

PHUSE US Connect 2021

Taking Better Control of ODS Outputs Using ODS Tagsets

- Kamlesh Patel, AVP
- Rang Technologies Inc, Piscataway, New Jersey
- Kamlesh.patel1@rangtech.com



Agenda

- Introduction: ODS Tagsets
- Background Concepts
- Various tips
 - Tip 1. Getting started and how it works
 - Tip 2. Identifying events
 - Tip 3. Using various options
- Summary
- Q&A

Introduction: ODS Tagsets

- ODS Output

Traditional ODS

Controls width, not height

Title/Footnotes goes in
RTF title/footnote section

Higher memory usage

ODS Tagsets

Controls width & height

Title/Footnotes one block

Low memory usage

Introduction: ODS Tagsets

- Supplied with many in-built
- To view list -

*1. To view all tagsets;

```
proc template;  
  list tagsets;  
run;
```

*To view source code of
tagsets;

```
proc template;  
  source tagsets.rtf;  
run;
```

Listing of: SASHELP.TMPL_EN		
Path Filter is: Tagsets		
Sort by: PATH/ASCENDING		
Obs	Path	Type
1	Tagsets	Dir
2	Tagsets.Chtml	Tagset
3	Tagsets.Config_debug	Tagset
4	Tagsets.Csv	Tagset
5	Tagsets.Csvall	Tagset
6	Tagsets.Csvbyline	Tagset

```
74          source tagsets.rtf;  
define tagset Tagsets.Rtf;  
  notes "This is the RTF.TPL definition";  
  
  define event changes;  
    start:  
      set $log_note  
        "NOTE: This is the RTF tagset (  
information.";   
      set $changelog[] "-";  
      set $changelog[] ".v1.11, 02/07/20";  
      set $changelog[] "VSPACE option ad";  
      set $changelog[] " ";
```

Background Concepts

- Event –
 - Occurrence of some specific action that decides what is to be written to the output file.
 - Defined by “define event” statement in Proc TEMPLATE
- Variable –
 - Like any other SAS variable, it holds data value

```
define event proc;  
  start:  
    do /if $order_repeat;  
      set $proc_name "Report" /if cmp( proc_name, "Report");  
    done;  
  finish:  
    unset $proc_name;  
end;
```

TIP I. LEARN TO CREATE SIMPLE TAGSETS

- STEP I-A > Just creating tagsets!!!

* 1-A > First Tagset Creation;

```
proc template;
```

```
    define tagset tagsets.learning;
```

```
        define event data; *Start of event;
```

```
            put value; *Putting value;
```

```
        end; *End of event;
```

```
    end;
```

```
run;
```

```
ods tagsets.learning file="learning.txt";
```

```
    proc print data=sashelp.class;
```

```
    run;
```

```
ods _all_ close;
```

- STEP I-A > Just creating tagsets!!!
Output looks like below:

```
File Edit Format View Help
AlfredM1469.0112.5AliceF1356.584.0BarbaraF1365.398.0CarolF1462.8102.5HenryM1463.
5102.5JamesM1257.383.0JaneF1259.884.5JanetF1562.5112.5JeffreyM1362.584.0JohnM125
9.099.5JoyceF1151.350.5JudyF1464.390.0LouiseF1256.377.0MaryF1566.5112.0PhilipM16
72.0150.0RobertM1264.8128.0RonaldM1567.0133.0ThomasM1157.585.0WilliamM1566.5112.
0
```

Data values are one after another

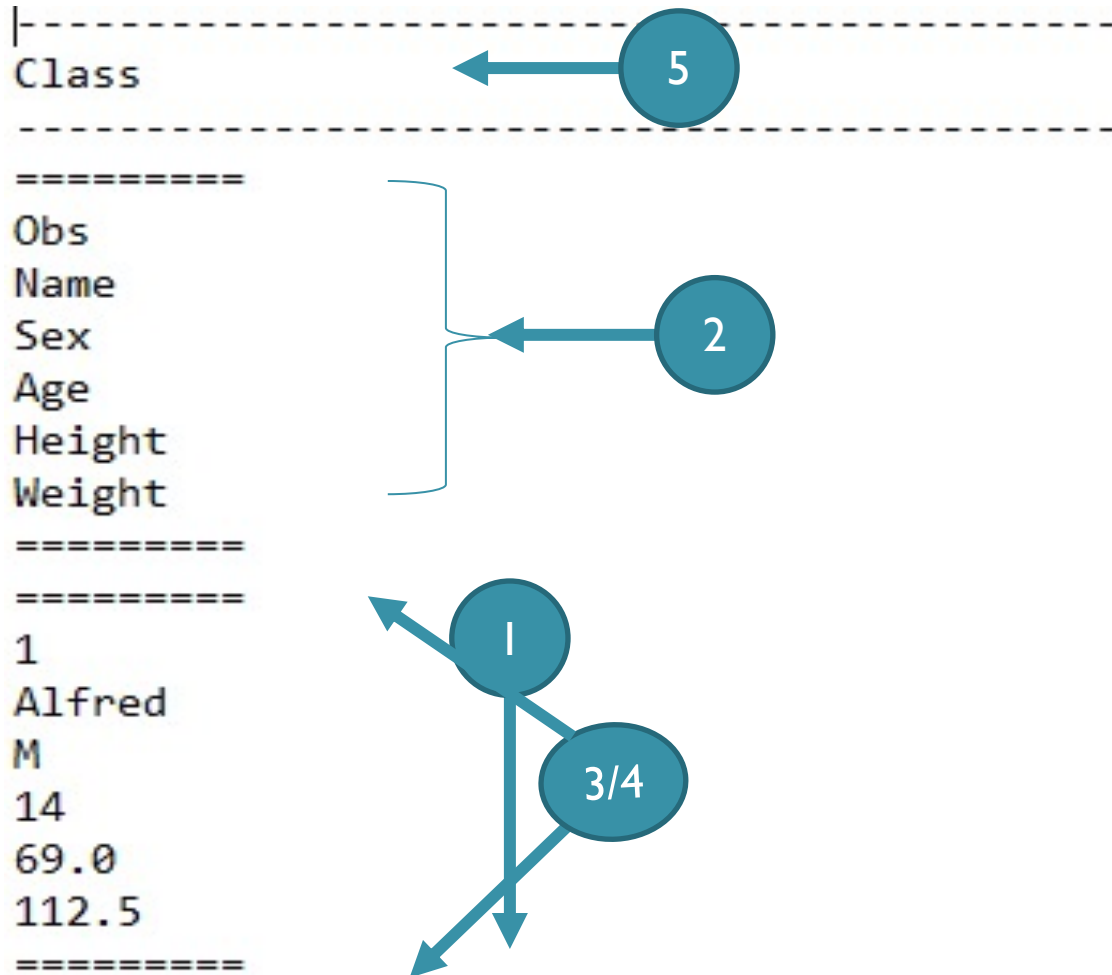
• STEP I-B > Adding more!!!

```
define tagset tagsets.learning;
  define event data;
    put value nl; * 1.n1;
  end;
  define event header; * 2;
    put value nl;
  end;
  define event row; * 3;
  start:
    put "=====" nl; * 4;
  finish:
    put "=====" nl;
  end;
  define event system_title; * 5;
  start:
    put "-----" nl;
    put Value nl;
  finish:
    put "-----" nl;
  end;
end;
```

1. nl > New Line
2. Header event
3. Row event
4. Put Statement / Start/finish
5. System_title event

- STEP I-B > New events & Text in line

Output looks like below:



1. nl > New Line
2. Header event
3. Row event
4. Put Statement / Start/finish
5. System_title event

- STEP I-C > Getting like table!!!

```
define event data ;  
  put VALUE "      " / if cmp( COLSTART , "1" );  
  put VALUE "      " / if cmp( COLSTART , "2" );  
  put VALUE "      " / if cmp( COLSTART , "3" );  
  put VALUE "      " / if cmp( COLSTART , "4" );  
  put VALUE "      " nl/if cmp( COLSTART , "5" );  
end;
```

```
put VALUE "      " / if cmp( COLSTART , "1" );
```

Data
Value

Horizontal Tab
after each value

1. IF Statement (Conditional)
2. cmp – compare function
3. COLSTART variable - Specifies the column number

- STEP I-C > New events & Text in line

Output looks like below:

```
-----  
Title 1: Class Data  
-----  
Name      Sex      Age      Height   Weight  
Alfred    M        14       69.0     112.5  
Alice     F        13       56.5     84.0  
Barbara   F        13       65.3     98.0  
Carol     F        14       62.8     102.5  
Henry     M        14       63.5     102.5  
James     M        12       57.3     83.0  
Jane      F        12       59.8     84.5  
Janet     F        15       62.5     112.5  
Jeffrey   M        13       62.5     84.0  
John      M        12       59.0     99.5  
Joyce     F        11       51.3     50.5  
Judy      F        14       64.3     90.0  
Louise    F        12       56.3     77.0  
Mary      F        15       66.5     112.0  
Philip    M        16       72.0     150.0  
Robert    M        12       64.8     128.0  
Ronald    M        15       67.0     133.0  
Thomas    M        11       57.5     85.0  
William   M        15       66.5     112.0
```

TIP 2. IDENTIFY EVENT OF INTEREST

```
ods listing;  
ods markup type=Text_Map file='Text_Map.xml';  
ods markup type=EVENT_Map file='Event_Map.xml';
```

```
proc print data=sashelp.class;  
run;
```

```
ods markup close;  
ods listing close;
```

```

- <table_head just="c" trigger_name="attr_out" event_name="table_head" index="IDX" output_label="Data Set SASHELP.CLASS" output_name="Print"
  colcount="1">
  - <row just="c" trigger_name="attr_out" event_name="row" index="IDX" output_label="Data Set SASHELP.CLASS" output_name="Print" colcount="1"
    section="head">
      <header just="r" class="header" trigger_name="attr_out" event_name="header" name="Obs" index="IDX" label="Obs" value="Obs"
        output_label="Data Set SASHELP.CLASS" output_name="Print" type="string" colcount="1" colwidth="8" col="1" precision="0" scale="0"
        data_row="0" row="1"> </header>
      <header just="l" class="header" trigger_name="attr_out" event_name="header" name="Name" index="IDX" label="Name" value="Name"
        output_label="Data Set SASHELP.CLASS" output_name="Print" type="string" colcount="1" colwidth="8" col="2" precision="0" scale="0"
        data_row="0" row="1"> </header>
      <header just="l" class="header" trigger_name="attr_out" event_name="header" name="Sex" index="IDX" label="Sex" value="Sex"
        output_label="Data Set SASHELP.CLASS" output_name="Print" type="string" colcount="1" colwidth="8" col="3" precision="0" scale="0"
        data_row="0" row="1"> </header>
      <header just="r" class="header" trigger_name="attr_out" event_name="header" name="Age" index="IDX" label="Age" value="Age"
        output_label="Data Set SASHELP.CLASS" output_name="Print" type="string" colcount="1" colwidth="8" col="4" precision="0" scale="0"
        data_row="0" row="1"> </header>
      <header just="r" class="header" trigger_name="attr_out" event_name="header" name="Height" index="IDX" label="Height" value="Height"
        output_label="Data Set SASHELP.CLASS" output_name="Print" type="string" colcount="1" colwidth="8" col="5" precision="0" scale="0"
        data_row="0" row="1"> </header>
      <header just="r" class="header" trigger_name="attr_out" event_name="header" name="Weight" index="IDX" label="Weight" value="Weight"
        output_label="Data Set SASHELP.CLASS" output_name="Print" type="string" colcount="1" colwidth="8" col="6" precision="0" scale="0"
        data_row="0" row="1"> </header>
    </row>
  </table_head>
- <table_body just="c" trigger_name="attr_out" event_name="table_body" index="IDX" output_label="Data Set SASHELP.CLASS" output_name="Print"
  colcount="1">
  - <row just="c" trigger_name="attr_out" event_name="row" index="IDX" output_label="Data Set SASHELP.CLASS" output_name="Print" colcount="1"
    section="body">
      <header just="r" class="rowheader" trigger_name="attr_out" event_name="header" name="Obs" index="IDX" label="Obs" value="1"
        output_label="Data Set SASHELP.CLASS" output_name="Print" type="double" colcount="1" colwidth="8" col="1" precision="0" scale="0"
        col_id="1" section="body" data_row="1" row="2" rawvalue="P/AAAAAAAAA="> </header>
      <data just="l" class="data" trigger_name="attr_out" event_name="data" name="Name" index="IDX" label="Name" value="Alfred"
        output_label="Data Set SASHELP.CLASS" output_name="Print" type="string" colcount="1" colwidth="8" col="2" precision="0" scale="0"
        col_id="2" section="body" data_row="1" row="2"> </data>
      <data just="l" class="data" trigger_name="attr_out" event_name="data" name="Sex" index="IDX" label="Sex" value="M" output_label="Data Set
        SASHELP.CLASS" output_name="Print" type="string" colcount="1" colwidth="1" col="3" precision="0" scale="0" col_id="3" section="body"

```

1. Event Name

table_head

row

header

table_body

Data

2. How they get triggered can be understood by looking this.

TIP 3. KNOW MORE OPTIONS

* Identity options available in ODS Tagsets ;

```
ods tagsets.rtf file='test1.html'  
options (doc="help");
```

```
proc print data=Sashelp.Class;  
run;
```

```
ods _all_ close;
```

=====

The RTF Tagset Help Text.

This Tagset/Destination helps with the creation of RTF files
for MS Word

=====

These are the options currently supported by this tagset.

Sample usage:

```
ODS TAGSETS.RTF OPTIONS(doc='Quick',  
                        CONTENTS='yes');
```

```
ODS TAGSETS.RTF OPTIONS(SPECIFIC_OPTION='value');
```

Debug_Level: No default value.

Current value: 0

Usage: OPTIONS(Debug_Level=1)

Description:

Determine what debugging information should be printed
to the log. The values expected are numeric and can be used to
take whatever action is needed. Used in tagsets being debugged,
but requires a local tagset to be modified.

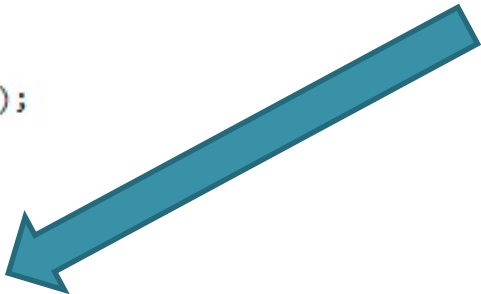
Doc: No default value.

Help: Displays introductory text and options.

Quick: Displays available options.

Settings: Displays Current settings.

Various options are
shown in log



1. TROWD: Inserts raw RTF specifications directly onto the table row header descriptions.

Similarly there are many other options TRHDR, TROWHDRCELL

RTF Specification

`trrh1000`



Number (twip)
gives height

```
ods tagsets.rtf file= 'learning.rtf'  
OPTIONS (TRHDR='\trrh1000');  
proc print data=Sashelp.Class noobs;  
run;  
ods _all_ close;
```

Name	Sex	Age	Height	Weight
Alfred	M	14	69.0	112.5
Alice	F	13	56.5	84.0



Summary

- ODS Tagsets - Powerful.
- Can be explore to full extent, even with some basic knowledge
- Can gain better control over output

Thanks....

Questions?