

## Paper CC-04

**The Last Line**

Brandon Barrett, Human Genome Sciences, Inc., Rockville, MD  
 Binoy Varghese, Cybrid, Inc., Wormleysburg, PA

**ABSTRACT**

Summary tables and listings form an integral part of a clinical study report. These reports are created using raw/analysis datasets. An essential practice in the Life Sciences industry is to create/re-create analysis datasets and reports in a sequential order at every extraction of raw data. This rule ensures data consistency. Although, almost all organizations have various checks to safeguard this policy, a simple solution is to reserve the last line (footnote10) of each report to list input datasets and corresponding date-time stamps, along with date-time of report creation. Having this footnote on the report increases the chances of an inconsistency being detected, not only by the programming team but by every other team that is involved in writing and reviewing the clinical study report. It also provides a trail to reviewers in case they are interested in examining the input datasets. The purpose of this paper is to present a technique that will automatically produce the last line of the report with data dependency and the date-time stamps. Additionally, it will list the report name and the program that created the report.

**INTRODUCTION**

Many organizations have the practice of listing a bread crumb in the footer of their reports that may consist of some or all of the components below:

- associate (username) who created the report.
- SAS® program name along with datetime of report creation.
- raw/analysis datasets and datetime associated with the datasets.

This can be achieved within reporting programs by using SAS® macros, SAS autocall macro facility and SAS automatic macro variables. In subsequent sections, we examine various types of reporting programs and how %read and %footer macro calls enable automatic creation of the bread crumb. The complete SAS code for %read and %footer macro is provided in the paper and can be used as is, in most cases.

**TYPES OF REPORTING PROGRAMS**

In clinical reporting, we usually come across reporting programs that create either a single report or a multitude of reports. The reusability of code depends on the similarities of the reports being created in terms of the data processing and output formatting that is shared by the reports.

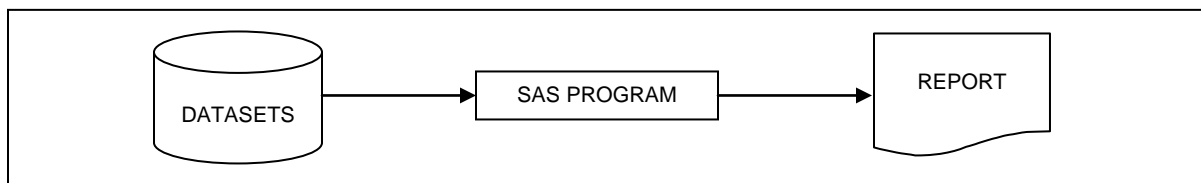
**ONE-TO-ONE REPORTING**

Fig 1. one-to-one reporting

Fig 1 illustrates one-to-one reporting – a single sas program is used to create a single report. In this case, footnote10 is used only once during the execution of the program for creating the report.

**ONE-TO-MANY REPORTING**

Fig 2 illustrates one-to-many reporting. The sas program contains one or more macro definitions each of which may produce a single report. Based on the parameters supplied within the macro calls, each of these macro calls may result in different reports produced by the same macro definition. In such a case footnote10 will be used more than once during execution of the program to create multiple reports.

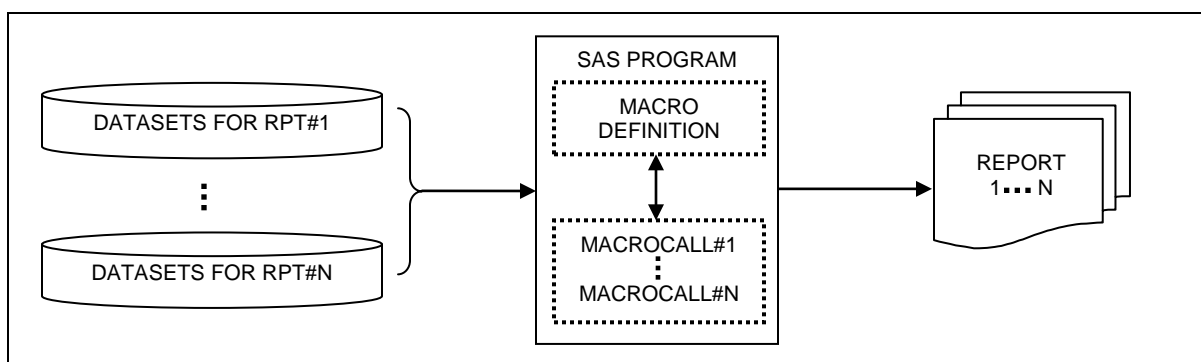


Fig 2. one-to-many reporting

## MACRO PARAMETERS AND HIGH LEVEL ALGORITHM

Automatic population of footnote10 is facilitated by the %read and %footer macro calls.

### %READ

The %READ macro is used to read datasets from analysis/extracts data library to the work data library. At the same time, it also saves the datetime meta information of the dataset.

The %READ macro has 3 parameters:

- ⇒ LIB – source library name (Required parameter)
- ⇒ DSN – input dataset name (Required parameter)
- ⇒ OUT – output dataset name (Optional parameter). If this parameter is not specified in the macro call, the dataset copied to the work library will have the same name as the input dataset.

The algorithm for %READ is

1. Create global macro variable BRDCRUMB, if determined as the first call during the execution of a program.
2. Read input dataset and save it in work data library.
3. Obtain datetime meta information of input data and append to BRDCRUMB macro variable along with input library name and dataset name.
4. Cancel footnote10 statement.

### %FOOTER

The %FOOTER macro does not have any input parameters. This macro assigns the value stored in BRDCRUMB macro variable to footnote10 statement.

The algorithm for %FOOTER is

1. Obtain username of associate.
2. Obtain program name and datetime of execution.
3. Assign information in #1, #2 and value of BRDCRUMB macro variable to define footnote10.
4. Reset BRDCRUMB macro variable to missing.

## SAS CODE

The complete sas code for the %READ and %FOOTER macro is listed below. Copy the code as is and save as READ.sas and FOOTER.sas.

### READ.SAS

```
%macro read(lib=,dsn=,out=&dsn);

/* create global macro variable BRDCRUMB at first call */
%if %symexist(brdcrumb) eq 0 %then %global brdcrumb;

/* save input dataset in work library */
data &out;
    set &lib..&dsn;
run;
```

```

/* capture meta information of input dataset */
ods listing close;
ods output Attributes=attri&dsn;
    proc contents data=&lib..&dsn;
    run;
ods output close;
ods listing;

/* update BRDCRUMB macro variable */
data _null_;
    length mystr $100;
    set attri&dsn;
    if label1 in ('Last Modified') then do;
        mystr=catt("%upcase(&lib..&dsn)",'(',put(nvalue1,datetime13.),')');
        if "%trim(&brdcrumb)" ne "" then mystr=catt("%trim(&brdcrumb)",',',
',trim(mystr));
        call symput('brdcrumb', trim(mystr));
    end;
run;

/* remove intermediate datasets */
proc datasets library=work nolist;
delete attri&dsn;
run;
quit;

/* reset footnotel0 statement */
footnotel0;

%mend read;

```

## FOOTER.SAS

```

%macro footer;

/* obtain program name and datetime of program execution */
proc sql noprint;
    select distinct catt(upcase(scan(scan(trim(left(xpath)), -1, "\"),1,'.')),
'.SAS(', compress("&sysdate:&systime"), '))'
    into: progname
    from sashelp.vextfl where index(upcase(xpath),'.SAS');
quit;

/* prepare footnotel0 */
footnotel0 "%upcase(&sysuserid)/%trim(&progname)/%trim(%nrstr(&brdcrumb))";

/* reset BRDCRUMB macro variable */
%let brdcrumb=;

%mend footer;

```

## MACRO SETUP

Most organizations use a hierarchical directory structure to organize datasets, tables, listings, graphs, programs and associated validation work as shown in Fig 3. Save READ.sas and FOOTER.sas macro to the "macro" directory.

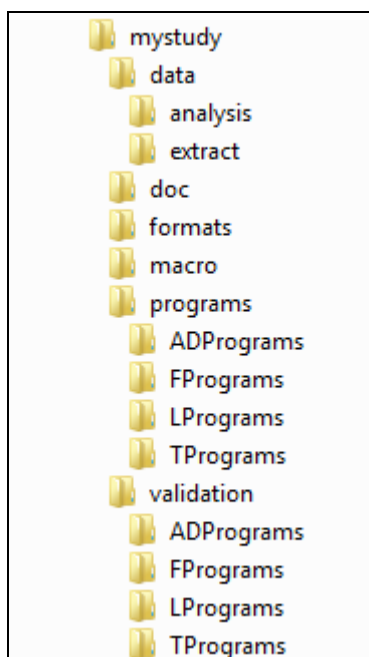


Fig 3. Typical directory structure

**AUTOEXEC.SAS FILE**

Specify the path to autocall macro library path in the autoexec.sas file:

```
options sasautos=(" <path to macro folder>" sasautos) mautosource;
```

**SAMPLE REPORTING PROGRAMS****ONE-TO-ONE REPORTING**

```
-- sas processing --;
%read(lib=an, dsn=dem);
-- sas processing --;
%read(lib=ext, dsn=ae);
-- sas processing --;
%footer;
-- report creation --;
```

**ONE-TO-MANY REPORTING**

```
%macro myrpt(lib1=an,dsn1=,lib2=an,dsn2=);

-- sas processing --;
%read(lib=&lib1, dsn=&dsn1);
-- sas processing --;
%read(lib=&lib2, dsn=&dsn2);
-- sas processing --;
%footer;
-- report creation --;

%mend myrpt;

-- create rpt#1 --;
%myrpt(dsn1=dem,lib2=ext,dsn2=ae);
-- create rpt#2 --;
%myrpt(dsn1=dem,lib2=ext,dsn2=lab);
```

## **SAMPLE FOOTNOTE10**

BVARGHESE/T\_DEM.SAS(11JUL11:10:53)/AN.DEM(11JUL11:10:17), AN.AE(11JUL11:10:30)

## **CONCLUSION**

The technique discussed in this paper makes 2 assumptions:

1. Each reporting program will read input datasets only using the %read macro call.
2. The %footer macro is called after reading in all input data and prior to report generation.

If these assumptions are not complied with, then footnote10 may contain incomplete or incorrect information.

This technique has been successfully tested using SAS version 9.2 on Windows Operating System.

## **CONTACT INFORMATION**

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

Brandon Barrett  
Human Genome Sciences, Inc.  
14200 Shady Groove Road  
Rockville, MD 20850  
[brandon\\_barrett@hgsi.com](mailto:brandon_barrett@hgsi.com)

Binoy Varghese  
Cybrid, Inc.  
1017 Mumma Road, Suite 105  
Wormleysburg, PA 17043  
[mailme@binoyvarghese.com](mailto:mailme@binoyvarghese.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.