

# Validation Summary using SYSINFO

Srinivas Vanam

Mahipal Vanam

Shravani Vanam

Percept Pharma Services, Bridgewater, NJ

## ABSTRACT

This paper presents a macro that produces a Validation Summary using SYSINFO automatic macro variable. The study lead who is managing a clinical study often needs to make sure that the datasets created during analysis are validated properly. In a general scenario, the lead distributes the datasets among programmers to either create primary or validation datasets. The primary and validation programmers create their assigned datasets following the Specifications in specified locations. This macro compares each dataset in primary location with its counterpart in validation location. After performing PROC COMPARE of each dataset, the result of the comparison is stored in an automatic macro variable "SYSINFO" that is supplied by the SAS system. This macro decodes the value of SYSINFO in order to interpret the comparison result and produces a summary report of validation of all the datasets present in Primary and Validation locations.

## ALGORITHM

The macro %SYSINFO is passed with 3 parameters PRIPATH, VALPATH and DESC.

1. **PRIPATH:** is the path of the directory where Primary datasets are created.
2. **VALPATH:** is the path of the directory where Validation datasets are created.
3. **COMMFL:** is a flag(Y/N) which tells whether comments should be added or not.

It then reads in the list of all datasets present in PRIPATH and VALPATH. It checks the common datasets present in both directories and performs PROC COMPARE on each dataset in PRIPATH with its counterpart in VALPATH. After the PROC COMPARE, the result of its comparison will be stored in an automatic macro variable called SYSINFO. It then decodes the value of SYSINFO and creates a dataset called "SUMMARY" which contains each dataset name and the status report of its comparison.

If the COMMFL flag is set to "Y", then SYSINFO calls another macro called COMMENTS which adds the comments for each of the difference reported in SUMMARY dataset and creates a final summary dataset calls FINALSUM which is printed in the output using PROC REPORT.

**SYSINFO codes:**

Bit	Condition	Code	Description
1	DSLABEL	1	Data set labels differ
2	DSTYPE	2	Data set types differ
3	INFORMAT	4	Variable has different informat
4	FORMAT	8	Variable has different format
5	LENGTH	16	Variable has different length
6	LABEL	32	Variable has different label
7	BASEOBS	64	Base data set has observation not in comparison
8	COMPOBS	128	Comparison data set has observation not in base
9	BASEBY	256	Base data set has BY group not in comparison
10	COMPBY	512	Comparison data set has BY group not in base
11	BASEVAR	1024	Base data set has variable not in comparison
12	COMPVAR	2048	Comparison data set has variable not in base
13	VALUE	4096	A value comparison was unequal
14	TYPE	8192	Conflicting variable types
15	BYVAR	16384	BY variables do not match
16	ERROR	32768	Fatal error: comparison not done

Most of the SYSINFO codes can be understood with the description provided in the above table except for code 2 which means “dataset types differ”. There are some special SAS datasets like CORR, ACE etc that can be created by using TYPE= option in the DATA statement. Usually these datasets are produced by some Statistical procedures. These datasets can be also be created with typical DATA and SET statement for demo purpose.

**Example:**

```

data pri_ds ;
  x = 1 ;
  y = 2 ;
run;
data val_ds(type = corr) ;
  x = 1 ;
  y = 2 ;
run;

```

In this example, PRI\_DS and VAL\_DS are identical except the difference that, their TYPEMEM will differ when we check SASHELP.VTABLE.TYPEMEM. These dataset types should not be confused with SASHELP.VTABLE.MEMTYPE which usually have the values of DATA/VIEW depending on whether the file is a SAS dataset or a SAS view.

The SAS System												
09:54 Wednesday, April 15, 2015 1												
Obs	libname	memname	memtype	dbms_	memtype	memlabel	typenem	crdate	modate			
489	WORK	PRI_DS	DATA				DATA	15APR15:10:29:53	15APR15:10:29:53			
490	WORK	VAL_DS	DATA				CORR	15APR15:10:29:53	15APR15:10:29:53			
Obs	nobs	obslen	nvar	protect	compress	encrypt	npage	filesize	pcompress	reuse	bufsize	
489	1	16	2	---	NO	NO	1	8192	0	no	4096	
490	1	16	2	---	NO	NO	1	8192	0	no	4096	
Obs	delobs	nlobs	maxvar	maxlabel	maxgen	gen	attr	indxtype	datarep	sortname	sorttype	
489	0	1	1	0	0	.	ON		NATIVE			
490	0	1	1	0	0	.	ON		NATIVE			
Obs	sortchar	reqvector								datarepname		
489		181F101122220032220102320432012222003E0000100301								WINDOWS_32		
490		181F101122220032220102320432012222003E0000100301								WINDOWS_32		
Obs	encoding		audit	audit_	audit_	audit_	audit_	num_	num_			
489	wlatin1	Western (Windows)	no	no	no	no	no	0	2			
490	wlatin1	Western (Windows)	no	no	no	no	no	0	2			

### OPTIONS SQLUNDOPOLICY = NONE

Setting this option suppresses the WARNING message which will appear when the PROC SQL's CREATE TABLE statement refers to same dataset as the one mentioned in FORM clause.

Example:

```
proc sql ;
  create table temp as
  select *
  from temp
;
quit;
```

### OPTIONS MPRINT:

This option prints the SAS code generated from a macro in the SAS log, which helps the programmer in debugging in case of any errors/warnings.

### OPTIONS MINOPERATOR MINDELIMITER='/:

IN operator cannot be used in a macro prior to SAS v9.2. SAS 9.2 provides the ability to use IN operator within macro code. In order to make use of this facility, the option MINOPERATOR has to be set. The default delimiter is a space. To have a custom delimiter like a comma, it has to be set using MINDELIMITER option.

Example:

**Without MINOPERATOR:**

```
%if &bit. eq 4 or &bit. eq 8 or &bit. eq 16 or &bit. eq 32 %then %do;
```

**With MINOPERATOR:**

```
%if &bit in (4,8,16,32) %then %do;
```

**SAMPLE OUTPUT**

Dataset	SYSINFO	Code	Bit	Description	Comments
A	.	.	.	Dataset in PRI not in VAL	
B	.	.	.	Dataset in VAL not in PRI	
DS_00	0	0	0	** Datasets are Identical **	
DS_01	1	1	1	Dataset labels differ	Label 1 / Label 2
DS_02	2	2	2	Dataset types differ	DATA / CORR
DS_03	4	4	3	Variable has different informat	col2(DATE9./YYMMDD10.)
DS_04	8	8	4	Variable has different format	col2(DATE9./YYMMDD10.)
DS_05	16	16	5	Variable has different length	col2(8/9)
DS_06	33	1	1	Dataset labels differ	Label 1 / Label 2
	33	32	6	Variable has different label	col2(Label 1/Label 2)
DS_07	64	64	7	PRI dataset has an observation not in VAL	# of observations: 2/1
DS_08	128	128	8	VAL dataset has an observation not in PRI	# of observations: 1/2
DS_11	1024	1024	11	PRI dataset has a variable not in VAL	col2
DS_12	2048	2048	12	VAL dataset has a variable not in PRI	col2
DS_13	4096	4096	13	A value comparison was unequal	# of values unequal: 6
DS_14	8192	8192	14	Conflicting variable types	col2(num/char)
DS_16	32768	32768	16	Fatal Error: Comparison not done	* To be Implemented *
X	.	.	.	Dataset in VAL not in PRI	

**IMPROVEMENTS**

1. This macro creates a "summary" dataset which lists each dataset with the differences and then the description is added later on by calling the macro %DESCRIPTION. But the description can be added at the time of creation of "SUMMARY" dataset, by using PROC FCMP's RUN\_MACRO() feature.
2. The code to handle SYSINFO codes 256, 512, 16384(BY group differences) has not been developed in the %DESCRIPTION macro keeping in mind that, most of the times PROC

COMPARE will be run without using ID or BY statement after sorting the datasets by KEY variables.

3. Additional code can be added to make use of ID statement in selected comparisons.

## CONCLUSION

This macro can be used for producing a Summary report that helps to make sure the datasets are validated properly.

## REFERENCES

1. *“A Few Quick and Efficient Ways to Compare Data”* by Abraham Pulavarti.
2. *“An Easy, Concise Way to Summarize Multiple PROC COMPAREs using the SYSINFO Macro Variable”* by Lex Fennell.

## CONTACT INFORMATION

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

Srinivas Vanam ([srinivasvanam@gmail.com](mailto:srinivasvanam@gmail.com))

Mahipal Vanam ([mahipal.vanam@perceptservices.com](mailto:mahipal.vanam@perceptservices.com))

Shravani Vanam ([shravsvanam@gmail.com](mailto:shravsvanam@gmail.com))



## ACKNOWLEDGEMENTS

Percept Pharma Services ([www.perceptservices.com](http://www.perceptservices.com))  
1031 US Highway 22W, Suite 304  
Bridgewater, NJ. 08807

I would like to take this opportunity to thank my mentor **Mr. Srinivas Artham**, a Java Developer who taught me and inspired me towards Programming. – Srinivas Vanam.

## TRADEMARK INFORMATION

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.

Other brand and product names are trademarks of their respective companies.

**APPENDIX: SOURCE CODE**

```

/*****
** Project:                MACROS                                **
** Title:                  Validation Summary using SYSINFO macro variable **
** Program:                sysinfo.sas                          **
** Programmer:            Srinivas Vanam                        **
**                        Mahipal Vanam                          **
**                        Shravani Vanam                         **
** Date:                  31-Mar-2015                           **
** Purpose:                To PROC COMPARE datasets present in 2 libraries and present a **
**                        Validation Summary.                     **
**                        ****                                     **
** Modification History:  ****                                     **
**                        ****                                     **
**                        ****                                     **
**                        ****                                     **
*****/

/*****
** Preprocessing **
*****/
/* Cleaning up the LOG and LST files */
dm "log; clear; lst; clear;" ;

/* Setting GLOBAL options */
options sqlundopolicy = none ;
options mprint ;
options minoperator mindelimiter = ',' ; /* For using IN operator in macros */

/* Creating formats */
proc format ;
  value sysfmt
    0   = " ** Datasets are Identical ** "
    1   = "Dataset labels differ"
    2   = "Dataset types differ"
    4   = "Variable has different informat"
    8   = "Variable has different format"
    16  = "Variable has different length"
    32  = "Variable has different label"
    64  = "PRI dataset has an observation not in VAL"
    128 = "VAL dataset has an observation not in PRI"
    256 = "PRI dataset has a BY group not in VAL"
    512 = "VAL dataset has a BY group not in PRI"
    1024 = "PRI dataset has a variable not in VAL"
    2048 = "VAL dataset has a variable not in PRI"
    4096 = "A value comparison was unequal"
    8192 = "Conflicting variable types"
    16384 = "BY variables do not match"
    32768 = "Fatal Error: Comparison not done"
;
run;

/*****
** sysinfo macro **
*****/
%macro sysinfo(pripath=,valpath=,commfl=N) ;

  /* Creating the library references */
  libname pri "&pripath." access = readonly ;
  libname val "&valpath." access = readonly ;

  /* Getting the list of datasets in each library */
  proc sql ;
    create table prids as

```

## Validation Summary using SYSINFO, continued

```
select memname
from dictionary.tables
where upcase(libname)="PRI"
order by memname
;
create table valds as
select memname
from dictionary.tables
where upcase(libname)="VAL"
order by memname
;
quit;

/* Finding common datasets */
data pri both val ;
merge prids(in=pri)
      valds(in=val)
      ;
by memname ;
if pri and not val then output pri ;
if pri and val      then output both ;
if not pri and val  then output val ;
run;

proc sql noprint ;
select memname into :dslist separated by ':'
from both
;
quit;

proc sql ;
create table summary as
select memname as dsname length=12,
      . as sysinfo,
      . as code,
      . as bit,
      "Dataset in PRI not in VAL" as desc length=50
from pri
UNION
select memname as dsname length=12,
      . as sysinfo,
      . as code,
      . as bit,
      "Dataset in VAL not in PRI" as desc length=50
from val
;
quit;

/* Performing PROC COMPARE on each dataset */
%let i = 1 ;
%let result = ;
%let dsname = %scan(&dslist,&i,":") ;
%do %while(&dsname. ne ) ;
  %put Dataset being compared: &dsname. ;

ods html file = "C:\Users\vanams\Desktop\SUG Papers\sysinfo\dummy.html" ;
ods listing close ;
proc compare base = pri.&dsname
             comp = val.&dsname
             ;
  /* Code for handling BY groups */
  %if &dsname. eq DS_16 %then %do;
    by sid ;
  %end;

run;
ods listing ;
ods html close ;

%put Result of PROC COMPARE: &sysinfo. ;
%let result = &result. [&dsname.:&sysinfo.] ;
```

Validation Summary using SYSINFO, continued

```

        %let code = &sysinfo. ; /* SYSINFO value will be reset when new DATA/PROC step is started
*/
data result ;
  length dsname $12 desc $50 ;
  dsname = "&dsname." ;
  sysinfo = &code. ;
  if &code. eq 0 then do ;
    code = 0 ;
    bit = 0 ;
    desc = put(&code.,sysfmt.) ;
    output;
  end;
  else do ;
    do j = 1 to 16 ;
      code = bAND(&code.,2**(j-1));
      if code > 0 then do ;
        bit = j ;
        desc = put(code,sysfmt.) ;
        output;
      end;
    end;
  end;
  drop j ;
run;

proc append base = summary data = result force ;
run;

%let i = %eval(&i. + 1) ;
%let dsname = %scan(&dslist,&i,":") ;
%end;
%put &result. ;

/* Copy the SUMMARY dataset to FINALSUM dataset */
data finalsum ;
  set summary ;
run;

%if &commfl. eq Y %then %do ;
  /* Calls the COMMENTS macro to add the comments for each description */
  %comments;
%end;

/* Printing the final Summary report */
ods html file = "C:\Users\vanams\Desktop\SUG Papers\sysinfo\summary.html" ;
title "Validation Summary using SYSINFO" ;
proc report data = finalsum nowindows headline headskip ls=150 ;
  column dsname sysinfo code bit desc %if &commfl. eq Y %then %do; comments %end; ;

  define dsname / width=7 id "Dataset" order ;
  define sysinfo / width=7 id "SYSINFO" ;
  define code / width=7 "Code" ;
  define bit / width=7 "Bit" ;
  define desc / width=50 "Description" ;
  %if &commfl. eq Y %then %do;
    define comments / width=100 "Comments" ;
  %end;

  break after dsname / skip ;
run;
title;
ods html close;

%mend;

/*****
** comments macro **
*****/
%macro comments ;

```



## Validation Summary using SYSINFO, continued

```

/* Getting the count and list of dataset and its differences */
proc sql noprint ;
  select strip(dsname)||":"||left(put(code,best.)) into: valsum separated by "-"
  from summary
  where code ne .
  ;
  select count(*) into: sumcnt
  from summary
  where code ne .
  ;
quit;
%put &sumcnt. &valsum. ;

%do i = 1 %to &sumcnt. ;
  %let row = %scan(&valsum.,&i.,"-") ;
  %put &row. ;
  %let dsn = %scan(&row.,1,":") ;
  %let code = %scan(&row.,2,":") ;
  %put &dsn. &code. ;

  /* Create description for code 0 */
  %if &code. eq 0 %then %do ;
    %let comm = ;
  %end;
  /* Create description for codes 1,2 */
  %else %if &code. in (1,2) %then %do ;
    %if &code. eq 1 %then %let type = memlabel ;
    %else %if &code. eq 2 %then %let type = typemem ;
    proc sql noprint ;
      select &type. into :value1
      from dictionary.tables
      where upcase(libname) = "PRI" and upcase(memname) = "&dsn."
      ;
      select &type. into :value2
      from dictionary.tables
      where upcase(libname) = "VAL" and upcase(memname) = "&dsn."
      ;
    quit;
    %let comm = %left(&value1.) / %left(&value2.) ;
  %end;
  /* Create description for codes 4,8,16,32,8192 */
  %else %if &code. in (4,8,16,32,8192) %then %do;
    %if &code. eq 4 %then %let type = informat ;
    %else %if &code. eq 8 %then %let type = format ;
    %else %if &code. eq 16 %then %let type = put(length,best.) ;
    %else %if &code. eq 32 %then %let type = label ;
    %else %if &code. eq 8192 %then %let type = type ;
    proc sql noprint ;
      create table ds1 as
      select name, &type. as value1
      from dictionary.columns
      where upcase(libname) = "PRI" and upcase(memname) = "&dsn."
      ;
      create table ds2 as
      select name, &type. as value2
      from dictionary.columns
      where upcase(libname) = "VAL" and upcase(memname) = "&dsn."
      ;
      create table diff as
      select a.name, a.value1, b.value2
      from ds1 as a, ds2 as b
      where a.name = b.name and a.value1 ^= b.value2
      ;
      select strip(name) || "(" || strip(value1) || "/" ||
      strip(value2) || ")" into :comm separated by " - "
      from diff
      ;
    quit;
  %end;
  /* Create description for codes 64,128 */

```

## Validation Summary using SYSINFO, continued

```
%else %if &code. in (64,128) %then %do;
  proc sql noprint;
    select nobs into :nobs1
      from dictionary.tables
      where upcase(libname) = "PRI" and upcase(memname) = "&dsn."
      ;
    select nobs into :nobs2
      from dictionary.tables
      where upcase(libname) = "VAL" and upcase(memname) = "&dsn."
      ;
  quit;
  %let comm = # of observations: %left(&nobs1.)/%left(&nobs2.) ;
%end;
/* Create description for codes 1024,2048 */
%else %if &code. in (1024,2048) %then %do;
  %if &code. eq 1024 %then %do;
    %let lib1 = PRI ;
    %let lib2 = VAL ;
  %end;
%else %if &code. eq 2048 %then %do;
  %let lib1 = VAL ;
  %let lib2 = PRI ;
%end;
  proc sql noprint;
    select name into :vars separated by " "
      from dictionary.columns
      where upcase(libname) = "%left(&lib1.)" and upcase(memname) = "&dsn." and
            name not in ( select name
                          from dictionary.columns
                          where upcase(libname)="%left(&lib2.)" and upcase(memname)="%&dsn."
                        )
      ;
  quit;
  %let comm = &vars. ;
%end;
/* Create description for code 4096 */
%else %if &code. eq 4096 %then %do;
  ods listing close;
  ods output comparesummary = compsum ;
  proc compare base = pri.&dsn.
               comp = val.&dsn.
               listall;
    run;
  ods output close;
  ods listing;
  data _null_ ;
    set compsum ;
    flag = index(batch,"Total Number of Values which Compare Unequal:") ;
    if flag > 0 then do;
      call symput("nodif",scan(batch,2,":,."));
    end;
  run;
%let comm = %str(# of values unequal: &nodif.) ;
%end;
/* Create descriptions for other code values */
%else %do;
%let comm = %str( * To be Implemented * ) ;
%end;

/* Adding the description to the FINAL SUMMARY dataset */
data dummy ;
  length dsname $12 comments $200 ;
  dsname   = "&dsn." ;
  code     = &code. ;
  comments = "&comm." ;
run;
proc sort data = dummy      ; by dsname code ; run;
proc sort data = finalsum  ; by dsname code ; run;
data finalsum;
  merge finalsum(in=a)
        dummy(in=b);
```

Validation Summary using SYSINFO, continued

```
        by dsname code ;
run;
%end;

%mend;

/* Calling the macro *
* pripath: Specifies the Primary datasets location.
* valpath: Specifies the Validation datasets location.
* commfl : Flag to print description.
*/
%sysinfo(pripath=%str(C:\Users\vanams\Desktop\SUG Papers\sysinfo\pri),
        valpath=%str(C:\Users\vanams\Desktop\SUG Papers\sysinfo\val),
        commfl=Y
        );
```