Tricks and Tips with SAS/IntrNet

Keith Cranford
Marquee Associates, LLC

Dan Hammarstrom
Attorney General of Texas, Child Support Division

Abstract

This paper presents various techniques that help make your SAS/IntrNet applications more useful for your users. Printing considerations, graphics enhancements, and interactivity are some of the topics covered. The techniques involve SAS code and options as well as HTML, JavaScript and VBScript.

Introduction

With the increase in use of SAS/IntrNet, there arises the need to share experiences. The goal of this paper is to share some of our experiences and how we learned to address different needs. This is not intended to be an exhaustive list, but merely a collection of techniques that developers might find useful. Some were taken and modified from the SAS website or imbedded within SAS documentation. All could probably be found documented elsewhere. The value here is to bring some of these together in one place, and possibly spur discussion of other techniques that SAS/IntrNet developers have discovered.

Each technique is presented with a general discussion of the need and how to address the need. SAS, HTML, JavaScript or VBScript are shown and discussed, as necessary. Complete program code for each technique is then included in the Appendix.

Waiting icon

As a user is waiting for a report to return to the browser, it is often useful to display a note indicating that something is happening. This is especially useful if the request may take more than a few seconds. If it is a very long request, you might also want to indicate an approximate time frame. This can be done using PROC TEMPLATE and the HEADTEXT= option on the ODS HTML statement.

Three pieces of code are required to accomplish this. First, add “tagattr = PleaseWait()’ to the “style StartUpFunction” of the PROC TEMPLATE you wish to use. This tells the browser to call the Javascript function PleaseWait().

The next item to be added to the PROC TEMPLATE is the html code you wish to display. This code is entered in the “style Body” section. You can use any html code that displays a message to the user, but you must include the following code:

```
<div id="hidepage" style="position:absolute; height:100%;width:100%;">
```

Although not required, it is nice to include an animated GIF in your splash screen, since this gives the appearance that something is happening.

The following is an example of PROC TEMPLATE code that meets these requirements.

```
proc template;
   define style styles.oag2;
      parent = styles.default;
      style StartUpFunction from StartUpFunction /
         tagattr = "PleaseWait()";
      style Body from Body /
         prehtml =
         '<div id="hidepage" style="position:absolute;
                                    height:100%;width:100%;">
            <br /><br /><br /><br /><br />
            <center>
```

```
                <img src="/ideas/images/ideas.gif">
                <br />
                <b><font face="Arial" color="#000099" size="3">
                Your Request is Being Processed, Please Wait . . .
                     </font></b>
                <br /><br />
                <img src="/ideas/images/clock.gif">
              </center>
            </div>';
      end;
    run;
```

To implement this, add Javascript code to the HEADTEXT= option of the ODS HTML statement that defines the PleaseWait() function referred to in the PROC TEMPLATE code above.

```
ods html
    body     = _webout(dynamic)
    style    = oag2
    headtext = "<script>
                function PleaseWait()
                {if (document.getElementById)
{document.getElementById('hidepage').style.visibility='hidden'
                  }
                }
                </script>";
```

Once all three items are in place a user will not see a blank screen when they have submitted a request, but instead sees the custom screen you created.

To summarize, a sequence of events takes place to makes this work. When a user submits a request, SAS sends some information back to the browser: (1) the Javascript code that is in the HEADTEXT= option and (2) the “prehtml” code from the PROC TEMPLATE. At this point only the “prehtml” code is being displayed. This will continue until SAS has completed the request and sends the results to the browser. At this point SAS sends the request to run the Javascript code in the HEADTEXT= option. This is accomplished with the “tagattr = PleaseWait()” code from the PROC TEMPLATE, which calls the Javascript that was written earlier and hides the text written between the <DIV> tags.

One final note concerns the HEADTEXT= option. This option has a character limit (although we have not been able to track down what it is), so you might consider using an external style sheet or JavaScript file to overcome this limit. This would also have the benefit of changing styles in one place, instead of each program, thus improving maintenance.

Waiting Icon – Version 2

If you have an aversion to working with PROC TEMPLATE, an alternative to the waiting icon is to use <DIV> tags directly in your program to divide the output from the waiting message. Advantages to this method are not having to create a new style template and having programs self-contained, which also allows easier customization.
The general idea to this method is the same as the first, except that the HTML code is moved into the program, instead of in PROC TEMPLATE. The waiting message is contained in one set of <DIV> tags that is initially shown. The output is contained in another set of <DIV> tags that is initially hidden. Once the output is complete, a JavaScript program is run that displays the output and hides the waiting message. The following code achieves this.

```
ods path oagdata.templat(read) sashelp.tmplmst(read);
ods html body=_webout(dynamic)
    style=oag
        ;
```

```
data _null_;
   file _webout;
   put '<div id="pleasewait" style="position:absolute; height:100%; width:100%;">';
   put '<br /><br /><br /><br /><br />';
   put '<center><img src="/ideas/images/ideas.gif"><br />';
   put '<b><font face="Arial" color="#000099" size="3">';
   put 'Your Request is Being Processed, Please Wait . . . .</font></b><br /><br />';
   put '<img src="/ideas/images/clock.gif"></center></div>';
   put '<div id="output" style="visibility:hidden">';
   run;

                                < Output code goes here >

data _null_;
   file _webout;
   put '</div>';
   put '<script language="JavaScript1.2">';
   put '   document.getElementById("pleasewait").style.visibility   = "hidden"';
   put '   document.getElementById("output").style.visibility       = "visible"';
   put '</script>';
   run;

ods html close ;
```

Additionally, the before output code and after output code could be located in separate files and then included at the appropriate locations in all programs.

Multiple graph viewer

There are situations where you wish to display several reports either in a sequence or, at least, one at a time with the user determining which to view. However, you do not want to make a call to SAS to generate each report. This technique produces all the reports in a single program, using <DIV> tags to layer the reports and some VBScript along with an HTML form to allow the user to view a selected report.

This technique first utilizes <DIV> tags to produce layers in the HTML page, with each layer containing a single report. The ID parameter is used to give a name to the section and a CLASS parameter is used to indicate whether the layer should “show” or “hide”. The following example code produces a summary statistics report in a section called “report1”, and it should be initially shown.

```
data _null_ ;
  file _webout ;
  put '<div  id="report1" class="show">' ;
run ;

title 'Summary statistics' ;
proc means data=test mean min max maxdec=1 ;
  class gender ;
  var score ;
  run ;

data _null_ ;
  file _webout ;
  put '</div>' ;
```

Additional reports can be layered in like manner.

To select which report to view, an HTML form such as the following can be used.

```
<form action="" method="post">
  <input name="RptSel" type="radio" value="report1"
     onClick="ShowHide(report1)" checked> Summary<br>
  <input name="RptSel" type="radio" value="report2"
     onClick="ShowHide(report2)">Listing
</form>
```

The radio buttons allow a user to shift between the Summary and Listing reports. The onClick parameter calls a VBScript program (included in the Appendix) that determines which of the reports is shown or hidden. The example here uses VBScript, but similar JavaScript code could be used as well.

Some additional considerations involve the SAS code and how ODS presents the reports in HTML. By default, ODS puts a <HR> or <P> tag between procedure output. This can be overridden by including the following statement in a PROC TEMPLATE style.

```
style html / 'pageBreakLine' = %nrstr("");
```

If SAS/Graph is used to produce multiple graphs, use the NAME= option to give each graph a different name. Otherwise, only the final graph is displayed.

SAS/Graph drill down and tool tips

Drill-down capabilities are now built into SAS/Graph, making it very easy to add this functionality. The HTML= option in the graph procedures allows you to specify a variable that includes HTML code. This variable can contain any valid HTML code, such as HREF to link to another page or call another SAS program. You simply add this variable to the data set that is used for graphing.

For example, you might wish to add a tool tip and a link to a corresponding listing, for bars on a bar chart. This is accomplished by first adding a new variable (TEXT in this example) to the data set.

```
data test ;
   set test ;
   length text $ 200 ;
   select (gender) ;
      when ('M') text='title="This is a bar for Males"' ||
                      'href="//csewebprod/cgi-bin/broker?_service=oag' ||
                      '&_program=oagprog.graph_listing.sas' ||
                      '&_debug=0&gender=M&submit=Submit" ' ||
                      'target="_blank"';
      when ('F') text='title="This is a bar for Females"' ||
                      'href="//csewebprod/cgi-bin/broker?_service=oag' ||
                      '&_program=oagprog.graph_listing.sas' ||
                      '&_debug=0&gender=F&submit=Submit"' ;
      otherwise ;
   end ;
```

The TITLE= will produce a tool tip when the cursor hovers over the bar associated with a particular gender, if the browser window is selected. The HREF= establishes a link to a listing program for the particular gender (notice the gender= parameter that is passed). In this example, clicking on the M(ale) bar will open a new window due to the target=”_blank” parameter, whereas the F(emale) bar will use the current browser window to display the listing.

To use this technique in a SAS/Graph procedure, include a HTML= option on the graph statement such as VBAR or HBAR with PROC GCHART or PLOT with PROC GPLOT.

```
proc gchart data=test ;
  vbar gender / width=10 space=5 minor=0
                html=text ;
  run ;
```

SAS then embeds a client side image map into the HTML, which provides the functionality of the tool tips and drill down.

Open link (new SAS program) in new window

Similar to the graph example above, there are many instances where you will want to have a link to supporting information. You might also wish to have the new report appear in a separate window, so that you can view both reports simultaneously. A JavaScript method, window.open(), will allow this functionality. This method also gives you control of the window size, scrollbars, etc.

```
<script language="Javascript">
  function poprpt() {
    window.open("//csewebprod/cgi-bin/broker?_service=oag
    &_program=oagprog.graph_listing.sas&_debug=0
    &gender=B&submit=Submit",
    '_blank','height=450,width=500,menubar=no,scrollbars=yes')
    return
  }
</script>
```

When called, this will run the listing program for both genders, open the report in a new window (_blank paramenter), will be 450x500 pixels without a menubar but with a scrollbar. To call this function, reference the poprpt() function in a HREF tag.

```
<a href='javascript:poprpt()'>Listing</a>';
```

Print page icon

Instead of relying on the user to use the toolbar or pull-down menus to print a report, it is nice to provide an icon or link on the page that does this. This can be done with a JavaScript command, similar to what was shown in the previous section. The window.print() method prints the current page, so you may include the following JavaScript function.

```
<script language="Javascript">
  function printpage(){window.print()}
</script>
```

To print the current report, reference the printpage() function in a <A> tag with onclick parameter. You could either set up a text link or use an image like the following example.

```
<a href="#" onclick="printpage()">
<img src="//csewebprod/ideas/images/print.gif"
     alt="Click Here to Print Page" width="53" height="42"
     border="none">
</a>
```

Repeat page header

Printing tables from a web page can be problematic since there are no inherent page breaks. This can be overcome using a style section in the HEADTEXT= option on the ODS HTML statement, such as the following.

```
ods html
    body=_webout(dynamic)
    headtext = "<style> thead {display:table-header-group}</style>";
```

The ‘thead’ class selector controls the appearance of a table header. In this case the value “display:table-header-group” indicates that the table header should be repeated on each page.

Downloading to Excel

Many times a user wants to download data to Excel to perform their own analyses. The APPSRV_HEADER function provides this capability. It accomplishes this by overwriting the default mime header, which defines the type of document that ODS generates.

```
%let RV = %sysfunc(appsrv_header(Content-type,
                                 application/vnd.ms-excel));
%let RV = %sysfunc(appsrv_header(Content-disposition,
                                 %str(attachment; filename=temp.xls)));

ods htmlcss
    body     = _webout
    headtext = "<style> .zero {mso-number-format:\@;}</style>";
```

You may then follow this ODS statement with a PROC PRINT or PROC REPORT that formats the output to Excel. When this code is run, the standard File Download dialog is shown, allowing you to either save the file or open it in Excel directly.

You should be aware that Excel does not, by default, maintain leading zeros. To overcome this you need to do two things. First, add the code shown in the HEADTEXT= option of the ODS statement above. This creates a 'zero' style, where the "mso-number-format:\@" tells Excel to keep the leading zeros. Second, add style information to your output in your PROC PRINT or PROC REPORT section, such as the following.

```
proc print data = test label;
   var emp_id / style={htmlclass="zero"};
   run;
```

In this example emp_id is eight characters long and has leading zeros as part of the number. When Excel displays the data, the leading zeros will display - "00012345" instead of "12345".

Conclusion

Publishing to the web presents new challenges, and SAS/IntrNet provides many tools to meet these challenges. However, it may take some digging in various sources to find the proper tool. Hopefully, this paper will serve as a start down that search path and as a spark for further discussion.

If you have questions, please feel free to contact us at:

Keith Cranford
President and Principal Consultant
Marquee Associates, LLC
kcranford@marquee-assoc.com
512.453.6140

Dan Hammarstrom
System Analyst
Attorney General of Texas, Child Support Division
Dan.hammarstrom@cs.oag.state.tx.us
512.640.6303

Appendix

Code for: Waiting icon

****** *waiting.sas* ******

```
options spool fmtsearch=(oagdata);

ods path oagdata.templat(read) sashelp.tmplmst(read);
ods html body=_webout(dynamic)
    style=oag2
    rs=none path=&_tmpcat(url=&_replay)
    headtext = "<script>function loadimages()
       {if (document.getElementById)
       { document.getElementById('hidepage').style.visibility = 'hidden'}}
         </script>"
        ;

data _null_ ;
       file _webout ;

       do i=1 to 30000000 ;
       end ;

       put '<center><font color="003399" size=5><b>'
```

```
          'Job finished!</b></font></center>' ;


ods html close ;
```

Code for: Waiting icon

****** *waiting2.sas* ******

```
options spool fmtsearch=(oagdata);

ods path oagdata.templat(read) sashelp.tmplmst(read);
ods html body=_webout(dynamic)
    style=oag
         ;

data _null_;
   file _webout;
   put '<div id="pleasewait" style="position:absolute; height:100%; width:100%;">';
   put '<br /><br /><br /><br /><br />';
   put '<center><img src="/ideas/images/ideas.gif"><br />';
   put '<b><font face="Arial" color="#000099" size="3">';
   put 'Your Request is Being Processed, Please Wait . . . .</font></b><br /><br />';
   put '<img src="/ideas/images/clock.gif"></center></div>';
   put '<div id="output" style="visibility:hidden">';
   run;

data _null_ ;
        file _webout ;

        do i=1 to 30000000 ;
        end ;

        put '<center><font color="003399" size=5><b>Job finished, again!</b></font></center>'
;
run;

data _null_;
   file _webout;
   put '</div>';
   put '<script language="JavaScript1.2">';
   put '   document.getElementById("pleasewait").style.visibility   = "hidden"';
   put '   document.getElementById("output").style.visibility       = "visible"';
   put '</script>';
   run;

ods html close ;
```

Code for: Multiple graph viewer

****** *multi_view.sas* ******

```
options spool fmtsearch=(oagdata);

 **-- Test data --**;
data test ;
  input id $1. @3 gender $1. @5 score 3.  ;
  label id='Id #'
        gender='Gender'
               score='Test Score' ;
  cards ;
1 M  85
2 M  93
3 F  98
4 F  84
5 F  73
;
run ;

ods path oagdata.templat(read) sashelp.tmplmst(read);
```

```
ods html body=_webout(dynamic)
    style=oag2
    rs=none path=&_tmpcat(url=&_replay)
    headtext = "<script>function loadimages()
    {if (document.getElementById)
    { document.getElementById('hidepage').style.visibility = 'hidden'}}
    </script>"
                ;

 **-- VBscript for showing/hiding layers --**;

    file _webout ;

        put '<style>' ;
        put '<!--' ;
        put '   .hide { display: none;   font-family: Arial; font-size: 8pt; }' ;
        put '   .show { display: inline; font-family: Arial; font-size: 8pt; }' ;
        put '-->' ;
        put '</style>' ;
        put '<script language="VBscript">' ;
        put '   Option Explicit';
        put '   DIM menuID,menuObj,report1,report2' ;

        put '     report1   = "report1"' ;
        put '     report2   = "report2"' ;

        put '   SUB ShowHide(menuID)' ;
        put '     menuObj   = menuID' ;

        put '     if document.all.item(menuObj).className = "show" then' ;
        put '        document.all.item(menuObj).className = "hide"' ;
        put '     else' ;
        put '        document.all.item(report1).className    = "hide"' ;
        put '        document.all.item(report2).className    = "hide"' ;
        put '        document.all.item(menuObj).className    = "show"' ;
        put '       end if' ;
        put '   END SUB' ;

        put '   SUB Hide()' ;
        put '        document.all.item(report1).className    = "hide"' ;
        put '           document.all.item(report2).className    = "hide"' ;
        put '   END SUB' ;
        put '</script>' ;



**-- Navigation --**;

        file _webout ;

        put '<table width="100%">' ;
        put '<tr>' ;
        put '<td width="10%">';
        put '<form action="" method="post">' ;
        put '<input name="RptSel" type="radio" value="report1" '
         'onClick="ShowHide(report1)" checked>Summary<br>' ;
        put '<input name="RptSel" type="radio" value="report2" '
         'onClick="ShowHide(report2)">Listing<br>' ;
        put '</form>' ;
        put '</td>' ;
        put '<td width="80%">' ;
        run ;

data _null_ ;
  file _webout ;

  put '<DIV  id="report1" class="show">' ;
run ;

title 'Summary statistics' ;
proc means data=test mean min max maxdec=1 ;
```

```
  class gender ;
  var score ;
  run ;

data _null_ ;
  file _webout ;

  put '</DIV>' ;
  put '<DIV  id="report2" class="hide">' ;


title 'Listing' ;

        column id gender score ;
        define id / 'ID' ;
        define gender / 'Gender' ;
        define score / 'Test/Score' format=8.0 ;
        run ;

data _null_ ;
        file _webout ;

        put '</DIV>' ;
        put '</td>' ;
        put '</table>' ;
        run ;

ods html close ;
```

Code for: Drill down and tool tips:

****** *graph_drill.sas* ******

```
options spool fmtsearch=(oagdata);

 **-- Test data --**;

  input id $1. @3 gender $1. @5 score 3.  ;
  label id='Id #'
        gender='Gender'
         score='Test Score' ;
  cards ;
1 M  85
2 M  93
3 F  98
4 F  84
5 F  73
;
run ;

ods path oagdata.templat(read) sashelp.tmplmst(read);
ods html body=_webout(dynamic)
    style=oag2
    rs=none path=&_tmpcat(url=&_replay)
    headtext = "<script>function loadimages()
     {if (document.getElementById)
     { document.getElementById('hidepage').style.visibility = 'hidden'}}
     </script>"
       ;

data test ;
        set test ;
        length text $ 200 ;
        select (gender) ;
      when ('M') text='title="This is a bar for Males"' ||
                  'href="//csewebprod/cgi-
bin/broker?_service=oag&_program=oagprog.graph_listing.sas' ||
                  '&_debug=0&gender=M&submit=Submit" target="_blank"';
      when ('F') text='title="This is a bar for Females"' ||
```

```
                'href="//csewebprod/cgi-
bin/broker?_service=oag&_program=oagprog.graph_listing.sas' ||
                '&_debug=0&gender=F&submit=Submit"';
         otherwise ;
       end ;
run ;

goptions device=gif xpixels=500 ypixels=320 goutmode=replace ftext=swissb htext=2;
pattern color=green ;

title f=swissb h=3 'Drill-down Example' ;
proc gchart data=test ;
  vbar gender / width=10 space=5 minor=0
                html=text ;
  run ;

ods html close ;
```

****** *graph_listing.sas* ******

```
options spool fmtsearch=(oagdata);

 **-- Test data --**;

  input id $1. @3 gender $1. @5 score 3.  ;
  label id='Id #'
        gender='Gender'
               score='Test Score' ;
  cards ;
1 M  85
2 M  93
3 F  98
4 F  84
5 F  73
;
run ;

ods path oagdata.templat(read) sashelp.tmplmst(read);
ods html body=_webout(dynamic)
    style=oag2
    rs=none path=&_tmpcat(url=&_replay)
    headtext = "<script>function loadimages()
     {if (document.getElementById)
     { document.getElementById('hidepage').style.visibility = 'hidden'}}
     </script>"
                ;

%macro listing ;

title "Listing" ;
proc report data=test nowd split='/' ;
  %if %superq(gender) ne B %then
    where gender="&gender" ;;
       column id gender score ;
       define gender / 'Gender' ;
       define id / 'ID' ;
       define score / 'Test/Score' format=8.0 ;
       run ;

%mend listing ;
%listing ;
ods html close ;
```

Code for Open link in new window:

****** *new_window.sas* ******

```
options spool fmtsearch=(oagdata);

 **-- Test data --**;
data test ;
  input id $1. @3 gender $1. @5 score 3.  ;
  label id='Id #'
        gender='Gender'
               score='Test Score' ;
  cards ;
1 M  85
2 M  93
3 F  98
4 F  84
5 F  73
;
run ;

ods path oagdata.templat(read) sashelp.tmplmst(read);
ods html body=_webout(dynamic)
    style=oag2
    rs=none path=&_tmpcat(url=&_replay)
    headtext = "<script>function loadimages()
     {if (document.getElementById)
     { document.getElementById('hidepage').style.visibility = 'hidden'}}
     </script>"
                ;

data _null_ ;
       file _webout ;
       put '<script language="Javascript">' ;
    put ' function popstat() {' ;

       put '   window.open("//csewebprod/cgi-
bin/broker?_service=oag&_program=oagprog.graph_listing.sas&_debug=0&gender=B&submit=Submit"'
;
       put ",'_blank','height=450,width=500,menubar=no,scrollbars=yes');" ;
       put 'return' ;
       put ' }' ;
       put '</script>' ;

       put "   <a href='javascript:popstat()'>" ;
       put '   Listing';
       put '   </a>';

run;

goptions device=gif xpixels=500 ypixels=320 goutmode=replace
         ftext=swissb htext=2;
pattern color=green ;

title f=swissb h=3 'Drill-down Example' ;
proc gchart data=test ;
  vbar gender / width=10 space=5 minor=0 ;
  run ;

ods html close ;
```

Code for Print page icon:

****** *print_page.sas* ******

```
options spool fmtsearch=(oagdata);

 **-- Test data --**;
data test ;
  input id $1. @3 gender $1. @5 score 3.  ;
  label id='Id #'
        gender='Gender'
               score='Test Score' ;
  cards ;
1 M  85
2 M  93
3 F  98
4 F  84
5 F  73
;
run ;

ods path oagdata.templat(read) sashelp.tmplmst(read);
ods html body=_webout(dynamic)
    style=oag2
    rs=none path=&_tmpcat(url=&_replay)
    headtext = "<script>function loadimages()
     {if (document.getElementById)
     { document.getElementById('hidepage').style.visibility = 'hidden'}}
     </script>"
                ;

data _null_ ;
       file _webout ;
       put '<script language="Javascript">' ;
       put ' function printpage(){window.print()}';
       put '</script>' ;

       put '<a href="#" onclick="printpage()">';
       put '<img src="//csewebprod/ideas/images/print.gif" '
          'alt="Click Here to Print Page" width="53" height="42" border="none">';
       put '</a>';

run;

goptions device=gif xpixels=500 ypixels=320 goutmode=replace
         ftext=swissb htext=2;
pattern color=green ;

title f=swissb h=3 'Drill-down Example' ;

  vbar gender / width=10 space=5 minor=0 ;
  run ;

ods html close ;
```

Code for Repeat page header:

****** *print_page2.sas* ******

```
options spool fmtsearch=(oagdata);

 **-- Test data --**;
data test ;
  input id  $2. @4 gender $1. @6 score 3. ;
  label id='Id #'
        gender='Gender'
               score='Test Score' ;
  cards ;
 1 M  85
 2 M  93
 3 F  98
 4 F  84
 5 F  73
...
36 M  91
37 M  87
38 F  92
39 F  94
40 F  63
;
run ;

ods path oagdata.templat(read) sashelp.tmplmst(read);
ods html body=_webout(dynamic)
    style=oag2
    rs=none path=&_tmpcat(url=&_replay)
    headtext = "<script>function loadimages()
     {if (document.getElementById)
     { document.getElementById('hidepage').style.visibility = 'hidden'}}
     </script>
     <style> thead {display:table-header-group}</style>"
        ;

data _null_ ;
       file _webout ;
       put '<script language="Javascript">' ;
       put ' function printpage(){window.print()}';
       put '</script>' ;

       put '<a href="#" onclick="printpage()">';
       put '<img src="//csewebprod/ideas/images/print.gif" alt="Click Here to Print Page"
width="53" height="42" border="none">';
       put '</a>';

run;

proc report data=test nowd ;
  column id gender score ;
  define id  / 'ID #' ;
  define gender / 'Gender' ;
  define score / 'Test Score' ;
  run ;

ods html close ;
```

Code for Downloading to Excel:

****** *excel_listing.sas* ******

```
options spool fmtsearch=(oagdata);

 **-- Test data --**;
data test ;
  input id $1. @3 gender $1. @5 score 3.  ;
  label id='Id #'
        gender='Gender'
               score='Test Score' ;
  cards ;
1 M  85
2 M  93
3 F  98
4 F  84
5 F  73
;
run ;


%let RV = %sysfunc(appsrv_header(Content-type, application/vnd.ms-excel));
%let RV = %sysfunc(appsrv_header(Content-disposition,
                  %str(attachment; filename=temp.xls)));
ods listing close;
ods path oagdata.templat(read) sashelp.tmplmst(read);
ods html
    body     = _webout
    style    = excel
    css
    headtext = "<style> .zero {mso-number-format:\@;}</style>";

proc print data = test label split='?'
             style(obs)= {foreground=white background=#000099 just=center}
             style(obsheader) = {foreground=white background=#000099
                                 just=center font_weight=bold}
             style(header)    = {foreground=white background=#000099
                                 just=center font_weight=bold};
    var id / style={htmlclass="zero" just=center};
    var gender / style={just=left};
    var score / style={just=center} ;
    label id     = "ID"
           gender = 'Gender'
           score  = 'Test?Score' ;
    run;

ods html close;
```