Paper 4916-2020

Applications of DOW-Loop: Extension to PROC REPORT

Neha Yadav and Pranathi Salla, Axio, a Cytel Company

ABSTRACT

A double DOW-loop in conjunction with PROC REPORT can be used to present order variable information across pages when vertical space is needed to separate blocks of related rows. When creating reports in PDF ODS destination, SPANROWS option is used to repeat the values of GROUP or ORDER variable across the pages. However, sometimes the values are not repeated as expected, especially when a BREAK or LINE statement is used.

The proposed solution uses SPANROWS and a compute block to change the height of the last row in a block of records. Preprocessing is needed since there is no LAST dot (LAST.variable) option in PROC REPORT. A double DOW-loop can be used for this processing. A double DOW-loop consists of two "DO-UNITL" statements. The first "DO-UNTIL" calculates the number of records per each "break" group (LAST.BREAK) and second "DO-UNITL" attaches the calculated number of records to each break group. When used with a compute block, the vertical spacing can be adjusted.

We will present an example that demonstrates the problem and uses the proposed solution using Base SAS® 9.4 in a windows environment. It is appropriate for programmers in any industry with a basic understanding of PROC REPORT. The discussion focus on the use of PDF, though limitations of RTF will be presented.

INTRODUCTION

While creating reports in PDF ODS destination by PROC REPORT, it is advisable to present the data in a format so that it is easy for the reviewers to read.

In PROC REPORT, ORDER variables always repeat across subsequent pages for LISTING destination, but never for RTF and occasionally for PDF when a certain condition is met. SPANROWS option could be used to repeat the value of an ORDER variable across the pages. However, these values are not repeated when a BREAK or LINE statement is used as these statements create their own cell which breaks the spanning cell into two.

To avoid this, a compute block can be used to create a fake blank line by adjusting the vertical spacing after the LAST.RECORD of the blocks of record without using the LINE statement. As PROC REPORT doesn't have Last dot (LAST.VARIABLE) option, a DOW-loop is used to do this preprocessing.

EXAMPLE I

All safety related data is collected in a clinical trial study for safety analysis. The Clinical laboratory evaluation is one of the most important assessments for safety consideration. Laboratory results can be presented in several summary tables. Selected quantitative clinical laboratory variables, i.e. hematology, biochemistry, coagulation and urinalysis summaries using mean, standard deviation, minimum, maximum and median by treatment group at each visit, shift tables and CTCAE (Common Terminology Criteria for Adverse Events) summaries are some of the widely used ones.

In most of the oncology studies, Laboratory results are graded using NCI-CTCAE, where possible. Grades typically range from 1 (least severe) to 5 (most severe), though not all grades are defined for some parameters. Parameters that have criteria available for both low and high values, i.e., hypo- and hyper-, are summarized for both criteria. The same subject can be counted for both values if the subject has different laboratory values meeting each criterion. Shift tables of NCI-CTCAE grade change from baseline to worst post-baseline grade for selected laboratory parameters are generally presented.

Figure 2 and 3 provides an example of such a summary for a parameter called ALT (Alanine Aminotransferase). An increased level to ALT value corresponds to an adverse event term of Alanine Aminotransferase Increased as per the CTCAE criteria. An example of the data is given in Figure 1.

	PARAMCD	sv1	sv3	sv2	row1	row2	row3	_1	_2
1	ALT	1	1	-1	Alanine aminotransferase increased	Baseline	N	65	35
2	ALT	1	1	1	Alanine aminotransferase increased	Baseline	Not Present	65 (100.0%)	34 (97.1%)
3	ALT	1	1	2	Alanine aminotransferase increased	Baseline	Grade 1	0 (0.0%)	1 (2.9%)
4	ALT	2	1.5	1.5	Alanine aminotransferase increased	Cycle 1 Day 15	N	63	35
5	ALT	2	1.5	1.5	Alanine aminotransferase increased	Cycle 1 Day 15	Not Present	32 (50.8%)	35 (100.0%)
6	ALT	2	1.5	2.5	Alanine aminotransferase increased	Cycle 1 Day 15	Grade 1	20 (31.7%)	0 (0.0%)
7	ALT	2	1.5	3.5	Alanine aminotransferase increased	Cycle 1 Day 15	Grade 2	5 (7.9%)	0 (0.0%)
8	ALT	2	1.5	4.5	Alanine aminotransferase increased	Cycle 1 Day 15	Grade 3	6 (9.5%)	0 (0.0%)
9	ALT	2	1.51	1.51	Alanine aminotransferase increased	Shift to Cycle 1 Day 15	N	63	35
10	ALT	2	1.51	1.51	Alanine aminotransferase increased	Shift to Cycle 1 Day 15	0 Grade Shift	32 (50.8%)	34 (97.1%)
11	ALT	2	1.51	2.51	Alanine aminotransferase increased	Shift to Cycle 1 Day 15	1 Grade Shift	20 (31.7%)	0 (0.0%)
12	ALT	2	1.51	3.51	Alanine aminotransferase increased	Shift to Cycle 1 Day 15	2 Grade Shift	5 (7.9%)	0 (0.0%)
13	ALT	2	1.51	4.51	Alanine aminotransferase increased	Shift to Cycle 1 Day 15	3 Grade Shift	6 (9.5%)	0 (0.0%)
14	ALT	2	2	2	Alanine aminotransferase increased	Cycle 2 Day 1	N	62	36
15	ALT	2	2	2	Alanine aminotransferase increased	Cycle 2 Day 1	Not Present	42 (67.7%)	36 (100.0%)
16	ALT	2	2	3	Alanine aminotransferase increased	Cycle 2 Day 1	Grade 1	19 (30.6%)	0 (0.0%)
17	ALT	2	2	4	Alanine aminotransferase increased	Cycle 2 Day 1	Grade 2	1 (1.6%)	0 (0.0%)
18	ALT	2	2.01	2.01	Alanine aminotransferase increased	Shift to Cycle 2 Day 1	N	62	36
19	ALT	2	2.01	2.01	Alanine aminotransferase increased	Shift to Cycle 2 Day 1	0 Grade Shift	42 (67.7%)	35 (97.2%)
20	ALT	2	2.01	3.01	Alanine aminotransferase increased	Shift to Cycle 2 Day 1	1 Grade Shift	19 (30.6%)	0 (0.0%)
21	ALT	2	2.01	4.01	Alanine aminotransferase increased	Shift to Cycle 2 Day 1	2 Grade Shift	1 (1.6%)	0 (0.0%)

Figure 1. Dataset for shift table which is shown in Figure 2 and 3

The dataset contains few numeric sorting variables and some text variables along with _1 and _2 columns which denotes the treatment arms.

CTC Event Term			Arm A (N=65)	Arm B (N=35)
Alanine aminotransferase increased	Baseline	N	65	35
		Not Present	65 (100.0%)	34 (97.1%)
		Grade 1	0 (0.0%)	1 (2.9%)
	Cycle 1 Day 15	N	63	35
		Not Present	32 (50.8%)	35 (100.0%)
		Grade 1	20 (31.7%)	0 (0.0%)
		Grade 2	5 (7.9%)	0 (0.0%)
		Grade 3	6 (9.5%)	0 (0.0%)
	Shift to Cycle 1 Day 15	N	63	35
		0 Grade Shift	32 (50.8%)	34 (97.1%)
		1 Grade Shift	20 (31.7%)	0 (0.0%)
		2 Grade Shift	5 (7.9%)	0 (0.0%)
		3 Grade Shift	6 (9.5%)	0 (0.0%)

Figure 2. Example of Laboratory Shift table (Page 1)

CTC Event Term			Arm A (N=65)	Arm B (N=35)
	Shift to Cycle 2 Day 15	N	57	33
		0 Grade Shift	34 (59.6%)	32 (97.0%)
		1 Grade Shift	21 (36.8%)	0 (0.0%)
		2 Grade Shift	2 (3.5%)	0 (0.0%)
	Cycle 3 Day 1	N	61	34
		Not Present	37 (60.7%)	34 (100.0%)
		Grade 1	24 (39.3%)	0 (0.0%)
	Shift to Cycle 3 Day 1	N	61	34
		0 Grade Shift	37 (60.7%)	33 (97.1%)
		1 Grade Shift	24 (39.3%)	0 (0.0%)

Figure 3. Example of Laboratory Shift table (Page 2)

The laboratory shift table (figure 2 and figure 3) is created using PROC REPORT step where we can see that the first column, i.e. Alanine Aminotransferase increased is not repeating on the subsequent page. While we expect that the order variable will repeat on subsequent pages, it does not as the merge tables are broken down as shown in figure 4 by adding the compute statement as shown in snippet code 1.

CTC Event Term			Arm A (N=65)	Arm B (N=35)
Alanine aminotransferase increased	Baseline	N	65	35
		Not Present	65 (100.0%)	34 (97.1%)
		Grade 1	0 (0.0%)	1 (2.9%)
	Cycle 1 Day 15	N	63	35
		Not Present	32 (50.8%)	35 (100.0%)
		Grade 1	20 (31.7%)	0 (0.0%)
		Grade 2	5 (7.9%)	0 (0.0%)
		Grade 3	6 (9.5%)	0 (0.0%)
	Shift to Cycle 1 Day 15	N	63	35
		0 Grade Shift	32 (50.8%)	34 (97.1%)
		1 Grade Shift	20 (31.7%)	0 (0.0%)
		2 Grade Shift	5 (7.9%)	0 (0.0%)
		3 Grade Shift	6 (9.5%)	0 (0.0%)
	Cycle 2 Day 1	N	62	36
		Not Present	42 (67.7%)	36 (100.0%)
		Grade 1	19 (30.6%)	0 (0.0%)
		Grade 2	1 (1.6%)	0 (0.0%)
	Shift to Cycle 2 Day 1	N	62	36
		0 Grade Shift	42 (67.7%)	35 (97.2%)
		1 Grade Shift	19 (30.6%)	0 (0.0%)
		2 Grade Shift	1 (1.6%)	0 (0.0%)

Figure 4. Default merged cells broken due to LINE statement

```
compute after sv3 ;
    line ' ';
endcomp;
```

Code 1. Inserting a blank line in between groups defined by SV3

The sole purpose of introducing this blank line is to create a better aesthetic. Otherwise, the code without the compute block results in the output provided in figure 5. This output is difficult to read.

CTC Event Term			Arm A (N=65)	Arm B (N=35)
Alanine aminotransferase increased	Baseline	N	65	35
		Not Present	65 (100.0%)	34 (97.1%)
		Grade 1	0 (0.0%)	1 (2.9%)
	Cycle 1 Day 15	N	63	35
		Not Present	32 (50.8%)	35 (100.0%)
		Grade 1	20 (31.7%)	0 (0.0%)
		Grade 2	5 (7.9%)	0 (0.0%)
		Grade 3	6 (9.5%)	0 (0.0%)
	Shift to Cycle 1 Day 15	N	63	35
		0 Grade Shift	32 (50.8%)	34 (97.1%)
		1 Grade Shift	20 (31.7%)	0 (0.0%)
		2 Grade Shift	5 (7.9%)	0 (0.0%)
		3 Grade Shift	6 (9.5%)	0 (0.0%)

Figure 5. Table generated using PROC REPORT but without inserting a blank line

Without using the compute statement (as in snippet code 1), it repeats the order variables on the next pages as expected but in order to have a presentable report, the single cell which spans all rows for the order value is broken due to LINE statement. The solution is to use a compute block to create a fake blank line by adjusting the vertical spacing after the LAST.RECORD of the blocks of record without using a LINE statement. As PROC REPORT doesn't have Last dot (LAST.VARIABLE) option, DOW loop is used to do this preprocessing as shown in snippet code 2.

The DOW loop preprocessing includes double "DO-UNTIL" statements where the first "DO-UNTIL" counts the total number of related records in each block (TROW) and second "DO-UNTIL" attaches the calculated number of records to each block. This is done in a data step prior the PROC REPORT. A dummy variable is defined in PROC REPORT (COUNT) as a computed variable which also counts the number of records in related block and when this count is same as the count of TROW, vertical spacing is adjusted using a call define statement. The complete code is shown in Appendix.

```
data final;
    do n=1 by 1 until(last.sv3);
    set rpt;
    by paramcd sv0 sv1 sv3 sv2 toxgr;
    if first.sv3 then trow=1;
    else trow+1;
    end;
    do until(last.sv3);
    set rpt;
    by paramcd sv0 sv1 sv3 sv2 toxgr;
    output;
    end;
run;
```

Code 2. Preprocessing using a DOW loop

This results in dataset shown in figure 6, where TROW is equal to the count of rows for each related block. In our example, it's for every distinct value of SV3.

	row1	row2	row3	trow
1	Alanine aminotransferase increased	Baseline	N	3
2	Alanine aminotransferase increased	Baseline	Not Present	3
3	Alanine aminotransferase increased	Baseline	Grade 1	3
4	Alanine aminotransferase increased	Cycle 1 Day 15	N	5
5	Alanine aminotransferase increased	Cycle 1 Day 15	Not Present	5
6	Alanine aminotransferase increased	Cycle 1 Day 15	Grade 1	5
7	Alanine aminotransferase increased	Cycle 1 Day 15	Grade 2	5
8	Alanine aminotransferase increased	Cycle 1 Day 15	Grade 3	5
9	Alanine aminotransferase increased	Shift to Cycle 1 Day 15	N	5
10	Alanine aminotransferase increased	Shift to Cycle 1 Day 15	0 Grade Shift	5
11	Alanine aminotransferase increased	Shift to Cycle 1 Day 15	1 Grade Shift	5
12	Alanine aminotransferase increased	Shift to Cycle 1 Day 15	2 Grade Shift	5
13	Alanine aminotransferase increased	Shift to Cycle 1 Day 15	3 Grade Shift	5

Figure 6. Final dataset before PROC REPORT with TROW variable

As shown in snippet code 3, a dummy variable is computed in PROC REPORT to keep count of the total number of records for each break variable (i.e. where a blank line is desired). When this count matches with the count of TROW variable (as highlighted in figure 7), call define statement is executed. This adjusts the vertical spacing of the cells, giving it an appearance of blank line without exactly breaking the merged cell, and hence resulting in repeating the ORDER variables to subsequent pages.

row1	row2	row3	trow	count
Alanine aminotransferase increased	Baseline	N	3	1
Alanine aminotransferase increased	Baseline	Not Present	3	2
Alanine aminotransferase increased	Baseline	Grade 1	3	3
Alanine aminotransferase increased	Cycle 1 Day 15	N	5	1
Alanine aminotransferase increased	Cycle 1 Day 15	Not Present	5	2
Alanine aminotransferase increased	Cycle 1 Day 15	Grade 1	5	3
Alanine aminotransferase increased	Cycle 1 Day 15	Grade 2	5	4
Alanine aminotransferase increased	Cycle 1 Day 15	Grade 3	5	5
Alanine aminotransferase increased	Shift to Cycle 1 Day 15	N	5	1
Alanine aminotransferase increased	Shift to Cycle 1 Day 15	0 Grade Shift	5	2
Alanine aminotransferase increased	Shift to Cycle 1 Day 15	1 Grade Shift	5	3
Alanine aminotransferase increased	Shift to Cycle 1 Day 15	2 Grade Shift	5	4
Alanine aminotransferase increased	Shift to Cycle 1 Day 15	3 Grade Shift	5	5

Figure 7. PROC REPORT resulted dataset with the dummy computed COUNT variable

```
compute before sv3; cnt=0; endcomp;
  compute count; cnt+1; count=cnt;
        if count=trow then do;call define(_row_,'style','style=[height=20pt]');end;
  endcomp;
```

Code 3. Dummy variable count created in PROC REPORT to calculate the number of records for each BREAK variable

The DOW loop with PROC REPORT processing results in repetition of ORDER variable values to next pages as shown in figure 8.

CTC Event Term			Arm A (N=65)	Arm B (N=35)
Alanine aminotransferase increased	Shift to Cycle 2 Day 15	N	57	33
		0 Grade Shift	34 (59.6%)	32 (97.0%)
		1 Grade Shift	21 (36.8%)	0 (0.0%)
		2 Grade Shift	2 (3.5%)	0 (0.0%)
	Cycle 3 Day 1	N	61	34
		Not Present	37 (60.7%)	34 (100.0%)
		Grade 1	24 (39.3%)	0 (0.0%)
	Shift to Cycle 3 Day 1	N	61	34
		0 Grade Shift	37 (60.7%)	33 (97.1%)
		1 Grade Shift	24 (39.3%)	0 (0.0%)

Figure 8. Laboratory Shift table Example (Page 2) after using the DOW loop preprocessing in conjunction with PROC REPORT

EXAMPLE II

Data listings is one of those which should be provided with unique identifiers on each page of the report. Everybody knows what goes in a data listing, the actual task of a programmer is to know how to organize and present the data in a way so that others can quickly and easily gather what they need to know from it specially in cases with large data listings.

ICH E3 Guideline requests to include many data listings and the most important ones are the safety summary ones such as Adverse Events listings. In the pharmaceutical industry AE summaries play a very significant role as a safety issue. The drug record is often used in conjunction with adverse events to determine whether adverse events were treatment emergent.

It is suggested to provide more thorough adverse event information specifically subjects identifiers and verbatim terms for each adverse event in the report, in Sections 14.3.1 and 16.2.7. Safety data is utmost important as reviewers concentrate on individual cases, which data listings helps to achieve this process.

The adverse event listings tend to have too many pages. Lots of data fields are combined into single column to assure the adverse event data for a subject is compact and reviewer can review all of a subject's events in few pages. However, since lots of data fields of interest with length responses can result in uneven wrapping onto new lines or uneven page breaks. Since the most important thing Is readability, it is always a good practice to show the important information such as treatment information and unique identifiers across the pages to make it easy for review purposes.

Figure 9 and 10 shows an example of adverse event listing where the values of treatment group and subject ID column values are not repeated on page 2 and onwards because of the LINE statement.

Treatment Group	Subject ID	System Organ Class/ Preferred Term/ Adverse Events	Start Date/ Time (Study Day)	Stop Date/ Time (Study Day)	Ongoing?	TEAE?	Toxicity Grade	Action Taken with Study Treatment	Other Action Taken with Subject	Relationship to Study Treatment	Outcome
TRT A	ABC-0004	Gastrointestinal Disorders/ Nausea/ Intermittent Nausea			No	Yes	Grade 1	Dose Not Changed	None	Not Related	Not Recovered/Not Resolved
		Infections And Infestations/ Urinary Tract Infection/ Urinary Tract Infection	19MAY2018 (97)		Yes	Yes	Grade 1	Dose Not Changed	None	Not Related	Recovered/ Resolved
		Vascular Disorders/ Intermittent Claudication/ Endovenous Ablation Of Right Great Saphenous Vein Due To Claudication/Leg Pain	30MAY2018 (108)		Yes	Yes	Grade 1	Drug Interrupted	None	Not Related	Recovered/ Resolved

Figure 9. Adverse Event Listing (Page 1)

Treatment Group	Subject ID	System Organ Class/ Preferred Term/ Adverse Events	Start Date/ Time (Study Day)	Stop Date/ Time (Study Day)	Ongoing?	TEAE?	Toxicity Grade	Action Taken with Study Treatment	Other Action Taken with Subject	Relationship to Study Treatment	Outcome
		Uncoded/ Uncoded/ Worsen Of Diabetes	04AUG2019 (136)		No	Yes	Grade 1	Dose Not Changed	None	Not Related	Not Recovered/Not Resolved
		Uncoded/ Uncoded/ Urinary Tract Infection	16AUG2019 (161)		Yes	Yes	Grade 1	Not Applicable	None	Not Related	Recovered/ Resolved
	ABC-0010	Injury, Poisoning And Procedural Complications/ Infusion Related Reaction/ Chest Pains Due To Infusion Reaction	19FEB2018/ 09:30 (19)	19FEB2018/ 11:30 (19)	Yes	Yes	Grade 3	Drug Withdrawn	Required Withdrawal From Study	Definitely Related	Recovered/ Resolved

Figure 10. Adverse Event Listing (Page 2) where the columns, Treatment Group and Subject ID are not repeated.

Using DOW loop along with PROC REPORT, as shown in snippet code 4, gives the appearance of a blank line without breaking the merging which provides better presentation of listing.

```
data final;
    do n=1 by 1 until(last.usubjid);
    set rpt;
    by trt01an usubjid aestdtc aestdtc socpt;
    if first.usubjid then trow=1;
    else trow+1;
    end;
    do until(last.usubjid);
    set rpt;
    by trt01an usubjid aestdtc aestdtc socpt;
    output;
    end;
run;
```

```
compute before usubjid; cnt=0; endcomp;
compute count; cnt+1; count=cnt;
      if count=trow then do;call define(_row_,'style','style=[height=40pt]');end;
endcomp;
```

Code 4. DOW loop in conjunctions with PROC REPORT

This results in the desired output as shows in figure 11.

Treatment Group	Subject ID	System Organ Class/ Preferred Term/ Adverse Events	Start Date/ Time (Study Day)	Stop Date/ Time (Study Day)	Ongoing?	TEAE?	Toxicity Grade	Action Taken with Study Treatment	Other Action Taken with Subject	Relationship to Study Treatment	Outcome
TRT A	ABC-0010	Injury, Poisoning And Procedural Complications/ Infusion Related Reaction/ Chest Pains Due To Infusion Reaction	19FEB2018/ 09:30 (19)	19FEB2018/ 11:30 (19)	Yes	Yes	Grade 3	Drug Withdrawn	Required Withdrawal From Study	Definitely Related	Recovered/ Resolved
		Injury, Poisoning And Procedural Complications/ Infusion Related Reaction/ Light Headedness Due To Infusion Reaction	19FEB2018/ 09:30 (19)	19FEB2018/ 11:30 (19)	Yes	Yes	Grade 2	Drug Withdrawn	Required Withdrawal From Study	Definitely Related	Recovered/ Resolved

Figure 11. Adverse events listing (Page 2) with repeated values of the columns

RESTRICTIONS

This application cannot be extended to RTF destination since RTF destination does not measure vertically. When PROC REPORT uses a GROUP or ORDER variable and the values span more than one page, the value is not repeated at the top of the second and subsequent pages as PROC REPORT sends table instructions to Microsoft Word but leaves much of the pagination up to the word processor. However, the new RTF tagset available in SAS® 9.2, in conjunction with the new SPANROWS option on the PROC REPORT statement, provides a way to repeat the value of a GROUP or ORDER variable if that variable's value spans more than one page.

CONCLUSION

An extension of the application of Double DOW loop in association with PROC REPORT is to repeat the values of order or group variables on subsequent pages when the record with same ORDER value span multiple pages. This is an efficient programming technique which eliminates the need of multiple data steps resulting in an aesthetically pleasing report.

REFERENCES

1. Richard Read Allen. Practical Uses of the DOW Loop, WUSS 2009 Proceedings
2. Nancy Brucken. One-Step Change from Baseline Calculations, and Other DOW Loop Tricks, MWSUG 2008 Proceedings
3. Mario Widel, Eli Lilly & Co, Indianapolis, IN, Good versus Better SDTM: Data Listings, PharmaSUG 2016 Proceedings
4. http://support.sas.com/kb/53/783.html (Usage Note 53783: GROUP or ORDER variables in PROC REPORT might not repeat with the SPANROWS option if BREAK or LINE statements are used)

ACKNOWLEDGMENTS

We would like to thank William Coar and Emily Woolley for the useful discussion and review of this paper.

CONTACT INFORMATION

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

Neha Yadav
Axio Research, LLC
2601 4th Ave #200
Seattle WA 98121
(206) 547-2829
nehay@axioresearch.com

Pranathi Salla
Axio Research,LLC
2601 4th Ave #200
Seattle WA 98121
(206) 547-2829
pranathis@axioresearch.com

APPENDIX I

```
****Pre-processing using double DOW loop before PROC REPORT;
   data final;
    do n=1 by 1 until(last.SV3);
    set rpt;
    by paramcd sv0 sv1 SV3 sv2 toxgr;
    if first.SV3 then TROW=1;
    else TROW+1;
    end;
    do until(last.SV3);
    set rpt;
    by paramcd sv0 sv1 SV3 sv2 toxgr;
    output;
    end;
   run;


****Proc Report;
   ODS escapechar='~';
   ODS LISTING CLOSE;
   ODS PDF file ="..\report.pdf" style=pdf notoc;

   Proc Report data=final nowd spanrows center headline nowindows missing
   split='!';
   column flag page paramcd sv0 row1 sv1 SV3 row2 sv2  toxgr row3 _1 _2
   TROW count ;

    define flag       / order order=data noprint ;
    define page       / order order=data noprint ;
    define paramcd    / order order=data noprint ;

    define sv0        / order order=internal noprint;
    define sv1        / order order=internal noprint;
    define SV3        / order order=internal noprint;
    define sv2        / order order=internal noprint;

    define toxgr      / order order=internal noprint;

    define row1       / order order=internal left " " style header=[just=L]
                      style(column)=[cellwidth=18%];
    define row2       / order order=internal left " " style header=[just=L]
                      style(column)=[cellwidth=35%];
    define row3       / order order=internal left " " style header=[just=L]
                      style(column)=[cellwidth=15%];


    define _1         / display "TRT A !&TRT1" style(header)=[just=C]
                      style(column)=[cellwidth=15% just=C vjust=top];

    define _2         / display "TRT B !&TRT2" style(header)=[just=C]
                      style(column)=[cellwidth=15% just=C vjust=top];

    define TROW       / display noprint;
    define count      / computed noprint;
```

```
compute before SV3; cnt=0; endcomp;

compute count; cnt+1; count=cnt;
    if count=TROW then do;
    call define(_row_,'style','style=[height=20pt]');end;
endcomp;


break before flag / page contents='';

run;

ODS PDF CLOSE;
ODS LISTING;
```