PharmaSUG 2017 - Paper QT10

Everyone can use a little Currency – when dependent data set updates silently make your analysis data set out of date.

Scott Worrell, PAREXEL International

ABSTRACT

Nearly every analysis data set has dependencies on multiple other data sets. The dependent data sets may be raw data sets or other analysis data sets. At times a dependent data set is updated, but the person in charge of updating the analysis data set is not notified of the modification to the dependent data set. This makes the analysis data set with the data dependency invalid. In turn, this condition produces data integrity issues in every Table, Listing or Figure (TLFs) that uses the analysis data set. Before running TLFs, it is a good idea to verify that all dependent data sets are older or the same age as the analysis data sets. The Data Currency Utility detects data set currency issues and reports them so that they can be corrected through an update run of the analysis data set creation program.

This presentation is appropriate for persons with all levels of SAS experience, but assumes basic understanding of SAS® Dictionary Tables. The utility was written and tested using SAS® 9.4 on the Linux operating system.

INTRODUCTION

Unresolved data set dependencies create data integrity issues for analysis datasets and for the Tables, Listings and Figures (TLFs) that are created from them. In a busy, fast-paced Clinical Programming environment, Programmers that are responsible for maintaining dependent datasets may fail to notify colleagues after they make a dependent data set update. Recreating an out of date analysis data set prior to creating TLFs can save time and effort as compared to debugging a display or resolving a Quality Control (QC) discrepancy caused by a lack of data currency.

Through this paper, you will be introduced to a utility that compares the last modification dates of dependent data sets and the analysis data sets that use them as input. The utility output will either confirm that there are no data dependency issues or it will highlight where issues exist so they can be resolved through an analysis data set re-run. The utility may be set up to report on multiple analysis data sets – even the full set of analysis data sets associated with a reporting effort.

DATA CURRENCY UTILITY FUNCTIONALITY

The Data Currency Utility has 3 primary functions:

1) It displays the full list of data sets and modification dates for the analysis data set being checked for currency (called the target library). It also displays this same information for every analysis or raw data set that the target data set uses as input (called a dependent library).
2) The Utility determines if a data set currency issue exists. A currency issue is defined as a case where any dependent data set was modified more recently than the target data set was last created.
3) The Data Currency Utility reports all error conditions and it also outputs the contents of all libraries referenced in the program. The primary output is a report showing the currency status of every dependent dataset / target dataset relationship. This report provides the paths to each referenced data set in the footnotes and it details the study and reporting effort being checked in the report header.

PROGRAM SET-UP

The following occurs during Program Set-up:

- All Global macro variables are declared

- All LIBNAMES are defined. This covers both target and dependent libraries.
- The PDF path and filename macro variables are assigned
- The Study and Reporting Effort macro variables are assigned

The code to perform program set-up is:

```
%let pdfpath=/path /* directory where final PDF report is to be stored */
%let pdfname=dataset_currency_report; /* filename for final PDF report.
System date is appended to this. */
%let study=xxxxx.; /* assign study ID for report title line 2 */
%let repeff=yyyyy;  /* assign reporting effort name for title line 2 */
%let path1=/path1;  /* path for arp libname */
%let path2=/path2;  /* path for dmp libname */
%let path3=/path3;  /* path for aesi libname */
%let path4=/path4;  /* path for aesi2 libname */

title1 'Data set Currency Check Utility';
title2 "Study: &study. Reporting Effort: &repeff.";
footnote1 "Currency utility was last run by:  &sysuserid.";
footnote2 "Libname arp points to:   &path1.";
footnote3 "Libname dmp points to:   &path2.";
footnote4 "Libname aesi points to:   &path3.";
footnote5 "Libname aesi2 points to:   &path4.";

libname arp “&path1.”;
libname dmp “&path2.”;
libname aesi “&path3.”;
libname aesi2 “&path4.”;
```

DATA SET CONTENTS MACRO

The functions performed by this macro (%getdict) include the following:

- Validate parameters that are passed into the macro
- Report on any errors if any are found.
- Obtain data set metadata from the SAS®Data Dictionary using PROC SQL
- List certain data set level contents from the library : data set name, label and last modified data
- Set a return code so that the macro status can be recorded onto the SAS® Log.

The code that follows shows how the SAS® Dictionary tables are queried to get the last data set modification date. Macro variable &lib is passed into the %getdict macro. The last modified timestamp is broken out into separate date and time fields to add clarity when reported:

```
proc sql noprint;
   create table &lib._md as
   select libname as LN label='Libname',
          memname as DSN label='Data set Name',
          memlabel as DS_LABEL label='Data set Label',
          modate as MOD_DT format=datetime22. label='Last modified date/time'
   from dictionary.tables
   where libname = "&uplib."
   order by dsn;
quit;

data &lib._md;
```

```
   set &lib._md;
   attrib mod_date label='Last modified date' format=date9.
          mod_time label='Last modified time' format=time8. ;
   mod_time = timepart(mod_dt);
   mod_date = datepart(mod_dt);
run;
```

DATA SET CURRENCY MACRO

The functions performed by this macro (%currchk) include the following:

- Validate parameters that are passed into the macro, report errors if any are found
- Use data set metadata captured in the contents macro to determine where data currency issues exist.
- Capture whether each target/dependent data set relationship has a currency issue or not.
- Set a return code so that the macro status can be recorded onto the SAS® Log.

The code that determines if a currency issue exists is as follows:

```
* validations passed.  Check dataset currency: target vs destination;
data combined1(drop=ln dsn);
   set dependent target;
   attrib ln_dsn format=$17. ;
   ln_dsn = strip(ln)||'.'||strip(dsn); /* create a 2 level dataset name for
report */
run;

* Put dependent and target data on one observation and assign separate target
and destination variable names ;
data combined2;
   set combined1;
   attrib ln_dsn2 label='Dependent Libname/Data set'
          mod_dt2 label='Dependent DS last mod date' format=datetime22.
          type2 label='Dependent Data set Type'
          ln_dsn label='Target Libname/Data set'
          mod_dt label='Target DS last mod date' format=datetime22.
          type label='Target Data set Type' ;
   retain ln_dsn2 mod_dt2 type2;
   if type='D' then do;
      ln_dsn2 = ln_dsn;
      mod_dt2 = mod_dt;
      type2 = type;
   end;
   if type = 'T' then
      output;
run;

* Determine dataset currency and set up corresponding message to report;
data combined3;
   set combined2;
   attrib MESSAGE format=$45. label='Comparison result' ;
   if mod_dt ge mod_dt2 then
      message = 'Update of target data set not required';
   else
      message = '** Update of target data set IS required **';
run;
```

REPORT CREATION MACRO

The functions performed by this macro (%currout) include the following:

- Combine the individual data currency results into a single data set for reporting.
- Sort the data to facilitate logical presentation in the PDF output report.
- Write output to a PDF report. Present currency issues first. Generate page breaks whenever the target data set changes.

The code to combine, sort and output the report is as follows:

```
* determine which of the report datasets exist;
   %do i=1 %to &limit.;
      proc sql noprint;
         select count(*) into :fin&i.
         from dictionary.tables
         where libname = "WORK" and memname = "FINAL_&i.";
      quit;
   %end;

   * append all report datasets that exist;
   data final;
      set
   %do i=1 %to &limit.;
      %if &&fin&i. ne 0 %then %do;
           final_&i.
      %end;
   %end;
   ;
   run;

   * Sort so that the dependent datasets needing a rerun appear first on
output report;
   proc sort data=final;
      by message ln_dsn ln_dsn2;
   run;

   * Write the report to an Adobe Acrobat PDF file ;
   ods listing close;
   ods pdf file="&pdfpath./&pdfname._PS17_&sysdate..pdf";
 *  ods pdf file="&pdfpath./&pdfname._&sysdate..pdf";
   options ls=130 orientation=landscape spool;
   title4 'Data set Currency Check Overall Results';
   title5 '*** Update of target dataset IS required ***';
   proc print label noobs;
      id ln_dsn ;
      by ln_dsn;
      where index(message,'*') > 0;
      pageby ln_dsn;
      var mod_dt ln_dsn2 mod_dt2;
   run;
   title5 'Update of target dataset is not required';
   proc print label noobs;
      id ln_dsn ;
      by ln_dsn;
      where index(message,'*') = 0;
      pageby ln_dsn;
      var mod_dt ln_dsn2 mod_dt2;
   run;
   ods pdf close;
   ods listing;
```

SAMPLE MACRO CALLS

It is recommended that the %getdict macro is called one time for each libname that has been defined. Here is an example call:

```
   %getdict(arp):  /* ARDATA library */
```

For the %currchk macro, one call is needed for every target and dependent data set relationship. Use data set creation code or data set QC code to determine which dependencies exist. Number the last macro variable sequentially.

Following is a set of %currchk macro calls for a single target dataset. In practice, you would code %currchk macro calls for all of your target analysis datasets:

```
* currchk macro parameters:  target libname, target dsn, target type (T or
t), dependent libname, dependent dsn, dependent type (D or d), output dataset
suffix ;

%currchk(arp,target1,T,aesi,depend1,D,1); /* arp.target1 calls */
%currchk(arp,target1,T,aesi,depend2,D,2);
%currchk(arp,target1,T,arp,depend3,D,3);
%currchk(arp,target1,T,arp,depend4,D,4);
%currchk(arp,target1,T,arp,depend5,D,5);
%currchk(arp,target1,T,aesi2,depend6,D,6);
```

Following is the one macro call required to output the data currency report. When calling macro %currout, pass in the total number of dependent data sets run through the %currchk macro. Here is an example call for %currout:

```
%currout(30); /* generate the report.  Modify the number passed into the
macro to the highest number in the %currchk series above. */
```

UTILITY OUTPUT AND OPPORTUNITIES FOR EXPANSION

The Data Currency Utility produces 3 types of output:

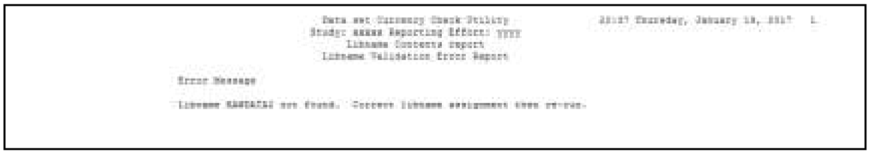
OUTPUT DESCRIPTIONS

1. A validation report detailing any error conditions encountered such as the following. This report appears in the file Currency.lst.
   - A macro parameter that did not pass validation – such as a data set type other than T (Target) or D (Dependent).
   - A dependent or target data set was not found in the library that was specified
   - A reference to a SAS library was found but it had not been established through a valid LIBNAME statement.
2. The Data Set Contents report. This report also appears within Currency.lst.
   - This report provides the data set names, labels and last modification dates for all datasets in the LIBNAMEs passed into the contents macro.
   - Ideally, every data set area referenced by a data currency utility library would appear in this report. This should cover the target library where the analysis datasets reside as well as every library where dependent data sets reside.

3. The Data Set Currency report. This report is in PDF format and is stored in Dataset_Currency_report_*ddmmmyy*.pdf where *ddmmmyy* represents the date the report was created.

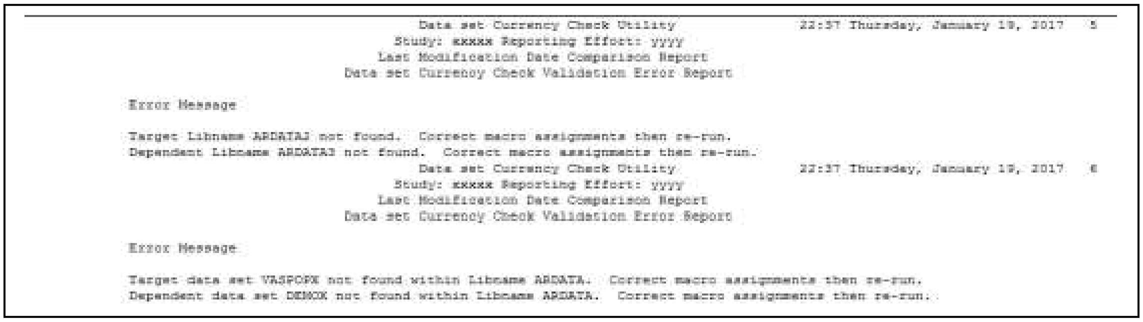
    - This report lists all of the data currency results that passed validation.
    - The report is organized into two sections in the following order:
        - Data Sets with currency issues are displayed first. These are the data sets where a re-run is needed.
        - Data Sets where there are no currency issues are in the second section.
    - Within each section, datasets are presented in the following order
        - Target Data Set is the primary sort variable. Both the libname and the dataset name are listed.
        - Dependent Data Set details form the secondary sort criteria.

OUTPUT SAMPLES



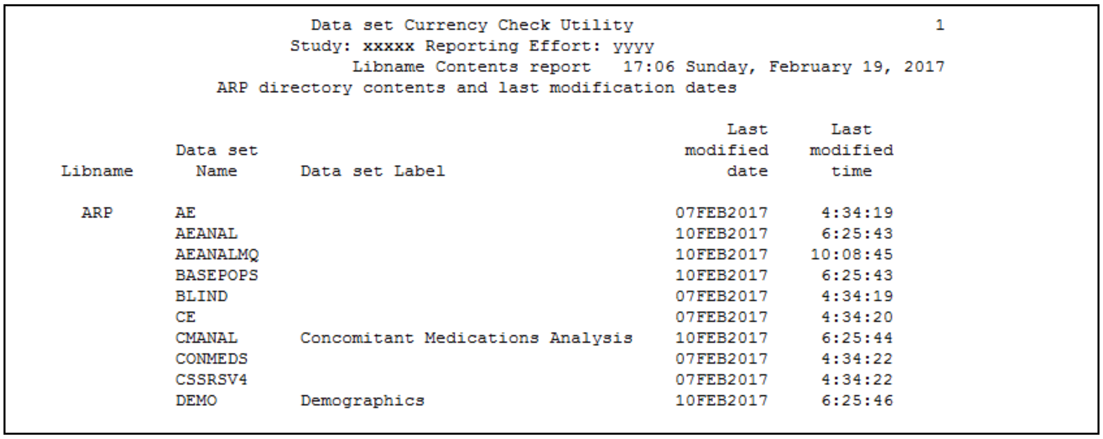
```
                    Data set Currency Check Utility            22:37 Thursday, January 19, 2017   1
                   Study: xxxxx Reporting Effort: yyyy
                         Libname Contents report
                     Libname Validation Error Report

Error Message

Libname XARDATA3 not found.  Correct libname assignment then re-run.
```

Output 1. Contents macro - Invalid Libname Assignment error



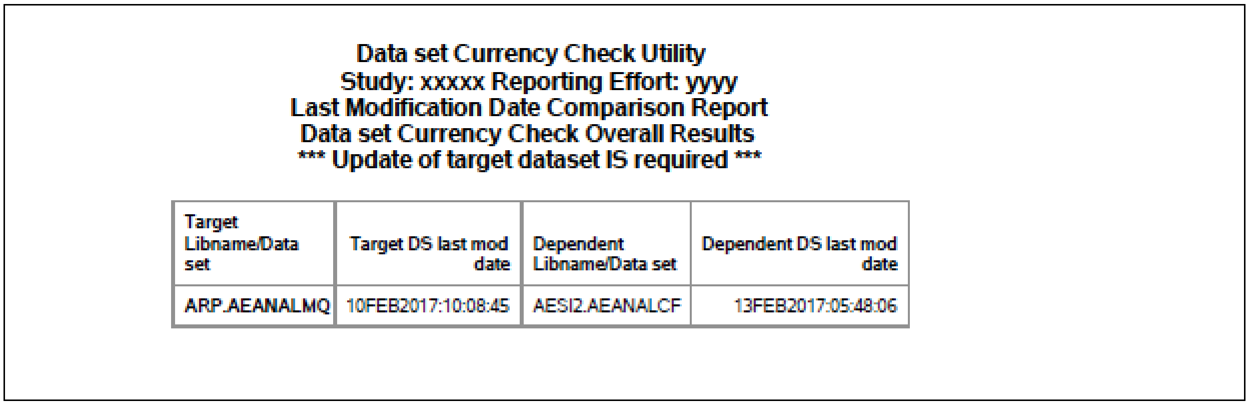
```
                          Data set Currency Check Utility            22:37 Thursday, January 19, 2017   5
                        Study: xxxxx Reporting Effort: yyyy
                      Last Modification Date Comparison Report
                   Data set Currency Check Validation Error Report

Error Message

Target Libname ARDATA3 not found.  Correct macro assignments then re-run.
Dependent Libname ARDATA3 not found.  Correct macro assignments then re-run.
                          Data set Currency Check Utility            22:37 Thursday, January 19, 2017   6
                        Study: xxxxx Reporting Effort: yyyy
                      Last Modification Date Comparison Report
                   Data set Currency Check Validation Error Report

Error Message

Target data set VASPOPX not found within Libname ARDATA.  Correct macro assignments then re-run.
Dependent data set DEMOX not found within Libname ARDATA.  Correct macro assignments then re-run.
```

Output 2. Currency macro – invalid libname assignment and data set not found condition



```
                          Data set Currency Check Utility                              1
                        Study: xxxxx Reporting Effort: yyyy
                               Libname Contents report    17:06 Sunday, February 19, 2017
                 ARP directory contents and last modification dates

                                                              Last       Last
                  Data set                                modified   modified
    Libname         Name      Data set Label                  date       time

      ARP         AE                                     07FEB2017    4:34:19
                  AEANAL                                 10FEB2017    6:25:43
                  AEANALMQ                               10FEB2017   10:08:45
                  BASEPOPS                               10FEB2017    6:25:43
                  BLIND                                  07FEB2017    4:34:19
                  CE                                     07FEB2017    4:34:20
                  CMANAL      Concomitant Medications Analysis 10FEB2017    6:25:44
                  CONMEDS                                07FEB2017    4:34:22
                  CSSRSV4                                07FEB2017    4:34:22
                  DEMO        Demographics               10FEB2017    6:25:46
```

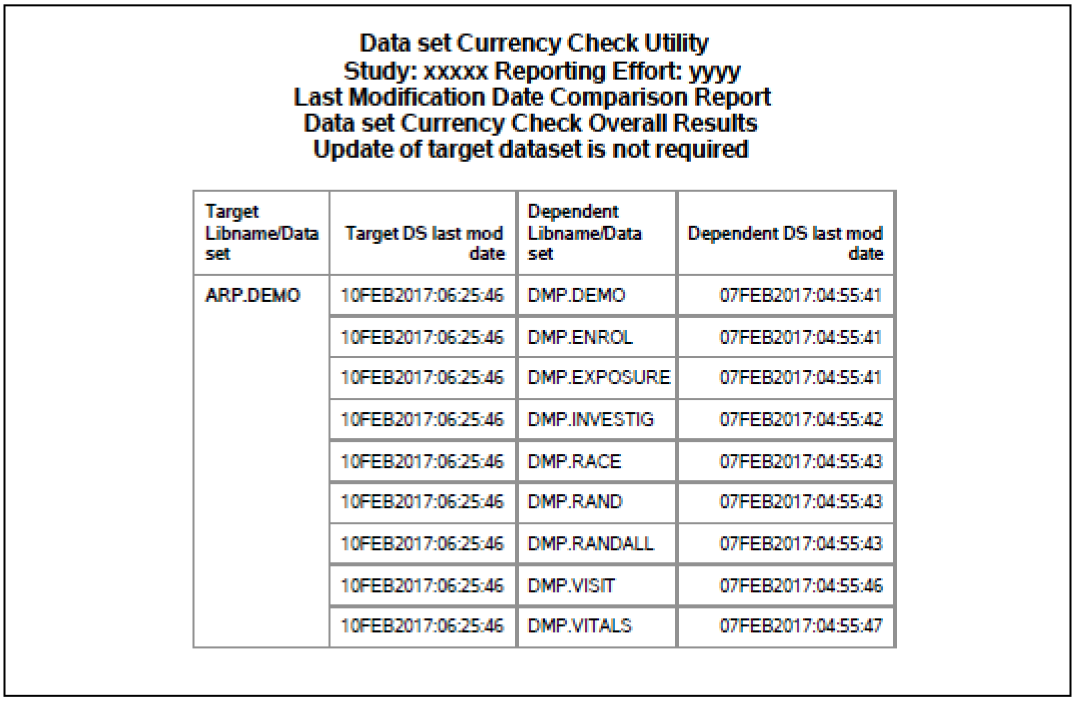
Output 3. Libname contents report – partial output for a single target library



Data set Currency Check Utility
Study: xxxxx Reporting Effort: yyyy
Last Modification Date Comparison Report
Data set Currency Check Overall Results
*** Update of target dataset IS required ***

Target Libname/Data set	Target DS last mod date	Dependent Libname/Data set	Dependent DS last mod date
ARP.AEANALMQ	10FEB2017:10:08:45	AESI2.AEANALCF	13FEB2017:05:48:06

Output 4. Data Currency output when a currency issue was found



Data set Currency Check Utility
Study: xxxxx Reporting Effort: yyyy
Last Modification Date Comparison Report
Data set Currency Check Overall Results
Update of target dataset is not required

Target Libname/Data set	Target DS last mod date	Dependent Libname/Data set	Dependent DS last mod date
ARP.DEMO	10FEB2017:06:25:46	DMP.DEMO	07FEB2017:04:55:41
	10FEB2017:06:25:46	DMP.ENROL	07FEB2017:04:55:41
	10FEB2017:06:25:46	DMP.EXPOSURE	07FEB2017:04:55:41
	10FEB2017:06:25:46	DMP.INVESTIG	07FEB2017:04:55:42
	10FEB2017:06:25:46	DMP.RACE	07FEB2017:04:55:43
	10FEB2017:06:25:46	DMP.RAND	07FEB2017:04:55:43
	10FEB2017:06:25:46	DMP.RANDALL	07FEB2017:04:55:43
	10FEB2017:06:25:46	DMP.VISIT	07FEB2017:04:55:46
	10FEB2017:06:25:46	DMP.VITALS	07FEB2017:04:55:47

Output 5. Data Currency output when a currency issue was not found

EXTENDING THE CURRENCY CONCEPT

The data currency concept presented in this utility could be extended to cover other programming situations where dependencies exist. A few examples are:

- Macros that are updated, but programs that call them have not been re-run to use the new version of the macro.
- One or more datasets or input files used in the creation of a Table, Listing or Figure (TLF) is updated but the TLF display has not been re-run to pull in data from the new dataset or file.

The Data Currency Utility could be enhanced to cover these situations or other situations where dependencies exist.

CONCLUSION

Data Currency issues can be a real issue in today's fast-paced, deadline driven clinical programming environment. In order to avoid lengthy and frustrating debugging efforts, the small investment of time required to prepare and run the Data Currency utility is time well spent.

After all - everyone knows that people rarely complain about having too much currency!

The full Data Currency program code including referenced macros is available within the PharmaSUG 2017 proceedings.

ACKNOWLEDGMENTS

The author gratefully acknowledges the review and input from PAREXEL International colleagues Scott Burroughs and Shannon LaBelle who each offered helpful comments during the development of this paper.

RECOMMENDED READING

- *The SQL Procedure: DICTIONARY Tables – SAS® OnlineDoc V8 (https://v8doc.sas.com/sashtml/proc/zsqldict.htm)*

CONTACT INFORMATION

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

Scott Worrell
PAREXEL International Durham, NC USA
scott.worrell@parexel.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.