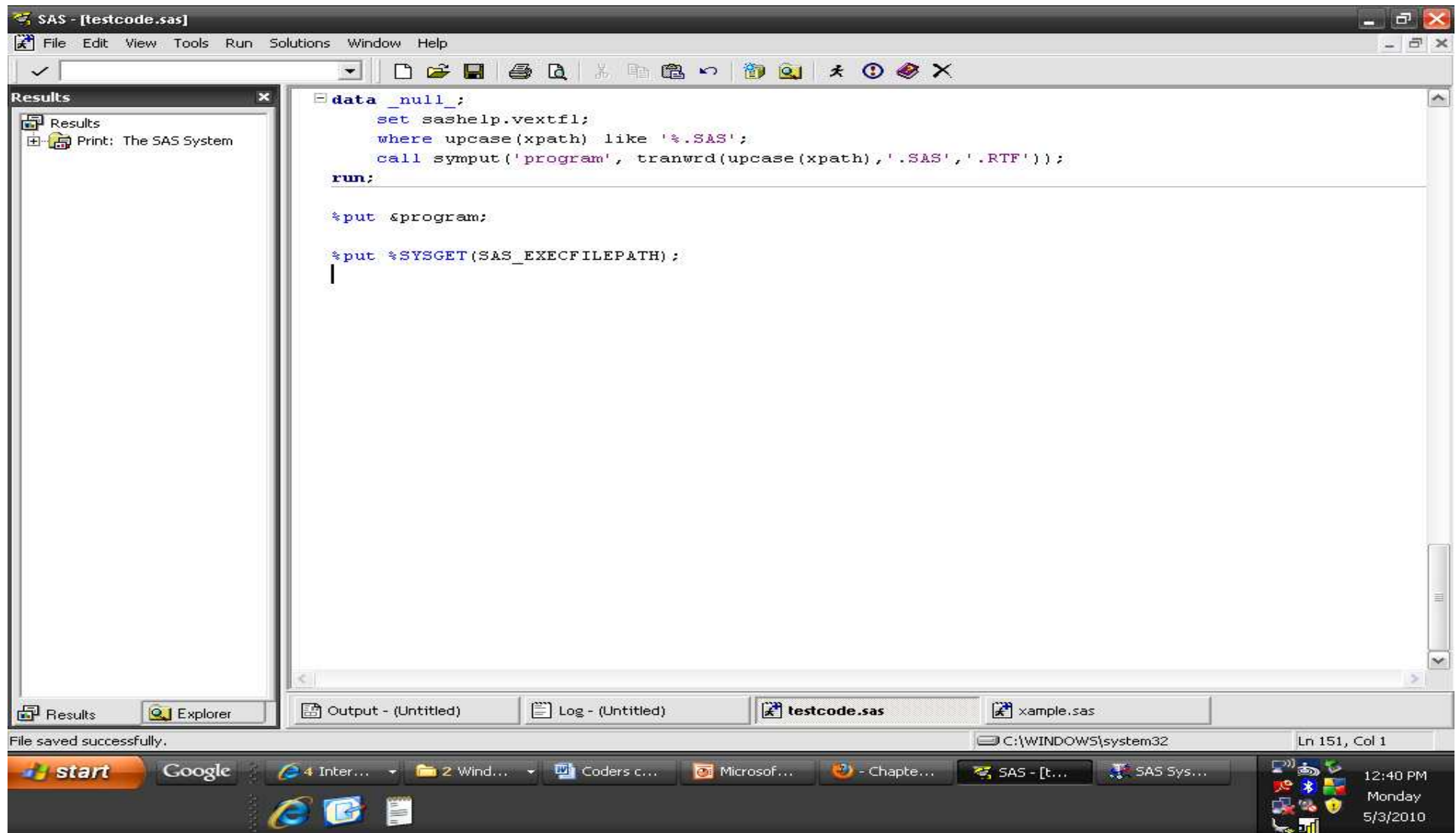
real time	0.00 seconds
cpu time	0.00 seconds

18

19 %put &outfile;

C:\DOCUMENTS AND SETTINGS\VICTOR\DESKTOP\BASUG\TESTCODE.RTF

...or does it? Look what happens when I open another program in my session and rerun my original program



I haven't changed anything about the code, I have just opened up another program in my session.
It now returns the wrong path and file!!

Program TESTCODE.SAS:

```
1 data _null_;  
2   set sashelp.vextfl;  
3   where upcase(xpath) like '%.SAS';  
4   call symput('outfile', tranwrd(upcase(xpath),'.SAS','.RTF'));  
5 run;
```

NOTE: There were 4 observations read from the data set SASHELP.VEXTFL.

WHERE UPCASE(xpath) like '%.SAS';

NOTE: DATA statement used (Total process time):

real time	0.04 seconds
cpu time	0.04 seconds

6

7 %put &outfile;

C:\PROJECTS\PRESENTATION\PROGRAMS\XAMPLE.RTF

This is dangerous. Order and FILEREF in VEXTFL are dependant on the arbitrary order in which the programs were opened. Here's why the problem happens:

Obs	fileref	xpath	xengine
1	#LN00003	TERMINAL	
2	#LN00005	C:\Documents and Settings\Victor\My Documents	
3	#LN00007	C:\Documents and Settings\Victor\Desktop	
4	FL1	C:\Documents and Settings\Victor\Desktop\basug\testcode.sas	
5	#LN00011	C:\Documents and Settings\Victor\Desktop\basug\testcode.sas	
6	FL2	C:\projects\presentation\programs\xample.sas	
7	#LN00015	C:\projects\presentation\programs\xample.sas	
8	#LN00001	C:\Program Files\SAS\SAS 9.1\core\sasmsg	
9	#LN00001	C:\Program Files\SAS\SAS 9.1\access\sasmsg	
10	#LN00001	C:\Program Files\SAS\SAS 9.1\graph\sasmsg	
11	#LN00001	C:\Program Files\SAS\SAS 9.1\stat\sasmsg	
12	#LN00002	TERMINAL	

`%SYSGET(SAS_EXECFILEPATH)` works regardless of how many program have been opened in the session, but only works in interactive mode

```
181 data _null_;  
182   Call symput('outfile2',tranwrd(upcase("%SYSGET(SAS_EXECFILEPATH)"),'.SAS','.RTF'));  
183 run;
```

NOTE: DATA statement used (Total process time):

real time	0.00 seconds
cpu time	0.00 seconds

```
184  
185 %put &outfile2;  
C:\DOCUMENTS AND SETTINGS\VICTOR\DESKTOP\BASUG\TESTCODE.RTF
```

If you cannot guarantee the platform and submission mode,
then a hybrid of the two methods is probably a good idea

```
proc sql noprint;
  select count(distinct xpath) into :prgs_count
  from sashelp.vextfl
  where upcase(xpath) like '%.SAS';
  ;
  select distinct xpath into :y
  from sashelp.vextfl
  where upcase(xpath) like '%.SAS';
  ;
quit;
```

```
%macro setpath;
  %if %eval(&prgs_count ne 1) %then %do;
    %put DATANOTE: Multiple SAS prgs in VEXTFL - switching to sas_execeflepath environment variable for path;
    %let pth_var = %sysget(SAS_EXECPATH);
  %end;
  %else %do;
    %put DATANOTE: SASHELP.VEXTFL used to determine path;
    %let pth_var = &y;
  %end;
%mend;
%setpath;
```

In Conclusion

- SASHELP views are very powerful tools for making code more generalizable; we have covered only a small fraction of their potential uses.
- The best way to get to know them is to browse the library and experiment.
- But be careful, especially where multiple platforms and submission methods can be involved!

Thank You

vpontes@inatecinc.com