

# %COMPARWS: Compare with summary: a macro using proc COMPARE to write a file of differences to edit and use for updates

Ronald Fehd, Centers for Disease Control and Prevention, Atlanta GA

## ABSTRACT

Data review can be viewed as a two-step process:

1. Compare: review differences between two data sets.
2. Exception Report: review data consistency with range and logic checks.

A data set may be updated with a file containing sets of these three statements -- ID, assignment and closure:

```
* if ID = 1 then do;
* <variable name> = <value>;
* end;
```

This paper reviews the proc COMPARE output data set and the manipulation of that data set which is necessary to write a report to a file which contains update statements in an easily editable form. Summary information is written at the end of the update file.

The output file is in this form:

```
if ID = 1 then do;
* VAR_A = 'A4Q3J';
* VAR_A = 'A4Q8J';
* dif = '...X.';
end;
```

where 'X' in the value of Dif indicates the position of difference of the values in the Base and Compare data sets.

## INTRODUCTION

Data review is a necessary step in data management. Its purpose is to provide accurate and consistent data for data analysis. SAS® software provides proc COMPARE as a useful tool to identify differences between two data sets. These data sets may be the result of two independent data entry steps or the results of scanning using different degrees of optical character recognition or one of each of these two processes.

With large survey data sets reviewing, indicating, and making corrections using a printed report from proc COMPARE can be tedious in two areas:

1. Low information per page: many pages in a report
2. Transferring corrections from paper to an update file necessitates retyping variable names and values; this, in turn, requires an edit-and-correct cycle.

This paper reports on a utility macro which fulfills these criteria:

1. A concise report per observation, easy to read and edit.
2. A difference report is written to file for later editing and subsequent use as an %included update file.
3. A summary report provided for data entry process review.

For a discussion of exception reports see Fehd 1998, DEMOXRPT.

### Assumptions about the data sets to be compared

Pairs of data sets to be used with this macro are expected to have the same structure, identifiers (ID), and number of observations.

The test in step 1 checks the ID list and exits if it is blank. The %NOBS macro is based on the %OBSNVAR macro (SAS Macro Language Reference, First Edition). In step 2 %NOBS provides the number of observations of the base and comparison data sets. Step 3 compares the number of observations in both data sets and exits if they are unequal. It is assumed that both data sets are sorted and unique on the IDs, else proc COMPARE will fail.

### Specifications of a correction file to be used for updates

In order for a text file to be usable as an %included update file, it must be written in the following format:

```
* if ID = 1 then do;           *1 ID;
* variable = <value>;       *2 variable assignment;
* end;                       *3 ID closure;
```

1. The ID line opens an if-do statement. This example shows a single numeric id. If the ID is character then the ID value must be enclosed in quotes. Similarly, if other secondary IDs are character, they must likewise be quoted. The first specification is:

*1. quote ID values, according to their type.*

This is accomplished by using the proc CONTENTS of the base data set and saving the type in a macro variable. Step 4 uses %ARRAY (Fehd 1997) to create macro variables of each of the IDs. In step 5 the type of each ID is saved as a macro variable.

2. This procedure was written to eliminate the errors associated with retyping variable names and values. As with IDs, variables listed must be quoted, or not, according to their type. This information is available in the proc COMPARE output data set and is used in the report writer in step 10. The first part of the variable assignment specification is:

*2.1 quote difference values, according to their type.*

This report lists two different values of the same variable, one from the base data set, the other from the comparison data set. Visual comparison of short sequences of letters or digits to locate differences is relatively easy. For sequences of five or more characters it may be necessary to offer a visual indicator of the position of the difference. The second part of the variable assignment specification is:

*2.2 for length greater than or equal to five, indicate where difference is*

3. Every open control statement must have closure. The last specification is:

*3. provide an end statement.*

These specifications are relatively simple to implement in a data \_NULL\_ report writer. We now review proc COMPARE and its options with a view to producing a data set which is usable by the report writer.

## Proc COMPARE: report and output data set

The prototype for this macro is the proc COMPARE Value Comparison Results report (SAS® *Procedures Guide, Version 6, Third Edition*). The transpose option is used to report by observation or ID. This report is paged by ID, which produces a minimum of one page per ID, no matter the number of variables reported different.

In step 6 proc COMPARE creates an output data set. The option *outall* writes four observations per ID with any difference. These are identified by the variable *\_Type\_*.

<i>_TYPE_</i>	explanation
-----	-----
BASE	value in base data set
COMPARE	value in compare data set
DIF	difference marker for character
PERCENT	difference of numeric variable

In step 7 the number of observation of the output data set is checked. If there are no differences, a message is written to the output file and the processing is ended.

Step 8: The difference data set is transposed and then made unique on the differences. This process produces the data set for the report writer.

### DATA \_NULL\_ report writer

Step 9: Values used in summary are created.

Step 10: The report writer now has a data set with IDs which can be used with first- and last-ID processing. This partially fulfills specifications 1 and completely fulfills specification 3. Macro Q adds appropriate quotes to the identifiers for the fulfilment of the rest of specification 1.

Variables are identified as either character or numeric by comparing the values of Dif and Percent. Appropriate quotes are provided when writing the detail lines; this fulfills specification 2.1. Specification 2.2 is met by an optional comment line which prints the Dif variable containing an 'X' at the position of difference.

### Summary information

This report provides several types of summary information. At the end of the print file is the following data: the number of IDs, variables and cells in the base and difference data sets, and percentages of each. This data is intended as feedback to the data entry process. The percentage of cells reported different is the data entry error rate.

An example of detail with summary data, from test data. Data entry error rate is at lower right: Pcnt : Cells

```

*;if NID1 = 3 then do;
* D3      = '.' ;
* D3      = '!' ;
*;end;*NDiff = 1 ;
**smry :  IDs      Vars      Cells;
** Base:   4        8        32;
** Diff:   3        4         4;
** Pcnt:   75%     50%     12.5%;

```

Other reports are printed only when a PRNTPFILE is provided. Summary of Ids: proc FREQ of IDs shows IDs in the comparison

report. This is used by data review staff to pull data collection forms. Summary of Variables: proc FREQ of Name shows which variables have most differences. Summary Crosstab: proc FREQ of Compare \* Base shows standardization of differences across the data set. Detail, by Variable: prints the COMPARWS data set sorted by variable Name. Each report is enabled by a macro parameter. The macro may be stored with any set of reports enabled.

### Usage:

In a production setting this macro is first used to write and print both a correction file and a paper report. The paper report is reviewed by colleagues responsible for data collection. The marked up copy is returned and the desired corrections are enabled in the file by removing the asterisks in front of the appropriate statements. These updates are applied to the data and a third data set created. The macro is then used to write only a paper report, which is used to verify that all indicated corrections were made.

```

Program COMPAR12: report differences
filename COMPRPT <file-spec>;
%COMPAREWS ( LIBRARY . <DATA1>
, LIBRARY . <DATA2> , IDNMBR
, PRNTPFILE=COMPRPT ) ;
note: writes report to file, prints report for review.

```

Process: review report, note corrections to be made.

Process: edit file: remove asterisks to enable corrections.

```

Program DATA3: apply corrections
DATA <DATA3>;
set <DATA2>;
%include COMPRPT;

```

```

Program COMPAR23: confirm corrections
%COMPAREWS ( LIBRARY . <DATA2>
, LIBRARY . <DATA3> , IDNMBR ) ;
note: print report to review and confirm corrections.

```

## CONCLUSION

Proc COMPARE provides a voluminous report of differences between two data sets. This routine uses proc COMPARE's output data set to provide a concise difference report which may be saved as a text file or printed for review. When corrections are necessary, the text file is edited instead of retyping variable names and values. Using the text file to update the data set eliminates the edit-and-correct cycle associated with manually retyping variables and values.

## REFERENCES

Fehd, Ronald (1997), "%ARRAY, construction and usage of arrays of macro variables," *Proceedings of the Twenty-Second Annual SAS Users Group International Conference*, 22, 447-450.

Fehd, Ronald (1998), "DEMOXRPT: macros for writing Exception Reports: perform range and logic checks on a data set; write file of exceptions to edit and use for updates" *Proceedings of the Twenty-Third Annual SAS Users Group International Conference*, 23.

SAS Institute Inc. (1997), *SAS® Macro Language Reference, First Edition*, Cary, NC: SAS Institute Inc.

SAS Institute Inc. (1990), *SAS® Procedures Guide, Version 6, Third Edition*, Cary, NC: SAS Institute Inc.

SAS is a registered trademark of SAS Institute, Inc. In the USA and other countries, © indicates USA registration.

**Author:** Ronald Fehd                   e-mail: RJF2@cdc.gov  
**Centers for Disease Control** PHPPPO/DLS MS-G25  
**Atlanta GA 30341-3724**                   voice: 770/488-4316  
**SAS-L archives:** send e-mail  
**to:** SAScontrib@SASserv.uga.edu  
**for %ARRAY**                   subject: cntb0031: download  
**for %COMPARWS**               subject: cntb0034: download  
**for %NOBS**                   subject: cntb0038: download  
**for DEMOXRPT**               subject: cntb0039: download

```
%MACRO NOBS(_MAC_VAR,DATA=.,_GLOBAL_=0);run;
%IF "&_MAC_VAR" = "" %THEN %LET _MAC_VAR = NOBS;
%IF &_GLOBAL_ %THEN %DO; %global &_MAC_VAR.; %END;
%IF &DATA = . %THEN %LET DSN = &SYSLAST; %*resolve mac-vars in name;
%ELSE %LET DSN = &DATA.;
%LET DSID = %sysfunc(open(&DSN));
%IF &DSID %THEN %DO;
%LET &_MAC_VAR. = %sysfunc(attn(&DSID,NOBS));
%LET NVARS = %sysfunc(attn(&DSID,NVARS));
%put NOBS returns nvars: "&NVARS"=<&&NVARS>;
%LET RC = %sysfunc(close(&DSID)); %END;
%ELSE %put Open for data set &DATA. failed - %sysfunc(sysmsg());
%put NOBS: "&_MAC_VAR"=<&&_MAC_VAR> data=&DSN; %*... NOBS*; %MEND;
/* ***** */
* MACRO: COMPARWS                               NOTE: uses macros ARRAY, NOBS
*
*
* USAGE: 1. %COMPARWS(DATA1,DATA2,IDO);
*       2. %COMPARWS(DATA1,DATA2,ID1 ID2);
*       3. %COMPARWS(DATA1,DATA2,IDO,PRNTFILE=<fileref>);
*       4. %COMPARWS(DATA1,DATA2,IDO,TESTING=1);
*       5. %COMPARWS(DATA1,DATA2,IDO,DROPLIST=Var1 Var2 .. VarN);
*       6. %COMPARWS(DATA1,DATA2,IDO,READBASE=XXXX,READCOMP=YYYY);
*       7. TITLE1 'title for this project';
*       %COMPARWS(DATA1,DATA2,IDO,TITLEN=2);
*
* DESCRIPTION: use proc COMPARE output data set
*               to write a file of differences
*               to be used as an %INCLUDE file for updates
*               and summary report(s) of differences
*
* PROCESS: 1. if IDLIST blank, print error-msg . . . . . exit *
*       2. %NOBS of BASE & COMPARE for error-check *
*       3. if error-checks, print error-msg . . . . . exit *
*       4. %ARRAY of ID(s) *
*       5. vars in IDLIST (character? numeric?) *
*       6. proc COMPARE out = COMPARWS *
*       7. %NOBS, if no differences, print exit-message . . . exit *
*       8. proc TRANSPOSE, rotate and unique date set *
*       9. make values used in summary *
*       10. write corex to PRNTFILE *
*       11. if wanted, print summary report(s) *
*
* NOTES: width of variables, in COMPARE output data set
*       depends on largest character variable in input data set
*       DROPLIST is provided to delete wide variables from comparison*
*!> >>>-> This routine will fail for large numbers of differences *!>
*!> where data set has one wide character variable *!>
*!> which increases the size of the COMPARE output data set *!>
*
* KEYWORDS: %ARRAY %NOBS COMPARE TRANSPOSE
*            corrections updates "output data set from procedure"
*
* Author: Ronald Fehd, B.S. C.Sc                   e-mail: RJF2@cdc.gov *
*       Centers for Disease Control PHPPPO DLS MS-G25 *
*       Atlanta GA 30341-3724                   voice: 770/488-4316 *
* * * * *
%MACRO COMPARWS(/*)
BASE       /* BASE     for proc COMPARE                   */
,COMPARE   /* COMPARE for proc COMPARE                 **
          ** either or both may be 2-level names       */
,IDLIST    /* list of primary-key(s) == by-vars         **
,PRNTFILE=PRINT/*output destination, default = PRINT   */
          /*may be 'external-file' -- note quotes -- or fileref */
,DROPLIST= /*list of vars to be dropped from comparison */
          /*recommend drop very wide character variables: len gt 12 */
,READBASE= /*password for BASE                         */
,READCOMP= /*password for COMPARE                     */
,SHRYIDS=1/*?print FREQ of IDS?                       */
,SHRYNAME=0/*?print detail of variables, by name?     */
,SHRYVARS=0/*?print FREQ of variables with differences? */
,SHRYXTAB=0/*?print FREQ cross-tabulation of Compare * Base? */
,TESTING=0/*?enable test msg and prints?             */
,TITLEN =1/*TITLE line #                               */
)/* autocall macro: options MautoSource Mstored SASMStore=LIBRARY; */
/store des = 'proc COMPARE w/summary of differences'/*
/*-----*/
```

**ACKNOWLEDGMENTS**

These routines were developed over a period of ten years while I crunched the numbers of survey data collected by the Model Performance Evaluation Program (MPEP) of the Division of Laboratory Systems, Public Health Practice Program Office of the Centers for Disease Control, and Prevention, Atlanta, Georgia. I wish to thank Sharon Blumer, David Cross, Thomas Hearn, and William Schalla of the MPEP group for their help in developing this routine. John Hancock, chief, Information Services Activity, encouraged me to write up these routines.

```
TITLE&TITLEN. "comparison of &BASE. and &COMPARE.";
%*1. if IDLIST blank, print err-msg, exit;
%IF &IDLIST = %THEN %DO; %put !!!!!!!!!!!!!!!!;
%put !! no ID-List !!; %put !!!!!!!!!!!!!!!!; %GOTO ENDOMACR; %END;
%*2. %NOBS of BASE & COMPARE for error-check;
%local BASENOBS; %NOBS(BASENOBS,data = &BASE );
%local COMPNOBS; %NOBS(COMPNOBS,data = &COMPARE);run;
%* NOBS mac-vars are local to %COMPARWS;
%*3. if err-checks print err-msg, exit;
%IF &BASENOBS ne &COMPNOBS/*-----*/
or not &BASENOBS
or not &COMPNOBS %THEN %DO; DATA _NULL_;
file &PRNTFILE;
put
%IF &BASENOBS ne &COMPNOBS %THEN ****number of obs differ, "
" &BASE. = &BASENOBS. "
" &COMPARE. = &COMPNOBS.;" /;
****no obs in &BASE.; " /;
****no obs in &COMPARE.; " /;
/*closure: put*/;
%GOTO ENDOMACR; %*.....; %END;
%*4. %ARRAY of ID(s): for use in macro loops;
%local I DIM_IDS;%LET DIM_IDS = 3;
%DO I = 1 %TO &DIM_IDS; %local IDS&I; %END;
%LET IDLIST = %upcase(&IDLIST.); %* proc CONTENTS.Name is upcase *;
%ARRAY(IDS,&IDLIST);run; %*ARRAY mac-vars are local to COMPARWS;
%*5. vars in IDLIST (character? numeric?);
proc CONTENTS data = &BASE.
%IF "&READBASE" ne "." %THEN (read=&READBASE.);
out = BASECONT (keep = Name Type) noprint;
DATA _NULL_;
length C_N $ 1;
do until(EndoFile);
set BASECONT end = EndoFile; %*-----*
*Type == (1=numeric, 2=character) from proc CONTENTS*;
if Type = 1 then C_N = 'N';
else C_N = 'C';
%DO I = 1 %TO &DIM_IDS;
if Name = "&IDS&I." then call symput("TYPE&I.",C_N); %END;
%*..... do until(EndoFile)*; end;
stop;
%*6; proc COMPARE
base = &BASE. (%IF "&DROPLIST" ne "." %THEN drop=&DROPLIST.;
%IF "&READBASE" ne "." %THEN read=&READBASE.);
compare = &COMPARE. (%IF "&DROPLIST" ne "." %THEN drop=&DROPLIST.;
%IF "&READCOMP" ne "." %THEN read=&READCOMP.);
out = COMPARWS
outall /* == outBASE outCOMP outDIF outPERCENT */
outnoequal /* write only obs with one or more differences*/
noprint;
by &IDLIST.;
%macro TESTPRNT(DATA,MSG); %*-----*
%IF &TESTING %THEN %DO; proc PRINT data=&DATA;
TITLE eval(&TITLEN.+1) "data=&DATA.: &MSG";
proc CONTENTS data = &DATA.;
run; TITLE eval(&TITLEN.+1); %END; %MEND;
%TESTPRNT(COMPARWS,output from proc COMPARE);
%*7. %NOBS, if no differences print exit-msg, exit;
%local ANYDIFF; %NOBS(ANYDIFF);run;
%IF not &ANYDIFF %THEN %DO; DATA _NULL_;
file &PRNTFILE;
```

```

                put '***no differences;';
                %GOTO ENDOMACR;                                %END;

%*8. TRANSPOSE: rotate & unique data to get one obs with one difference;

proc TRANSPOSE data = COMPARWS (drop = _Obs_)
              out = COMPARWS (drop = _Label_);
  id _Type_;
  by &IDLIST;
  var _all_;          %TESTPRT(COMPARWS,transposed w/all variables);

DATA COMPARWS;
do until(EndoFile);
  set COMPARWS end = EndoFile;
  where Base ne Compare and _Name_ ne '_TYPE_';
  output; %*..... do until(EndoFile)*; end;
stop;          %TESTPRT(COMPARWS,transposed w/only differences);

%*9. make values used in summary;

proc FREQ data = COMPARWS;%do cross-tab of IDs *;
  tables
    &IDS1
  %IF &DIM_IDS ge 2 %THEN
    %DO I = 2 %TO &DIM_IDS;          * &&IDS&I.          %END;
    / list noprint out = DIFFFIDS;
  tables _Name_ / list noprint out = DIFFVARS;

%local DIFFFIDS ; %NOBS(DIFFFIDS,data = DIFFFIDS );
%local DIFFVARS ; %NOBS(DIFFVARS,data = DIFFVARS);
%local DIFFCFLS ; %NOBS(DIFFCFLS,data = COMPARWS);
%local BASEVARS ; %NOBS(BASEVARS,data = BASECONT);run;

%*10. write corex to PRNTFILE; options pageno=1;

DATA %IF "&PRNTFILE" eq "PRINT" %THEN _NULL_ ;
      %ELSE COMPARWS; /*closure: DATA*/ ;
drop B Q C1 C2 C3 Dif Percent NDiff;
length Q $ 1;
retain B -1 NDiff 0 N 1;%*N: counter for proc PRINT sum*;
file &PRNTFILE;
do until(EndoFile);
  set COMPARWS end = EndoFile;
  by &IDLIST;

  if Dif ne Percent then Q = " ";/** == var is numeric **/
  else Q = " ";/** == var is char **/

  %MACRO Q(N);%IF &&TYPE&N. = C %THEN " ";/*return quote*/ %MEND;

  if first.&&IDS&DIM_IDS then do; NDiff = 0; /*.....*/
    put " ";if &IDS1 = " %Q(1) &IDS1 +B %Q(1)
    %IF &DIM_IDS. ge 2 %THEN
      %DO I = 2 %TO &DIM_IDS.; " and &&IDS&I = " %Q(&I) &&IDS&I +B %Q(&I)
      %END;
      " then do;"; %*.....*if first.*; end;
  NDiff + 1;
    put ' ' _Name_ @13 '=' @15 Q +B Base +B Q ' ';
    / ' ' _Name_ @13 '=' @15 Q +B Compare +B Q ' ';
  if length(Dif) gt 5
  then put ' * differnc' @13 '=' @15 Q +B Dif +B Q ' ';

  if last.&&IDS&DIM_IDS then put ' *;end;*NDiff = ' NDiff ' ';
  %IF "&PRNTFILE" ne "PRINT" %THEN %DO; output; %END;
  %*..... do until(EndoFile)*; end;

%*11. print summary percentages at end of difference file;

%macro MAKEPCNT(VAR,NUM,DENOM);%create %, trim value to &D decimals -*;
%local D I TRIMLEN; %LET D = 3; %LET TRIMLEN = 0;
%LET &VAR = %sysevalf(100 * &&NUM / &&DENOM);
%LET I = %index(&&VAR,.);%*if dot in value trim number of decimals;
%IF &I %THEN %DO; %LET TRIMLEN = %val(%length(&&VAR) - &I);
%IF &TRIMLEN gt &D %THEN %LET TRIMLEN = &D;
%LET &VAR = %substr(&&VAR,1,&I + &TRIMLEN); %END;
%*PUT &VAR = &&VAR;%*.....*;%MEND;

```

```

%local BASECELS;%LET BASECELS = %eval(&BASENOBS * &BASEVARS);
%local PCNTNOBS;%MAKEPCNT(PCNTNOBS,DIFFFIDS ,BASENOBS);
%local PCNTVARS;%MAKEPCNT(PCNTVARS,DIFFVARS,BASEVARS);
%local PCNTCELS;%MAKEPCNT(PCNTCELS,DIFFCFLS,BASECELS);
retain C1 12 C2 22 C3 32;
put ****smry : " @C1 'Ids' @C2 'Vars' @C3 'Cells';";
put **** Base:" @C1 '&BASENOBS' @C2 '&BASEVARS' @C3 '&BASECELS';";
put **** Diff:" @C1 '&DIFFFIDS' @C2 '&DIFFVARS' @C3 '&DIFFCFLS';";
put **** Pcnt:" @C1 '&PCNTNOBS%' @C2 '&PCNTVARS%' @C3 '&PCNTCELS%';";
stop;

%IF "&PRNTFILE" ne "PRINT" %THEN %DO;%*.....;
%*11. print desired summary report(s);options pageno=1;

%IF &SMRYIDS %THEN %DO;%*from step 9;
proc PRINT data = DIFFFIDS;sum Count;
  TITLE%eval(&ITITLE.+1) "summary: IDs listed"; %END;

%IF &SMRYVARS %THEN %DO;%*from step 9;
proc PRINT data = DIFFVARS;sum Count;
  TITLE%eval(&ITITLE.+1) "summary: Variables listed"; %END;

%IF &SMRYXTAB %THEN %DO;
proc FREQ data = COMPARWS;
  tables Compare * Base / list missing;
  TITLE%eval(&ITITLE.+1) "summary: Compare * Base"; %END;

%IF &SMRYNAME %THEN %DO;
proc SORT data = COMPARWS;
  by _Name_ Base Compare &IDLIST.;
proc PRINT data = COMPARWS;sum N;
  by _Name_; id _Name_;
  TITLE%eval(&ITITLE.+1) "detail: by Var-Name"; %END;

data _NULL_ ;%*print detail list*; options pageno=1;
  TITLE%eval(&ITITLE.+1);
  file PRINT;
  do until(EndoFile);%local LRECL;%LET LRECL = 72;
    infile &PRNTFILE end = EndoFile pad recl = &LRECL.;
    input @1 Line Schar&LRECL.;
    put @1 Line Schar&LRECL.; %*do until(EndoFile)*; end;
stop; %*..... %IF "&PRNTFILE" ne "PRINT"; %END;
%ENDOMACR: run;TITLE%eval(&ITITLE.);%*.....*;%MEND;
/* test data section: enable by ending this line with slash (/) ----**
DATA A B; input @1 nID1 1. @1 ID1 $CHAR1.
           @2 nID2 1. @2 ID2 $CHAR1.
           @3 N1 3. @3 D1 $CHAR22.
           @33 D2 $CHAR22. @33 D3 $CHAR1.; output A;

  if ID2 = '0' then substr(D2,21,1) = '@';
  if ID2 = '2' then do; substr(D1,21,1) = '@';
    N1 = 124; end;
  if ID2 = '3' then D3 = '!'; output B; cards;
1123.....
12123.....
20123.....
33323.....
/*test usage: enable by ending this line with slash (/) ----**
options uprint nocenter;
%COMPARE(A,B, ); %*test error when no by-list;
%COMPARE(A,B,NID1,TESTING=1);%*show intermediate data sets;
%COMPARE(A,B,NID1); %*test numeric ID;
%COMPARE(A,B, ID1); %*test character ID;
%COMPARE(A,B, ID1 ID2 ); %*2 char IDs;
%COMPARE(A,B,NID1 NID2 ); %*2 numeric IDs;
%COMPARE(A,B,NID1 ID2 NID2); %*3 IDs: combination of numeric + char;
%COMPARE(A,B,NID1,PRNTFILE=SASAUTOS(ZCOMPARE));%*OpSys=MVS;
*filename ZCOMPARE 'c:\sas\zcompare.txt'; %*OpSys=WIN;
%COMPARE(A,B,NID1,PRNTFILE=ZCOMPARE);
/*example usage of file created by this program as an update file ****
*data NEW;
* set OLD;
* %include <fileref> / source2;
/*.....*/

```