

SESUG Paper 239-2018

SAS Generated Dashboard - Useful Tool for A SAS Programmer

Pratap Kunwar, Emmes Corporation

ABSTRACT

Being savvy in SAS technical skills is a key to becoming a top SAS programmer. At the same time, it is important to possess other supporting tools that can provide related project and other information instantly during programming. A SAS-generated, HTML-based dashboard page with relevant information and instant navigation to project folders, people, other documents and any related html links is highly desirable, resulting in time saved time during the work day.

In this presentation, first I will share my dashboard that I have generated using SAS. Second, because of its popularity among my colleagues and non-programmers, I will show how I have set-up a SAS program for non-programmers, so a dashboard can be generated from Excel using the SAS interface, without having to open SAS in interactive mode.

INTRODUCTION

Having supportive documents accessible during programming is key to accomplishing programming tasks quickly. Programmers often do various work arounds with third-party tools to organize project files, folders, file paths and web links to have information handy when needed. This paper will describe how one can use Base SAS to create a personal html page that is full of supporting information and links, as shown in Figure 1 below.

pkunwar

Study1	Study2	Study3	Study4	Study5	Study6	Study7	Study8	Study9	Study10	Macros	Formats	Conferences	MLINKS	CSR	SMC	IND	PICKLIST	Books	STAT	SAS	Standardization	
Study1	Links	Information								People												
	Protocol	path\study\wp\study1_posted_studymaterials\protocol								Branch	ABC				AC	Name1						
	STAT	path\study\stat\study1								Group	XYZ				CSA	Name5						
	DM	path\study\dm\study1								Platform	ABC				DMG	Name1						
	Common	path\study\common\study1								Formats	PATH\CUP\ABC\SAS				DMG2	Name2						
	WP	path\study\wp\study1								eCRF	PATH\STUDY\DM\STUDY1\ADVANTAGEEDCANNOTATED ECRFS				DRI	Name9						
	WR-SAS	path\webreports\nonwalk\study1\sas\source								ProdData	PATH\CUP\ABC\SAS\12034				PMG	Name3						
	WR-RPT	path\webreports\nonwalk\study1\reports								TestData	PATH\CUP\ABC\TEST\SAS\12034				SAS Programmer	Pratap Kunwar						
	WR-PUB	path\webreports\study\study1								Sites	Name1, Name2, Name3				Statistician	Name6						
	WR-HR-REQ	path\abc\study1 - nonwalk								LabSite	Name1, Name2				Statistician2	Name7						
	WR-DM-FL	path\study\dm\study1\web_report_spreadsheets								RepSite	Name1											
	WR-Template									Agenda/Minutes												
	Website	https://emmes.cid.com/node/zzzzzz																				
	R-Subj-Profile	path\study\stat\study1\subject_profile																				
	R-IND	path\study\stat\study1\20_ind_reports																				
	R-SMC	path\study\stat\study1\21_smc_reports																				
	R-IR																					
	R-CSR	path\study\stat\study1\24_final_reports																				
	R-CSR-Sheet																					
	R-Manuscript																					
Picklist	path\study\stat\study1\11_picklists																					

Figure 1: SAS generated page

The tabs shown above can display any category of your choice from projects, protocols, favorite websites or list of SAS books. Information source can be implemented from meta data, or direct entry to SAS, or through excel as shown in Figure 2. This all depends on various factors, such as availability of meta data, programming expertise or simply personal choices. Excel based entry is suitable for non-programmers and for those who want to generate a dashboard without opening SAS in interactive mode.

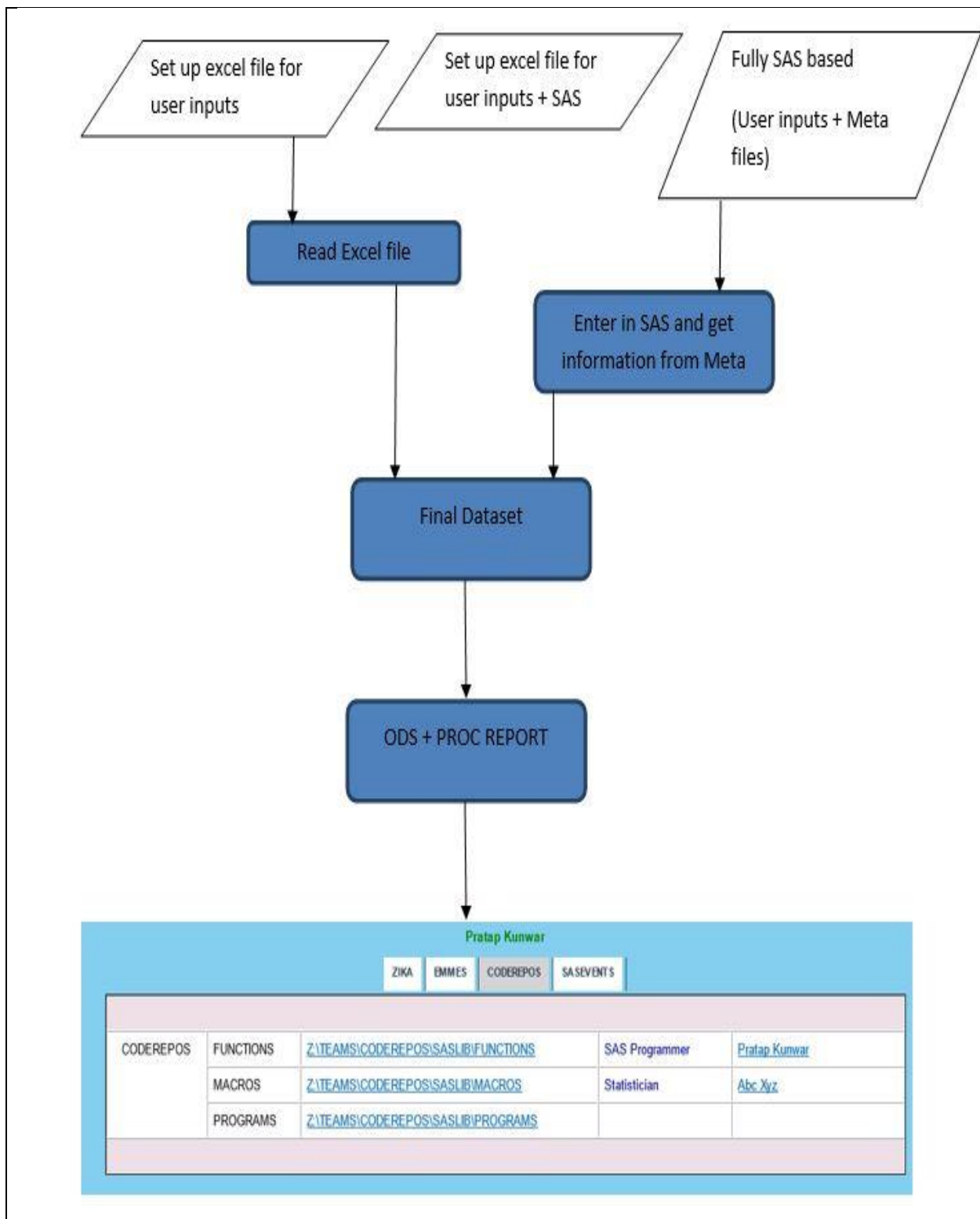


Figure 2. Flow diagram

Data Sources:

Option 1: Entry in Excel

An excel file can be set-up as shown below in Display 1. The 1st column (Order) is for ordering purposes in output html page (Figure 2). The 2nd column (Category) is Tab names. The 3rd column (Sub_category) is the 2nd column of the output page. The 4th column(Path) is the 3rd column of the output page. The 5th column (Title) is for 4th column of output page. The 6th column is for 5th column of the output page and the 7th column (Profile_Name) for page title.

Order	Category	Sub_Category	PATH	Title	Full_Name	Profile_Name
1	ZIKA	Programs	Z:\ZIKA\SAS\PROGRAMS	SAS Programmer	Pratap Kunwar	Pratap Kunwar
1	ZIKA	Reports	Z:\ZIKA\SAS\REPORTS	Statistician	Abc Xyz	
2	EMMES	Home	http://my.emmes.com/	SAS Programmer	Pratap Kunwar	
2	EMMES	ESUG	Z:\TEAMS\SAS USER GORUP			
3	CODEREPOS	FUNCTIONS	Z:\TEAMS\CODEREPOS\SASLIB\FUNCTIONS	SAS Programmer	Pratap Kunwar	
3	CODEREPOS	MACROS	Z:\TEAMS\CODEREPOS\SASLIB\MACROS	Statistician	Abc Xyz	
3	CODEREPOS	PROGRAMS	Z:\TEAMS\CODEREPOS\SASLIB\PROGRAMS			
4	SASEvents	SESUG	http://www.sesug.org/SESUG2018/index.php		Pratap Kunwar	
4	SASEvents	MWSUG	http://www.mwsug.org/2018/index.html			
4	SASEvents	SASGF	https://www.sas.com/en_us/events.html			

Display 1. Setting up excel file

Option 2: Entry in SAS

Another method is creating datasets directly in SAS as shown below in Display 2.

```
data z1;
do Category= 'CODEREPOS';
  do Sub_Category= 'FUNCTIONS', 'MACROS', 'PROGRAMS', 'SAS Programmer',
    'Statistician';
    output;
  end;
end;

do Category = '1000', '2000', '3000', '4000';
  do Sub_Category='Sub_Categoryegoryocol', 'STAT', 'DM', 'Common', 'WP', 'SMC',
    'CSR', 'IND', 'DMG', 'DRI', 'PMG', 'SAS Programmer', 'Statistician';
    output;
  end;
end;
run;

data z2;
set z1;

if Sub_Category= 'CODEREPOS' then
do;
  if upcase(Sub_Category)='FUNCTIONS' then
    value="Z:\TEAMS\CODEREPOS\SASLIB\FUNCTIONS";
  else if upcase(Sub_Category)='MACROS' then
    value="Z:\TEAMS\CODEREPOS\SASLIB\MACROS";
  else if upcase(Sub_Category)='PROGRAMS' then
    value="Z:\TEAMS\CODEREPOS\SASLIB\PROGRAMS";
  else if upcase(Sub_Category)='SAS Programmer' then value="Pratap Kunwar";
  else if upcase(Sub_Category)='Statistician' then value="Abc Xyz";
end;
run;
```

Display 2. Direct entry to SAS

Option 3: Meta Datasets

This scheme is totally dependent on the availability of meta datasets in your environments. In clinical environments, one can easily pull information such as site info, folder location, enrollment information from master datasets and then split by protocols (i.e. tabs) in output page. This scheme works well when meta data is available and study folders have standard folder structure. Otherwise, it might require longer programming codes.

Programming Steps:

Reading Excel File(s):

The following code shown in Display 3 below can be used to read the excel file. If the folder contains multiple xls files for multiple profiles, one can easily set up to read only yours with a macro variable `%let name=&sysuserid;` This will read one that matches the computer user ID, provided there is a xls file name `&sysuserid.xlsx`

```
%let xlsdir=&path.\xlsdir;
%let name=&sysuserid; or %let name=pkunwar;
libname xlsin "&xlsdir.\&name..xlsx" mixed=yes;
DATA xfile1;
    set xlsin."sheet1$"N;
    if order=0 then delete; /*not to read*/
    rename order=ord;
RUN;
libname xlsin clear;
```

Display 3. Importing excel based profile

Processing Imported Dataset:

The above imported file can be processed further as shown in Display 6 below.

```
DATA rptout;
    set xfile1;
    path=path;
    Category=upcase(Category);
    Sub_Category=Sub_Category;

    if not missing(full_name) then zname3=
cats("http://my.emmes.com/users/", lowercase(cats(first(full_name),
    scan(compress(full_name, '-'), 2))));

    format temp1 $500.;
    temp1=path;

RUN;
```

Display 6. Macro

In my environment, we have my employee profile on an html page at <http://my.emmes.com/users/pkunwar>", so I wanted to have a hyper link with first initial and last name. This might not be available, or may require different settings in other organizations.

Generating Final HTML Page:

Once the dataset is processed, we get a Macrovariable with a list of tabs as shown in the top portion of Display 7. Then we loop each tab within the output html page as shown below.

```
%let fsize=2;
%let tabs=0;
%let tabsc=;
proc sql;
    select distinct category into: tabs separated by ' ' from rptout order by ord;
    select distinct cats(' ', category) into: tabsc separated by ', ' from rptout order
        by ord;
    select propcase(profile_name) into: profile_name from rptout;
quit;
%let ntab = %sysfunc(CountW(&tabs));
%put &tabs;
%put &tabsc;
%put &ntab;

title;
footnote;
filename body "&rptdir.\pkunwar.html";
data _null_;
    file body;
    put '<!DOCTYPE html>';
    put '<html>';
    put '<head>';
    put "&<font font color=green face=arial style=line-height:0pt><center><h5>
        pkunwar </h5></font>";
run;

filename body "&rptdir.\pkunwar.html" mod;
ods tagsets.tableeditor path="&rptdir." body=body (title="&profile_name")
style=custom
options(frozen_headers="yes" pageheight="800" web_tabs="&tabsc"
web_tabs_just="center" background_color="skyblue" top_margin="15");

%macro tabs;
    %do i = 1 %to &ntab;
        %let tab=%scan(&tabs,&i);

        proc report data=rptout nowd spanrows noheader
            style(report)=[font_face=arial cellspacing=.5 outputwidth=14in font_size=&fsize.]
            style(header)=[background=lightgrey font_weight=bold font_size=&fsize.]
            style(column)=[background=white font_face=arial just=center font_size=&fsize.]
            style(lines)=[background=pkwh font_weight=bold foreground=blue just=left
                cellheight=.02in font_size=&fsize.];

            where category="&upcase(&tab)";
            columns link category templ sub_category path;
            define category/display order;
            define templ/display noprint;
            define link /display noprint;

            define sub_category/display style(column)={just=left};
            define path/display style(column)={just=left};
```

```

compute sub_category;
  call define(_col_, "url", temp1);
endcomp;

compute link;
  if link=0 then call define(_row_, "style",
    "style=[backgroundcolor=lightblue fontweight=bold]");
endcomp;

compute before _page_/style={just=1 vjust=top font_weight=bold background=pkwh};
  line ' ';
endcomp;

compute after comm/style={just=1 vjust=top foreground=green}; ;
  line comm $500.;
endcomp;

run;

%end;
%mend tabs;
%tabs;

ods tagsets.tableeditor close;
filename body clear;

```

Display 7. Final HTML Output

Setting up for non-programmers:

The Excel based method is most suitable for non-programmers. You can have a folder (Folder 1) for xls files where non-programmers create their personal excel files. A .bat file linked to the SAS programs located in SAS folder (Folder 2) can set up to read either all *.xlsx files or *.xlsx file matching with [&sysuserid](#) provided there is a xls file name [&sysuserid.xlsx](#). Output html files are then generated in the Profile folder (Folder 3) as shown in Figure 3 below.

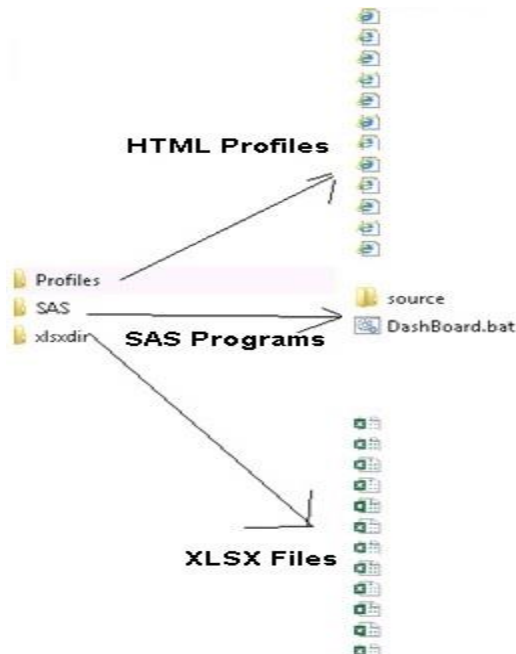


Figure 3: Sample folder setting for non-programmers

CONCLUSION

In this paper, I have presented multiple ways to create a profile page that is used for relevant information and instant navigation to project folders, people, other documents and any related html links.

ACKNOWLEDGMENTS

The author would like to thank my colleagues at the EMMES Corporation for their feedback and encouragement. The author would also like to thank Ryan Bratt for proofreading this paper.

CONTACT INFORMATION

Your comments/questions/criticisms are valued and encouraged. Please contact the authors at:

Pratap Kunwar
The Emmes Corporation
401 N Washington St.
E-mail: pkunwar@emmes.com