Paper 226-2009

A Framework to Customize Frame: Building an Interactive Report Web Portal by ODS

Xin Wei, Hoffmann-La Roche, Nutley, NJ

ABSTRACT

Clinical SAS® programmers generate hundreds, if not thousands pages of plain-text tables and listings on the daily basis. Identifying critical information from this vast volume of data, such as the patient who is a super responder to a novel RA treatment, an alarming adverse event, is not a trivial task. Therefore, it is in high demand to develop an interactive, user-friendly interface through which you could easily cross-examine various sections of a clinical report including tables, listings and figures by simply point-clicking on browser. In this paper, I will describe a framework that involved various web techniques from ODS and SAS procedure to customize the frame files created by ODS HTML/TAGSETS so that it can serve as an integrated web interface for an end user to review table/listing interactively.

Keywords: Web Report, ODS, Tagsets, PROC REPORT, PROC TEMPLATE, HTML

INTRODUCTION:

The current clinical trial reporting systems still heavily use the plain text summary tables and patients listings generated by PROC REPORT and PROC TABULATE. One downside of this type of “old-fashion” report is the difficulty to cross-examine the different sections of report and the lack of user interactivity. For instance, once you read the frequency table of a particular serious adverse event (SAE), you may want to immediately see the full AE listings and 12 lead ECG plot for all the individual patients who experience this type of SAE. This is not always an easy thing to do particularly when you are holding a report with hundreds and thousands pages, which is very common for late stage clinical trial. Bookmark of PDF report improves the readability but does not offer full solution.

Recently, lots of efforts have been devoted to having SAS run the analysis and reporting on real time. This requires licensing SAS/Intrnet or a bit more skills involving Common Gateway Interface (CGI) supported by a web server if you do not want the additional cost for SAS/Intrnet. The dynamic web reporting not only provides the user-friendly interface but also the real time summarization of the most updated, sometimes un-cleansing raw data. However, a potential caveat is that the results viewed by end users would be generated on fly without adequate validation.

One useful compromise between these two approaches is to create the static html report that resides on web server and have the full user interactivity. Since these html tables and graphics are created prior to the end user being granted access, they can be validated by QA/QC to meet the compliance requirement in the same standard way the plain text report is validated. In this paper, I use a single SAS macro to create such a web report interface for a simplified mock clinical study with three level hierarchies in its table of contents. First of all, the report may have multiple sections, for instance, patient disposition, efficacy analysis, safety analysis etc. Under each section title, the content of report is ordered by the number of table, listing and graphs. Finally, a grand table could be further split into a series sub-tables. For example, an electrocardiogram (ECG) summary should consist of tables for QT_interval, QRS complex, T-wave etc., all of which are different measures in ECG category.

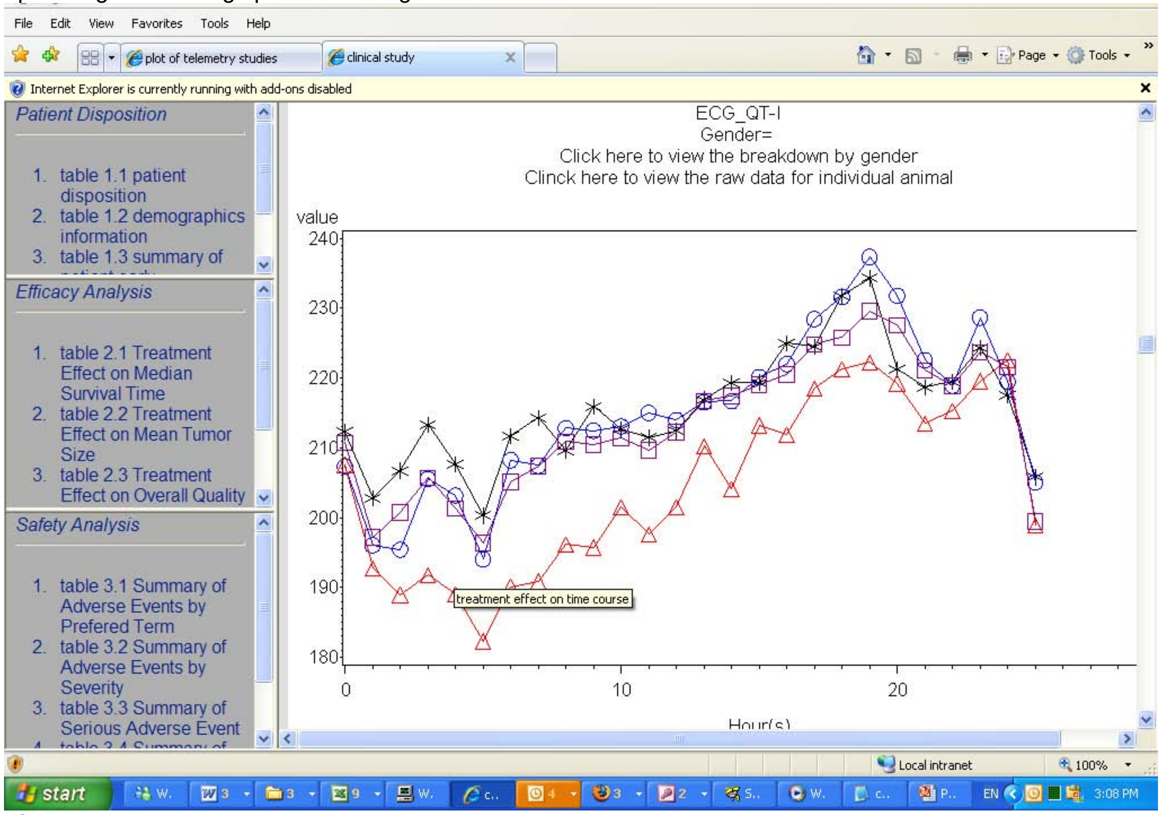
CONCEPT

1. Firstly, the table of contents and output body files for each section of a clinical report is created by standard ODS html or tagsets statement. PROC TEMPLATE is used to amend the appearance and hyperlink of table of contents generated by ODS to make it better resemble a real web portal.

2. The three table of contents files for patient disposition, efficacy, safety and the main body file of one of these three sections (depending on which of parts you want the end user to see at the first glance) are wrapped together into a single frame files that serve as a main entry to the full on-line report.

3. The links to the body of report are embedded into table of contents by the default built-into function of ODS. The links between different plots are created by “link=” options in title and footnote statement of SAS GRAPHICS procedure. The links between the dots on plots and the specific sections of tables are constructed by “html=” options of plot statement from PROC GPLOT and “style=[url=’ ’]” options from REPORT edure.

We expect to see all html files and their related graphics saved in the designated windows directory or web server after every step above is completed. A main frame file opens a web page as shown in Figure 1 in which you can navigate through all sections of a clinical report by point-clicking the table ID/name on the left and viewing the upcoming table/listings/plots on the right.


File Edit View Favorites Tools Help
plot of telemetry studies
clinical study
Page Tools
Internet Explorer is currently running with add-ons disabled
ECG_QT-I
Gender=
Click here to view the breakdown by gender
Clinck here to view the raw data for individual animal
value
Hour(s)
treatment effect on time course
Local intranet
100%
start
3:08 PM

Patient Disposition

1. table 1.1 patient disposition
2. table 1.2 demographics information
3. table 1.3 summary of

Efficacy Analysis

1. table 2.1 Treatment Effect on Median Survival Time
2. table 2.2 Treatment Effect on Mean Tumor Size
3. table 2.3 Treatment Effect on Overall Quality

Safety Analysis

1. table 3.1 Summary of Adverse Events by Prefered Term
2. table 3.2 Summary of Adverse Events by Severity
3. table 3.3 Summary of Serious Adverse Event

Figure 1

DEMONSTRATION

STEP 1: customize the table of contents

ODS html statements creates body file, table of contents and a frame file that stacks the table of contents and body of report together. This default function of ODS serves as the corner stone for our application. The following is the basic statements:

```
filename odsout 'c:\test';
ods listing close;
ods html body='body.html' contents='contents.html' frame='frame.html'
    path=odsout url=none;

proc report XXXXX; run;

ods html close;
ods listing;
```

Here the filename statement assigns the path that holds all the ODS output to the filename “odsout”. In the ODS statement, “path=odsout” option specifies that all upcoming ODS output files will be saved under that path. “url=none” exempts the need to specify the full path of output files every single time. Figure 2 demonstrates two levels of hierarchy in the table of contents. Each master table is created by a separate procedure (Report or Gplot) and its subtitles represent the values of by variables.

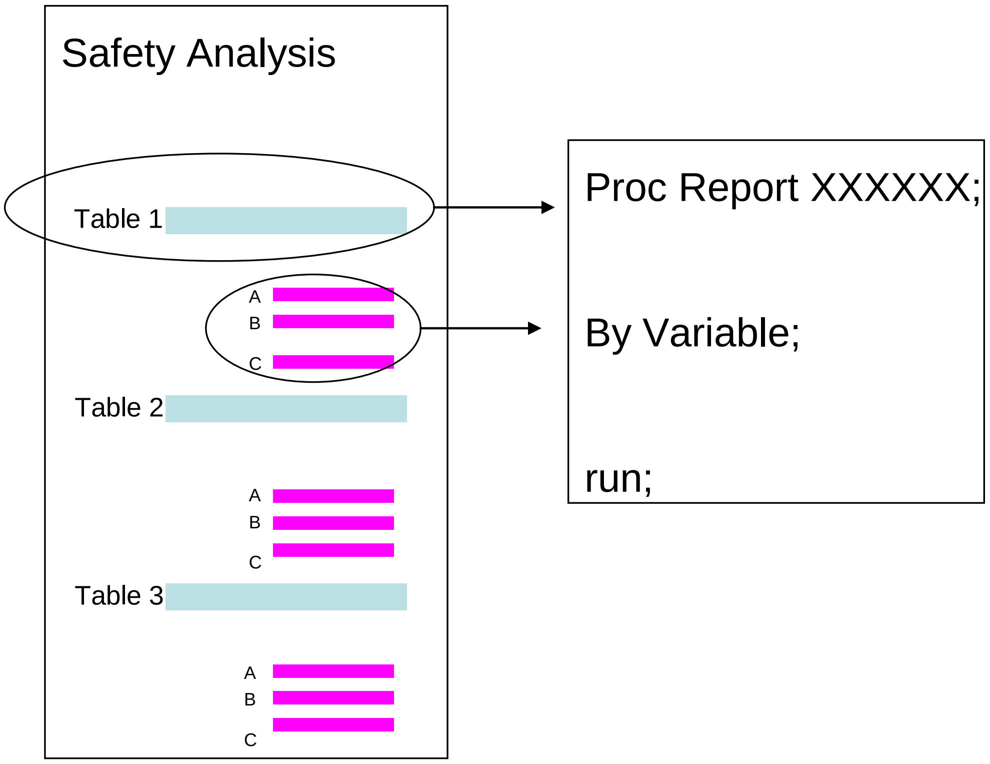


Figure 2

Figure 3 is an example of table of contents file generated by default ODS setting, which is quite distant from what we hope to achieve for a real web portal.

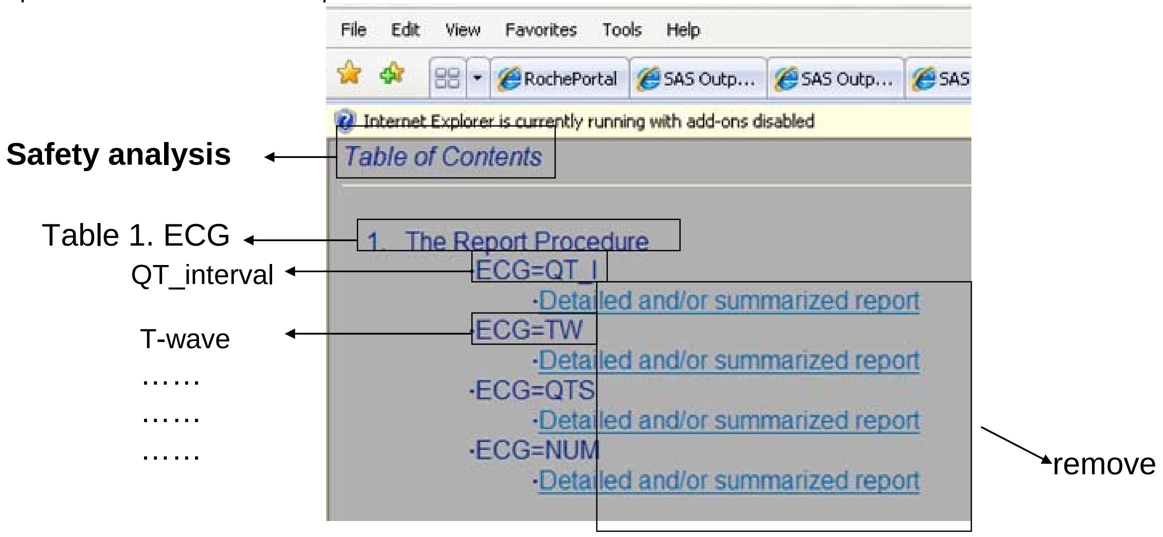

File Edit View Favorites Tools Help
RochePortal
SAS Outp...
SAS Outp...
SAS
Internet Explorer is currently running with add-ons disabled
Safety analysis
Table of Contents
Table 1. ECG
1. The Report Procedure
QT_interval
ECG=QT_I
·Detailed and/or summarized report
T-wave
ECG=TW
·Detailed and/or summarized report
……
·ECG=QTS
·Detailed and/or summarized report
……
·ECG=NUM
……
·Detailed and/or summarized report
remove

Figure 3

We may want to do several things to improve the default.

1. Replace the generic “table of contents” with your own title for this section, such as ‘Safety Analysis’ or “Statistical Analysis of XXXXX’. This can be done by modifying the pretext in the style setting of PROC TEMPLATE as follows. In the following ODS html/tagsets statements, “style styles.test” need to be added in order to inform ODS of the update of style definition.

```
 Proc templatep;
  define style styles.test;
     parent=styles.default;
     style contenttitle from contenttitle /
     pretext = " Statistical Report of Telemetry Data";
  end;
run;
```

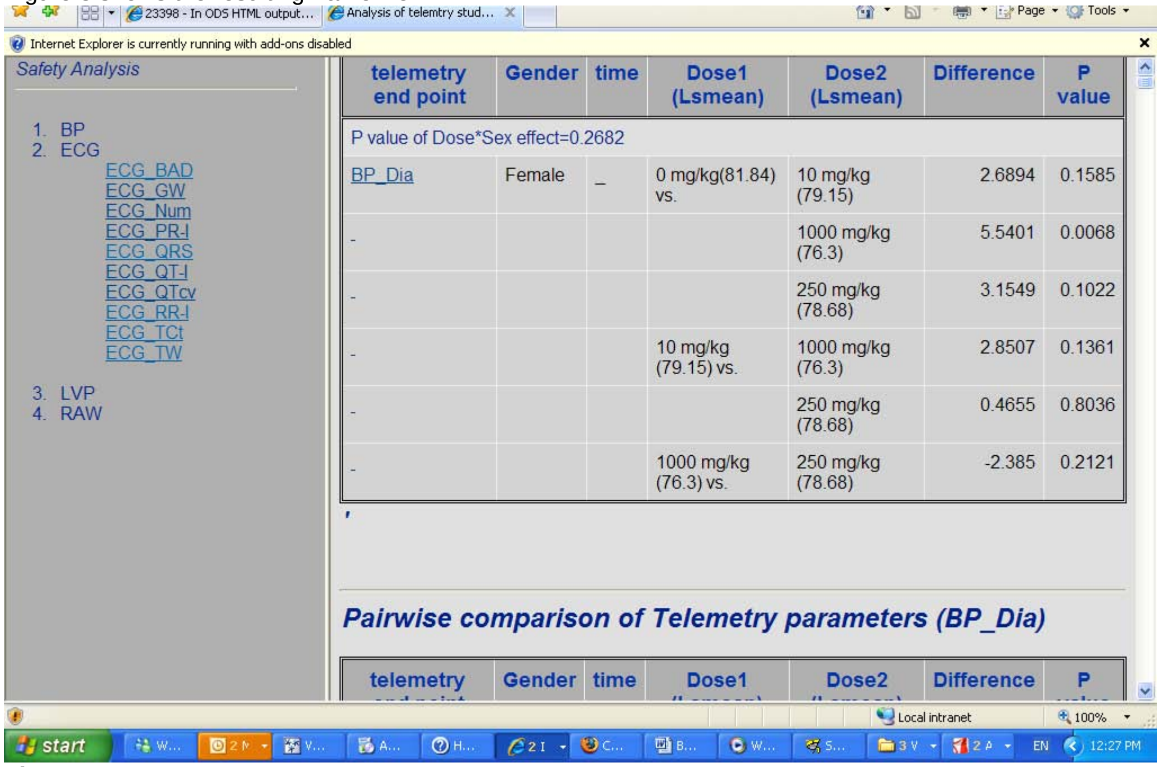
2. Replace the generic “The Report procedure” with your own table name. This can be done by using the ODS label statement prior to the execution of PROC REPORT.

```
ods label "ECG”;
```

3. Under each table name, the default subtitles appear in the format of “ECG=XXX”. A short description “Detailed and/or summarized report” hidden under the subtitle can be displayed once the subtitle is clicked by mouse. This short description contains the link to the corresponding portion of the report. What we want to do is a) to delete the “ECG=” from the subtitle so that only the real value of ECG variable, such as “QT_Interval” become the subtitle b) move the link from the description to the subtitles so that you can get the QT_Interval report by directly clicking the “QT_Interval” c) remove the description “Detailed and/or summarized report”. ODS tagsets allows a greater level of control over the SAS output than HTML by providing various options to modify the existing template. The following code from SAS website revises the default table of contents page:

```
 Proc template;
  define tagset tagsets.noequal;
  parent=tagsets.html4;
    define event list_entry;
      start:
    /* add link to the by variable */
        put "<li";
        putq " class=" HTMLCLASS;
        put ">";
        trigger do_link /if listentryanchor;
        trigger do_list_value /if ^listentryanchor;
        trigger hyperlink finish / if !cmp(htmlclass,"ContentName");
      finish:
  /* add a space only for a new . */
        put "<br>"  NL / if cmp(htmlclass,"ContentName");
        put "</li>" NL;
    end;
    define event hyperlink;
      start:
        put "<a href=""" URL;
        put """";
        putq " target=" HREFTARGET;
        put ">";
        put VALUE / if !cmp(htmlclass,"bycontentfolder");
      finish:
        put "</a>" NL;
    end;

    /* remove the by varuable = from the subtitle */
    define event do_list_value;
      eval $list index(reverse(value),"=")-1;
      eval $byval reverse(substr(reverse(value),1,$list));
      trigger hyperlink / if cmp(htmlclass,"bycontentfolder");

    put $byval;
     end;

    end;

run;
```

Figure 3 shows the resulting frame file:


23398 - In ODS HTML output...
Analysis of telemtry stud...
Page
Tools
Internet Explorer is currently running with add-ons disabled
Local intranet
100%
start
12:27 PM

Safety Analysis

1. BP
2. ECG
   - ECG_BAD
   - ECG_GW
   - ECG_Num
   - ECG_PR-I
   - ECG_QRS
   - ECG_QT-I
   - ECG_QTcv
   - ECG_RR-I
   - ECG_TCt
   - ECG_TW
3. LVP
4. RAW

telemetry end point	Gender	time	Dose1 (Lsmean)	Dose2 (Lsmean)	Difference	P value
P value of Dose*Sex effect=0.2682						
BP_Dia	Female	_	0 mg/kg(81.84) vs.	10 mg/kg (79.15)	2.6894	0.1585
-				1000 mg/kg (76.3)	5.5401	0.0068
-				250 mg/kg (78.68)	3.1549	0.1022
-			10 mg/kg (79.15) vs.	1000 mg/kg (76.3)	2.8507	0.1361
-				250 mg/kg (78.68)	0.4655	0.8036
-			1000 mg/kg (76.3) vs.	250 mg/kg (78.68)	-2.385	0.2121

Pairwise comparison of Telemetry parameters (BP_Dia)

telemetry	Gender	time	Dose1	Dose2	Difference	P

Figure 4

STEP 2: Add additional table of contents to the frame file

Report procedure with by statement only generates two levels of link hierarchy, which are the safety categories (Electrocardiogram, blood pressure etc.) and measure parameters under each category, respectively. One option to add additional level of hierarchy to the report, such as efficacy analysis and patient disposition, is to place an extra table of contents into the first frame file when this new table of contents has already been customized with the steps we described above. The following statement in Template procedure does so:

```
     define event content_frame;
         break /if ^exists( contents_name);
         put "<frameset frameborder=""yes"" framespacing=""1""
              rows=""66%,*"">" NL;
         put "<frame marginwidth=""4"" marginheight=""0""";
         putq " src=" URL;
         put " name=""contents"" scrolling=";
         putq "auto" /if ^exists( CONTENTSCROLLBAR);
         putq CONTENTSCROLLBAR;
         putq " title=" contents_title;
         put " title=""The Table of Contents""" /if
              ^exists( contents_title);
         put $empty_tag_suffix;
         put ">" NL;
              set $newtoc $options["NEWTOC_LINK"];
         put "<frame marginwidth=""4"" marginheight=""0"" src=""";
         put $newtoc;
              put """";
         put " name=""test"" scrolling=""auto"" title=""Added link"">"
             NL;
         put "</frameset>" NL;
      end;
```

This code add new link to the area of table of contents on the left of html frame. We then need to modify the ODS statement to specify the newly added table of contents file as follows:

```
ods tagsets.noequal body="body.html"
```

```
     contents='contents.html' frame='frame.html'(title="Safety Analysis")
     style=styles.test options(newtoc_link="contents2.html")
      path=odsout gpath=odsout(url=none);
```

Here “contents=” specifies the table of contents for the current section of a report, such as safety analysis. “newtoc_link=” within options specifies the newly added table of contents for additional section, such as “Efficacy Analysis”.

STEP 3: Building an interactive HTML body files

Customizing table of contents gives user the ease of browsing the different sections of a lengthy report. Furthermore, when you find some important or interesting data point at one section of a report, we hope that you can click them on the web report to cross-examine other related data points that might be informative but scatter throughout the whole study. Figure 4 shows an example where a time course plot for mean value, a time course plot for individual patient and the contrast p value table by time are linked together in a web portal. This is can be done by embedding hyperlink to the data point on body file.

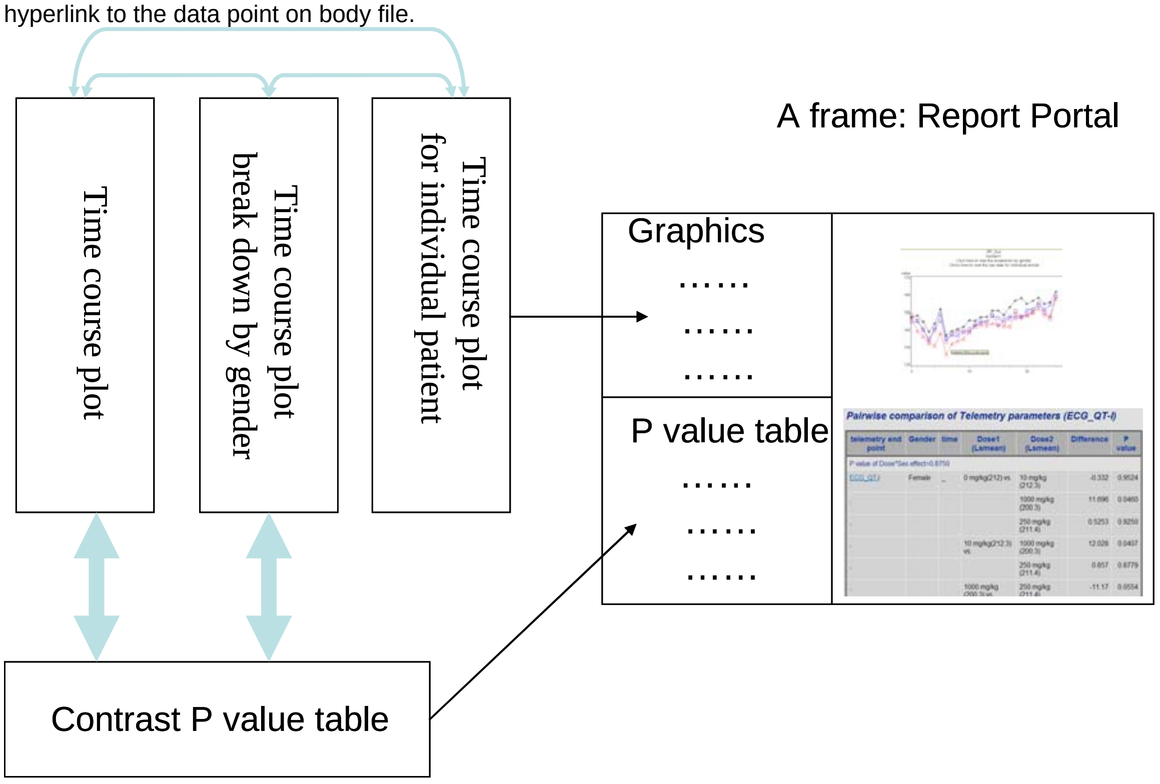

Time course plot
Time course plot break down by gender
Time course plot for individual patient
A frame: Report Portal
Graphics
……
……
……
P value table
……
……
……
Pairwise comparison of Telemetry parameters (ECG_QT-I)
Contrast P value table

Figure 5

1. **Anchor SAS html output and create variable containing link name in data step**

In ODS statement, adding “anchor="XXX1"” labels the SAS html output with ascending number. This numerical anchor will later instruct html web page to display the proper section of a HTML report. In data step, we also need to create a variable carrying the link. In the example of ECG, we hope to click one data point from a time course plot to get the pair-wise contrast p value table at the same time point. The following code labels the HTML table by parameter name and time.

```
ods tagsets.noequal file='pvalue_table.html' anchor='XXX1' XXXXXXX;

 proc sort data=pvalue; by parameter; run;
 proc report data=pvlaue nowd;
     column (parameter time XXXXXXXXXXXXXXXXX);
        by parameter;
        define parameter/order order=data  XXXXXXX;
        define time     /order order=data  XXXXXXX;

       XXXXXXXXXXXXXXXXXXX;

        break after time/page;
```

```
run;
ods tagsets.noequal close;
```

Both by and break statements instruct SAS ODS to incrementally label the html table with anchor ID. For instance, the final page of table will have anchor ID 72 if there are totally 3 parameters and 24 time points. The following code creates link variable in dataset for time course plot.

```
Proc sortdata=plot_data; by parameter time; run;
data plot_data;
     set plot_data;
     by parameter time;
     if first.time then i+1;
     html_var='pvalue_table.html#XXX'||trim(left(put(i,best.)));
run;
```

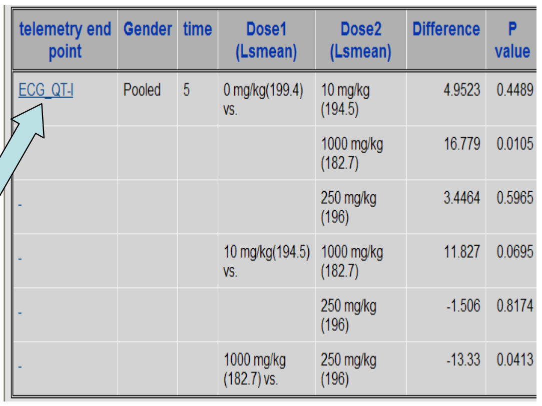
2. Link plot to table

In the Gplot procedure, “html=” option is used to embed the link to the data point on graph.

```
ods tagsets.noequal body="plot.html" anchor="fg1" path=odsout gpath=odsout(url=none);
goptions reset=all ftext="Arial" htext=1.5 vsize=6.5 in hsize=9 in gsfmode=replace
device=gif gsfname=gsasfile rotate=landscape colors=(black);
proc gplot data=plot_data;
     by parameter;
     plot mean*time=treatment/html=html_var;
run;
ods tagsets.noequal close;
```

3. Link table to plot

It would be convenient to check the plot while reviewing summary table by clicking the parameter name (Figure 6). Here we use format to embed the hyperlink pointing to a plot into the parameter name on the table.



telemetry end point	Gender	time	Dose1 (Lsmean)	Dose2 (Lsmean)	Difference	P value
ECG_QT-I	Pooled	5	0 mg/kg(199.4) vs.	10 mg/kg (194.5)	4.9523	0.4489
				1000 mg/kg (182.7)	16.779	0.0105
-				250 mg/kg (196)	3.4464	0.5965
-			10 mg/kg(194.5) vs.	1000 mg/kg (182.7)	11.827	0.0695
-				250 mg/kg (196)	-1.506	0.8174
-			1000 mg/kg (182.7) vs.	250 mg/kg (196)	-13.33	0.0413

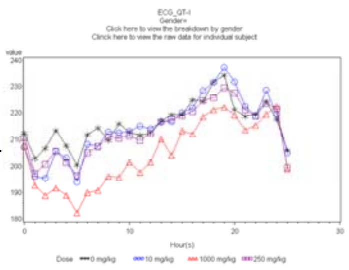


Figure 6

First we create the link variable and use it as the format for the parameter name.

```
proc sort data=pvalue; by parameter; run;
data pvalue;
     set pvalue;
     by parameter;
     if first.parameter then i+1;
     html_plot='plot.html#'||trim(left(put(i,best.)));
     fmtname='$param_plot_html';
run;
 proc sort nodupkey out=html_fmt; by parameter; run;
 proc format cntlin=pvalue(rename=(parameter=start html_plot=label)); run;
```

In the data step, we create the variable “html_plot” that holds the link and then define this variable as the format for the parameter name. The parameter name (parameter) and format variable (html_plot) are renamed as start and label in “cntlin=” option as required. FORMAT then writes this format into library so you do not need to manually define them. In PROC REPORT, “style=[url=]” option is used in define statement to embed the hyperlink.

```
proc report data=pvlaue nowd;
```

```
     column (parameter time XXXXXXXXXXXXXXXXX);
     by parameter;
     define parameter/order order=XXXXX XXXXXXX
                      style=[url=$param_plot_html.];
     define time/order order=data XXXXXXXXXXXXX;

     XXXXXXXX;

     break after time/page;
run;
```

4. Link plot to plot

“html=” option in plot statement of Gplot edure allows the link from one point on graphs to a different files. In fact, both footnote and title statements in SAS procedure provide option to set up hyperlink. This is useful to establish a link between individual graphs. The syntax is the followings:

```
Title link=”XXXX.html" 'Click here to view XXXXXX';
```

It is straightforward to insert a fixed link by this syntax. In order to vary the link destination based on by statement, we need the "#byval()" function. As an example, we may want to click the title of plot with pooled gender to view the separate graphs for different gender (as shown in Figure 7).

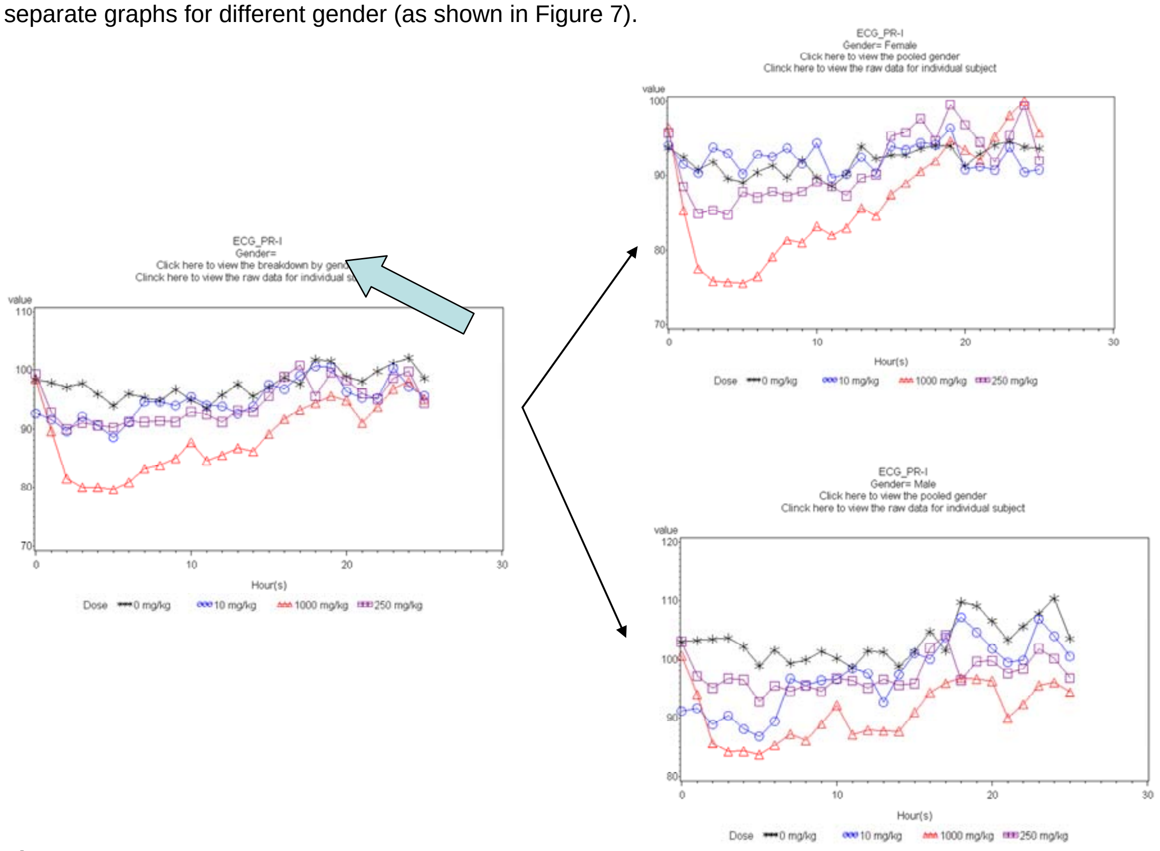


Figure 7

We first create variables that hold the link.

```
proc sort data=plot_data; by parameter sex; run;
data plot_data;
     set plot_data;
     by parameter;
        if first.parameter then i+1;
        link1="pooled.html#fg"||trim(left(put(i,best.)));
        link2="gender.html#fg"||trim(left(put(2*i-1,best.)));
run;
```

We then generate two body files, pooled_plot.html and separated_plot.html where the parameter to parameter link is established between these two files.

```
ods tagsets.noequal file='pooled_plot.html' anchor=1 XXXXXXXXX;
proc sort data=plot_data; by parameter link2; run;
proc gplot data=plot_data;
     by parameter;
        plot mean*time=treatment/html=html_var;
        where sex='pooled';
        title "#byval(link2)" 'click here to view the graphs break down
                              by gender';
run;
ods tagsets.noequal close;

ods tagsets.noequal file='separated_plot.html' anchor=1 XXXXXXXXX;
proc sort data=plot_data; by parameter link2 sex; run;
proc gplot data=plot_data;
     by parameter;
     plot mean*time=treatment/html=html_var;
     where sex in ('female', 'male');
     title "#byval(link1)" 'click here to view the graph for pooled
                            gender';
run;

ods tagsets.noequal close;
```

Note that the difference between link1 and link2 is because pooled gender only has one plot for one parameter while the gender break down has two graphs (male and female) for the same parameter.

DISCUSSION:

The techniques used in this paper to modify HTML page created by SAS ODS heavily rely on your in-depth understanding of ODS TEMPLATE procedure. One more flexible alternative, given that you are well versed with HTML language syntax, is to re-write the HTML source file in plain text by using DATA STEP. For instance, in page 3 figure3, we see the need to remove the “Detailed and/or summarized report” from the table of contents. This has been done by a relatively complex PROC TEMPLATE syntax shown in page 4. However, if you know what to change in HTML source file in order to achieve what you want, you can simply fix the source file in DATA STEP.

The first data step from the following codes reads the contents.html file created in the previous procedure into a SAS data set “portal” where a single column “raw” holds all the lines of HTML text contents. In the second data step, we identifies the line in HTML syntax that contains “Detailed and/or summarized report” by index function and get rid of it. The last data _null_ recreates the table of contents file by exporting the modified SAS dataset. Now open the updated contents.html, you will see the result you want.

```
data portal;
     infile "XXXX\contents.html" truncover lrecl=256;
     input raw $ 1-200;
run;
data portal;
     set portal;
     if index(lowcase(raw),'detailed and/or summarized report') then delete;
run;
data _null_;
    set portal;
    file "XXXX\contents.html";
    put raw;
run;
```

CONCLUSION:

In this paper, we describe a set of ideas and techniques that develop a HTML frame file created by SAS ODS into a web interface where the end users can interactively browse/cross-examine the table and body of contents of a clinical report.

REFERENCE:

- Base SAS 9.1.3 edures Guide, Second Edition
- SAS/GRAPH 9.1 Reference
- SAS 9.1.3 Output Delivery System: User's Guide SAS 9.1.3 Output Delivery System: User's Guide

CONTACT INFORMATION

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

Name:	Xin Wei
Enterprise:	Hoffmann La Roche Ltd
Address:	340 kingsland street
City, State ZIP:	Nutley, NJ, 09110
Work Phone:	973-235-2520
Fax:	973-235-2134
E-mail:	xin.wei@roche.com