

Don't miss the hidden gems among the new features of SAS ODS for report creation

Patrick René Warnat, HMS Analytical Software GmbH, Heidelberg, Germany

ABSTRACT

Good reporting tools are very important for programmers in the pharmaceutical industry. The introduction of the output delivery system (ODS) has eased the creation of reports with SAS. In the newest versions of SAS new features were added to ODS that enhance the possibilities regarding reporting.

This paper will focus on the code needed to apply and combine selected features of ODS which are helping a lot in improving reporting possibilities: the usage of ODS Layout to arrange report elements, adding text and lists with PROC ODSTEXT and PROC ODSLST and styling your reports with CSS. In addition, it is shown how the new report writing interface can be used to implement highly customizable report elements.

See how to apply these helpful new ODS features that are off the beaten track of ODS usage.

INTRODUCTION

Shipping with the various sub-versions of SAS 9, new features of the SAS output delivery system have been introduced that are very helpful for report creation. The new features are not revolutionary, but very helpful improvements and enhancements of SAS ODS.

This paper focuses on features that improve report programming with SAS in regard to following aspects: controlling the layout of reports (ODS Layout and Report Writing Interface), applying styling attribute to report elements (application of cascading stylesheets) and generating report output using a data driven approach (report writing interface and ODS TEXT and LIST).

In the following paragraphs, the selected new features are shortly described, application examples are shown and recommendations for further reading are given.

ODS LAYOUT

A new feature experimentally introduced in SAS version 9.13 and productive with SAS version 9.2 is ODS LAYOUT. Using ODS LAYOUT, it is easily possible to arrange output objects on an output page of a page-oriented output destination. Two main concepts are supported: gridded layouts or absolute layouts. In a gridded layout, first a grid is defined with a certain number of columns (and optionally rows), then, the grid cells are filled with output objects in a certain order.

A simple example of SAS code to generate a report using a gridded layout:

```
ods pdf file="C:\TEMP\ODS_Layout_Gridded.pdf";

/** define gridded layout with two columns */
ods layout start columns=2;

/** headline text is put into first row of layout, spanning two layout columns
***/
ods region column_span=2;
ods pdf text="A headline";

/** first table (layout row two, layout column one) */
ods region;
proc print data=sashelp.class (where=(sex='F'));
run;

/** second table (layout row two, layout column two) */
ods region;
proc print data=sashelp.class (where=(sex='M'));
```

PhUSE 2014

```
run;

ods layout end;
ods pdf close;
```

With an absolute layout, layout areas are generated with a fixed size and positioned with absolute coordinates within the current page of the output destination.

An example of SAS code to generate a report using an absolute layout:

```
ods pdf file="C:\TEMP\ODS_Layout_absolute.pdf";

/** define an absolute layout */
ods layout start;

/** first region:
    8 centimeters from left and 14 centimeters from top
*/
ods region width=10cm height=9cm x=8cm y=14cm;
proc print data=sashelp.class (where=(sex='F'));
run;

/** second region */
ods region width=10cm height=9cm x=1cm y=5cm;
proc print data=sashelp.class (where=(sex='M'));
run;

ods layout end;
ods pdf close;
```

More information on ODS LAYOUT can be found in the following paper: [1].

PROC ODSTEXT AND PROC ODSLST

The two new ODS procedures named ODSTEXT and ODSLST simplify the output of text paragraphs and (textual) lists. The first procedure, ODSTEXT can be used to define templates for text items of the ODS output. Using the DATA= option of ODSTEXT, a SAS data set can be used to fill a text template with values. Here a simple example, where a paragraph template with is filled with values out of two columns of an input set. The template is filled twice, one time for every row of the input set:

```
ods pdf file="C:\TEMP\ODS_ODSTEXT.pdf";
data work.text;
  length paragraphNumber 8;
  length text $32;
  paragraphNumber = 1; text = 'test text 1'; output;
  paragraphNumber = 2; text = 'test text 2'; output;
run;

proc odstext data=work.text;
  p "This is paragraph" || put(paragraphNumber,2.) || ", with item: " || text;
run;
ods pdf close;
```

The ODSLST procedure is very similar to the ODSTEXT procedure but has its main emphasis on the output of bullet point lists where the list items can be filled by an input data set. In the following simple example a list item with the value of the variable name is created for every row of the input data set, in addition a sub-list item with data from other variables of the same row is written to the output:

```
ods pdf file="C:\TEMP\ODS_ODSLIST.pdf";
proc odslist data=sashelp.class;
  item;
```

PhUSE 2014

```
p "child name: " || name;
list;
  item;
    p "age of child: " || put(age, 2.);
    p "height of child: " || put(height, 3.);
  end;
end;
end;
run;
ods pdf close;
```

More information on ODS TEXT and ODS LIST can be found in the SAS ODS User's Guide.

THE NEW REPORT WRITING INTERFACE (RWI)

The new report writing interface helps to create highly customizable report elements. It comprises a new data step syntax enabling fine-grained control how data set elements are written to the active output destinations.

This new technique has two main application areas: controlling the layout of table output or creation of data-driven layouts that define the output of report pages and for example create one report report page for every record of an input file. Both of these applications are well described in the following paper: [2].

The new syntax of the report writing interface follows an object oriented style: During a data step, a output object is declared, and an "messages" are send to the output object. A simple example, putting a simple text to the output:

```
ods pdf file="C:\TEMP\ODS_ODSRWI1.pdf";
data _null_;
  declare odsout outobj();
  outobj.format_text(data: 'Hello!');
  outobj.delete();
run;
ods pdf close;
```

In another example, a simple table is written to the output:

```
ods pdf file="C:\TEMP\ODS_ODSRWI2.pdf";
data _null_;
  declare odsout o();
  o.table_start();
    o.row_start();
      o.format_cell(text: 'text 1');
      o.format_cell(text: 'text 2');
    o.row_end();
    o.row_start();
      o.format_cell(text: 'text 3');
      o.format_cell(text: 'text 4');
    o.row_end();
  o.table_end();
  o.delete();
run;
ods pdf close;
```

These examples are just a glimpse at the possibilities the RWI is offering, more examples are given here: [3, 4]

STYLING YOUR REPORTS WITH CSS

Styling of reports is normally done using SAS styles. Styles can be edited using the TEMPLATE procedure. Starting with SAS 9.2, cascading stylesheets (open specification published by the World Wide Web Consortium, W3C) can be used to describe how output should be formatted. This can be applied not only to the output destination HTML but also to other output destinations. In many organizations, using the CSS syntax has the advantage that (at least a big part) of style descriptions can be reused that might be already available from web projects.

One can use CSS similar to SAS styles:

```
ods pdf cssstyle='C:\TEMP\css1.css' file="C:\TEMP\ODS_CSS.pdf";
```

PhUSE 2014

```
proc print data=sashelp.class;
run;
ods pdf close;
```

The corresponding CSS file contains the following:

```
.body {
  background-color: #FFFFFF;
  color: #000000;
}
.data {
  background-color: #FFA500;
  color: #000000;
  font-family: "Lucida Grande", Arial, sans-serif; font-size: 11pt;
}
.header {
  background-color: #D0D0D0;
  color: #000000;
  font-family: "Lucida Grande", Arial, sans-serif;
  font-size: 12pt;font-weight: bold;text-align: left;
}
.rowheader {
  background-color: #D0D0D0;
  color: #000000;
  font-family: "Lucida Grande", Arial, sans-serif;
  font-size: 12pt;
  font-weight: bold;
  text-align: left;
}
.table {
  border: 1px solid #000000;
  border-collapse: collapse;
  border-spacing: 0px;
}
```

More on the possibilities using CSS can be found here: [5]

CONCLUSION

To conclude, the ODS features discussed here offer new possibilities for SAS report programmers. The shown approaches will hopefully help you implementing reports more effectively with SAS ODS. More examples of the application of the described ODS features can be found in various papers (see references as listed below).

REFERENCES

web links as accessed on 22th September 2014:

- [1] <http://www2.sas.com/proceedings/sugi31/159-31.pdf>
- [2] <http://support.sas.com/resources/papers/proceedings14/SAS330-2014.pdf>
- [3] <http://support.sas.com/resources/papers/proceedings14/1689-2014.pdf>
- [4] <http://support.sas.com/resources/papers/proceedings14/1362-2014.pdf>
- [5] <http://support.sas.com/resources/papers/proceedings11/297-2011.pdf>

CONTACT INFORMATION

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

Dr. Patrick R. Warnat
HMS Analytical Software GmbH
Rohrbacher Str. 26
69115 Heidelberg
Germany
<http://www.analytical-software.de/en/start/>

PhUSE 2014

Brand and product names are trademarks of their respective companies.