ABSTRACT

Although version control on the files, datasets or any document can be challenging, COMPARE® Procedure provides an easy way to compare two files and indicate the differences between them. The paper utilizes the comparison results from PROC COMPARE® and builds it into a SAS® macro to highlight changes between files in terms of addition, deletion or an update to a record in a convenient excel format. Some common examples where this utility can be useful is comparing CDISC Controlled Terminology (CT) release versions, comparing Medical dictionary versions like MedDRA, WHODrug, or comparing certain Case Report Form (CRF) data like Adverse Events (AE) to review new events being reported at various timepoints for data monitoring purposes.

INTRODUCTION

COMPARE® Procedure is widely used to exhibit differences between two files. It has several useful options to control the results that are of interest to the user. The macro %diff_report discussed in the paper utilizes PROC COMPARE results along with some data manipulations to produce a very comprehensive excel report using the ODS EXCEL® system.

MACRO DESCRIPTION

The macro definition is as below:

```sas
%diff_report(newds = , /* Name of the current or latest dataset */
             oldsds = , /* Name of the previous dataset to compare with */
             unqkey = , /* Unique list of variables to identify records */
             vars2cmp = , /* List of variables to compare, default is ALL */
             outdsn = /* Name of the output dataset or excel file */
            );
```

Here the parameter “unqkey=” assumes a list of variables [separated by space (“ “) as delimiter] which are able to uniquely identify the records in the datasets. These variables support the comparison in the form of ID variables in COMPARE Procedure. If no unique key(s) is provided, the program will abort and request to re-assess this parameter again.

The other parameter “vars2cmp=” takes a list of variables [separated by space (“ “) as delimiter] on which the comparisons are to be made. If no value is provided, then it includes all the variables in the dataset for comparison.

Sample call looks like:

```sas
%diff_report(newds = NEW,
             oldsds = OLD,
             unqkey = Var1 Var2 Var3,
             vars2cmp = , /* default is ALL */
             outdsn = DIFF
            );
```

```sas
%diff_report(newds = NEW,
             oldsds = OLD,
             unqkey = Var1 Var2 Var3,
             vars2cmp = Var4 Var5 Var6 Var7,
             outdsn = DIFF
            );
```
MACRO REPORTS

Sample report created from the macro is shown below (Display 1 & Display 2)

<table>
<thead>
<tr>
<th>Type</th>
<th>CodeList Code</th>
<th>Code</th>
<th>Extensible</th>
<th>CT Name [Short]</th>
<th>CT Name [Long]</th>
<th>CDISC_Submission_Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Added</td>
<td>C100129</td>
<td>C125795</td>
<td>Yes</td>
<td>QSCAT</td>
<td>Category of Questionnaire</td>
<td>COPD ASSESSMENT TEST</td>
</tr>
<tr>
<td>Added</td>
<td>C100129</td>
<td>C125795</td>
<td>Yes</td>
<td>QSCAT</td>
<td>Category of Questionnaire</td>
<td>ASEX-MALE</td>
</tr>
<tr>
<td>Added</td>
<td>C100129</td>
<td>C125795</td>
<td>Yes</td>
<td>QSCAT</td>
<td>Category of Questionnaire</td>
<td>ASEX-FEMALE</td>
</tr>
<tr>
<td>Deleted</td>
<td>C85047</td>
<td>C81019</td>
<td>Yes</td>
<td>LTESTCD</td>
<td>Laboratory Test Code</td>
<td></td>
</tr>
<tr>
<td>Deleted</td>
<td>C85047</td>
<td>C74753</td>
<td>Yes</td>
<td>LTESTCD</td>
<td>Laboratory Test Code</td>
<td></td>
</tr>
<tr>
<td>Update</td>
<td>C85491</td>
<td>C89926</td>
<td>Yes</td>
<td>MICROORG</td>
<td>Microorganism</td>
<td>SALMONELLA ENTERICA SUBSP. ENTERICA SEROVAR TYPHI</td>
</tr>
</tbody>
</table>

Display 1: Comparison between CDISC Controlled Terminology (CT) versions

Display 1 highlights records which were added, deleted and updated when comparing two versions of the CDSIC CT file.

<table>
<thead>
<tr>
<th>Type</th>
<th>SubjectID</th>
<th>Site</th>
<th>Aeterm</th>
<th>Aestdtc</th>
<th>Aenstdtc</th>
<th>Aetoxgr</th>
</tr>
</thead>
<tbody>
<tr>
<td>Update</td>
<td>A12345</td>
<td>100</td>
<td>Nausea</td>
<td>7/10/2018</td>
<td>7/15/2018</td>
<td>Mild</td>
</tr>
<tr>
<td>Added</td>
<td>A12345</td>
<td>100</td>
<td>Myalgia</td>
<td>8/21/2018</td>
<td>8/27/2018</td>
<td>Mild</td>
</tr>
<tr>
<td>Added</td>
<td>A12345</td>
<td>100</td>
<td>Bleeding</td>
<td>9/8/2018</td>
<td>9/6/2018</td>
<td>Moderate</td>
</tr>
<tr>
<td>Added</td>
<td>A12345</td>
<td>100</td>
<td>Abdominal Pain</td>
<td>10/1/2018</td>
<td>10/10/2018</td>
<td>Mild</td>
</tr>
<tr>
<td>Added</td>
<td>A12345</td>
<td>100</td>
<td>Congestion</td>
<td>11/1/2018</td>
<td>11/2/2018</td>
<td>Mild</td>
</tr>
</tbody>
</table>

Display 2: Comparison between Adverse events data versions

Display 2 highlights records which got modified when comparing two Adverse Events (AE) datasets at different timepoints.

A new column “Type” is added as the 1st column to indicate the type of change noted while comparing the two files. Some visual enhancements are made with the background color corresponding to the type of change and a comment box (SAS style- “flyover”) is added to display the old value in comparison with the current value for a given cell when the type of change is “Update”.

Please refer to APPENDIX for the program flow to create these reports.

CONCLUSION

The paper discussed a SAS macro which can compare two files and highlight changes between them in an easy to read excel file. Users are encouraged to enhance this macro to compare multiple datasets at the same time when stored at different sub-directories.

REFERENCES


CONTACT INFORMATION

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

Name: Abhinav Srivastava
Enterprise: Gilead Sciences, Inc.
E-mail: srivastvaabhinav@gmail.com

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**

```sas
%macro diff_report(newds    = ,    /* Name of the NEW dataset */
                   oldds    = ,    /* Name of the OLD dataset */
                   unqkey   = ,    /* List of key variables */
                   vars2cmp = ALL, /* List of variables to compare */
                   outdsn   =      /* Name of the output dataset */
);
 /*=== Step 1: Add Checks ===*/
%* 1) Check all required Input parameters are provided;
<.. SAS code ..>;

%* 2) Check if &unqkey identifies records uniquely, otherwise ABORT;
<.. SAS code ..>;

%* 3) Check if &var2cmp parameter is provided or defaulted to ALL;
*     - If defaulted to ALL, &vars2cmp stores all variables for comparison
   EXCEPTION those listed in &unqkey
<.. SAS code ..>;

/*= Step 2: Run Compare (Restrict variables to &unqkey and &vars2cmp only)=*/
proc compare data=&newds  comp=&oldds
   out=outcomp outnoequal outbase outcomp outdif noprint;
   id &unqkey;
   var &vars2cmp;
run;

/*=== Step 3: Check for Added/Deleted Records ===*/
%* 1) Create a macro variable "lastkey" which holds the last variable from
   &unqkey. Example: If unqkey=var1 var2 var3 then lastkey is 'var3' ;
%* 2) Sort the dataset by &unqkey and _type_;
   _TYPE_ is created by COMPARE Procedure with values-BASE, COMPARE and DIF
%* 3) Flag Added/Deleted records ;
   data add_del;
   set outcomp;
   by &unqkey _type_;
   length change $10;
   if first.&lastkey and last.&lastkey then
do;
   if _type_="BASE" then
   do; change='ADDED'; end;
else if _type_="COMPARE" then
   do; change='DELETED'; end;
   end;
run;

/*=== Step 4: Check for Updated Records ===*/
%* 1) Transpose dataset to easily identify current and previous values;
   proc transpose data= add_del out= t_add_del;
   by &idvar change;
   id _type_;
   var &vars2cmp;
quit;
```
%* 2) Flag Change;
    data chg (where=(change ^= ' '));
    set t_add_del;
    by &unqkey;
    length chgtxt chgval $1000;
    diff_num = indexc(dif,"0123456789"); /* Identifies numeric mismatches;
    diff_char= index(dif,"X");          /* Identifies character mismatches;
    if diff_char>0 or diff_num>0 then
      do;
      change='UPDATED'; chgtxt = strip(_name_)||">OLD:"||strip(compare);
      chgval=strip(chgval)||'#'||strip(chgtxt);
      end;
      <.. more lines of code ..>
run;

/**** Step 5: Combine and create Final dataset ====*/
data &outdsn;
  <.. merge datasets- chg, &newds and &oldds by &unqkey >
  <.. more lines ..>
run;

/**** Step 6: Prepare EXCEL report ====*/
ods excel file = "./&outdsn";
<... ods excel options ...>
proc report data= &outdsn;
  <... column and define statements ...>
/* Loop through all variables to style Added/Deleted/Updated records*/
%let dsid = %sysfunc(open(&outdsn.,i));
%do i=1 %to %sysfunc(attrn(&dsid,nvars));
  %let varn=%sysfunc(varname(&dsid,&i));
  compute &varn./char length=20; *- compute block start;
  * <... if record was Added ..> ;
  call define(_col_,"style","style=[font_weight=bold background=lightgreen]"));
  * <... if record was Deleted ..> ;
  call define(_col_,"style","style=[font_weight=bold background=lightred]"));
  * <... if record was Updated ..> ;
  call define(_col_,"style","style=[font_weight=bold background=lightgray
    flyover=' "||strip(chgtxt)||" ']"));
  endcomp ; *- compute block end;
%end;
run;
%mend diff_report;