PharmaSUG 2013 - Paper PO16

TLFs: Replaying Rather than Appending

William Coar, Axio Research, Seattle, WA

ABSTRACT

In day-to-day operations of a Biostatistics and Statistical Programming department, we are often tasked with generating reports in the form of tables, listings, and figures (TLFs). Some requests come in the form of a small number of TLFs whereas others are more substantial in magnitude. Regardless, creating a single document for distribution and review may be required after all TLFs have been completed.

A common setting in the pharmaceutical industry is to develop SAS code in which individual programs generate one or more TLFs in some standard formatted output such as RTF or PDF with a common look and feel. Furthermore, programs are developed over time, with the production run being in batch mode. The result is a set of TLFs completed at different times. Creation of a final (single) document with a properly sectioned and hyperlinked Table of Contents as well as dynamic page numbering may be desired. The ability to deliver a single document also greatly simplifies document management and electronic review for many end users.

Many options have been proposed that post-process individual RTF or PDF files. An alternative approach which uses ODS Document is introduced. Unlike many other techniques, ODS Document uses intermediate files called item stores that are independent of ODS destination. This technique has proven successful across multiple projects in our specific setting, and continues to show promise in other applications as well.

INTRODUCTION

Although the setting for the application presented may be quite specific, it is not uncommon in the pharmaceutical industry. Individual programs generate one or more TLFs via text (LST) files or through some Output Delivery System (ODS) destination such as RTF or PDF. These programs are developed over time, and submitted in batch mode for production version of the reports. The final procedure for all summary tables and listings is Proc Report. Figures result from Proc SGRender using custom graphic templates. The other SG Procedures can be used successfully as well, though the flexibility for customization associated with Graph Template Language may be preferable. The application presented has been implemented in a Windows environment with SAS 9.2 (and more recently 9.3).

EXISTING METHODS

A number of methods to create a single document from a set of individual TLF files exist ([1], [2], [3], [4]). A few have been discussed within the past year at PharmaSUG 2012 ([7],[8]) and the Western Users of SAS Software ([6]). The more common setting is for reports to exist in RTF format, though at least one method exists for individual PDF files.

When individual reports are in RTF format, numerous applications to create an appended report exist. Some consist of:

- Manipulation of RTF field codes and setting the text of RTF files into a single file using SAS
- Using Dynamic Data Exchange (DDE) to open Word and insert bookmarks and files into a single document from within SAS
- Using Visual Basic (VB) macros to insert individual files into a single document within Word
- Combinations or variations of DDE and VB

Far fewer applications have been presented for appending individual PDF files:

- Manually inserting individual files and creating bookmarks within a professional version of Adobe Writer

Other methods exist as well, though not discussed here. Almost all methods presented require post processing RTF or PDF files created at a previous run in SAS. Not all methods provide a hyperlinked Table of Contents (ToC) or bookmarking to allow for easy navigation. Lastly, some require additional software licenses such as Microsoft Word or Adobe Writer thereby requiring knowledge base beyond SAS programming.

Should the existing outputs exist in the form of traditional LST (text) files, some of these applications can still be used. However, discussion of LST files is omitted since the application presented here requires ODS.

REPLAY: A TWO STEP APPROACH

The replay approach is an alternative to post-processing LST, RTF, or PDF files. The first of this two step process uses ODS Document to create item stores for each TLF. These item stores house the instructions and data used by the procedures called within a block of ODS Document statements. The second step uses Proc Document to combine, restructure, and replay these item stores into a single document within an ODS destination. Since all the item stores are replayed within a single ODS destination, the result is a single file that contains all the reports that encompass the TLFs.

ODS DOCUMENT

As previously mentioned Proc Report is generally used in the creation of each table or listing. As with the infamous ODS sandwich for creating an RTF or PDF, we use an ODS Document sandwich to create the item store. Prior to the Proc Report, an ODS Document statement opens the item store. After the Proc Report, the item store is closed with a concluding ODS Document *close* statement. The ODS Document sandwich is necessary for figures as well, which are produced with Proc SGRender. Additional details on a SAS item store can be found in [9].

One of the power attributes of the SAS item stores created in this application are that they are independent of ODS destination and style template. The original SAS programs can create RTF files, yet the item stores can be replayed directly to PDF using a different style template at a later time without rerunning the original program. In this application, permanent item stores are created for each Proc Report and Proc SGRender. The location is defined through a LIBNAME statement. Furthermore, item store names are easily automated as are the file names of the individual reports created with the batch execution.

The details of this first step are best explained through examples:

```
ODS Document name=istore.&PGRM (write);

proc report data=tns_rpt center headline headskip nowindows missing
      contents="Table 1: Subject Disposition";
      column flag sv1 sv2 row1 _1 _2 _3;
      define flag  /order order=data noprint ;

      Other define statements

      break before flag / page contents='';
      run;

ODS Document close;
```

As seen above, the ODS document statements are straightforward. The initial statement (re) creates the item store using the *write* option. The second ODS statement closes the item store. Lawhorn provides much detail regarding the additional updates to Proc Report in [5]. There are three noteworthy updates to the standard Proc Report:

1. The *contents*= option is used to define the text used for the hyperlinks and bookmark labels. Other SAS procedures may use the *description*= option.
2. The first variable in the column statement is a dummy flag variable that has a single value for every record. It is defined as an order variable and accompanied by the noprint option.
3. The break before statement is necessary for the proper hyperlinks in the ToC and Bookmark structure.

These additional programming statements, although seemingly tedious, are easily added to existing SAS programs with minimal effort. Once in place, it is easily carried forward to new programs with the use of Cut-&-Paste.

Graph Template Language(GTL) and Proc SGRender can be used to create high quality custom graphics starting with SAS 9.2. In addition to the ODS Document sandwich, the graph template must also be stored. When the item store from Proc SGRender is replayed at a later time, the graph template must exist somewhere in the ODS search path.

```
ODS Document name=istore.&PGRM (write);

proc sgrender data=fp template=&PGRM
              description="Figure 1: Cumulative Enrollment";
run;

ODS Document close;
```

As seen in this example, creating an item store for a figure that uses SGRender is straightforward.

PROC DOCUMENT

The second step of the Replay approach requires the use of Proc Document. In this application, there are two procedure calls. The first procedure call reads in all the item stores from the creation of the individual TLFs. It also restructures the components to allow for a more customized Table of Contents and Bookmark structure. The second procedure call replays all or part of the restructured item store. It is this procedure call within an ODS destination sandwich that allows us to replay all the individual item stores into a single document. The result is a single document that has all the TLFs with a hyperlinked table of contents and bookmarks of the desired structure.

The details of reading in and restructuring item stores is less straightforward. See Lawhorn [5] for more detailed explanation. At first impression the code may appear complicated. However, once there is a basic understanding of the structures of these item stores, and an understanding of the necessary structure for the desired output, the code is manageable. Below is example code used to create the combined item store in Display 3: Combined item store with restructuring:

```
proc document name=work.app(write);
    copy \istore.table1\Report#1 to ^;
    copy \istore.table2\Report#1 to ^;
    move Report#2\Report#1 to Report#1;
    delete Report#2;
    setlabel Report#1 “Tables”;

    copy \istore.figure1\Sgrender#1 to ^;
    copy \istore.figure2\Sgrender#1 to ^;
    move Sgrender#2\SGRender#1 to SGRender#1;
    delete SGRender#2;
    setlabel SGRender#1 “Figures”;
run;
```

In the above example there are two sections in the ToC: Tables and Figures. As is, there are two tables and two figures. The copy statement is used to read in each item store. The move and delete statements are used to restructure. Once the combined item store is created, the Tables and/or Figures can be replayed to an ODS destination.

```
proc document name=work.app;
    replay Report#1;
    replay Sgrender#1;
run;
quit;
```

With the appropriate options in the ODS destination statements, a hyperlinked ToC (in RTF) and Bookmarks (in PDF) are automatically generated. When the RTF file is initially opened in MS Word, the ToC will exist but will appear empty. Within MS Word, the user will need to refresh/update the Table of Contents to populate all the underlying RTF fields.

RESULTS

Visual aids to the structure of the individual and combined item stores may be useful to better understand the restructuring in the Proc Document procedure call.

EXAMPLE: SINGLE ITEM STORE

A simple proc document statement can be used to see the structure of (any) item store.

```
proc document name=istore.table1;
       list / levels=all;
run;
quit;
```

If the item store *table1* was created from Proc Report, the results will be of the form:

Listing of: \Istore.table1

Order by: Insertion

Number of levels: All

Obs	Path	Type
1	**\Report#1**	Dir
2	**\Report#1\Report#1**	Dir
3	**\Report#1\Report#1\Report#1**	Table

Display 1: Output from LIST statement in Proc Document.

For each Proc Report, there are two directory levels and one table. With this structure in mind, examples of combined and restructured item stores are presented below.

EXAMPLE: COMBINED ITEM STORE

SAS code previously presented combines two tables (from Proc Report) and two figures (from Proc SGRender). If we were to only copy the individual item stores without restructuring with move/delete statements, the combined item store would have the following structure:

Listing of: \work.app

Order by: Insertion

Number of levels: All

Obs	Path	Type
1	**\Report#1**	Dir
2	**\Report#1\Report#1**	Dir
3	**\Report#1\Report#1\Report#1**	Table
4	**\Report#2**	Dir
5	**\Report#2\Report#1**	Dir
6	**\Report#2\Report#1\Report#1**	Table
7	**\Sgrender#1**	Dir
8	**\Sgrender#1\SGRender#1**	Graph
8	**\Sgrender#2**	Dir
9	**\Sgrender#2\SGRender#1**	Graph

Display 2: Combined item store WITHOUT restructuring

As each item store is copied in, the report (or graph) sequence number is automatically incremented. As is, the elements of the combined item store can be replayed into a single document. However, the Table of Contents and Bookmarking would not have the desired structure. With the MOVE and DELETE statement, the restructured item store has the following levels:

Listing of: \work.app
Order by: Insertion
Number of levels: All

Obs	Path	Type
1	**\Report#1**	Dir
2	**\Report#1\Report#1**	Dir
3	**\Report#1\Report#1\Report#1**	Table
4	**\Report#1\Report#2**	Dir
5	**\Report#1\Report#2\Report#1**	Table
6	**\Sgrender#1**	Dir
7	**\Sgrender#1\SGRender#1**	Graph
8	**\Sgrender#1\SGRender#2**	Graph

Display 3: Combined item store with restructuring

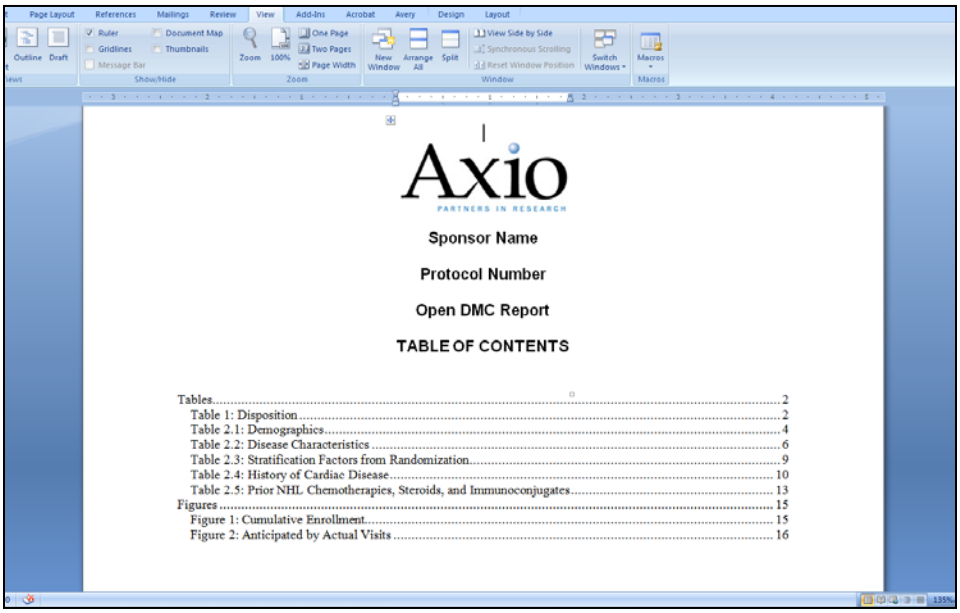
As seen in the restructured item store, the sectioning is defined by directory levels Report#1 and SGRender#1. The SETLABEL statement is used to label these sections, as seen in the final product. Knowledge of creating even more directories and moving around within an item store allows for additional customized sectioning, but the details are beyond the scope of this poster.

FINAL PRODUCT

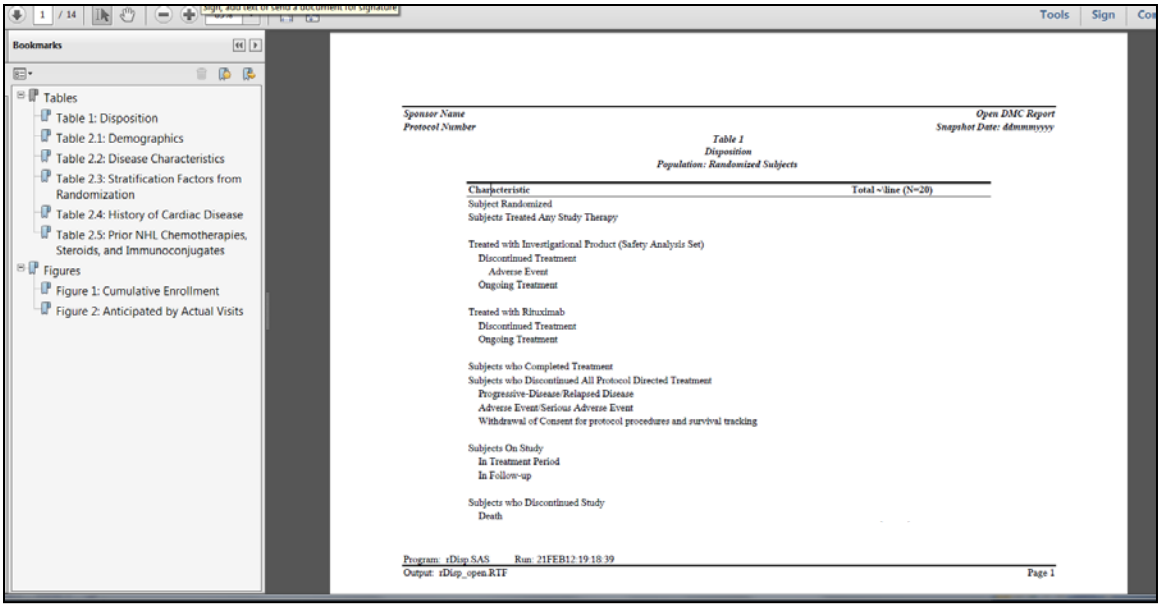
The final replayed report that has all the TLFs results from the replay statement(s) in the final Proc Document wrapped in ODS RTF or ODS PDF statements. In order to obtain a table of contents in RTF, the TOC_DATA and CONTENTS=ON options should be specified. The NOGTITLE and NOGFOOT are also used when incorporating figures into the final document which place titles/footnotes in the header/footer rather than in the image. No specific options are needed for PDF with SAS 9.2 as NOGTITLE and NOGFOOT are not available for the PDF destination until SAS 9.3.

A more custom Table of Contents can be created with additional features of the style template (such as inclusion of company logo), project name, etc.

Examples of the first page of the final replayed report are provided below. Only the first page is provided for obvious reasons as they demonstrate a hyperlinked table of contents in RTF and bookmarks in PDF.



Display 4: Hyperlinked ToC from RTF destination.



Display 5: Hyperlinked Bookmarks from PDF Destination.

As seen in the PDF example, there is additional work to be done. In particular, it is common to use in-line RTF codes when the initial output is assumed to be RTF. However, this removes some of the item store's independence of ODS destination. A solution is to replace in-line RTF codes with ODS escape character codes. In the above example, ~\line can be replaced by ~n to achieve the desired result. Additionally, a border is used to separate the footnotes. This border is part of the style template. In the author's experience, this border always appears (incorrectly) above the last footnote whenever the PDF destination used, so it is seems to be a function of the destination rather than with ODS Document. These are two examples of periodic troubleshooting that arise with the expanded use of the replay technique.

CONCLUSIONS

As shown in the above examples, it is possible to obtain a single document of all TLFs with a hyperlinked table of contents and/or bookmarks with at least one level of sectioning. No additional software is required for generation of PDF files. Microsoft Word is used to open the final RTF file to select/update the table of contents. The replay approach is also showing potential with providing patient listings by site and study with a very specific structure (with additional levels in bookmarking) defined by FDA.

At first glance, the replay approach may seem intimidating especially for programmers with limited experience with item stores. The basic examples presented above do not require in-depth knowledge of item stores. Starting with a single item store, combining two item stores and restructuring are best understood by trying. Adding the code for ODS Document statements can fit into existing programs and processes. The Proc Document step will require additional efforts to implement and understand. The trade off is the ability to have a single document with desired structure and content for electronic distribution through a stable process.

Although the replay application is presented in the context of Proc Report and Proc SGRender, ODS Document and Proc Document offer much flexibility for many other SAS procedures as well. However, the application presented evolved from a very specific need in a very specific setting. These techniques have been used across multiple projects and have shown much promise.

REFERENCES

[1] Shannon, D. “To ODS RTF and Beyond”, SUGI 27, Paper 1-27

[2] Osowski, S., Fritchey, T. “Hyperlinks and Bookmarks with ODS RTF”, Pharmasug 2006, Paper TT21

[3] Tong, C. “ODS RTF: Practical Tips”, NESUG 16, Paper AT007

[4] Gilmore, J. “Using Dynamic Data Exchange with Microsoft Word”, SUGI 22, Paper 308

[5] Lawhorn , B. 2011, Let’s Give ’Em Something to TOC about: Transforming the Table of Contents of Your PDF File, SAS Global Forum

[6] Coar, W. “Appending Reports: A Review and Fresh Look”, WUSS 2012, Paper FP-70

[7] Anbazhagan, S, and Patel, S., “A SAS Macro Utility to Append SAS-Generated RTF Outputs and Create the Table of Contents”, PharmaSUG 2012, Paper AD12

[8] Gupta, A, “Watermarking and Combining Multiple RTF Outputs in SAS”, PharmaSUG 2012, Paper CC06

[9] SAS 9.2 Output Delivery System: User’s Guide, Understanding Item Stores, Template Stores, and Directories, Available at: http://support.sas.com/documentation/cdl/en/odsug/61723/HTML/default/viewer.htm#a003112776.htm

ACKNOWLEDGMENTS

The author would like to thank Nate Derby, Denka Markova, Darrin Despain, and David Kerr for their review and insights contributing to the poster. The author would also like to acknowledge the many programmers publishing information on the internet, and apologizes to those not properly referenced above, citing the inherent nature of internet searching makes it difficult to track so many individual ideas.

CONTACT INFORMATION

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

William Coar, PhD
Biostatistician/Director of Central Region Operations
Axio Research, LLC
11001 West 120th Avenue, Suite 400
Broomfield, CO 80021
Email: williamc@axioresearch.com