

Tricks for testing Standard macro components in SAS Software.

1. L. U. Jørgensen
Novo Nordisk A/S, Copenhagen,
Denmark

Poster number PP18

Overview:

- Background:
- 5-7 persons working on a daily basis with updates of standard components
- Portfolio of 130 standard programs to produce derived data and TFL outputs.

Save time - Save output - Save the log

- Version 1 of program _v01:
- Version 2 of program _v02:

```
1 %macro standard_program_v01(year=);
2 %if %sysfunc(year(%sysfunc(today(),8.)) = %sysfunc(strip(&year))) %then
3 %do;
4   data event;
5     length event $100;
6     event = "PhUSE. Budapest %sysfunc(strip(&year))";
7     output;
8     event = "Conference 2.0.";
9     output;
10    event = "From Attending to Collaborating";
11    output;
12  run;
13 %end;
14 %else
15 %do;
16   data event;
17     length event $100;
18     event = "PhUSE. %sysfunc(strip(&year))";
19     output;
20     event = "Conference X.0.";
21     output;
22     event = "PhUSE text for next year";
23     output;
24  run;
25 %end;
26 options nocenter;
27 proc print data=event noobs label;
28   var event /style=jjust=left FONT_STYLE=italic COLOR=white background=orange;
29   label event = "Headlines";
30 run;
31 %mend standard_program_v01;
32 %standard_program_v01(year=2012); *Test case 1
33 %standard_program_v01(year=2013); *Test case 2
```

```
1 %macro standard_program_v02(year=);
2 %if %sysfunc(year(%sysfunc(today(),8.)) = %sysfunc(strip(&year))) %then
3 %do;
4   data event;
5     length event $100;
6     event = "PhUSE. Budapest %sysfunc(strip(&year))";
7     output;
8     event = "Conference 2.0.";
9     output;
10    event = "From Attending to Collaborating";
11    output;
12  run;
13 %end;
14 %else
15 %do;
16   data event;
17     length event $100;
18     event = "PhUSE. New City %sysfunc(strip(&year))";
19     output;
20     event = "Conference 3.0.";
21     output;
22     event = "PhUSE will be more and more interaktiv";
23     output;
24  run;
25 %end;
26 options nocenter;
27 proc print data=event noobs label;
28   var event /style=jjust=left FONT_STYLE=italic COLOR=white background=orange;
29   label event = "Headlines";
30 run;
31 %mend standard_program_v02;
32 %standard_program_v02(year=2012); *Test case 1
33 %standard_program_v02(year=2013); *test case 2
```

- Save OUTPUT in datasets:
- USE:
- 1 libname run_v01 'C:\users\liuj\run_v01';
- 2 data run_v01.event_case1;
- 3 set work.event;
- 4 run;
- Save LOG in file and convert LOG to datasets:
- USE:
- 1 filename test 'C:\users\liuj\run_v01.log';
- 2 proc printto log=test;
- 3 run;
- 4 read logfile ----> datasets

Headlines:
PhUSE. Budapest 2012
Conference 2.0:
From Attending to Collaborating
Headlines:
PhUSE. 2013
Conference X.0:
PhUSE. The text for next year

- Save OUTPUT in datasets:
- USE:
- 1 libname run_v02 'C:\users\liuj\run_v02';
- 2 data run_v02.event_case1;
- 3 set work.event;
- 4 run;
- Save LOG in file and convert LOG to datasets:
- USE:
- 1 filename test 'C:\users\liuj\run_v02.log';
- 2 proc printto log=test;
- 3 run;
- 4 read logfile ----> datasets

Headlines:
PhUSE. Budapest 2012
Conference 2.0:
From Attending to Collaborating
Headlines:
PhUSE. New City 2013
Conference 3.0:
PhUSE. This year more interaktiv

Compare _v01 with _v02:
USE:
1 proc compare data=run_v01.event_case1
2 compare=run_v02.event_case1
3 noprint
4 out=work.comp outnoequal;

- Trick
- Save the output from all test cases in SAS datasets
- Use SAS proc compare to test with outputdatasets
- Stack them and test observations

- Trick
- Remove anything time dependent from log e.g.: real time, cpu time, options, macvars, workmems, datetimestamp
- Transform log to a dataset
- Use SAS proc compare on datasets

Automatic validation.

- Write macro to save all parameteres from standard component on execution:

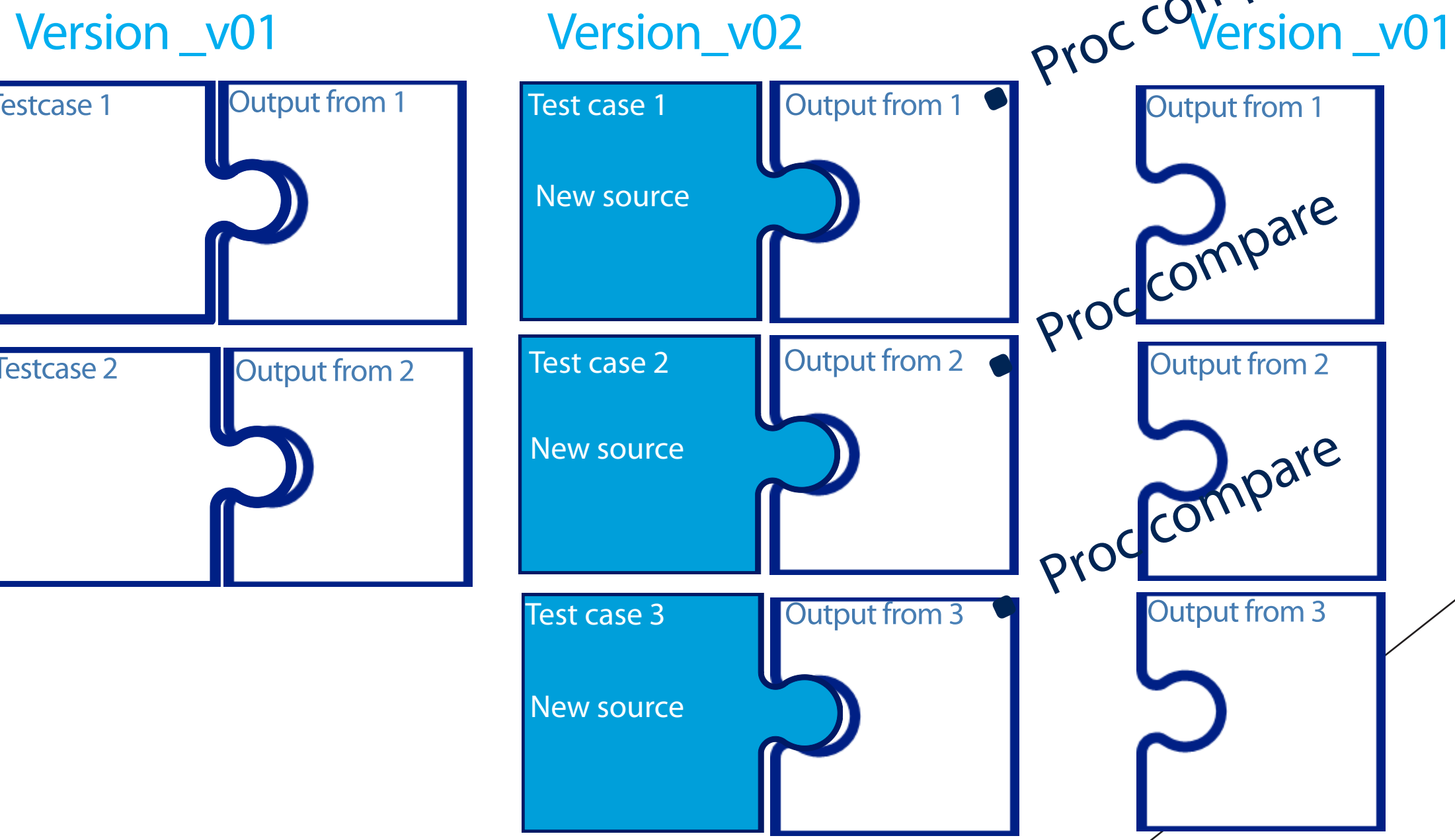
```
1 %macro save_parms_v01(
2   pgmname =
3 );
4   data work.input_parms;
5     length pgmname $50;
6     set sashelp.vmacro;
7     pgmname = "%nrbrquote(&pgmname)";
8   ;
9   run;
10 %mend save_parms_v01;
11 %save_parms_v01(
12   pgmname = lis_hypo_te_v01.sas
13 );
```

Partial output:

	A	B	
1	macroname	parameter name	value
2	LIS_HYPO_TE_V01	INC_MESI	
3	GLOBAL	ID	16270400
4	GLOBAL	ACCESS_MACRO	P:\general\Standard_Program_Library\Macros
5	GLOBAL	X_SEL_SUBJ_HYPO	TRT_EMERGENT = "Y"
6	GLOBAL	_CLIENTAPP	SAS Enterprise Guide'
7	GLOBAL	_CLIENTUSERID	liuj'
8	GLOBAL	_CLIENTUSERNAME	LIUJ (Lina Ulkjær Jørgensen)'
9	GLOBAL	FONT_SIZE	9
10	GLOBAL	INSTANCE_PATH	P:\nn1250\NN1250-3583\current\stats
11	GLOBAL	TRIAL_POPULATION	FAS
12	GLOBAL	DELETION_FLAG	
13	GLOBAL	STD_LISTING_LIB	-P:\general\Standard_Program_Library\Listing
14	GLOBAL	OUTPUT_FOLDER	-P:\nn1250\NN1250-3583\current\stats\outp
15	GLOBAL	TITLE_FINAL	---Treatment Emergent Hypoglycaemic Episod
16	GLOBAL	TRL_ARM_SHORT_LB	TRL_ARM_SHORT_LB
17	GLOBAL	TREATMENT	"IDeg OD" "IGlar OD"
18	GLOBAL	_SDDFOOTER1	nn1250/NN1250-3583/current/stats
19	GLOBAL	_SDDFOOTER2	17SEP2012:16:56:04 - lis_hypo_te_v01.sas/l_h
20	GLOBAL	DAY_TE	7
21	GLOBAL	TITLE_TEXT	Treatment Emergent Hypoglycaemic Episodes
22	GLOBAL	CLEAN_UP	N
23	AUTOMATIC	SYSUSERID	liuj
24	AUTOMATIC	SYSVER	9.3

- Trick
- Save all parameters used on execution
- Check against protocol and programming plan
- E.g. Check similar treatments throughtout a study

Hotfix Update component



Branchtesting

- Robustness
- Branch test the source:

Line No	A	B	C	D	E	F	G	H	I	J	K
1	TRUE	FALSE	Branch description								
2	14	2	%if %sysfunc(year(%sysfunc(today(),8.)) = %sysfunc(strip(&year))) %then								
3		2	%else								

- Trick
- Write every if/then/else statement in the program to Excel
- A test program must hit all lines
- A utility, writes all if/then/else statement to Excel and all True cases

