ABSTRACT

This paper addresses two key areas: (1) creating templates; and (2) changing the table of contents in SAS Web design. Example code for changing defaults for the table of contents is provided. Also, a key to template design and example templates are included. This paper is meant to assist developers designing web pages using SAS. An example application with graphs is presented in this paper.

INTRODUCTION

SAS’s Output Delivery System (ODS) and SAS/Intrnet™ allows the SAS programmer to create dynamic web enabled reports with little modification to their current code. To design custom web pages is a little more complicated. This paper is meant to assist the SAS programmer in designing the application as they desire and also has tips for adding the finishing touches. The templates created in this paper can serve as a key when creating an application for the web. Example code is also provided for several procedures showing how to change labels in the table of contents. Lastly, an example application is included that uses one of the templates in this paper. This is a dynamic application with drillable graphs.

TEMPLATE STYLES

Templates can easily be created by modifying existing templates provided by SAS. Code for an existing template can be viewed. See example that follows:

```sas
proc template;
  source styles.brick;
run;
```

There are several papers that give a thorough explanation on template design. Since the details have been covered in other papers, they will not be addressed here. Instead, code for modifying essential features of the template is provided. Papers covering specifics of template design in detail are listed in the reference section. The templates produced here are based on sample code from SAS’s web page.

MODIFYING TEMPLATE STYLES

The templates in this paper show how to change many features that permit an individual to create a custom application. Features include, changing the background of the body, the contents and the title; adding a specific title to the table of contents; removing numbers from the table of contents; adding extra space between items in the table of contents; adding graphics (logos); removing the default by line; removing the border from graphs; and lastly adding links. An example showing how to specify the by line in the table of contents through the procedural code is also provided.

The example code that follows creates a new template from an existing template. The effects of this new template are included in comments within the code, and can be seen by comparing Figures 1 and 2.

Example code for template_a:

```sas
proc template;
define style
styles.template_a/store=work.templat;
  parent = styles.default;
  /*removes default by line from table of contents*/
  replace bycontentfolder from bycontentfolder/
    prehtml=_undef_
    pretext='<!--'
    posttext='-->';
  /*background of body set to white*/
  replace body from document /
    background = white;
  /*background of table of contents set to white*/
  replace contents from document /
    background = white;
  /*change title in table of contents*/
  replace text /
    'content title' = "ANALYSIS OF ROSE DATA";
end;
run;
```

CHANGING THE TABLE OF CONTENTS WITHIN PROCS

Other changes can also be made to the table of contents within the actual PROCs. Example code follows that shows how replace the procedure names, how to add a specific by line, and how to change the label of the links listed below the procedure name in the table of contents.

Example code for PROC REPORT:

```sas
proc report data=rosedata nowd CONTENTS= 'All Rose Data';
column type zone year blooms;
define Type / group 'Rose Type';
define year / group 'Year';
define zone / group 'Growing Zone';
define blooms /analysis mean 'Average Blooms'
  format =3.;
title 'Number of Blooms';
ODS PROCLABEL = 'Number of Blooms';
run;
```

Example code for PROC TABULATE:

```sas
proc tabulate data=rosedata CONTENTS= 'Rose Data';
class type zone year;
var blooms;
table type*zone all, mean*blooms*(year all)/CONTENTS = 'Average Number of Blooms';
title 'Average Number of Blooms';
```
ODS PROCLABEL = 'Blooms by Type';
run;
quit;

Example code for PROC GCHART:

proc gchart data = rosedata;
/*specify by line for table of contents*/
hbar3d blooms/DESCRIPTION = 'Bar Charts for #byval(year)'
levels=10
cframe=cxdedded
freq freqlabel='Frequency'
caxis=black
group=Type
shape=cylinder;
title1 'Chart of Blooms';
by year;
ODS PROCLABEL = 'Yearly Graphs for Rose Data';
run;
quit;

Figures 1 and 2 contrast the results using the default output and default template, versus output from the procedures above using the DESCRIPTION, CONTENTS, and ODS PROCLABEL options and the modifications in template_a.

CONTINUING WITH TEMPLATE MODIFICATION

This is a good start, but additional changes such as adding graphics and links, and fine tuning the table of contents are probably desired. Code addressing these areas follow.

Example code for template_b:

/*template for dynamic web app*/
ODS PATH work.templat (update)
asuser.templat (READ)
sashelp.tmplmst (READ);
proc template;
define style
styles.template_b/store=work.templat;
parent = styles.default;

replace colors /
  "tableborder"=white /*outline of table and graphs*/
  "headerfg"=dark blue  /*heading text in table*/
  "databg"=CXDEDDED /*background of cell tables*/
  "datafg"=black /*data in table cells*/;

replace Text /
 'Content Title' = "ANALYSIS OF ROSE DATA";

/*change font sizes in table of contents*/
/*font for top title*/
replace contents from contents /
  font=(tahoma, 5);
/*font for procedure title*/
replace contentprocname from indexprocname /
  font=(roman,4);
/*remove numbering or bullets */
bullet=none;
/*font for label replacing procedure title*/
replace contentproclabel from contentproclabel /
  font=(roman,4);
/*font for label below procedure title*/
replace contentfolder from contentfolder /
  font=(roman,3);
/*font for listed items and add a space between items in table of contents*/
replace contentitem from indexitem /
  font=(roman,3);
posthtml='<p';
/*font for by line*/
replace Bycontentfolder from Bycontentfolder /
  font=(roman,3);
/*adding graphics-logos with links to the top*/
replace Body from Document /
  prehtml="<TABLE width="100%" align=center>
  TD align=bottom align=left >
  <font face="verdana, arial, helvetica, sans-serif" size="1.5">
GARDENING LINKS
<A href="http://www.rosegardening.com/"
  target="_parent"></IMG
AN EXAMPLE APPLICATION

This application provides an example of the dynamic graphs created with SAS/Intrnet using template_b. The graphs produced are based on sample code on SAS’s web page.

The following code tells SAS to create html and send it to the browser as a frame:

```sas
ods listing close;
ods html path= &_tmpcat (url=&_replay)
frame=_webout(dynamic)
contents=con.html
body=bod.html
RS=none
newfile=page
style=template_b;
/*insert proc code here*/
ods html close;
```

Example code for creating four drillable pie charts:

```sas
/*delete old charts*/
goptions device=gif570;
proc greplay nofs igout=work.gseg;
delete _all_;
run;
proc sort data=roses;
by type;
run;
/*options for pie charts*/
goptions reset=global gunit=pct
device=gif570 ftext=swiss ftitle=swissb
ctitle = black htitle=7 htext=5.5 hsize=2
vsize=2;
/*add links to data set here called 'HREF'*/
data roses;
set roses;
/* The link variable. */
length HREF $200.;
/* Assign values to HTML variables. */
if type='English' then HREF='href="'||&_replay||'English_data.html"'||"';
else if type='Tea' then HREF='href="'||&_replay||'Tea_data.html"'||"';
else if type='Yellow' then HREF='href="'||&_replay||'Yellow_data.html"'||"';
else if type='Grand' then HREF='href="'||&_replay||'Grand_data.html"'||"';
goptions nodisplay;
/*pie charts by types*/
proc gchart data=roses;
pie year / DESCRIPTION = 'Pie Chart'
sumvar =blooms
context=black
coutline=black
noheading
nolegend
ascending
value=outside
html=HREF
 discrete;
```

Results of this template can be seen in Figure 3. Using template_b added the graphics (or logos) to the top of the body page with links. It also added links to the bottom of the body page. Changes to the table of contents include adding a space between links, changing the font and removing the bullets in the table of contents. Other aspects of template_b are shown in Figures 6, 7 and 8.

![Figure 3](image.png)
by type;
Title1 "#Byval(type)";
ODS PROC LABEL = "BLOOMS";
options nobyline;
/*Reset HSIZE and VSIZE back to their default values*/
goptions hsize=0 vsize=0 device=gif570;

/* Set an order so that all observations will be out of range */
axis3 order=(-99 to -90 by 1);

/* Create a Shared title */
proc gslide;
title1 h=5.5 'Pie Charts Showing Blooms';
footnote h=3.5 'Click on pie chart(s) for the data for each type of rose.';
ODS PROC LABEL = "Blooms by Year for Each Type";
run;
quit;
/* Turn DISPLAY on */
goptions display;

/* Create 5 panel template,
Replay entries into panels */
proc greplay nofs igout=work.gseg tc=templatecat
;tdef five des='Pie Charts'
1/llx=1 lly=10
ulx=1 uly=50
urx=50 ury=50
lrx=50 lry=10

2/llx=1 lly=50
ulx=1 uly=90
urx=50 ury=90
lrx=50 lry=50

3/llx=50 lly=50
ulx=50 uly=90
urx=99 ury=90
lrx=99 lry=50

4/llx=50 lly=10
ulx=50 uly=50
urx=99 ury=50
lrx=99 lry=10

5/llx=0 lly=0
ulx=0 uly=100
urx=100 ury=100
lrx=100 lry=0

color=white;

template five;
treplay 1:gchart 2:gchart1 3:gchart2
4:gchart3 5:gslide;
run;
quit;
/*print data for each type for link*/
proc print data =roses noobs CONTENTS = 'Data Collected';
var blooms height year;
where type like 'English';
Title1 h=4 "Data for English Roses";
ODS PROC LABEL = "English Roses";
ods html body=English_data.html;
run;
quit;

proc print data =roses noobs CONTENTS = 'Data Collected';
var blooms height year;
where type like 'Tea';
Title1 h=4 "Data for Tea Roses";
ODS PROC LABEL = "Tea Roses";
ods html body=Tea_data.html;
run;
quit;

proc print data =roses noobs CONTENTS = 'Data Collected';
var blooms height year;
where type like 'Yellow';
Title1 h=4 "Data for Yellow Roses";
ODS PROC LABEL = "Yellow Roses";
ods html body=Yellow_data.html;
run;
quit;

proc print data =roses noobs CONTENTS = 'Data Collected';
var blooms height year;
where type like 'Grand';
Title1 h=4 "Data for Grand Roses";
ODS PROC LABEL = "Grand Roses";
ods html body=Grand_data.html;
run;
quit;

Figure 4 shows the body page for the pie charts using template_b. Figure 5 shows the frame page. The code above will display results as a frame (table of contents and body).
The piecharts in this application are drillable. That is clicking on a particular piechart will link to the corresponding page showing data for each type of rose. An example is shown in Figure 3.

Another effect of template_b is removing the outline around the graphs. This was done by specifying the same color as the background color (white) for the “tableborder.” Figure 6 shows the default, and Figure 5 shows the results of setting the tableborder to white.

CONCLUSION

SAS ODS features allow developers to tailor web applications by modifying existing templates and changing default labels in the procedural code.

WEB SITES

http://www.ukans.edu/cwis/units/ippbr/ksdata/sas.shtml
http://www.sas.com/md/base/topics/templateFAQ/Template.html
http://www.sas.com/service/techsup/sample/sample_graph.html

REFERENCES


ACKNOWLEDGMENTS

I am grateful to my colleagues at INDUS Corporation for their assistance in web design.

CONTACT INFORMATION

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

Veronica Y. Rath
INDUS Corporation
1953 Gallows Road
Vienna, VA 22182
Phone: (703) 506-6700
Fax: (703) 506-6776
Email: veronica.rath@induscorp.com
http://www.induscorp.com

SAS and SAS/Intrnet are registered trademarks of the SAS Institute Inc., Cary, NC, USA.