

PhUSE 2012

Paper TS07

A long way down from an output-window to the bottom of pdf - some shortcuts

Karl Ernst Siegler, CRS-Mannheim GmbH, Grünstadt, Germany

ABSTRACT

This presentation describes the introduction of tables, listings and figures generated as pdf-files with the SAS® output delivery system (ods) in a CRO (contract research organization) environment.

CRS Clinical Research Services GmbH is a full service CRO located in Germany. Approx. 50 trials per year are processed in the department "Clinical Data Management" - most of them Phase I studies. This requires strong standardization of planning and programming processes.

The previous CRS standard for tables, listings and figures was to use the content of a SAS output window with subsequent modifications by word-processing macros. With growing demands on documentation, traceability and with the electronic common technical document (eCTD) requirements, it was inevitable to abandon this process. In 2010 it was decided to deliver pdf-files instead of word documents. After defining requirements, finding feasible realizations and re-writing standard operation procedures (SOPs), the process was implemented several months later and experience was collected. The main specifications and some solutions will be presented throughout this paper.

INTRODUCTION

Once the decision was made to deliver all tables, listings and figures used in clinical trial reports as pdf-files, discussions started. The new process should fit in the existing environment. On one hand, the company exists for more than 30 years and therefore there were several restrictions based on existing standards and habits.

On the other hand, CRS had to convince customers (resp. sponsors) to accept other procedures, than those they were used to. Therefore the appearance of pdf-output had to be as close as possible to that of the former output.

Beyond internal and external definitions, need arose to incorporate the eCTD-specifications. They will be summarized in the next section. After having defined all needs, the programming experience started.

In a first step a satisfactory way to produce tables, listings and figures as pdf-files was launched. In addition, it turned out to be important to display raw SAS output, e.g. from the mixed procedure. This is sometimes needed for the technical output section in the clinical trial report. It was noticed, that tables, listings and "raw" SAS output were easily embedded into the pdf approach. It was more difficult for graphics. This was not as smooth as it was done for tables and listings. Most of this work was done with simple modifications outside the SAS procedures, i.e. writing output into pdf files and adapting SAS options.

Additional work was necessary to introduce ods statements within SAS procedures. For example the column widths within the report procedure had to be defined as "style(column)=[cellwidth=2.0cm]" instead of "width=8".

In a second step, the main issues were the definition of appropriate bookmarks in pdf-files and the set up of a comprehensive style template. It was challenging to get pdf-bookmarks in the way they should appear. This will be described in detail in this paper.

After nearly 12 months experience, some things are running smooth and some things are going to be improved. This will be summarized in the conclusions section.

ELECTRONIC COMMON TECHNICAL DOCUMENT

The definition of the SAS/ods environment for tables, listings and figures is in some extent triggered by the "Electronic Common Technical Document Specification - Appendix 7 Specification for submission Formats". The current version of this document was published 2008. Meanwhile it is state of the art in CRO activities to work according these requirements.

The following list summarizes the most important issues, which CRS incorporated into the new process. The numbers in brackets refer to the page numbers in the eCTD document.

- File type and size: pdf-files < 100 MB (p. 7-1), this is not really restrictive for SAS output. Experience shows that this is more than sufficient for output objects.
- Fonts: Courier fonts for data and optional Times New Roman for titles, footnotes etc. (p. 7-1).
- Font size: Courier 9pt and Times New Roman 11pt (company's interpretation of p. 7-2).

PhUSE 2012

- No colors in text and graphics (company's interpretation of p. 7-2).
- Page orientation landscape for tables, listings and figures (company's interpretation of p. 7-2).
- Paper size A4 with top 5cm, bottom 3cm, left 3cm, right 3cm margins (company's interpretation of p. 7-2).
- Unique headers and footers for identification of a document (p. 7-2). This is up to now left to the Medical Writing Department in order to ensure a consistent appearance throughout the whole clinical trial report.
- Bookmarks for tables, listings and figures. "The bookmark hierarchy should be identical to the tables of contents with no additional bookmark levels beyond those present in the table of contents" (p. 7-3).
- Although "only the internal page numbers of the document are expected (1-n)" (p. 7-4), there was a boisterous claim for "page x of y" for each table, listing or figure.

THE MOST IMPORTANT SPECIFICATIONS

A further part of CRS' motivation for inventing new processes is to streamline SAS programming and reporting. On one hand there is a reduction in interfaces with other software, e.g. word-processing macros, and SAS programming is more straightforward. On the other hand it is vital to have "ready to report" SAS output with minimal need for formatting and aligning.

Before making any decision, it is necessary to obtain a good knowledge of the existing environment in order to make any improvements. Beyond this, it is important to deliver with constant quality for those working with summary tables (e.g. physicians). As they do not have any immediate benefit from revisions in SAS programming, it might be hard to get any indulgence from this side, if e.g. the quality decreases.

From the present experience, the following issues are most critical in leaving the output window and turning towards ods.

- Will SAS/ods be utilized for pdf or rtf output? There are good reasons, to move to rtf output first and then convert it to pdf. One important reason is the familiarity of most users with comments and track change functionality.
- Which procedures (proc report, proc print, proc tabulate, graphic procedures) will mainly be used to generate output?
- Will one file be generated for each output object (i.e. table, listing or figure) or will the output objects written into files as complete subsections (e.g. pharmacokinetics or laboratory assessments)?
- Which degree of standardization is to be applied? This affects the scope of the style templates. Are decisions (e.g. column widths) left to the programmer or do exhaustive specifications exist?
- Do not under-estimate familiarity!
- Which structural elements are requested (page x of y, bookmarks, table of contents, etc.) after the decision for pdf (or rtf) was made?

The overview shows, what is easily done (+) with regular ods/pdf statements and what appears to be more difficult (-).

	page x of y	customized bookmarks	table of contents
one pdf-file for each object	+	+	-
several objects in one pdf-file	-	+	+

At CRS the general organization stipulates that a separate SAS program is written for each table, listing or figure. This decision is not critical for the invention of the ods/pdf environment. It is straightforward to adapt the planned programming work to most conditions.

The appearance of output objects has to be identical with previous standards. This is quite easy to implement, as the contents of the tables and the syntax within SAS procedures is not affected by the shift away from the output window towards pdf. In phase I there are repeated claims for sub- and superscripts and for special characters in the notations for pharmacokinetic parameters. It is nearly no extra effort to add this to the pdf files.

After an extensive search on SAS web-pages, solutions for tables, listings and figures generated with proc print and proc report are achieved. The use of proc tabulate is followed up with less priority. During the development process it was found out that put statements in data steps for generating "empty" output objects, e.g. "Table not applicable", have to be covered as well.

DESIGN YOUR OWN STYLE WITH THE TEMPLATE PROCEDURE

After gathering all specifications and making all necessary decisions, the practical work continues with the company's template procedure. A carefully designed style, created with the template procedure, is a central component for a unified track to generate ready to use SAS/ods output.

Without any style template, the pdf output is formatted according to pre-specified definitions. Those are usually not applicable for clinical trial reports. Therefore most demands on fonts, appearance of tables, alignments of headers and columns, page margins etc. are defined within the template procedure. The template procedure is well

PhUSE 2012

documented and once the user gets familiar with it, the requested appearance of a specific output is easily done. There is no need to define a complete style from scratch. The CRS-style was not created as a "stand-alone" style with all necessary definitions. Rather the predefined "journal" style is used as a parent style and modified according to the company's requirements. Currently the CRS template procedure comprises approx. 200 lines of SAS code. Not all single items will be presented in detail. A short example code of a template procedure used for the definition of a company specific style for tables, listings and figures is given below. The definitions are not complete, nevertheless the code works as a starting point for own experiments.

```
proc template;
  define style COMPANY_STYLE;
    parent = styles.journal;
    replace fonts /
    /*Titles from title statement*/
    'TitleFont' = ("Courier",9pt)
    /* Titles from proc statement*/
    'TitleFont2' = ("Courier",9pt)
    /*table row headings*/
    'HeadingFont' = ("Courier",9pt);
    /*set page margins*/
    replace Body from Document /
    topmargin = 5cm
    leftmargin = 3cm
    rightmargin = 3cm
    bottommargin = 3cm;
    /*appearance of data*/
    style data /
    font = ("Courier",9pt)
    background=white
    foreground=black
    just=left;
  end;
run;
```

It turned out that "Courier New" font exhibits some unwanted properties, e.g. not searchable text in pdf-files. Therefore "Courier" should be used. For further information on eCTD compliant styles refer to Hull (2010).

CUSTOMIZED BOOKMARKS FOR TABLES, LISTINGS AND FIGURES

After having defined a CRS style, the first tables and listings are reviewed. The appearance is very satisfying, and now the work on structural elements starts. Once one is used to handle pdf files, the bookmarks are more familiar than tables of contents. It is not straightforward to define bookmarks in the way that is defined in the eCTD. Detailed examples will be given for some SAS procedures often used for drawing tables, listing and figures. In these examples it is assumed, that listings are generated with the print procedure, tables with the report procedure and figures with the gplot procedure. Other SAS procedures might require further customizations in order to adapt the pdf-bookmarks. All examples are used with SAS for windows, version 9.2 on XP_PRO platform.

INTRODUCTION TO EXAMPLES

The examples given in this paper are based on the SASHELP.CLASS data, which is available with the current SAS software. All examples run with some options, which are given for consistency reasons. One might adapt the options as needed. The same holds for the "escapechar", which is a control character for SAS/ods. The sort procedure is used once, because some by-statements will appear in the prepared examples.

```
options nodate nonumber nocenter orientation = landscape papersize = A4;
ods escapechar = "^";
  proc sort data=SASHELP.CLASS out = EXAMPLE;
    by SEX;
  run;
```

The papersize option should be used, because the linesize and pagesize options do not have the same effect on pdf files as they have on the output window.

PhUSE 2012

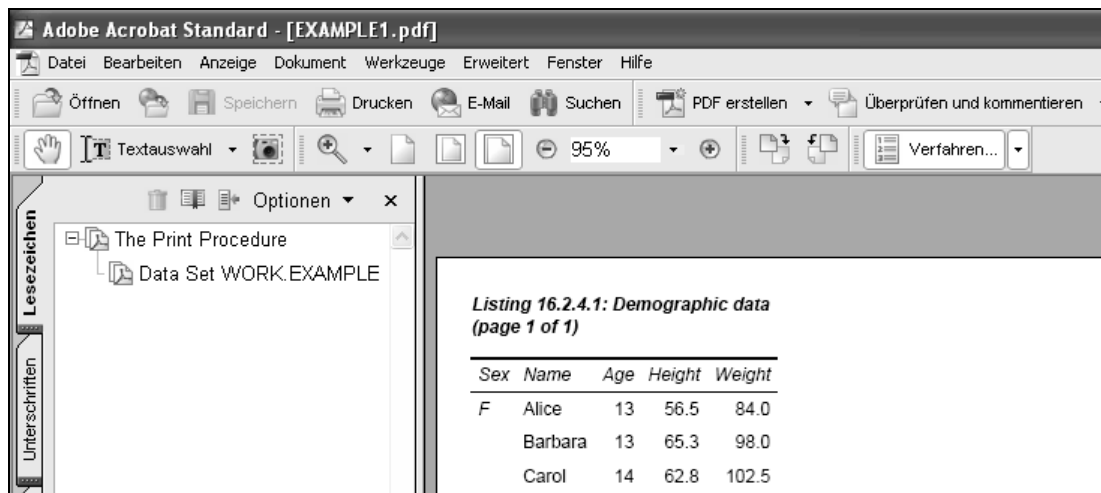
THE REGULAR WAY TO GENERATE PDF-BOOKMARKS

In these examples, the focus is on bookmarks, not on the appearance or contents of the displayed tables. Once it is decided, that bookmarks are to be used in pdf-files, the first steps are very simple and yield pdf-bookmarks at once. This is done in the example below, where merely the ods pdf statement is used.

```
ods pdf file = "P:\TEMP\EXAMPLE1.pdf" style = journal;
title1 justify=left "Listing 16.2.4.1: Demographic data";
title2 justify=left "(page ^{thispage} of ^{lastpage})" justify =right "&SYSDATE9";
proc print data=EXAMPLE noobs;
  var NAME AGE HEIGHT WEIGHT;
  id SEX;
  by SEX;
run;
ods pdf close;
```

Unfortunately the bookmarks are not in the shape as intended in eCTD. The content of the bookmark does not show a meaningful title with respect to the contents of the table. Moreover the default bookmark (in this case) has got two levels. Nevertheless it is seen, that no extra statements are necessary to request bookmarks.

The "justify" statement does not show any effect in the output window. It aligns titles (and footnotes) in pdf-files. Different alignments (left and right) might be combined in one title.



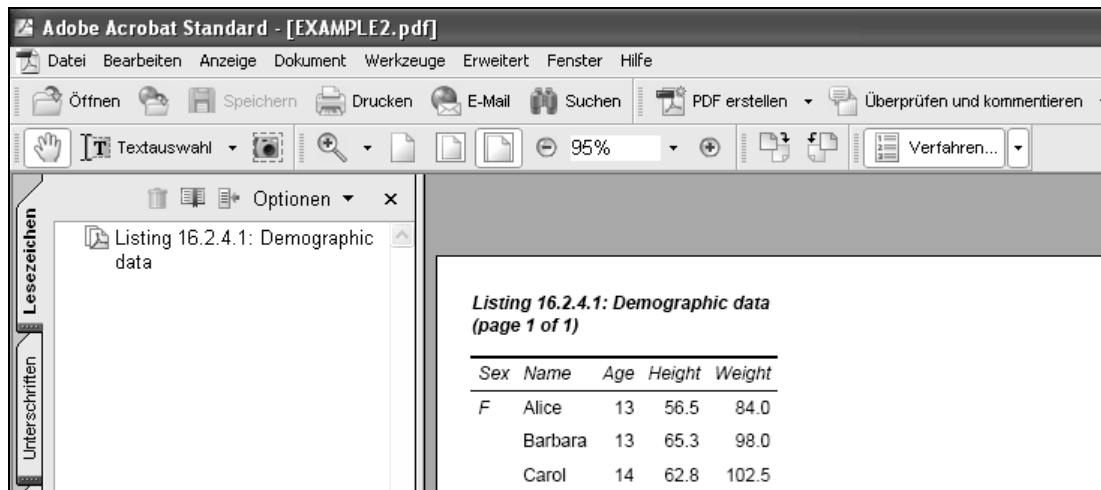
CUSTOMIZED BOOKMARKS FOR LISTINGS GENERATED WITH THE PRINT PROCEDURE

In the next example, the procedure from the previous example is repeated, but several statements are added. The macro variable "MY_TITLE" is defined and used twice: First as a title statement which appears above the listing and second as the procedure label, defined in the ods proclabel statement. With this, the bookmark receives an appropriate content. Another modification in comparison to the first example is the introduction of a contents="" request in the procedure statement. This omits the second entry in the bookmark.

```
ods pdf file = "P:\TEMP\EXAMPLE2.pdf" style = journal;
%let MY_TITLE = %str("Listing 16.2.4.1: Demographic data");
ods proclabel &MY_TITLE;
title1 justify=left &MY_TITLE;
title2 justify=left "(page ^{thispage} of ^{lastpage})" justify =right "&SYSDATE9";
proc print data=EXAMPLE noobs contents = "";
  var NAME AGE HEIGHT WEIGHT;
  id SEX;
  by SEX;
run;
ods pdf close;
```

Now the bookmark is in the intended shape, which is shown below. The title of the table appears in the bookmark and it contains only one entry.

PhUSE 2012



CUSTOMIZED BOOKMARKS FOR TABLES GENERATED WITH THE REPORT PROCEDURE

In the third example, the data from the previous example are used and displayed; now using the report procedure. This procedure allows more features and it is preferred in displaying more complex tables. The syntax is as before with defining a procedure label and using the contents option. Moreover, a break has to be introduced to erase subordinate entries in the pdf-bookmark.

```
ods pdf file = "P:\TEMP\EXAMPLE3.pdf" style = journal;
%let MY_TITLE = %str("Table 14.1.1: Descriptive statistics of demographic data");
ods proclabel &MY_TITLE;
title1 justify=left &MY_TITLE;
title2 justify=left "(page ^{thispage} of ^{lastpage})" justify =right "&SYSDATE9";
proc report data=EXAMPLE nowd spanrows contents = "";
  column SEX NAME AGE HEIGHT WEIGHT;
  define SEX / group;
  break before SEX / page contents = "";
run;
ods pdf close;
```

The output is not displayed; it is very similar to the one above. A more elaborated example is given next.

CUSTOMIZED BOOKMARKS FOR TABLES GENERATED WITH MORE THAN ONE REPORT PROCEDURE

Under some circumstances, it might happen that two report procedures are necessary in order to produce one table in a clinical trial report. This is illustrated in the next example. The setting is as before. Two tables are made - one for male and one for female participants. The first report procedure is like the one above with an additional where-statement (just for the illustration). This takes care for a bookmark with merely one entry. The second report procedure is preceded by the ods pdf nobookmarkgen statement. With this, the following bookmarks are suppressed and only the one from the first procedure persists. The page x of y feature works within these tables.

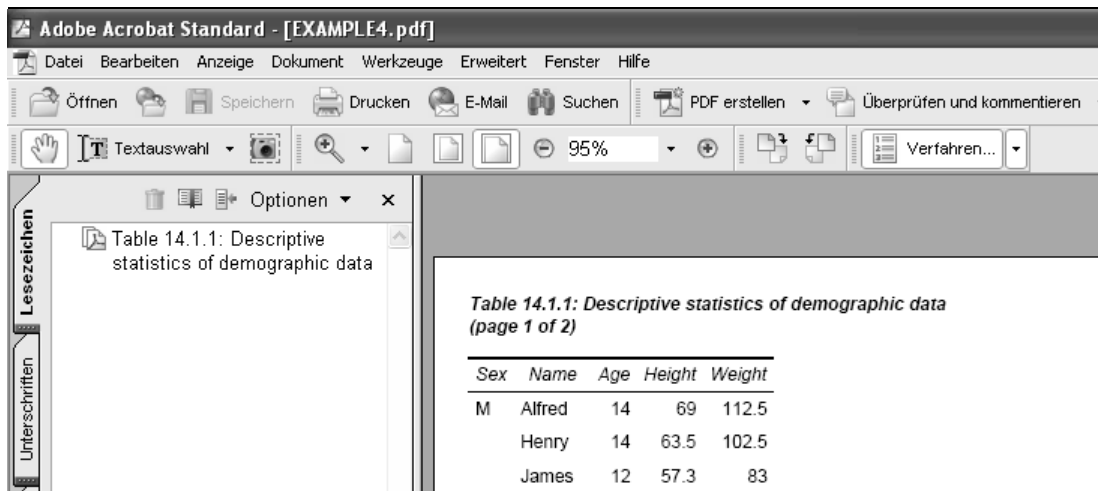
```
ods pdf file = "P:\TEMP\EXAMPLE4.pdf" style = journal;
%let MY_TITLE = %str("Table 14.1.1: Descriptive statistics of demographic data");
ods proclabel &MY_TITLE;
title1 justify=left &MY_TITLE;
title2 justify=left "(page ^{thispage} of ^{lastpage})" justify =right "&SYSDATE9";
proc report data=EXAMPLE nowd spanrows contents = "";
  column SEX NAME AGE HEIGHT WEIGHT;
  define SEX / group;
  break before SEX / page contents = "";
  where SEX = "M";
run;

ods pdf nobookmarkgen;
proc report data=EXAMPLE nowd spanrows contents = "";
  column SEX NAME AGE HEIGHT WEIGHT;
  define SEX / group;
```

PhUSE 2012

```
break before SEX / page contents = "";
where SEX = "F";
run;
ods pdf close;
```

The result is shown below. The table consists of two pages, one for male and one for female study participants. The contents="" statements suppress the second level bookmarks. Therefore the break is necessary. If no break is intended to be used, a possible workaround is to define a variable with a constant value which is not displayed and to break before this variable.



CUSTOMIZED BOOKMARKS FOR FIGURES GENERATED WITH THE GPLOT PROCEDURE

Graphics should have the same appearance as tables and listings. Therefore the bookmarks have to be controlled as well. Especially when using by-statements, one entry for each realization of the by-variable appears in the bookmarks. In order to get it in the desired shape, it is necessary to file the graphic output in an output file. Afterwards the filed graphic is dumped with the greplay procedure. This is illustrated in the following lines. The use of proclabel and title statements is as described above. The page x of y feature does not work within these graphics.

```
ods pdf file = "P:\TEMP\EXAMPLE5.pdf" style = journal;
%let MY_TITLE = %str("Figure 14.1.1: Scatter plot of height by weight");
goptions nodisplay;
title1 justify=left h=9pt &MY_TITLE justify =right h=9pt "&SYSDATE9";
proc gplot data=EXAMPLE gout=EXAMPLE5;
  plot WEIGHT * HEIGHT / des = "";
  by SEX;
run;
quit;
```

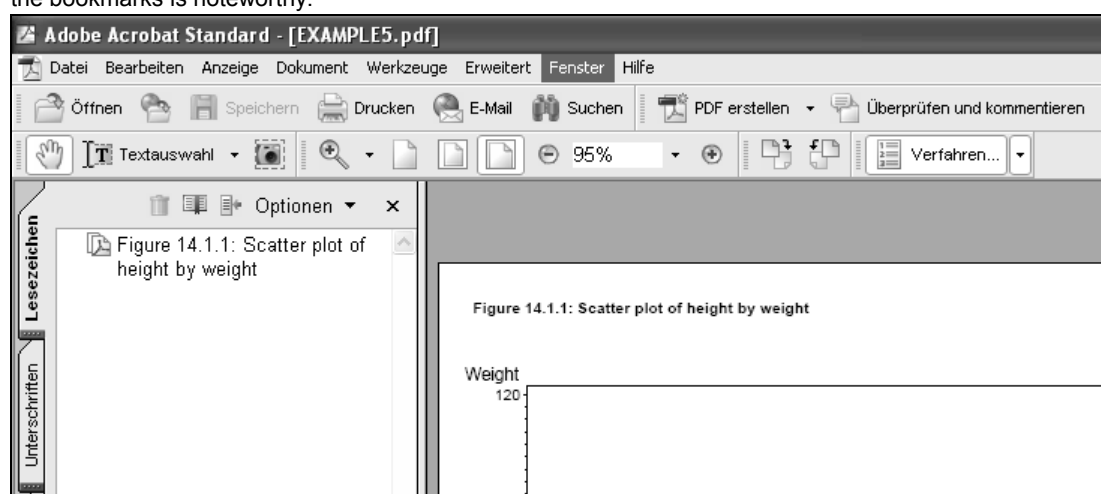
Note that no output is created yet, because the nodisplay option is active.

Starting from here, the graphics output is displayed. The greplay procedure creates the output and uses the procedure label MY_TITLE. With this, one gets a graphic output with a one-level bookmark. It is essential to define all graphic elements before the first gplot procedure (e.g. axes, labels, fonts etc.). Otherwise, the plot might not have the desired appearance.

```
goptions display;
ods proclabel &MY_TITLE;
proc greplay igout = EXAMPLE5 nofs;
  replay _all_;
  delete _all_;
run;
quit;
ods pdf close;
```

PhUSE 2012

Although appearing quite complicated at first glance, this process works very well. Especially with the use of by-statements, where each realization of the by-variable generates an extra level in the bookmark, the gain in simplifying the bookmarks is noteworthy.



WHAT IS MISSING, WHAT IS DIFFICULT?

PUTTING IT ALL TOGETHER

After generating one pdf-file for each table, listing or figure, the files are to be assembled with e.g. Adobe Acrobat® Standard software. This works quite well as long as the filenames have a relation to the table-numbers, since the filenames are displayed as an additional bookmark after bringing together the pdf-files.

This depends strongly on the software applied to merge pdf-files and is not within the scope of this presentation.

STRANGE OR MISSING

Some features are unfamiliar or missing when moving from the output window to pdf files and some things do not work as simple as expected. Three examples are given as an impression of unforeseen hassle. This is not intended to prevent anyone from using ods/pdf features. It merely underlines the importance of careful plans and tests to avoid half-done deliverables.

Empty lines in tables:

While implementing the new process, it was noticed that in the report procedure the "break after VARNAM / skip" does not work. It was realized that this is a very important structural element and omitting it is nearly impossible. A possible, but not comfortable workaround is:

```
compute after VARNAM;
line @1 " " ;
endcomp;
```

Page x of y in graphics:

Unfortunately page x of y does not work with the graphic procedures as described above. A possible workaround is to employ some tricks involving #byval1 #byvaln if by statements are used. This is not satisfactory.

Unexpected results:

In some rare cases, the appearance of tables or listings is corrupted on single pages. This is reproducible with no simple workaround. Therefore it seems to be very important to do a formal check on every page after pdf-output processes are introduced. The following excerpt exhibits a short line through the variables "Subject No" and "PID" which was not expected and it merely appears on this single page.

Listing 16.2.7.1: Adverse Events				
(page 7 of 8)				
Subject No.	PID	Adverse event (Modified Term)	Adverse Event (Reported term)	Onset Date/Time
12-123	112233	DROWSINESS	DROWSINESS	2012-06-02
[AB]		PRURITUS	PRURITUS	2012-06-03
12-456	445566	TIREDDNESS	TIREDDNESS	2012-06-04
[AB]				
12-789	778899	DROWSINESS	DROWSINESS	2012-06-05
[AB]				

PhUSE 2012

CONCLUSION

The present experience is encouraging: The new processes are saving time. The acceptance of the customers resp. sponsors (approx. 15 at this time) is very high. Only one company refused the new standard, but meanwhile they revised their SOPs and they are in line now. A few companies asked for the same process with rtf files, which is a future project. Nevertheless in-text tables are already made with ods/rtf.

The feedback from sponsors was positive or (more often) neutral. They accepted the output, which indeed looks like the previous output and they got used to handle pdf-files. Acceptance of bookmarks in pdf-files was very low, a table of contents was frequently requested.

The clinical staff did not appreciate the new process. On one hand they were missing the table of contents and the possibility to write comments within the familiar word-processing software. On the other hand, data management introduced other alterations at the same time. They were triggered by changes in responsibilities and implementation of CDSIC standards. Some essential parts of standard output were modified as CDISC was introduced. Clinical staff did not distinguish between the different changes. If anything occurred to be spurious, pdf-output was blamed for it. Right from the beginning, SAS programmers could work more efficient within the new process. However, it was quite hard to get them used to ods-statements within SAS procedures. Leaving the need to apply other software for formatting SAS output was a large gain and this improves traceability as well. With growing experience of the SAS programmers, the appearance of tables and listings improves and it turns out that - compared to an output window - much more is possible.

Beyond translating the pdf-process to an rtf-process, there are some more plans:

1. A deeper understanding of customizing bookmarks and tables of contents in order to come closer to the needs of our readers (explore proc document possibilities).
2. To fiddle about the page headers with the objective to display company logos and trial numbers with no external (i.e. outside SAS) efforts - perhaps with "ods layout" and "ods region" functionalities.

REFERENCES

ICH M2 EWG (2008) Electronic Common Technical Document Specification, ICH eCTD Specification V 3.2.2 [http://estri.ich.org/eCTD/eCTD_Specification_v3_2_2.pdf]

Haworth, L.E., Zender, C.L., Burlew, M.M. (2009) Output delivery system: The basics and beyond. SAS Institute Inc., Cary, NC.

Hull, R. (2010) I've Got the Styles You Need, SAS Global Forum 2010, Synteract, Inc., Carlsbad, CA

Lawhorn, B. (2011) Let's give 'em something to TOC about: Transforming the table of contents of your PDF file, SAS Global Forum 2011, SAS Institute Inc., Cary, NC.

ACKNOWLEDGMENTS

Thanks to Karin Petra Schell who furnished our new home while I was writing this paper.

CONTACT INFORMATION

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

Author Name	Karl Ernst Siegler
Company	CRS Clinical Research Services Mannheim GmbH
Address	Richard-Wagner-Strasse 20
City / Postcode	67269 Grünstadt
Work Phone:	+49 6359 899 379
Fax:	+49 6359 899 352
Email:	karl-ernst.siegler@crs-group.de
Web:	www.crs-group.de

Brand and product names are trademarks of their respective companies.