Paper 225

A Table Production System That Meets the Challenges of Tomorrow Using SAS/AF® Software and the Report Procedure

By David Trenery, Hoechst Marion Roussel, Denham, UK

Abstract

This paper describes a table production system of modular design to cope with change: changing sources of information, table production software, and management requirements. The modular design also enables individual pieces to be developed, tested and run in isolation. The core of this system is a standardised SAS data set, which contains one observation for each row of the table. This along with a report template (which has the style and layout of the table) is passed to the table production software. A control SAS data set drives the system and contains a table identifier, the name of the program that produces the standardised data set, the report template name along with additional information. An AF application acts as an interface to get information from the control set, run the program and sends the resulting data set and template name to the table production software. Draft tables are created using Proc Report. The design of the system made it easy to change the final table production software. Different software was tried including Proc Report. The ultimate goal was a single Microsoft® Word document containing all the tables, making use of different fonts and type sizes.

Introduction

I work for a large pharmaceutical company, where the table production process can easily get out of hand. Large numbers of tables are requested and the specs for these tables change constantly. I was about to start programming the tables for my largest trial yet. The project was over a long time scale, many other programmers would be assisting me and the data would not be finalised until the last moment. The final tables would ultimately reside in a Microsoft Word document but how they would get there was still undecided. We needed a system that would impose some structure on the table production process, enable us to start work on the tables yet be flexible enough to deal with all the changes in requirements and software ahead of us.

Our previous table production process

In previous studies at our site the programmer would consult the list of tables, and create a program using the SAS® system that would create one or more of these tables after consulting some table specifications. Many of these tables were similar and most fell into one of half a dozen different styles. Final table output might be produced in one of the following methods:

- cut and paste of SAS output into Microsoft Word
- running a Microsoft Word macro on a file containing SAS output
- an in-house mailmerge application

Other potential methods

It was also suggested we use an in-house application developed by the global head office or that we could send processed data to Microsoft® Access, which would then produce a Microsoft Word document. The enhanced PROC REPORT in the next release of SAS sounded like it could do the trick. Too late for this project, but there for subsequent studies.
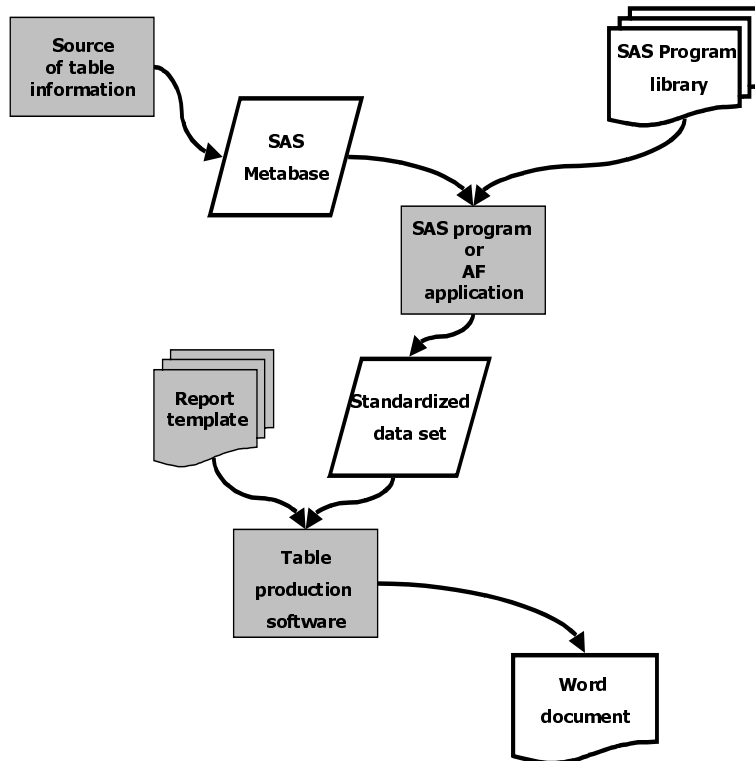
The solution

To cope with future requirements I designed a modular table production process table (see figure 1). A control data set contained a list of tables; a program library contained programs which when run would produce a standardised dataset containing the content of one table. A template library contained a report template which when applied to the standardised dataset would produce the final.

Until a decision was made on the final table production process this data set was sent to one of half a dozen different PROC REPORT templates to produce the draft tables. Meanwhile trials were carried out on:

- sending the standardised data set to Microsoft Access via PROC DBLOAD and producing the tables there
- saving the PROC REPORT output to a file and applying a Microsoft Word macro to format the table to suit Microsoft Word

Figure 1. Diagram of the table production process

Note that the grey sections are modules that can be replaced deal with changing data sources, table production software and preferred method of working.

The Standardised data set.

In the process of creating a report we take data and give this to the report generator that creates the report. I have always preferred to summarise and massage the data first before giving it to the report generator. This way you can easily see the report content through a simple Proc Print or Fsview (SAS/FSP® software). It also gives you much greater control and makes it easier to change the report generator. My standardised data set is basically a crude version of the report. With one observation per non-blank row of the report (including titles and footnotes) and variables storing the columns of data in the report, titles and footnotes and additional control variables determining the layout of the report. I stored both titles and footnotes in one 200-character variable with a control variable determining whether it was a title, a footnote or a note to appear immediately at the foot of the report.
Other control variables determined page, section and sub-section breaks and the ordering of the rows within the

table. My page variable was of the form 'Page 1 of 23' and in addition to determining pagination was placed at the bottom left of each page. A solid line split sections with a double underline at the end of the last section. A bank line split sub-sections. One variable indicated the font to use in the Microsoft Access versions of the report.

The Control data set

I adapted an existing Microsoft®Excel spreadsheet containing the table numbers, titles and other information to drive the process.
The control data set consisted of the following information: table title and number, the name of the program which creates the standardised data set, and the name of the report template to use to create the table. Table number used both in the title of the table and to name any resulting files. I used other variables in the control data set to document and track the table production process.

Standardised SAS data set

```
              Control variables          Columns appearing in table

PAGE        COUNT RECTYPE ROWTEXT              NEW_DRUG        OLD_DRUG       PVALUE
              1 T        Table 6.5.1 Baseline demogr...
              2 T        Characteristic
1 of 1        3 S        Study-1a             N=172           N=251
1 of 1        4 FQ       Sex
1 of 1        5 FQ       Male                 124 ( 72%)      162 ( 65%)     0.4110
1 of 1        6 FQ       Female                48 ( 28%)       89 ( 35%)
1 of 1        7 FQ       Age
1 of 1        8 FQ       <= 65                 89 ( 52%)      117 ( 47%)
1 of 1        9 FQ       65 +                  83 ( 48%)      134 ( 53%)
1 of 1       10 UV
1 of 1       11 UV                            172             251            0.5814
1 of 1       12 UV                             64.4(11.92)     65.6(10.96)
1 of 1       13 UV                             65              66
1 of 1       14 UV                             38-86           32-85
             15 F1        * p-values are determined u...
             16 F1        chi-squared test for catego...
             17 F2        SOURCE: HMR - STAT_CDM -dem...

                                             Optional variables
```

The program library

All the programs used to create the standardised data sets were stored in one directory.
Each program created only one standardised data set for each run. Information supplied by the control dataset was passed to the program. As well as placing some of this information, such as titles, inside the standardised data set, information from the control data set could be used in the program as a switch to subset the data. For example some of my programs check if the word 'fatal' appears in the title and then produces the table for either all adverse events or only fatal ones. Thus several runs of one program can produce several different tables.

The template library

The report templates contain the information used to turn a standardised data set into the desired report. We have two template libraries, one containing templates used in Proc Report, the other those used in Microsoft Access. When using Proc Report as the table generator I found that much of code was similar for each different report type. By breaking up the Proc Report code into three parts (top, middle and bottom). I found I only need a new middle section for each type of report. My report templates consisted only of a few statements which where %INCUDED inside a PROC REPORT step.

These where the COL statement, and the DEFINE statements for the non-control (display) variables. Producing tables using Proc Report from the standardised data sets turned out to be an extremely quick and easy process. One of my colleagues found that it was also easy to create the tables in Microsoft Access by applying templates to a DB4 file of the standardised data.

Interface between the control data set, SAS program library and the standardised data set.

To link the system together I can either use a SAS/AF® application or a batch job. The AF application is extremely useful to develop the programs and produce a small number of reports, while I prefer to use the batch job to produce many reports a once.
The AF application can:

- open the source of table information for update.
- enables one or many tables to be selected.
- runs the appropriate SAS programs, which create the standardised data sets.
- create tables using PROC REPORT.
- send PROC REPORT tables to Microsoft Word and run a Word macro to get the right page size, fonts etc.
- export the data in a form for other software packages to create the tables.

Example of final table produced by PROC REPORT and sent to Microsoft Word

```
SPONSOR:      BIG DRUG COMPANY
PROTOCOL NO:  WONDER DRUG 536372
                        Table 6.5.1 Baseline demographic characteristics (MITT, N=597)

_____
Characteristic                 Statistic                                              *p-value
                                              Wonder Drug              Old Drug
_____

    Study-1a                                  N=172                    N=251

    Sex
        Male                   N(%)              124 ( 72%)               162 ( 65%)         0.4110
        Female                 N(%)               48 ( 28%)                89 ( 35%)

    Age
        <= 65                  N(%)               89 ( 52%)               117 ( 47%)         0.7808
        65 +                   N(%)               83 ( 48%)               134 ( 53%)

                               N              172                      251                  0.5814
                               Mean(SD)          64.4(11.92)              65.6(10.96)
                               Median         65                       66
                               Range          38-86                    32-85

=======================================================================================================
* p-values are determined using the Kruskal-Wallis test for continuous variables and the        Page 1 of 1
chi-squared test for categorical variables


SOURCE: HMR - STAT_CDM -demogtab- - Status: DRAFT - Responsible: David Trenery (08JUL98, 18:19:42)
```

Conclusion

The table production system described here is modular and its core consists of a control data set, a library of programs that produce the standardised data sets, and report templates that when applied to the standardised datasets produce the reports. I found this design imposed the structure necessary for the successful development of a large number of tables by many programmers. It still gave us the flexibility to change elements of the table production process, particularly at the final table production stage. Indeed we found the process so easy that we produce reports both with Proc Report and Microsoft Access.

David Trenery
Hoechst Marion Roussel, Broadwater Park, Denham, Uxbridge, Middx UB9 5HP,UK
(0044) 1895-837803
David.Trenery@hmrag.com