

SAS[®] Proc Compare as a Validation Tool

Rob Krajcik

Bristol-Myers Squibb Company

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 registered trademarks or trademarks of their respective companies.

What Does Proc Compare Do?

Compares:

- the contents of two SAS data sets
- selected variables in different data sets
- variables within the same data set.

Determines matching variables and matching observations.

Matching variables are variables with the same name or variables you pair with the VAR and WITH statements.

Matching observations are observations that have the same values for all ID variables you specify or that occur in the same position in the data sets.

What Information Does it Provide?

Proc Compare generates the following information:

- whether matching variables have different values.
- whether one data set has more observations than the other.
- what variables the two data sets have in common.
- how many variables are in one data set but not in the other.
- whether matching variables have different formats, labels or types.
- a comparison of the values of matching observations.

Proc Compare Statements

```
PROC COMPARE <base= compare= option(s)>;  
BY <DESCENDING> variable-1 ...<DESCENDING> variable-n> NOTSORTED>;  
ID <DESCENDING> variable-1 ...<DESCENDING> variable-n> NOTSORTED>;  
VAR variable(s);  
WITH variable(s);
```

```
base=<xyz(keep= | drop= | rename= | where=)>  
compare=<zyx(keep= | drop= | rename= | where=)>
```

LABEL, ATTRIB, FORMAT, and WHERE statements can be used, as well as any global statements.

If you omit COMPARE=,
then you must use the VAR and WITH statements.

Proc Compare Selected Options

- `brief` – suppresses the four default summary reports (data set summary, variables summary, observation summary, and values summary) and produces a short comparison summary.
- `listobs` – lists all observations found in only one data set.
- `listvar` – lists all variables found in only one data set.
- `maxprint=total` | (per-variable, total) – specifies the maximum number of differences to print (default is 50,500).
- `noprint` – suppresses all printed output (use with `out=` option).
- `nosummary` – suppresses the four default summary reports.
- `novalues` – suppresses the value comparison results report.
- `outbase` | `outcomp` | `outdif` – writes `base` | `comp` | `dif` results to output data set specified in the `out=` option.
- `outnoequal` – suppresses writing observation to the output data set when all values in the observation are judged equal.
- `transpose` – prints the value differences report by observation instead of by variable.

Compare a Data Set with Itself

```
proc compare base=sashelp.class compare=sashelp.class; run;
```

The COMPARE Procedure

Comparison of SASHELP.CLASS with SASHELP.CLASS (Method=EXACT)

Data Set Summary

Dataset	Created	Modified	NVar	NObs
SASHELP.CLASS	12MAY04:22:53:56	12MAY04:22:53:56	5	19
SASHELP.CLASS	12MAY04:22:53:56	12MAY04:22:53:56	5	19

Variables Summary

Number of Variables in Common: 5.

Observation Summary

Observation	Base	Compare
First Obs	1	1
Last Obs	19	19

Number of Observations in Common: 19.

Total Number of Observations Read from SASHELP.CLASS: 19.

Total Number of Observations Read from SASHELP.CLASS: 19.

Number of Observations with Some Compared Variables Unequal: 0.

Number of Observations with All Compared Variables Equal: 19.

NOTE: No unequal values were found. All values compared are exactly equal.

Build Two Data Sets

```
proc sort data=sashelp.class out=class1;  
by name;  
run;
```

```
proc sort data=sashelp.class out=class2;  
by weight;  
run;
```

```
data class2(rename=(name=nam height=ht weight=wt sex=gender age=age_yrs));  
  set class2; /*Rename vars, change some names, add a duplicate */  
  if name eq 'Jane'  
  then name='Janet';  
  if name eq 'Ronald'  
  then name='Robert';  
  if name='Henry'  
  then output;  
  output;  
run;
```

```
proc compare base=class1 compare=class2 novalues listvar;  
run;
```

First Comparison Results

The COMPARE Procedure

Comparison of WORK.CLASS1 with WORK.CLASS2 (Method=EXACT)

Data Set Summary

Dataset	Created	Modified	NVar	NObs
WORK.CLASS1	29OCT07:10:37:50	29OCT07:10:37:50	5	19
WORK.CLASS2	29OCT07:10:37:50	29OCT07:10:37:50	5	20

Variables Summary

Number of Variables in Common: 0.

Number of Variables in WORK.CLASS1 but not in WORK.CLASS2: 5.

Number of Variables in WORK.CLASS2 but not in WORK.CLASS1: 5.

Listing of Variables in WORK.CLASS1 but not in WORK.CLASS2

Variable Type Length

Name	Char	8
Sex	Char	1
Age	Num	8
Height	Num	8
Weight	Num	8

Listing of Variables in WORK.CLASS2 but not in WORK.CLASS1

Variable Type Length

nam	Char	8
gender	Char	1
age_yrs	Num	8
ht	Num	8
wt	Num	8

WARNING: The data sets WORK.CLASS1 and WORK.CLASS2 have no variables in common. There are no matching variables to compare. Comparisons of data values not performed.

Second Comparison Results

```
proc compare base=class1 compare=class2 novalues;  
var name sex age height weight;  
with nam gender age_yrs ht wt;  
run;
```

The COMPARE Procedure

Comparison of WORK.CLASS1 with WORK.CLASS2 (Method=EXACT)

Values Comparison Summary

Number of Variables Compared with All Observations Equal: 0.

Number of Variables Compared with Some Observations Unequal: 5.

Total Number of Values which Compare Unequal: 71.

Maximum Difference: 62.

All Variables Compared have Unequal Values

Variable	Type	Len	Compare	Len	Ndif	MaxDif
Name	CHAR	8	nam	8	17	
Sex	CHAR	1	gender	1	8	
Age	NUM	8	age_yrs	8	12	3.000
Height	NUM	8	ht	8	17	17.700
Weight	NUM	8	wt	8	17	62.000

Third Comparison

```
proc compare base=class1 compare=class2(rename=(nam=name));  
id name; /* Use variable name to match observations */  
var sex age height weight;  
with gender age_yrs ht wt;  
run;
```

Third Comparison Results

The COMPARE Procedure

Comparison of WORK.CLASS1 with WORK.CLASS2 (Method=EXACT)

Data Set Summary

Dataset	Created	Modified	NVar	NObs
WORK.CLASS1	29OCT07:10:07:30	29OCT07:10:07:30	5	19
WORK.CLASS2	29OCT07:10:07:30	29OCT07:10:07:30	5	20

Variables Summary

Number of Variables in Common: 1.

Number of Variables in WORK.CLASS1 but not in WORK.CLASS2: 4.

Number of Variables in WORK.CLASS2 but not in WORK.CLASS1: 4.

Number of ID Variables: 1.

Number of VAR Statement Variables: 4.

Number of WITH Statement Variables: 4.

ERROR: The ID variable values do not match at observation 14 in the base data set WORK.CLASS1 and observation 3 in the comparison data set WORK.CLASS2.

(When one or both data sets are not sorted by the ID variables, or when NOTSORTED is specified, the observations must match one-to-one.)

NOTE: The current ID values in WORK.CLASS1 and WORK.CLASS2 are:

Name=Mary.

Name=James.

NOTE: Comparison aborted.

Fourth Comparison

```
proc sort data=class2;  
    by nam age_yrs;  
run;
```

```
proc compare base=class1 compare=class2(rename=(nam=name));  
id name; /* Use variable name to match observations */  
var sex age height weight;  
with gender age_yrs ht wt;  
run;
```

Fourth Comparison Results

The COMPARE Procedure

Comparison of WORK.CLASS1 with WORK.CLASS2 (Method=EXACT)

Data Set Summary

Dataset	Created	Modified	NVar	NObs
WORK.CLASS1	29OCT07:10:07:30	29OCT07:10:07:30	5	19
WORK.CLASS2	29OCT07:10:31:10	29OCT07:10:31:10	5	20

Variables Summary

Number of Variables in Common: 1.

Number of Variables in WORK.CLASS1 but not in WORK.CLASS2: 4.

Number of Variables in WORK.CLASS2 but not in WORK.CLASS1: 4.

Number of ID Variables: 1.

Number of VAR Statement Variables: 4.

Number of WITH Statement Variables: 4.

WARNING: The data set WORK.CLASS2 contains a duplicate observation at observation number 6.

NOTE: At observation 6 the current and previous ID values are:

Name=Henry.

NOTE: Further warnings for duplicate observations in this data set will not be printed.

Observation Summary ...

Number of Observations in Common: 17.

Number of Observations in WORK.CLASS1 but not in WORK.CLASS2: 2.

Number of Observations in WORK.CLASS2 but not in WORK.CLASS1: 3.

Number of Duplicate Observations found in WORK.CLASS2: 3.

Fifth Comparison

```
proc sort data=class1;  
  by name age;  
run;
```

```
proc compare base=class1 compare=class2(rename=(nam=name age_yrs=age)) listobs;  
id name age; /* Use variable name to match observations */  
var sex height weight;  
with gender ht wt;  
run;
```

Fifth Comparison Results

The COMPARE Procedure

Comparison of WORK.CLASS1 with WORK.CLASS2 (Method=EXACT)

Comparison Results for Observations

Observation 6 in WORK.CLASS2 not found in WORK.CLASS1: Name=Henry Age=14.
Observation 7 in WORK.CLASS1 not found in WORK.CLASS2: Name=Jane Age=12.
Observation 8 in WORK.CLASS2 not found in WORK.CLASS1: Name=Janet Age=12.
Observation 18 in WORK.CLASS2 not found in WORK.CLASS1: Name=Robert Age=15.
Observation 17 in WORK.CLASS1 not found in WORK.CLASS2: Name=Ronald Age=15.

Observation Summary

Observation	Base	Compare	ID
First Obs	1	1	Name=Alfred Age=14
Last Obs	19	20	Name=William Age=15

Number of Observations in Common: 17.

Number of Observations in WORK.CLASS1 but not in WORK.CLASS2: 2.

Number of Observations in WORK.CLASS2 but not in WORK.CLASS1: 3.

Number of Duplicate Observations found in WORK.CLASS2: 1.

Total Number of Observations Read from WORK.CLASS1: 19.

Total Number of Observations Read from WORK.CLASS2: 20.

Number of Observations with Some Compared Variables Unequal: 0.

Number of Observations with All Compared Variables Equal: 17.

NOTE: No unequal values were found. All values compared are exactly equal.

Compare Vars in One Data Set

```
data class;
  set sashelp.class;
  wait=weight;
  waite=weight;
  if sex='M'
  then do;
    wait=wt*1.05;
    waite=wt*1.1;
  end;
run;

proc compare base=work.class;
  id name;
  var weight weight;
  with wait waite ;
  where sex eq 'M';
run;
```

Compare Vars in One Data Set

Data Set Summary

Dataset	Created	Modified	NVar	NObs
WORK.CLASS	31OCT07:16:00:34	31OCT07:16:00:34	8	19

Values Comparison Summary

Number of Variables Compared with All Observations Equal: 0.
Number of Variables Compared with Some Observations Unequal: 2.
Number of Variables with Missing Value Differences: 2.
Total Number of Values which Compare Unequal: 20.
Maximum Difference: 0.

All Variables Compared have Unequal Values

Variable	Type	Len	Compare	Len	Ndif	MaxDif	MissDif
Weight	NUM	8	wait	8	10	0	10
Weight	NUM	8	waite	8	10	0	10

Value Comparison Results for Variables

Name	Base Weight	Compare wait	Diff.	% Diff
Alfred	112.5000	.	.	.
Henry	102.5000	.	.	.
James	83.0000	.	.	.

Check for Format Ranges

```
proc format lib=work;
value $zipst
  '01000' - '02799' = 'MASSACHUSETTS'
  '02800' - '02999' = 'RHODE ISLAND'
  '03000' - '03899' = 'NEW HAMPSHIRE'
  '03900' - '04999' = 'MAINE'
  '05000' - '05999' = 'VERMONT'
  '06000' - '06999' = 'CONNECTICUT'
;
quit;

proc format lib=work cntlout=format_data; quit;
title1 "Check for Format Ranges";
proc compare base=format_data;
id fmtname label;
var start;
with end;
run;
```

Check for Format Ranges

The COMPARE Procedure

Comparisons of variables in WORK.FORMAT_DATA (Method=EXACT)

Data Set Summary

Dataset	Created	Modified	NVar	NObs
WORK.FORMAT_DATA	29OCT07:11:35:24	29OCT07:11:35:24	21	6

Values Comparison Summary

Number of Variables Compared with All Observations Equal: 0.
Number of Variables Compared with Some Observations Unequal: 1.
Total Number of Values which Compare Unequal: 6.

All Variables Compared have Unequal Values

Variable	Type	Len	Compare	Len	Label	Compare Label	Ndif	MaxDif
START	CHAR	5	END	5	Starting value for format	Ending value for format	6	

Value Comparison Results for Variables

FMTNAME	LABEL	START	END
		Starting value for format	Ending value for format
		Base Value	Compare Value
ZIPST	MASSACHUSETTS	01000	02799
ZIPST	RHODE ISLAND	02800	02999
ZIPST	NEW HAMPSHIRE	03000	03899
ZIPST	MAINE	03900	04999
ZIPST	VERMONT	05000	05999
ZIPST	CONNECTICUT	06000	06999

Connecticut Airport Data

Source: <http://www.airnav.com>

City	FAA	ICAO	Airport	Elev	Longest Runway
Bridgeport	BDR	KBDR	Igor I. Sikorsky Memorial Airport	9	4677
Danbury	DXR	KDXR	Danbury Municipal Airport	458	4422
East Haddam	42B		Goodspeed Airport	9	2120
Ellington	7B9		Ellington Airport	253	1800
Groton/New London	GON	KGON	Groton-New London Airport	9	5000
Hartford	HFD	KHFD	Hartford-Brainard Airport	18	4417
Marlborough	9B8		Salmon River Airfield	540	2000
Meriden	MMK	KMMK	Meriden-Markham Municipal Airport	103	3100
New Haven	HVN	KHVN	Tweed-New Haven Airport	14	5600
New Milford	11N		Candlelight Farms Airport	675	2900
Oxford	OXC	KOXC	Waterbury-Oxford Airport	726	5800
Plainville	4B8		Robertson Field	200	3612
Putnam	C44		Toutant Airport	770	1756
Simsbury	4B9		Simsbury Airport (Simsbury Tri-Town Airport)	195	2205
Warehouse Point	7B6		Skylark Airpark	120	3242
Waterbury	N41		Waterbury Airport (Waterbury-Plymouth Airport)	850	2005
Willimantic	IJD	KIJD	Windham Airport	247	4278
Windsor Locks	BDL	KBDL	Bradley International Airport	173	9510

Long Character Strings

The COMPARE Procedure

Comparison of WORK.AIRPORTS1 with WORK.AIRPORTS2 (Method=EXACT)

Values Comparison Summary

Number of Variables Compared with All Observations Equal: 3.

Number of Variables Compared with Some Observations Unequal: 2.

Total Number of Values which Compare Unequal: 5.

Maximum Difference: 10.

Variables with Unequal Values

Variable	Type	Len	Label	Ndif	MaxDif
Airport	CHAR	60		3	
Runwaylng	NUM	8	Longest Runway	2	10.000

Value Comparison Results for Variables

	Base Value	Compare Value
FAA	Airport	Airport
	_____+	_____+
4B9	Simsbury Airport (Si	Simsbury Airport (Si
HVN	Tweed-New Haven Airp	Tweed-New Haven Aire
N41	Waterbury Airport (W	Waterbury Airport (W

Issue: Only the first 20 Chars are shown; 12 if using the TRANSPOSE option.

Long Character Strings

```
proc sort data=airports1; by faa; run;  
proc sort data=airports2; by faa; run;
```

```
proc compare base=airports1 compare=airports2 out=comp_airp  
             noprint outnoequal outbase outcomp outdif;  
id faa;  
var city airport runwyln;g;  
run;
```

```
proc print noobs data=comp_airp;  
id faa;  
var runwyln;g airport;  
run;
```

Long Character Strings

FAA	Runwyng	Airport
4B9	2205	Simsbury Airport (Simsbury Tri-Town Airport)
4B9	2215	Simsbury Airport (Simsbury Tri-Towne Airport)
4B9	10XXXXXXXXXX.....
HVN	5600	Tweed-New Haven Airport
HVN	5601	Tweed-New Haven Aireport
HVN	1XXXXX.....
N41	2005	Waterbury Airport (Waterbury-Plymouth Airport)
N41	2005	Waterbury Airport (Waterbury-Plimoth Airport)
N41	EX..XXXXXXXXXXXX.....

OUTDIF prints:

A . for matching characters variables; An X for non-matching character vars.

An E for Matching numeric variables; the difference for non-matching numerics.

Proc Compare Recap

- If the variables between the two data sets have different names, use VAR and WITH options to specify which variables to match to. Or rename variables on either the BASE= or COMPARE= option.
- When comparing two data sets, the observations (or ID variables being matched) need to appear in the same order. Variables do not have to be in the same order.
- ID variables come in handy when identifying non-matching observations.
- An ERROR message is issued when the data sets aren't sorted by the ID variables.
- Use the BRIEF option to suppress the summary reports, if you don't need them.
- NOVALUES is helpful for reducing printed output, if you don't need to see details of non-matching observations.
- LISTVAR shows which variables are in one data set but not the other.
- LISTOBS shows which observations are in one data set but not the other.
- TRANSPOSE can be helpful when tracking specific observations.
- You can compare variables within a data set with Proc Compare.
- Seeing “All values compared are exactly equal.” does not always mean the two data sets are exactly identical. You need to look at the total number of observations being compared.
- When comparing long strings, create an output data set and review it.

About the Speaker

Rob Krajcik
Principal Analyst II

Bristol-Myers Squibb
5 Research PKWY
Wallingford, CT 06492-7660

(203) 677-6125 (phone)
(203) 677-6197 (fax)

robert.krajcik@bms.com