

Paper 045-2007

Group Headers by Template with Optional Columns

Hsiwei Yu, Ming Tech, Flushing, New York

Dong-Min Shen, Ph. D., Schering-Plough, Kenilworth, New Jersey

Proc Template is a solution to produce textual report with requirements of complicated group column heading. Our reports have a pattern of varying number of columns, so instead of creating one template for a different number of columns, we use the optional column capability in Proc Template to produce a permanent template. Report program dynamically instantiates this base template to produce report with different number of columns or toggle on/off groups of columns. We reduced program complexity and execution time.

Introduction

We are required to produce a report with group headings in sequential order (Fig. 1 and 2). The requirement is columns are of three groups, i.e. tests, acceptable ranges, and measurements. The first two groups can have up to three columns, the third group can have many columns, and the second group may be absent. Our solution is to use Proc Template with Data Null Put _ODS_.

Simple Solution

For textual output, Proc Template defines a template with many statements, options, and suboptions. And a Data Null Put _ODS_ step prints the report. Based on data and user selection, we can create different templates for each combination of groups and columns. It is extra work because not only the Proc Template code needs to adjust to its data conditions but also the Data Null step. Complicated program logic is employed in two places. Also create Proc Template every time wastes time.

Program Outline

Get data according to user selection, then *dynamically* generate the following statements.

```
Proc Template; /* Part 1: Define a template. */
Define column group1A ..;
Define column group1B ..;
Define column group2A ..; ..
Define column group3A ..; ..
/* Group heading spans across all columns in its group. */
Define header headerGroup1 .. start= group1A; end= group1B; ..
Define header headerGroup2 .. start= group2A; end= group2B; ..
Run;

Data _null_; /* Part 2: Print with that template. */
File print ods=( .. columns= ( group1A= realName1 group1B= realName2 .. );
Put _ods_;
Run;
```

Features of Proc Template

A column defined in Proc Template can be printed (or bound to) more than once in the data null step. Similarly, a column defined in Proc Template but the data null step may

choose not to print it by not binding it with a real variable name (Fig. 3). We'll call it *optional column* for a column defined once in Proc Template but is printed several times. Secondly, we want group heading to spans across all columns in its group. In Define Header statement, the optional column is referenced in either the start or end suboption. When printed multiple times, the group header text automatically spans all columns.

If a group has only one column, then the start and end suboption must refer to the same value, Dynamic statement is the solution. During print, Dynamic statement is given the correct start or end column name.

This new approach has a permanent template that only needs to be defined once, say during installation. Report simply reuses this template. Only the Data Null print step has complex logic to adjust to many possible user selections. This promotes software reuse, reduces execution time, and follows good programming practice.

Improved Program Outline

Define once a re-usable template.

```
Proc Template;
Dynamic dynGroup1End dynGroup2Start;
Define column group1A ..; /* Print only once, an anchor. */
Define column group1B ..; /* Optional column. */
Define column group2A ..; /* Optional column. */
Define column group2B ..; /* Print only once, an anchor. */
/* Group heading spans across all columns in its group. */
Define header headerGroup1 .. start= group1A; end= dynGroup1End; ..
Define header headerGroup2 .. start= dynGroup2Start; end= group2B; ..
Run;
```

Get data per user input then dynamically generate *only* the appropriate Data Null statements.

```
Data _null_;
File print ods=( .. Columns= ( group1A= realName1 group1B= realName2 ..)
Dynamic= (
dynGroup1End= 'group1B' /* Optional column, not real variable name */
dynGroup2Start= 'group2A' /* Optional column, not real variable name */
) /* Dynamic's close parentheses */
);
Put _ods_;
Run;
```

Conclusion

We are told the new ODS Data Step Object is even better than this, but we are not allowed to use experimental features (as of SAS 9.1). Through this exercise, we gained more understanding of the many useful features in Proc Template. It is an interesting and rewarding learning experience for us.

Contact Information

Hsiwei Yu of Ming Tech in Flushing, NY, can be reached by email,
hsiwei_yu@yahoo.com

Dong-Min Shen, Ph. D. of Schering-Plough, Kenilworth, NJ, email address is dong-min.shen@spcorp.com

Fig. 1 Three group headings, each heading spans across all columns in its group.

Report Title						
Group 1 Heading		For the 2nd set		For the 3rd set		
G1 Fixed Column	G1 Optional 2	G2 Optional 1	G2 Fixed Column	G3 Fixed Column	G3 Optional 1	G3 Optional 2
Test	Specific Item	Acceptable Range 2	Acceptable Range	Measure 1	Measure 2	Measure 3

Fig. 2 The number of columns in a group may vary.

Report Title					
Group 1 Heading			For the 2nd Group	For the 3rd Group	
G1 Fixed Column	G1 Optional 1	G1 Optional 2		G3 Fixed Column	G3 Optional 1
Test	Category	Specific Item	Acceptable Range	Measure 1	Measure 2

Fig. 3 The second group is dynamically suppressed.

Report Title				
Group 1 Heading		For the 3rd set		
G1 Fixed Column	G1 Optional 2	G3 Fixed Column	G3 Optional 1	G3 Optional 2
Test	Specific Item	Measure 1	Measure 2	Measure 3

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.