Create bookmarked PDFs using ODS

Aruna Kumari Panchumarthi, Novartis Pharmaceuticals Corporation, EH, NJ-USA

Jacques Lanoue, Novartis Pharmaceuticals Corporation, East Hanover, NJ- USA

ABSTRACT

When clinical studies are performed in pharmaceutical industry, huge amounts of statistical appendices like listings, tables, analyses and graphs are created which form a part of electronic submissions. Typically these outputs are produced using standard macros, there's very little need for actually getting our hands dirty with ODS. However occasionally there will be requests to produce outputs that do not fit in the standard 'CSR' output model. SAS ODS and the compatibility of PROC REPORT with ODS DOCUMENT can be a very creative method of processing a large number of reports and graphs.
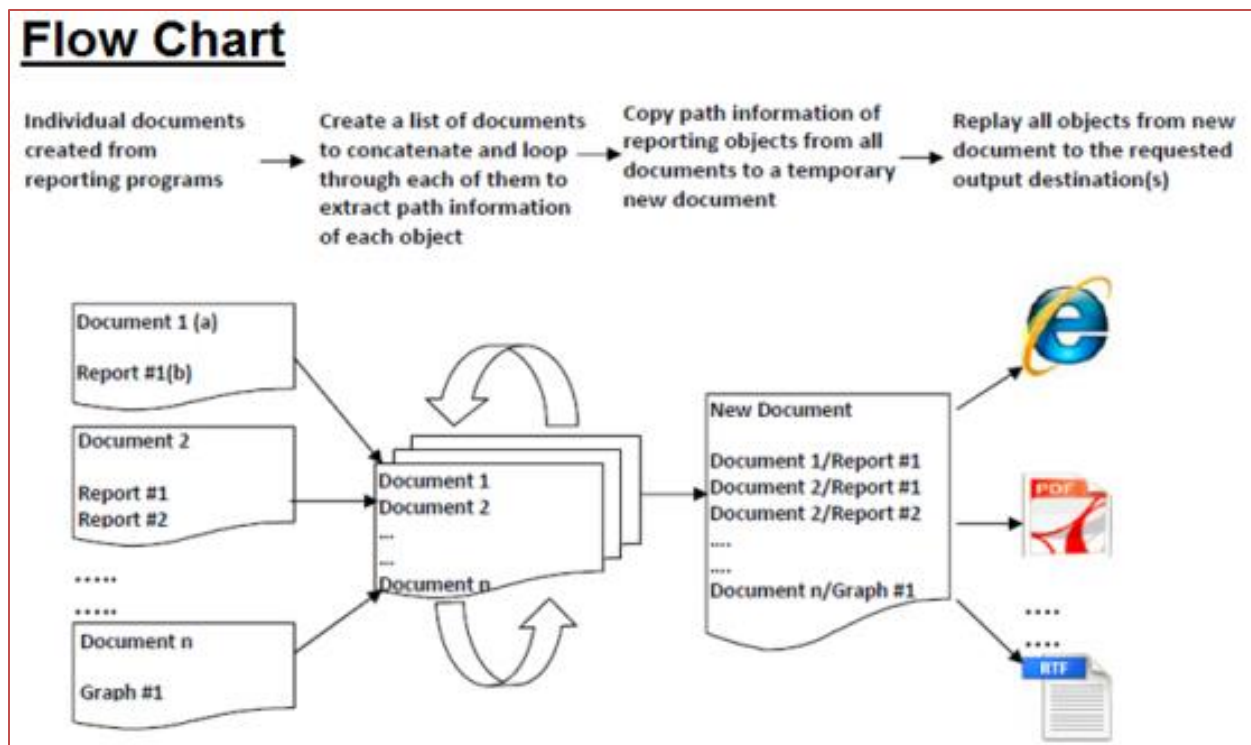
INTRODUCTION

As part of clinical trial reporting it is required to combine all reports in a single document like PDF/RTF for easy delivery and review. With ODS DOCUMENT it is possible to accomplish this task with relative ease.

PROC DOCUMENT, which is an enhancement to ODS, provides greater flexibility to modify/update reports. Apart from selectively displaying listing/graphs in the report, we can utilize ODS DOCUMENT capability to concatenate files and replay it in different destinations as a single document. This presentation will explain an automated process that combines various types of SAS output into PDF files, containing bookmarks.

PURPOSE/REQUIREMENT

To facilitate timely selection of clinical sites for potential FDA inspection as part of the NDA/BLA and/or supplementary review processes and used in the site inspection Audit. It is now a standard request for every NDA/BLA by FDA OSI, similar requests coming from EMA and other HA. Following a FDA request, we had to compile together several individual listings already provided by site subgroup into one unique pdf file which was reorganized by site and then listing for a selection of sites only, with corresponding bookmarks.

This paper discusses the evolution of a method to create the desired Subject data listings organized by site for each study (pdf) using the existing listing programs with minimal changes and enhancing the existing program to minimize the manual updates.

Flow Chart

| Individual documents created from reporting programs | → | Create a list of documents to concatenate and loop through each of them to extract path information of each object | → | Copy path information of reporting objects from all documents to a temporary new document | → | Replay all objects from new document to the requested output destination(s) |

Document 1 (a)

Report #1(b)

Document 2

Report #1
Report #2

* * * * *

* * * * *

Document n

Graph #1

Document 1
Document 2

Document n

New Document

Document 1/Report #1
Document 2/Report #1
Document 2/Report #2
....
....
Document n/Graph #1

LEGACY PROCESS

Previous process was very time consuming and lot of manual work for publishing group. We had to regenerate all listings for each site i.e. one RTF file per listing/site and ask publishing group to create the PDF and the bookmarks.

NEW PROCESS WITH ODS DOCUMENT

ODS DOCUMENT is an ODS destination. But, unlike "report" destinations, such as HTML, RTF, and PDF that create output files that are viewed and rendered with a third-party viewer, ODS DOCUMENT is not a SAS report destination. It is more like the ODS OUTPUT destination—in other words, a "data" destination. The ODS DOCUMENT destination holds output objects that are based on a SAS procedure or process. It holds output objects in their original creation structures, but you can rearrange the hierarchy and structure of the objects stored by ODS DOCUMENT. ODS documents are stored in a proprietary format (a document store) and can be viewed only with SAS software. The ODS DOCUMENT destination holds the output objects in such a way that they are able to be replayed or rerun to one of the report destinations. The ODS DOCUMENT destination is a SAS proprietary destination—to view or modify what's in the document store, you have to use either the ODS DOCUMENT window or PROC DOCUMENT.

Compiled all listings by site subgroup into one temporary pdf file (bookmarks are the automatic ones : Listing1>Site1,2,3,…,n ; Listing2>Site 1,2,3,…,n). Export into a txt file the list of bookmarks with theirs levels and paths. Create a new pdf file with custom folders and custom hierarchy based on the original list of bookmarks and PROC DOCUMENT to rearrange and replay the outputs. Bookmarks are renamed automatically with the site number or the title of the listing depending on the level.

Figure 1: Bookmarks in original pdf created with ODS and execution of listings for each site
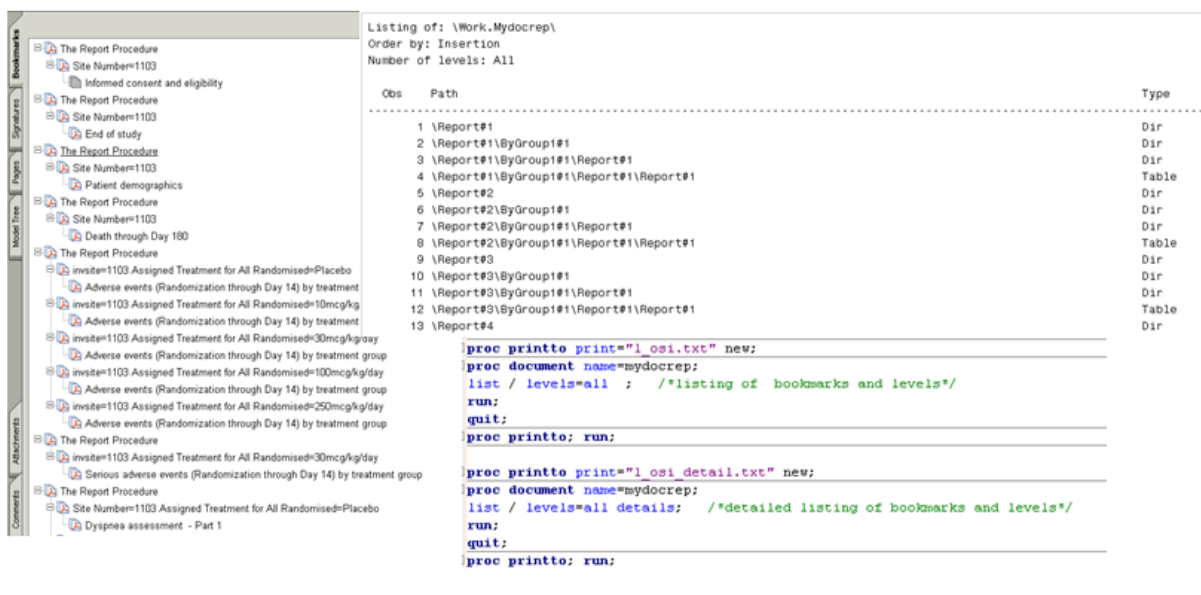
CREATE CUSTOM FOLDERS

Save output in an ODS document store using ODS document and Create custom folders and a custom folder hierarchy using PROC DOCUMENT to rearrange and replay your output. PROC DOCUMENT is a powerful procedure for Enhancement of ODS, providing greater flexibility to modify/update reports. It maintains an ongoing journal of your output that you can replay at any time without re-running the original analysis.

Several advantages: By delaying the tasks of printing and changing layout and appearance, you can focus on your main work. When it comes time to present the output, you can experiment with a variety of destinations, output formats, and style options, all without making any changes to or re-running your original code.

The highlighted PROC DOCUMENT program code below creates the new document store, NEWORDER. The MAKE statement creates one directory or folder for each site.

Next, the DIR statement causes an internal directory navigator to navigate back to the root directory of the document store. The syntax of this statement may look familiar to DOS or UNIX shell programmers who are familiar with navigation using DIR . or DIR .. syntax to navigate up and down the folder paths. Once positioned at the root directory (using ^^ instead of ..), the DIR statement causes navigation to the SITE_ directory. If the LISTING destination is open, the current directory location will be echoed in the Output window as a result of the DIR statement before the COPY statement. For example, the COPY statement copy objects from WORK.MYDOCREP into WORK.NEWORDER. It will be looped though for each site.

Code to create custom folders

```
/*define new order of bookmarks*/

        /*create one directory per site in which we will put each listing for that site*/
proc document name=neworder(write);
        make &listsite. ;
run;

        /*define each subdirectory which will correspond to one listing per site*/
        /*set the label */
        /*and copy the listing from original bookmark into new one*/
options mlogic;
%do s=1 %to %eval(&nbsite.);
        dir ^^;
        dir SITE_&&site&s. ;
        /**dir ;*/
        %do r=1 %to %eval(&&nbreps&s.);
                %let r2=%eval(&&&&bmrepn&s._&r.-((&s.-1)*26));
                %IF %eval(&&&&bmgrpt&s._&r.) EQ 1 %then %do;
                        make \work.neworder\SITE_&&site&s.#1\RepBG&r2. ;
                        setlabel \work.neworder\SITE_&&site&s.#1\RepBG&r2.#1  "%trim(&&titlelist&r2)";
                        dir  \work.neworder\SITE_&&site&s.#1\RepBG&r2. ;
                        copy \work.mydocrep&&&bmpath&s._&r. to ^;
                        dir ^^ ;
                %END;
                %ELSE %DO;
                        %IF %eval(&&&&bmgrp&s._&r.) EQ 1 %then %do;
                                make \work.neworder\SITE_&&site&s.#1\RepBG&r2. ;
                                setlabel \work.neworder\SITE_&&site&s.#1\RepBG&r2.#1  "%trim(&&titlelist&r2)";
                                dir  \work.neworder\SITE_&&site&s.#1\RepBG&r2. ;
                        %end;
                        copy \work.mydocrep&&&bmpath&s._&r. to ^;
                        %if %eval(&&&&bmgrp&s._&r.) EQ %eval(&&&bmgrpt&s._&r.) %then %do;
                                dir ^^ ;
                        %end;
                %END;
        %end;
%end;
```

REPLAY THE DOCUMENT

Once rearrange the objects you can replay the new document store to open LISTING, HTML and PDF destinations. PROC DOCUMENT code to REPLAY an ODS document, as shown in Figure 2 below.

Figure 2: REPLAY Code and the output

```
ods pdf file="&genpgm./export/pgm_saf/&FINALOUTPUT..pdf";

proc document name=neworder ;
    replay;
run;

ods pdf close;
run;
```


Some of ODS's advantages include the following:

- Creation of a navigational system.

- Creation of output objects from almost all procedures

- Creation of data sets from output object

- Control of format and style of report – font, color, etc.

- Easy distribution (free acrobat reader)

FURTHER IMPROVEMENTS

The process of manually opening each program and searching for the places that need to be updated is time consuming and prone to errors. By using Perl Regular Expression and CALL EXECUTE statements to enhance the existing program. Perl Regular Expression is a powerful tool that can be used to scan files for matches with an identifiable pattern and replace them with customized choices.

BATCH PROCESS CSR PROGRAMS

Existing CSR programs are batch processed to remove unwanted macros and statement using Perl Regular Expressions. PRXPARSE finds the exact match specified in the string for example: %autorun or %u_prerep etc. PRXMATCH in the next statement returns the first position in a string expression; with PRXCHANGE you can replace the string with a specified value. It's same as TRANWARD but one advantage of using PRXCHANGE over TRANWRD is that you can search for strings using wild cards. Note that you need to use the "s" operator in the regular expression to specify the search and replacement expression.

```
pid1=prxparse('/%autorun|%u_prerep|%u_postrep|%s_prerep|%s_postrep/i');
if prxmatch (pid1, line_ori) then line_new=prxchange('s/%/%*/',-1, line_ori);
```

Add options in proc report, rename final dataset

Using Perl Regular Expressions changed the options in proc report dynamically and delete temporary datasets, add where condition for selection of sites, by site subgroup, piece of code for handling no data to be shown for a given site, etc.

```
pid2=prxparse('/proc report\b.+?;/i');
pid3=prxparse('/data\b\s*?\=/i'); %regexp_check(pid3);

if prxmatch (pid2, line_ori) then do;
    line_new="<update done in next datastep>";
    CALL PRXSUBSTR (pid3, line_ori, pos_pid3 , length_pid3);
    dsfinal=scan(substr(line_ori,pos_pid3+length_pid3),1);
    call symputx ("dsfinal&i.",dsfinal);
end;
```

FLOWCHART
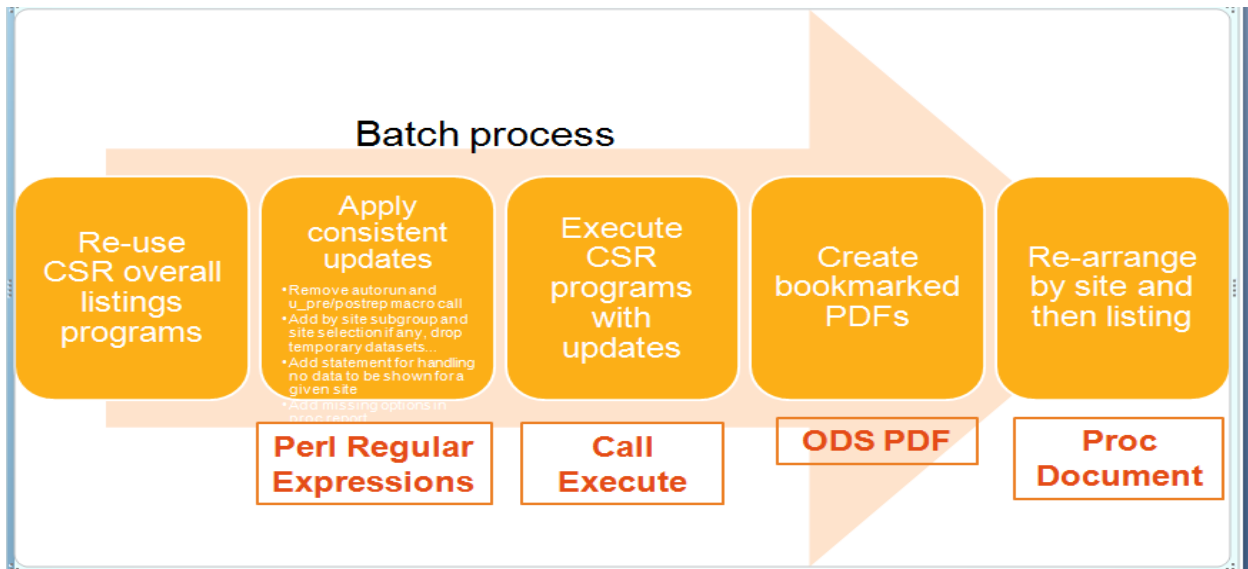
CONCLUSION

As you can see the examples in this paper, PERL Regular Expressions, CALL EXECUTE, ODS Document and PROC DOCUMENT create many opportunities for customizing your reports. You can change ODS destinations, system options, and macro variable between replaying documents to give variants of the report. By having greater control over the report details such as color, font, size, justification, order and labels, the format and quality of the report will be enhanced. The creation of custom styles and templates with PROC TEMPLATE facilitates more visually appealing output.

ACKNOWLEDGMENTS

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:
Name: Aruna K Panchumarthi
Organization: Novartis Pharmaceuticals
Address: 1 Health Plaza
City, State ZIP: East Hanover, NJ 07936.
Email: Kumari.sai@novartis.com

Name: Jacques Lanoue
Organization: Novartis Pharmaceuticals
Address: 1 Health Plaza
City, State ZIP: East Hanover, NJ.
Email: Jacques.Lanoue@novartis.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.