

What is the Definition of Global On-Demand Reporting within the Pharmaceutical Industry?

Eric Kammer, Novartis Pharmaceuticals Corporation, East Hanover, NJ

ABSTRACT

It is not uncommon in the pharmaceutical industry to have standardized reporting for data management cleaning activities or clinical review. However, certain steps have to be taken into consideration when programming standardized reports, so they run properly for all studies. One particular troublesome area is when the global reports are run on-demand by non-programmers. As new data becomes available, original test cases that were used could be different, and the programs could fail. This paper will identify techniques to assist programmers on what to do when developing code, so programs can run successfully when unexpected surprises with data occur running SAS® reports in an on-demand environment and give the customers information to assist in understanding the results.

INTRODUCTION

Global on-demand data cleaning reports are reports that can be run on any study and most likely by a non-programmer. Traditional programming allows a programmer to take SAS® programs and revise it according to the study needs. This is true for program templates that are copied to one's study and then changed in order to get them running properly. Most programmers have stable or non-changing data for analyses from snapshots, interim locks, or final database locks. A global on-demand report for data cleaning will have different data every day. This can even include "no data" if run early in a study, or different data, as it is being refreshed in the database. The data can even be missing or "bad data" if run early in the data cleaning process. If a SAS® error occurs, you do not have the luxury of debugging or looking at the SAS® log, as the messages will occur right in front of the user's eyes and even possibly fail. This can be embarrassing and cause unnecessary headaches. A global on-demand data cleaning report will be used by a non-programmer and run on current data during the course of the trial. These users will send the reports to clinical or other data managers in the process of cleaning data assuming the results are what should be expected.

A similar idea is defensive programming in trying to prevent issues that will occur in someone's program but in a global environment, and because non-programmers involved have to be taken to an even higher extreme.

Types of issues

One issue with user driven reports is when the report shows that there are no exceptions. One way of assisting the user is to show the number of records processed vs. number of occurrences, as it is important when reviewing reports. As an example, a user might run an exception report and see "No observations found". However, is this a false positive; "where the reason there are no records found is because there is actually no data to support the condition in the final dataset"?

One way to show if there was nothing to process is to give a message when there are no observations found in a report.

```
%macro noreport(in=);
/*****/
/* Check for zero observations                               */
/* in the dataset to be reported                           */
/*****/
  DATA _null_;
    CALL SYMPUT('_nobs',LEFT(PUT(nobs,best.)));
    STOP;
    SET &in nobs=nobs;
  RUN;
  %IF &nobs EQ 0 %THEN %DO;
    DATA noobs;
      LABEL text='***';
      text='*****'; OUTPUT;
      text='*** There are no observations for this report ***'; OUTPUT;
  %END;
%END;
```

What is the Definition of Global On-Demand Reporting within the Pharmaceutical Industry?

continued

```
text='*****'; OUTPUT;
RUN;
PROC PRINT LABEL noobs; RUN;
%END;
%mend noreport;
```

Another problem when running a report against a SAS® dataset is: has this dataset been created yet or did the user run the report too early in the data cleaning process? Does the file area being analyzed exist in the SAS® environment? If the data does exist, when was the last data updated or created?

Check if the data area was created.

```
%global yes_no;
%let yes_no=NO;
%macro doit(myfiler=);
%if %sysfunc(fileexist(&myfiler)) %then %let yes_no=YES;
%else
%put The SAS file &myfiler does not exist.;
%mend;
```

Showing the number of records processed gives a count on data available for the report, and if the report showed that no observations existed, can prove there was data to be used for the report. The number of records processed allows the user to review the information.

```
%macro numobs(in=);
/*****
/* shows number of records processed in a dataset */
*****/
proc contents data = &in out = counts noprint;
run;

data _null_;
set counts end=eof;
if eof then do;
x=compress(put(nobs,best.));
file print;
put "*****";
put "***" nobs "observations processed for dataset &in *****";
put "*****";
end;
RUN;
%mend numobs;
```

Show the date and time of the data created and number of observations.

What is the Definition of Global On-Demand Reporting within the Pharmaceutical Industry?

continued

```
%macro doobs(in=);
proc contents data = &in out=newdms noprint;run;
data _null_;
  set newdms end=eof;
  if eof then do;
    call symput('crddate',put(crdate,datetime20.));
    call symput('nobs',compress(put(nobs,Best10.)));
  end;
run;

data _null_;
file print;
put "*****";
put "&nobs observations processed for dataset &in ";
put "Data was created on &crddate *****";
put "*****";
RUN;
%mend;

%doobs(in=dmcount);
```

Alternatively, a variable that you are analyzing might be completely empty or does not contain the data it is supposed to, such as, if derived database information is still not available (e.g. age between 0 - 30) or only having certain lab categories such as chemistry, hematology, or urinalysis. In this case, there are records processed, but there is no data to support the algorithm as possibly the derivation was not available at the time of the data update, or only chemistry and hematology exist. The user might even run the program on another study that they are assigned to by mistake thinking the results are acceptable. A frequency or range display can be used if data is presented showing the user that “no records met the condition”, and what data actually was available for the report. The extra benefit of having user specified parameters is that an opposite condition can be used for test purposes, which will prove to the user that the results given are true but not currently available with existing data (e.g If age < 18 vs. if age < 50).

The following data was available for AGE, but no data was found to meet the conditions of the report or for the lab category, and no values were found for urinalysis.

AGEDRV.	Lab Category
0	Biochemistry
30	Hematology
31	
44	
50	
59	
61	
101	

Do not assume as it is best to pass variables or conditions using parameters and filters from a driver program, or have some metadata approach to use the variables that apply.

```
Problematic:
proc sort data=data_s.lb(keep= USUBJID LLBCAT) out=lab_all;
where LLBCAT="HEMATOLOGY";
by USUBJID;
run;
```

Better;

What is the Definition of Global On-Demand Reporting within the Pharmaceutical Industry?

continued

```
%let vars = USUBJID LLBCAT;
%let cond = LLBCAT="HEMATOLOGY";
%let dsn = lb;
%let byvar = USUBJID;

proc sort data=data_s.&dsn (keep=&vars) out=lab_all;
where &cond;
by &byvar;
run;
```

While it is very efficient to only keep those variables required, if you do not have some method ahead of time of checking to make sure the variables exist then the report could fail. Databases and standards change, so one needs to find ways of making the programs flexible, so they can be adaptable to new standards to avoid error messages.

ERROR: The variable DOSADM in the DROP, KEEP, or RENAME list has never been referenced.

If considering only a unit of 'g/dL' then if this unit does not appear, a message should be printed displaying the values found (e.g. g/dL, g/L), so the user knows that there were values found outside of the consideration and a footnote or title should be given to indicate

Note: Only values of g/L were considered for the program.

What is the Definition of Global On-Demand Reporting within the Pharmaceutical Industry?

continued

```
%macro novaluec(indsn=,invar=);
/*****
/* shows the range of values for character values */
*****/
data _null_;
  file print;
  put "*****";
  put " * No Data selected for &invar from dataset &indsn          ";
  put " ** Here are values collected for variable &invar          *****";
  put "*****";
  RUN;

proc freq data = &indsn;
tables &invar;
run;
%mend novaluec;

%macro novaluen(indsn=,invar=);
/*****
/* shows the values for a numerical result */
*****/

proc means data = &indsn noprint;
var &invar; output out = meany min=min max=max;
run;

data _null_;
  set meany;
  file print;
  put "*****";
  put " * No Data selected for &invar from dataset &indsn          ";
  put "Records for &invar-->Mimimum=" min " Maximum=" max "from &indsn";
  put "*****";
  RUN;
%mend novaluen;

%macro noreslt(theinn=,thevar=);
/*****
/* if no data supported algorithm then display the results */
*****/
proc contents data = &theinn out=newdms noprint;
run;
%let charnum=0;
data _null_;
  set newdms;
  if upcase(name)=upcase("&thevar") then
    call symput('charnum',put(type,1.));
run;
%if &charnum=2 %then %do;
  %novaluec(indsn=&theinn, invar=&thevar);
%end;
%else %if &charnum=1 %then %do;
  %novaluen(indsn=&theinn, invar=&thevar);
%end;
%else %do;
%put Note: The Variable name was not found on Dataset &theinn Variable
&thevar;
%end;
%mend;
```

What is the Definition of Global On-Demand Reporting within the Pharmaceutical Industry?

continued

Many times when an on-demand report is run, a file is created from SAS®, either as a SAS® listing, Excel®, or pdf. If a file is created for later review, the description, the user name, date, time with seconds have to be included, otherwise, if two users run the report accidentally at the same time, you will have interference from the filenames and can actually lock the on-demand program while being executed. The naming convention is also helpful if you are creating an audit trail of the report usage.

Each of these outputs poses problems especially if large record sizes are generated, which are difficult to work with.

Simple messages such as "warning it could take a while for the Excel® sheet to open, as there are >50,000 rows of data present maybe rerun and subset your data". Such a warning could then make the user suspect that something is wrong with the data or even the report output.

If there is an issue, it might be best to have the program stop and give a message to the user that the data did not support the current condition or something went wrong.

How will the user know the program failed and did not run? One important feature is to actually print a message in the output area to let the user know that the program started, processed a certain amount of data, finished executing, and is in the final stages of processing.

These messages will actually help the programmer debug any issues from the program, and facilitate a more accurate review of the output.

If the program completes successfully, summarize the various messages that occur running SAS® programs (e.g. warnings, uninitialized, important notes), as non-programmers will not understand these.

```
2 OCCURRENCES OF POSSIBLE ERRORS FOUND 00:26 Thursday, March 27, 2014

Line      Description
6        WARNING: Unable to copy SASUSER registry to WORK registry.
1078     WARNING: Variable DOSADM is uninitialized
```

Users or programmers can make mistakes in assumptions, so it is important to decide with upper or lower case for character data and show the assumptions made to produce the report.

However, if the program assumes a certain entry, then a message should be given to let the user know that an expectation was not met. Obviously, this cannot be done on each field; just key conditions.

As mentioned, another very important feature of assisting users and meeting the requirements of global reports is having user entered prompts that actually can control running the program. This can give flexibilities in creating subset reports such as baselines and lab parameters. This can result in less data being printed or adding important considerations such as dates, patients, and/or data milestone indications.

Baseline values could be visit 2 or 3, are good examples that change from study to study and are needed sometimes in order for reports to run properly. Variables might also change to indicate specifics (e.g. total daily dose, dose administered vs. actual dose). Optional variables vs. mandatory variables, and then the user who knows the study better can assess what is needed. In this way, user parameters can control certain elements of your global reports.

Here is an example of code for a dataset when the variable or dataset selected does not exist.

continued

```
%macro novardsn(invar=,indsn=,liber=);
proc contents data = &liber.._all_ out = &liber(keep=memname name) noprint;
run;

proc sort nodupkey data=&liber;
  by memname name;run;

data varfound dsnfound;
  set &liber;
  if trim(uppercase(name)) = trim(uppercase("&invar")) then output varfound;
  if trim(uppercase(memname)) = trim(uppercase("&indsn")) then output dsnfound;
run;

%let nbs=0;
%let dsn=0;
data _null_;
  set varfound end=eof;
  if eof then call symput('nbs',put(_n_,10.));
run;

data _null_;
  set dsnfound end=eof;
  if eof then call symput('dsn',put(_n_,10.));
run;

data _null_;
  file print;
%if &dsn=0 %then %do;
  put "***Dataset required &indsn not found***";
%end;
%else %if &nbs=0 %then %do;
  put "***Variable &invar not found on dataset &indsn***";
%end;
run;
%mend;
```

The report, when run by the user, should give a complete history of what happened at that point in the time, as each day, the data will be different from the database. The data might be extracted to SAS® datasets or taken directly from an Oracle® database.

Is the output draft or production, time and date run, location of report in the programming environment, name of the program, study that was processed, raw files or analysis files, user that ran the report, number of records processed, date of the last update to the database to show how recent data is, was data available for the variables, is the dataset or variable required found, and assumptions such as filters or parameters? All of the concepts considered in this paper are not part of the main body of the program but added separately. As mentioned, there can be user entered parameters before the program begins execution (e.g. baseline visit numbers) or a driver file at the beginning or end of the program whichever makes sense, and this can control the program, so a minor change does not require change control or additional documentation.

CONCLUSION

Data cleaning reports pose special problems when run by non-programmers “on-demand”, as new data arrives on an ongoing basis. It is important to take special steps while programming to let the users know why a particular report might have produced either no results or different results than expected. Using basic SAS® functionality, a programmer can present messages to the user giving answers as to why report output was produced. Messages are not only helpful to the user, but if a question is asked; a programmer can then assist the user as why the report behaved the way it did. The same concepts can be applied to each report, so extra programming or change control is not required. Basically, an entire history of the report regarding data can be displayed which allows any data review report to be more functional, and in the end, having fewer questions by the user, management, and the reviewer of the results of the reports. This includes showing the user that the report has begun, various steps of execution, giving data results as a summary to clarify the report or no data results, and at the end, summarizing the status and any error messages of the report.

CONTACT INFORMATION

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

Name: Eric Kammer
Enterprise: Novartis Pharmaceuticals
Address: One Health Plaza
City, State ZIP: East Hanover, NJ 07936
E-mail: eric.kammer@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.