

Table Numbers - Chaos to Simplicity

Michael Stout, DePuy Orthopaedics, Inc, a Johnson & Johnson Company
Warsaw, IN

ABSTRACT

Using the SAS® ODS® facility with automatic table numbering is an excellent method for packaging reports (tables) into a single output destination. Some reports are large and may contain hundreds of tables produced by SAS® procedures and data step processing. Managing large numbers of tables can be challenging and difficult. It can be extremely difficult to insert, delete or re-arrange tables without affecting table numbers. Although ODS® does a good job of bundling reports (tables), it does not automatically assign table numbers. This paper shows one method that may be used to dynamically assign table numbers while allowing the programmer to move from a state of chaos to simplicity.

BACKGROUND

Data Management produces several reports and needed a way to easily change table numbers. In the past we manually updated table numbers. This involved making changes to SAS® programs. It was tedious, time consuming and required a lot of quality assurance to make sure table numbers were correct; especially when massive updates were done to the report that involved deleting, adding or rearranging tables. This led to the development of the Set_Table_Titles macro.

MACRO

The Set_Tables_Titles macro is used to describe the first, second and third title lines. Only the third title line may be passed as a parameter to this macro.

SYNTAX: Set_Table_Titles macro (Title3Text= ,TableDesc=)

The first title line is automatically created by this procedure and will contain the table number. The second title line is assigned the value of macro variable &title2text which should be set prior to running this macro. Use a %let statement to assign the report name (ie. %let title2text = PharmaSUG Attendance Summary Report). It may contain the study name, report name or any other text such as the system date and will be the same for each table in the report.

This procedure has two parameters: Title3Text and TableDesc. Title line three contains text that clearly describes the table and is assigned to named parameter Title3Text. Parameter TableDesc will contain special codes that tell the macro how to display the table number.

SPECIAL CODES

Special codes are used and consist of two parts that are separated by a dash. The dash is optional when part two is blank. The first part of the code is used to format the table number. The second part of the code is used to reset the value assigned to segments of the table number.

Part One (FORMAT)

T	Display a Table number
TC	Display Table number and upper case letter
TCN	Display Table number, upper case letter and number
TCNX	Display Table number, upper case letter, number, and lower case letter

Part Two (RESET)

-T	Add one to segment one and reset all other segments
-C	Assign next upper case letter and reset remaining segments
-N	Add one to the third segment and reset the forth segment

Here are some sample codes that may be used to control the table numbers. If the second part of the code is omitted, the right most table number is incremented by 1 (ie. 1 → 2 and 1A → 1B).

T	Table 1
TC-T	Table 2A
TC	Table 2B
TC	Table 2C
TC-T	Table 3A
TCN	Table 3A.1

TABLE NUMBER

The table number has the following format and can consist of up to four distinct values. A dot is used to separate the third and fourth values of the table number, when they are used.

<N><A-Z>.<N>.<a-z>

where N is a number,
A-Z is an upper character from A-Z
a-z is a lower case letter from a-z

USING THE MACRO

```
/* PRE-PROCESSING */
%global auto_table auto_Subtab1 auto_Subtab2 auto_Subtab3 title2text;
%let auto_table = 0;
%let auto_Subtab1 = 0;
%let auto_Subtab2 = 0;
%let auto_Subtab3 = 0;
%let title2text = PharmaSUG SAS Users Group;

/* ASSIGN OUTPUT DESTINATION */
ODS RTF FILE='C:\TEMP\PharmaSUG.rtf';

/* CREATE OUTPUT */
%Set_Table_Titles(Title3Text=Summary Report,tableDesc=T); *1*;
proc print data=TableList;
run;
%Set_Table_Titles(Title3Text=Summary Report,tableDesc=TC-T); *2A*;
proc print data=TableList;
run;
%Set_Table_Titles(Title3Text=Summary Report,tableDesc=TC); *2B*;
proc print data=TableList;
run;
%Set_Table_Titles(Title3Text=Summary Report,tableDesc=TCN-C); *2C.1*;
proc print data=TableList;
run;
%Set_Table_Titles(Title3Text=Summary Report,tableDesc=TCN); *2C.2*;
proc print data=TableList;
run;
%Set_Table_Titles(Title3Text=Summary Report,tableDesc=TCN); *2C.3*;
proc print data=TableList;
run;
%Set_Table_Titles(Title3Text=Summary Report,tableDesc=TCNX); *2C.3.a*;
proc print data=TableList;
run;
```

```

%Set_Table_Titles(Title3Text=Summary Report,tableDesc=TCNX); *2C.3.b*;
proc print data=TableList;
run;
%Set_Table_Titles(Title3Text=Summary Report,tableDesc=TCNX); *2C.3.c*;
proc print data=TableList;
run;
%Set_Table_Titles(Title3Text=Summary Report,tableDesc=TCN-C); *2D.1*;
proc print data=TableList;
run;
%Set_Table_Titles(Title3Text=Summary Report,tableDesc=TCN); *2D.2 *;
proc print data=TableList;
run;
%Set_Table_Titles(Title3Text=Summary Report,tableDesc=TC-T); * 3A *;
proc print data=TableList;
run;
%Set_Table_Titles(Title3Text=Summary Report,tableDesc=TC); * 3B *;
proc print data=TableList;
run;
%Set_Table_Titles(Title3Text=Summary Report,tableDesc=T); * 4 *;
proc print data=TableList;
run;
%Set_Table_Titles(Title3Text=Summary Report,tableDesc=T); * 5 *;
proc print data=TableList;
run;

/* Close Output Destination */
ODS RTF close;
run;

```

CONSIDERATIONS

This program uses the SAS® function called COLLATE®. This function only works on Microsoft® Windows® operating systems. Care must be used when assigning table numbers since there are only twenty-six letters in the alphabet. This means the second and forth segments of the table number are limited to 26 values. If more than 26 characters are used, the system will display special characters.

CONCLUSION

Incorporating dynamic table numbers into reports with a large number of tables can reduce the amount of time and energy required to make changes to reports. Since table numbers are dynamic, it is easy to add, delete and rearrange tables. Reduce your stress level and move from a state of chaos to simplicity.

REFERENCES

SAS® Institute Inc. (1996). *SAS® Companion for the Microsoft Windows Environment, Version 6*. Cary, NC:SAS® Institute Inc.

ACKNOWLEDGMENTS

I would like to thank John Tescula, SAS Programmer, for helping to write and test the SAS code used to automate table numbering for tables, figures and listings. I would also like to thank Paul Voorhorst for reviewing the file paper.

CONTACT INFORMATION

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

Michael Stout
Team Lead, SAS Programming

DePuy Orthopaedics, Inc.
PO Box 988
700 Orthopaedic Drive
Warsaw, Indiana 46581-0988
Work Phone: (574) 372 7379
Fax: (574) 371 4950
E-mail: Mstout2@dpyus.jnj.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.

SAS CODE

```
/* Assign titles to table. This macro is called by Set_Table_Titles. */
%macro Set_Titles;
  title1 "&Text1";
  Title2 "&title2text";
  Title3 "&title3text";
run;
%mend;

%macro Set_Table_Titles(Title3Text= ,TableDesc= );
  data Title_null;
  length tab_name $ 50;
  parm2 = scan("&TableDesc",2,'-');
  parm1 = trim(left(scan("&TableDesc",1)));

  T_name = &Auto_Table;
  t_char = &Auto_SubTab1;
  T_num = &Auto_SubTab2;
  t_small_char = &Auto_SubTab3;

  /*
  Check the second part of code. This code is used
  bump up the table number and reset values for all segments
  that come after that segment. For example, code TCNX-T with
  a value of 2C.4c would become 3A.1a. The two is bumped up
  to 3 and all other segments are reset to their lowest value.

  The second part of the parameter use a T, C or N. All other
  values are ignored. T is the first segment, C the second and N
  is the third segment of the table number.
  */

  /* Code for Second segment of table number */
  If parm2 = 'C' then do;
    If parm1 = 'TCN' then do;
      t_char = t_char + 1;
      t_num= 0;
      t_small_char = 0;
    end;
    If parm1 = 'TCNX' then do;
      t_char = t_char + 1;
      t_num = 1;
    end;
  end;
end;
```

```

        t_small_char = 0;
    end;
end;

/* Code for Third segment of table number */
If parm2 = 'N' then do;
    If parm1 = 'TCNX' then do;
        t_num = t_num + 1;
        t_small_char = 0;
    end;
end;

/* Code for First segment of table number */
if parm2 = 'T' then do;
    if parm1 = 'TCNX' then do;
        t_name = t_name + 1;
        t_char = 1;
        t_num = 1;
        t_small_char = 0;
    end;
    else if parm1 = 'TCN' then do;
        t_name = t_name + 1;
        t_char = 0;
        t_num = 0;
        t_small_char = 0;
    end;
    else if parm1 = 'TC' then do;
        t_name = t_name + 1;
        t_char = 0;
        t_num = 0;
        t_small_char = 0;
    end;
end;

/* Determine the Table number that will be assigned to Title Line 1. */
if parm1 = 'T' then do; /* and parm2 = '' */
    t_char = 0;
    t_num = 0;
    t_small_char = 0;

    t_name = T_name + 1;
    tab_name = 'Table ' || left(put(t_name,6.));
    parm2 = '';
end;
else if parm1 = 'TC' then do;
    t_char = T_char + 1;
    t_num = 0;
    t_small_char = 0;
    tab_name = 'Table ' || trim(left(put(t_name,6.))) ||
collate(64+t_char,,1) ;
end;
else if parm1 = 'TCN' then do;
    t_num = T_num + 1;
    t_small_char = 0;
    tab_name = 'Table ' || trim(left(put(t_name,6.))) ||
collate(64+t_char,,1) ||
    '.' ||
    trim(left(put(t_num,6.))) ;
end;

```

```

end;
else if parml = 'TCNX' then do;
    t_small_char = T_small_char + 1;
    tab_name = 'Table ' || trim(left(put(t_name,6.))) ||
               collate(64+t_char,,1) ||
               '.' ||
               trim(left(put(t_num,6.))) ||
               '.' ||
               collate(96+t_small_char,,1);
end;
else tab_name = "Table &Auto_Table";

call symput('Text1',trim(left(TAB_NAME)));
call symput('Auto_Table', t_name);
call symput('Auto_SubTab1',t_char);
call symput('Auto_SubTab2',t_num);
call symput('Auto_SubTab3',t_small_char);
run;

/* Set Titles Lines that will be used by the table */
%Set_Titles
%mend;

```