Macro to Conduct Consistency Checks
Walter H. Hufford, Jr., Novartis Pharmaceuticals Corporation, East Hanover, NJ

ABSTRACT
Pharmaceutical organizations are transitioning from legacy data standards to the CDISC Study Data Tabulations Model (SDTM) and the Analysis Data Model (ADaM) in anticipation of the FDA mandating their use in electronic submissions. Moving to these new data standards is fraught with challenges and there is usually a steep learning curve followed by a bumpy implementation. Although most Data Managers and Statistical Programmers working in the pharmaceutical industry have been exposed to these data models, most are not experts in them yet. This often leads to inconsistent data set and variable attributes - the nemesis of all Statistical Programmers. PROC COMPARE is a useful tool when comparing 2 datasets for inconsistencies; however, it does not provide the ability to compare more than 2 datasets simultaneously. This paper presents a generic macro designed to identify data set and variable attribute discrepancies across N datasets simultaneously and export those discrepancies into a user friendly Excel format for quick review and resolution.

INTRODUCTION
The macro presented in this paper uses several basic SAS® programming techniques to dynamically compare N datasets across N studies and produce a variable attribute discrepancy report. These include PARMBUFF, use of SAS dictionary files, CNTLIN to build a dynamic format, the ODS ExcelXP tagset and style elements in PROC REPORT. Two discrepancy report examples are provided. The first example compares variable attributes across four different studies. The second example compares variable attributes from three studies against a “gold-standard” or “target”.

START OF MACRO
The only input required by the macro is a list of LIBREFs containing the datasets to compare for each study. The PARMBUFF option allows the macro to accept a varying number of LIBREFs. The code below displays the macro opening statement using the PARMBUFF option, a simple %PUT statement to feedback the value of the macro parameter to the LOG and a %LET statement which obtains the name of the first LIBREF from the list.

```
%macro StrucChk /parmbuff ;
%put syspbuff contains: &syspbuff ;
%let num=1 ;
%let study=%scan(&syspbuff,&num) ;
```

MACRO PARAMETER VALIDATION
All macros should behave nicely even if users do not. This is easily achieved via default parameters and soft landings when macro execution halts due to user error. The only user defined parameter required for this macro contains a list of LIBREFs for each study to be compared so a default is not possible. The code below displays three parameter checks users are most likely to trip on. The macro quietly halts execution and provides a friendly (and hopefully useful) description of the user error should any of the rules be broken.

```
%let e = 0 ;
%if &syspbuff= %then %do;
  %put ERROR:  You must supply a parameter to macro StrucChk. ;
  %let e = 1 ;
%end;
%let chk=1 ;
%let chklib=%scan(&syspbuff,&chk) ;
%do %while(&chklib ne) ;
  %if (%sysfunc(libref(&chklib))) %then %do;
    %put ERROR:  Invalid LIBREF - &chklib does not exist ;
    %let e = 1 ;
  %end ;
  %let chk=%eval(&chk+1) ;
  %let chklib=%scan(&syspbuff,&chk) ;
%end ;
```
%if &chk-1 < 2 %then %do ;
  %put ERROR: Macro requires at least 2 LIBREFS to compare ;
  %let e = 1 ;
%end ;
%if &e = 1 %then %goto exit ;
%put NOTE: StrucChk checks completed successfully ;

DATASET ATTRIBUTES

PROC SQL is used to interrogate the SAS dictionary files and create temporary datasets containing variable attribute information for each study (e.g., STUDY1, STUDY2...). A simple DO LOOP rolls thru each of the studies provided to the macro. DO WHILE checks execution at the bottom of the loop so once the last study is processed no more looping occurs. The same task could have been accomplished using the SAS help files or PROC CONTENTS.

%do %while(&study ne) ;
  proc sql ;
  create table STUDY&num as
  select libname, memname, name, type, length, format, label
  from dictionary.columns
  where LIBNAME="%upcase(&study)"
  order by memname, name, type, length, libname ;
  quit ;
  %let study_name&num=%UPCASE(%scan(&syspbuff,&num)) ;
  %let num=%eval(&num+1) ;
  %let study=%scan(&syspbuff,&num) ;
%end ;
  %let num=%eval(&num-1) ;

IDENTIFY DISCREPANCIES

The temporary datasets containing variable attributes are merged together BY the variable attributes under comparison. The IN= values from the MERGE statement are used to create a series of binary numbers. These binary numbers are then used to form a base (2) representation (e.g., 1111) which is stored in the DIS_FLG variable in decimal format (see yellow highlighted code below). By back converting to the base (2) representation, the DIS_FLG variable is used to identify variable attribute discrepancies as well as the source contributors. Assuming four studies being compared, if all four have the same attributes for a variable DIS_FLG = 15 (e.g., 1111 = 2³ + 2² + 2¹ + 2⁰ = 15); if only the first and third LIBREF have the same attributes for a variable DIS_FLG = 5 (e.g., 0101 = 2² + 2⁰ = 5).

The red highlighted code below performs a variable attribute consistency check against a “gold-standard”. By default the macro assigns the first study provided by the user as the “gold-standard” against which all other studies are compared.

data check ;
  merge %do I = 1 %to %eval(&num) ;
  STUDY&i (IN=in&i rename=(libname=lib&i ))
  %end ;
  by memname name type length format label ;
  array targ [&num] _temporary_ ( %do i = 0 %to %eval(&num - 1) ;
    %eval(2**i)
  %end ;)
  if _N_ = 1 then do ;
    gtot = sum( %do J=1 %to %eval(&num - 1) ;
      targ[J],
    %end ;
    targ[ %eval(&num)]) ;
    call symput("tot",Gtot) ;
  end ;
  dis_flg = 0 ;
  %do K = 1 %to %eval(&num) ;
    if in&k then dis_flg = dis_flg + targ(&k) ;
    if in1 and not in&k then do ;
      ds&k._lib = lib&k ;
      ds&k._dis_flg = "Non-match with Standard" ;
    end ;
  %end ;
%end ;
Macro to Conduct Consistency Checks, continued

```sas
end;
else if not in1 and in&k then do;
  ds&k._lib = lib&k;
  ds&k._dis_flg = '++++++++ Extra ++++++++';
end;
%end;
run;
```

**DYNAMIC FORMAT**

The objective of the macro is to identify variable attribute inconsistencies across studies and note them in a discrepancy file. For example, only study F2302s contains a variable named ADSLAGEU or studies F2306s and F2309s contain a variable named ADSL.RACE without the $RACEF format. The code below dynamically builds a format for the new DIS_FLG variable to associate the DIS_FLG variable value with a text description of the studies containing the unique variable attributes.

```sas
%local strt i a x;
%let strt = 1;

data cntlin (keep=fmtname type start label);
  retain sp ' ' fmtname 'dis_cde' type 'N';
  %do i = 1 %to &num;
    do var&i = &strt to &num;
      %do z = 1 %to &num;
        if var&i = &z then varb&i = %eval(2**%eval(&z-1));
        if var&i = &z then varbn&i = "&study_name&z";
      %end;
      start = sum(of varb1-varb&i);
      label = catx(sp,of varbn1-varbn&i);
      output;
    %let strt = %str(%(var&i + 1 %));
  %end;
%end;
run;
```

**DISCREPANCY REPORT**

Finally, the code below produces the variable attribute discrepancy report. The ODS ExcelXP tagset is used to produce an XML file that may be opened in Excel. Style elements are incorporated into the PROC REPORT to produce traffic-lighting. This highlights rows requiring attention of the reviewer. Recall the example above of four studies being compared, all four have the same attributes for a variable so DIS_FLG = 15 (e.g., 1111 = 2^3 + 2^2 + 2^1 + 2^0); if only the first and third LIBREF have the same attributes for a variable DIS_FLG = 5. For this example, the TOT macro variable resolves to 15. The COMPUTE block in the PROC REPORT is used to perform a CALL DEFINE statement containing style elements used to change the color of the row being displayed. If all studies have the same attribute for a variable then DIS_FLG = &TOT and the row color is set to light blue and identifies the row as a match not requiring further review or action. If only a single study has an attribute for a variable then the color of the row is set to red drawing reviewer attention. All other row colors are set to yellow since more than 1 but less than 4 studies have the variable attribute. In these examples yellow and red do not always indicate an issue but probably require review. Note that if all 4 studies define a variable attribute incorrectly then a blue row indicates a consistent approach but not necessarily a correct approach!

```sas
ods listing close;

ODS tagsets.ExcelXP path="/vob/&irproju./&protno./report/pgm_a/"
  file="PharmaSUG_Discrepancy_Traffic_Lighting_Example.xml"
  style=Printer options(embedded_titles='yes'
    embedded_footnotes='yes' print_header='&C&A&RPage &P of &N'
    print_footer='&RPrinted &D at &T' autofilter='all'
```
Macro to Conduct Consistency Checks, continued

```latex
orientation='landscape' autofit_height='yes'
absolute_column_width=18');

TITLE1 "Attribute Consistency Checks for Studies &syspbuff"
FOOTNOTE "Project: SuperDrug123"

ODS tagsets.ExcelXP options(sheet_name='Example 1');

proc report data=check nowindows headline headskip;
  column memname name type length label format dis_flg light;
  define memname / display 'Member';
  define name / display 'Variable';
  define type / display 'Type';
  define length / display 'Length' LEFT;
  define format / display 'Format';
  define label / display 'Label';
  define dis_flg / display 'Members Contributing Attributes'
    format=dis_cde. LEFT;
  define light / computed noprint;

  compute light;
  compute light = 0;
  if dis_flg in(0 %to %eval(&num - 1) ; %eval(2**&i) %end ;)then do;
    call define(_row_,"style","style=[background=lightRED]");
  end;
  else if dis_flg = &tot then do;
    call define(_row_,"style","style=[background=lightBLUE]");
  end;
  else do;
    call define(_row_,"style","style=[background=YELLOW]");
  end;
  endcomp;
run;
```

### Attribute Consistency Checks for Studies (F2302s, F2305s, F2306s, F2309s)

<table>
<thead>
<tr>
<th>Member</th>
<th>Variable</th>
<th>Type</th>
<th>Length</th>
<th>Label</th>
<th>Format</th>
<th>Members Contributing Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADLS</td>
<td>AGEU</td>
<td>char</td>
<td>20</td>
<td>Age Units</td>
<td>SAGEU</td>
<td>F2302S F2305S F2306S F2309S</td>
</tr>
<tr>
<td>ADLS</td>
<td>ARM</td>
<td>char</td>
<td>80</td>
<td>Description of Planned Arm</td>
<td>F2302S F2305S F2306S F2309S</td>
<td></td>
</tr>
<tr>
<td>ADLS</td>
<td>COMPLFL</td>
<td>char</td>
<td>20</td>
<td>Compliers Population Flag</td>
<td>F2302S F2305S F2306S F2309S</td>
<td></td>
</tr>
<tr>
<td>ADLS</td>
<td>RANFL</td>
<td>char</td>
<td>20</td>
<td>Randomized Population Flag</td>
<td>F2302S F2305S F2306S F2309S</td>
<td></td>
</tr>
<tr>
<td>ADLS</td>
<td>PASFL</td>
<td>char</td>
<td>20</td>
<td>Full Analysis Set Population Flag</td>
<td>F2302S F2305S F2306S F2309S</td>
<td></td>
</tr>
<tr>
<td>ADLS</td>
<td>IT</td>
<td>char</td>
<td>20</td>
<td>Intent-To-Treat Population Flag</td>
<td>F2302S</td>
<td></td>
</tr>
<tr>
<td>ADLS</td>
<td>ITFL</td>
<td>char</td>
<td>20</td>
<td>Intent-To-Treat Population Flag</td>
<td>F2302S</td>
<td></td>
</tr>
<tr>
<td>ADLS</td>
<td>PRPROFL</td>
<td>char</td>
<td>20</td>
<td>Per-Protocol Population Flag</td>
<td>F2302S F2305S F2306S F2309S</td>
<td></td>
</tr>
<tr>
<td>ADLS</td>
<td>RACE</td>
<td>char</td>
<td>20</td>
<td>Race</td>
<td>F2302S F2305S F2306S F2309S</td>
<td></td>
</tr>
<tr>
<td>ADLS</td>
<td>RACEGR1</td>
<td>char</td>
<td>20</td>
<td>Pooled Race Group 1</td>
<td>F2302S F2305S F2306S F2309S</td>
<td></td>
</tr>
<tr>
<td>ADLS</td>
<td>RACEGRIN</td>
<td>num</td>
<td>8</td>
<td>Pooled race group</td>
<td>F2302S F2305S F2306S F2309S</td>
<td></td>
</tr>
<tr>
<td>ADLS</td>
<td>SAFED</td>
<td>char</td>
<td>20</td>
<td>Safety Population Flag</td>
<td>F2302S F2305S F2306S F2309S</td>
<td></td>
</tr>
<tr>
<td>ADLS</td>
<td>SEX</td>
<td>char</td>
<td>20</td>
<td>Sex</td>
<td>F2302S F2305S F2306S F2309S</td>
<td></td>
</tr>
<tr>
<td>ADLS</td>
<td>SITEGR1</td>
<td>char</td>
<td>40</td>
<td>Pooled Site Group 1</td>
<td>F2302S F2305S F2306S F2309S</td>
<td></td>
</tr>
<tr>
<td>ADLS</td>
<td>SITEGRIN</td>
<td>num</td>
<td>8</td>
<td>Pooled Site Group</td>
<td>F2302S F2305S F2306S F2309S</td>
<td></td>
</tr>
<tr>
<td>ADLS</td>
<td>SITEID</td>
<td>char</td>
<td>20</td>
<td>Study Site Identifier</td>
<td>F2302S F2305S F2306S F2309S</td>
<td></td>
</tr>
<tr>
<td>ADLS</td>
<td>STUDYID</td>
<td>char</td>
<td>40</td>
<td>Study Identifier</td>
<td>F2302S F2305S F2306S F2309S</td>
<td></td>
</tr>
<tr>
<td>ADLS</td>
<td>SUBJID</td>
<td>char</td>
<td>40</td>
<td>Subject Identifier for the Study</td>
<td>F2302S F2305S F2306S F2309S</td>
<td></td>
</tr>
<tr>
<td>ADLS</td>
<td>TR1T1P</td>
<td>char</td>
<td>80</td>
<td>Planned Treatment for Period 1</td>
<td>F2302S F2305S F2306S F2309S</td>
<td></td>
</tr>
<tr>
<td>ADLS</td>
<td>TR1T1P2</td>
<td>char</td>
<td>80</td>
<td>Planned Treatment for Period 2</td>
<td>F2302S</td>
<td></td>
</tr>
<tr>
<td>ADLS</td>
<td>TR1T2P</td>
<td>char</td>
<td>80</td>
<td>Planned Treatment for Period 1 (N)</td>
<td>F2302S F2305S F2306S F2309S</td>
<td></td>
</tr>
<tr>
<td>ADLS</td>
<td>TR1T2P2</td>
<td>char</td>
<td>80</td>
<td>Planned Treatment for Period 2 (N)</td>
<td>F2302S</td>
<td></td>
</tr>
<tr>
<td>ADLS</td>
<td>TR1T3P</td>
<td>char</td>
<td>80</td>
<td>Planned Treatment for Period 1 (N)</td>
<td>F2302S F2305S F2306S F2309S</td>
<td></td>
</tr>
<tr>
<td>ADLS</td>
<td>TR1T3P2</td>
<td>char</td>
<td>80</td>
<td>Planned Treatment for Period 2 (N)</td>
<td>F2302S</td>
<td></td>
</tr>
<tr>
<td>ADLS</td>
<td>TR1T4P</td>
<td>char</td>
<td>80</td>
<td>Planned Treatment for Period 3</td>
<td>F2302S</td>
<td></td>
</tr>
<tr>
<td>ADLS</td>
<td>SUBJURD</td>
<td>char</td>
<td>40</td>
<td>Unique Subject Identifier</td>
<td>F2302S F2305S F2306S F2309S</td>
<td></td>
</tr>
<tr>
<td>ADLS</td>
<td>USUBJID</td>
<td>char</td>
<td>40</td>
<td>Unique Subject Identifier</td>
<td>F2302S F2305S F2306S F2309S</td>
<td></td>
</tr>
</tbody>
</table>

Project: SuperDrug123
Display 1. Variable Attribute Discrepancy Report

DISCREPANCY REPORT USING A "GOLD-STANDARD" OR TARGET

The previous example compares four studies none of which are considered the "gold-standard" or "target" which all studies must adhere to. The PROC REPORT below creates a discrepancy report using the first study provided by the user as the "gold-standard" against which all other studies are compared. Once again style elements are incorporated into the PROC REPORT to produce traffic-lighting. However, this example uses a style element on the DEFINE statement of the PROC REPORT to change the color of (e.g., highlight) individual cells rather than entire rows. The CALL DEFINE within the COMPUTE block of the PROC REPORT is used to change the color of every other row to facilitate user review across a row.

```sas
ODS tagsets.ExcelXP options(sheet_name='Example 2');
proc format ;
  value $sigb 'Non-match with Standard' = 'RED'
    '++++++++ Extra ++++++++ ' = 'BLUE';
run ;
proc report data=check nowindows headline headskip ;
  column memname name type length label format
ds2_dis_flg ds3_dis_flg ds4_dis_flg light ;
  define memname / display ;
  define name / display ;
  define type / display 'Type' ;
  define length / display 'Length' LEFT ;
  define format / display 'Format' ;
  define label / display 'Label' ;
  define light / computed noprint ;
  define ds2_dis_flg / display 'F2305s'
    style={font_weight=bold foreground=$sigb.} ;
  define ds3_dis_flg / display 'F2306s'
    style={font_weight=bold foreground=$sigb.} ;
  define ds4_dis_flg / display 'F2309s'
    style={font_weight=bold foreground=$sigb.} ;
  compute light ;
    line_count + 1 ;
    if mod(line_count,2)=0 then do ;
      call define(_row_, "style","style=[background=lightYELLOW]");
    end ;
  endcomp ;
run ;
ODS tagsets.ExcelXP CLOSE ;
```
Macro to Conduct Consistency Checks, continued

**CONCLUSION**

The concepts contained in the macro presented in this paper might be considered as an alternative solution to PROC COMPARE when comparing variable attributes across more than two studies simultaneously. In addition, it could be used by SAS Programmers during development of standard datasets (e.g., SDTM, ADaM, corporate standard) to ensure they are “on-target”. The code used to derive the DIS_FLG variable values could also be implemented in other applications that require identification of contributing data such as in oncology studies where several datasets are interrogated to identify date of death (e.g., ECOG, Subject Summary, Grade 5 Adverse Events).

**REFERENCES**


**ACKNOWLEDGMENTS**

I would like to thank John Wenston, Christina Scienski and Gregory Ligozio for their review and helpful suggestions.

**CONTACT INFORMATION**

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

Name: Walter H. Hufford, Jr.
Enterprise: Novartis Pharmaceuticals Corporation
Address: One Health Plaza
City, State ZIP: East Hanover, NJ 07936-1080
E-mail: Walter.Hufford@Novartis.com
Web: www.Novartis.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.