GV003

Routine Web Reporting with Simple SAS® ODS®

Ashley H. Sanders, Animal Improvements Program Lab - USDA, Beltsville, MD

ABSTRACT

Data provided via the Internet today is no longer simply a value added product, it is often the only product. Besides being available to an infinitely expandable audience, it can have the advantages of speed, cost effectiveness, and flexibility. With minimal knowledge of SAS® ODS® it is possible to automate the distribution of standard data reports via the Internet. Routine summaries of dairy genetic evaluations, provided by the Animal Improvement Programs Laboratory of USDA, demonstrate tools for Internet reporting using SAS ODS. Beginning with data preparation, basic ODS HTML, and PROC REPORT statements, reports can be enhanced with PROC TEMPLATE and additional HTML. Programs are automated using macros. Reports are integrated within a complete website and are accessible by search engines, while older versions are automatically archived.

INTRODUCTION

The introduction of the Output Delivery System exponentially increases the possibilities for handling and presenting data with the SAS system. One area where it is particularly useful is for the generation of routine reports. ODS is flexible enough to efficiently provide high quality print, internet, and spreadsheet output, whether the audience is one, one hundred, or the general public.

This paper provides the tools to create regular summary reports, distributed over the internet. Topics include creating indexed HTML reports, using simple macros to customize reports based on the data, and providing alternative file formats such as .rtf or .xls.

GETTING STARTED

Providing reports over the internet saves on costs, increases content flexibility, and makes material available to almost any audience at the click of a button. It is the most efficient form for reports requiring wide distribution, but it is also valuable for providing information to a small audience rapidly and effectively. Give some thought to a few questions of format and delivery before you begin programming.

- Who is the audience?
  - Knowledge of content and Internet
  - Computer capabilities
- Is the report integrated with other Internet content?
  - Documentation and explanatory material
  - Reference material
  - Connections to other data
- How often is this report generated?
  - Does it require an accessible archive?
- Are alternative data formats needed?
  - Spreadsheet
  - Wordprocessing or print

REQUIRED INFORMATION

Some information about the SAS server system and the Internet server where reports will be available is needed.

SYSTEM INFORMATION

Internet site and Internet server information

- Identify the root URL (e.g. http://aipl.arsusda.gov) and what will be the complete URL of the pages being created.
- If files can be written directly to the Internet server (via a network connection), identify the file path for the Internet site from the computer running SAS.
- If files cannot be written directly to the web server, identify the location where files should be written for later promotion to the web server.

SAS system information

- Identify the location of the default autocall libraries (directories where SAS looks for stored macros called from another program).

  This list is found in the SAS config (SASV8.CFG) file in the -SET SASAUTOS entry:

```
-SET SASAUTOS  (
    "!sasroot\core\sasmacro"
    "!sasext0\assist\sasmacro"
    "!sasext0\eis\sasmacro"
    "!sasext0\ets\sasmacro"
    "!sasext0\gis\sasmacro"
    "!sasext0\graph\sasmacro"
    "!sasext0\iml\sasmacro"
    "!sasext0\intrnet\sasmacro"
    "!sasext0\or\sasmacro"
    "!sasext0\qc\sasmacro"
    "!sasext0\share\sasmacro"
    "!sasext0\stat\sasmacro"
  )
```

  A directory where created macros are stored can be added to the list.

- The location of the default and created libraries of template stores:
  - sashelp.tmplmst contains the default templates.
  - sasuser.templat (PC) or libraries.templat (UNIX/LINUX) usually contain created templates.
- SAS OnlineDoc provides comprehensive information about PROC TEMPLATE and most ODS procedures, so know where it is available on your system.

HTML RESOURCE

Although not much knowledge of HTML is needed to get Internet pages directly from SAS, learning the use of just a few basic *tags* will provide much greater control over how the output looks, and how it fits in with the rest of your Internet site. Having a good basic HTML reference handy is very helpful.

WRITING THE PROGRAMS

BASIC ODS HTML

Any program that generates output can create HTML documents with only a few additional commands.

First, include the two lines:

```
ODS _ALL_ CLOSE;
ODS LISTING;
```

at the top of the program. Although not required, these lines ensure that any ODS destinations opened by other processes earlier in a session are closed, and that the standard output destination, LISTING, is open (this isn't usually necessary if SAS is not running in interactive mode, but it can't hurt). In addition include the line:

```
ODS PATH    work.template(update)
            library.templat(read)
            sashelp.tmplmst(read);
```

work.template provides space for temporary templates created and used within the session or program. library.templat (usually sasuser.templat in PC SAS) is the location of any previously created templates, and sashelp.tmplmst is the location of SAS standard templates.

Before the output procedure statements(PROC REPORT etc.), the listing destination is closed and the HTML destination is opened.

```
ODS LISTING CLOSE;
ODS HTML
     BODY=”bullrank.html” (title=”Sires sorted
          by Net Merit”)
     METATEXT=’name=”keywords” content=”dairy,
          sires, merit”’
     HEADTEXT=’<!-- program_name.sas -->’
     PATH=’/external/web/bul/’ (url=’’)
     STYLE=BarrettsBlue;
run;
```

Notice that the ODS HTML parameters are all in one statement. The BODY option specifies the .html file that will be created. The *title* option creates a <TITLE> HTML tag in the document <HEAD>. This is what appears in the title bar of most Internet browsers, and is used by some search engines, so it is a good idea to include it. Likewise, METATEXT creates a <META> tag in the document <HEAD> with additional indexing information. HEADTEXT is included in the HTML document <HEAD> “as is” so this tag can be used to include comments (defined by the <!-- --> tags) such as the name of the SAS program that generated the output. PATH indicates the full path to use when writing the BODY file. This would be the path used to access the Internet server from the computer running SAS, or the location where files are written for later promotion to the Internet server. The *url* option identifies how this should be identified as part of any links created by SAS in your output. Specifying the empty quotes causes relative links to be created (discussed later). STYLE must be found in one of the locations identified in the ODS PATH statement.

All output for procedures that follow will be written to the .html file named in the ODS statement. To complete the process, After the output procedures, close the .html file and re-open the usual output destination:

```
ODS HTML CLOSE;
ODS LISTING;
run;
```

PROC TEMPLATE BASICS

The format of all ODS output (actually, all SAS output) is controlled by a style template. Besides the default style definition, several other styles are available with base SAS, and it is possible to create and store new styles. To see a list styles available from SAS, run the program:

```
PROC TEMPLATE;
  list styles;
run;
```

The output should look something like this:

```
Listing of: SASHELP.TMPLMST
Path Filter is: Styles
Sort by: PATH/ASCENDING

Obs    Path        Type
1     Styles                Dir
2     Styles.BarrettsBlue  Style
3     Styles.Beige         Style
4     Styles.Brick         Style
7     Styles.Default               Style
8     Styles.Minimal               Style
9     Styles.NoFontDefault Style
10   Styles.Printer        Style
11   Styles.Rtf            Style
12   Styles.Sasweb         Style
14   Styles.Theme          Style
15   Styles.fancyPrinter   Style
16   Styles.sansPrinter    Style
```

By applying changes to the default template, it is relatively simple to creating a customized template, and a few changes in the default style definition are required to enable some HTML formatting.

Run the program:

```
PROC TEMPLATE;
  SOURCE Styles.Default;
run;
```

The default style definition (quite long) will be written to the log and contains a series of statements like this:

```
style fonts
 "Fonts used in the default style" /
  'TitleFont2' = ("Arial, Helvetica,
       Helv",4,Bold Italic)
  'TitleFont' = ("Arial, Helvetica,
       Helv",5,Bold Italic)
  'StrongFont' = ("Arial, Helvetica,
       Helv",4,Bold)
  'EmphasisFont' = ("Arial, Helvetica,
       Helv",3,Italic)
  'FixedEmphasisFont' = ("Courier",2,Italic)
  'FixedStrongFont' = ("Courier",2,Bold)
  'FixedHeadingFont' = ("Courier",2)
  'BatchFixedFont' = ("SAS Monospace,
       Courier",2)
  'FixedFont' = ("Courier",2)
  'headingEmphasisFont' = ("Arial, Helvetica,
       Helv",4,Bold Italic)
  'headingFont' = ("Arial, Helvetica,
       Helv",4,Bold)
  'docFont' = ("Arial, Helvetica, Helv",3);
```

A simple change to make from the default template is to eliminate the alternate font definitions. The fonts definition statement (above) includes three alternative fonts (and a size and typeface descriptor) for most elements. The fonts are looked for, in order, on the browser’s machine to use in displaying the HTML page. If none of the listed choices is found, the browser will substitute its default font (which could produce strange results). By default, the font code is inserted around the output in every cell of the report table, which can result in a lot of code. Since Arial and Times New Roman are almost universally available, limiting your font definitions to one of these can greatly reduce the size of your output files. To create a style definition with single font choices, the program looks like this:

```
PROC TEMPLATE;
  DEFINE STYLE webdeflt;
     PARENT=styles.default;
     REPLACE fonts /
     'TitleFont2' = ("Arial",4,Bold Italic)
     'TitleFont' = ("Arial",5,Bold Italic)
     'StrongFont' = ("Arial",4,Bold)
     'EmphasisFont' = ("Arial",3,Italic)
     'FixedEmphasisFont' =
          ("Courier",2,Italic)
     'FixedStrongFont' = ("Courier",2,Bold)
     'FixedHeadingFont' = ("Courier",2)
     'BatchFixedFont' = ("Courier",2)
     'FixedFont' = ("Courier",2)
     'headingEmphasisFont' = ("Arial",4,Bold
          Italic)
     'headingFont' = ("Arial",4,Bold)
     'docFont' = ("Arial",3);
  end;
run;
```

Notice that the whole ‘fonts’ style attribute, even the elements that are unchanged (e.g. FixedStrongFont), are included because they are part of the fonts definition element.

In order to include some HTML characters in the SAS code one other change to the default template is required. Add a second REPLACE statement to the above program:

```
REPLACE Document from Container /
  htmldoctype = "<!DOCTYPE HTML PUBLIC ""-
  /W3C//DTD HTML 3.2 Final//EN"">"
  htmlcontenttype = "text/html"
  protectspecialchars = off
  linkcolor = colors('link2')
  visitedlinkcolor = colors('link1');
```

Setting protectspecialchars to ‘off’ causes HTML code included in title statements etc. to be written to the .html file as HTML code, rather than interpreted and included as characters to display.

A few other changes to the default template, allowing greater flexibility in creating tables of contents will be discussed later. Temporary changes to the template definition that affect output generated during the open SAS® session can also be made.

USING PROC REPORT OR PROC PRINT

PROC PRINT will usually provide adequate report results for displaying simple data tables in HTML, but some features of ODS that are available in PROC REPORT can not be used in PROC PRINT. PROC REPORT can also be used to include summary rows, group data within a table, or produce output that requires calculations on the data. Developing familiarity with PROC REPORT is worthwhile for any programmer that has to provide summarized results for someone else to review or use. PROC TABULATE can also be used to create Internet reports, but, like PROC PRINT, most of its features are included in the PROC REPORT

Some of the effects available in either PROC PRINT or PROC REPORT (with protectspecialcharacter = off in the style template) are:

- Use labels that include HTML code.
- Create variables that include HTML code including links to other pages or data
- Apply styles to the whole output table in the PROC statement
- Use titles and footnotes that include HTML code
- Create a table of contents that displays in a frameset

SIMPLE ENHANCEMENTS

USING SIMPLE MACROS FOR MULTIPLE CLASS OUTPUT

If the same reports are frequently produced for standard sub-classes of data, use a macro to create an HTML report for each class. In the example of dairy evaluation reports, most reports are reproduced each quarter for each of the six major dairy breeds. The data step and the ODS output steps are all within the macro *breed*.

```
%macro breed (brd, breed, tb);
     data &tb (where=(brdevl=”&brd” and status
          in (‘A’,’F’));
             set lib.pta&run;
             if pctl = ‘-‘ then pctl = ’00’;
             if pctl = ‘+’ then pctl = ’100’;
             npctl = input(pctl,2.);
             relyld = round(100*relyld,1);
     run;

     ODS HTML
        BODY=”&tb.nm.html” (title=”&Breed
           Sires sorted by Net Merit &omonth.,
           20&nyear)
        PATH=”/external/web/bul/current” (url=
           “”)
        STYLE=webdeflt;
     run;

     PROC REPORT data=&tb;
             Title “Top &breed Sires”
     quit;

     ODS HTML CLOSE;
     ODS LISTING;
%mend breed;
%breed (hol,Holstein,HO);
%breed (jer,Jersey,JE);
```

An .html file for each breed is written to the directory for current reports (archiving previous reports is discussed later). Use %if %then sub-setting for code that should be selectively executed for certain sub-groups.

Another advantage to using macros when generating reports, is that when several variations on a variable or category are desired on the report, they can be associated in the macro definition. For instance the variable brd is stored as a three letter code (hol, ayr, etc) Formats can be used to produce variations on this in the output tables, but using a macro with three variables defined allows the full name to be used in the title statement, and the HTML file to be named using the two letter designation.

TABLE OF PAGES (FRAMES)

One way to organize and access related reports is with a table of pages (TOP) that contains links to each .html, and is displayed in an html *frame set*. SAS ODS has built in utilities for doing this, and the standard output can be enhanced with a little creativity.

```
ODS HTML
     BODY=”nmtab1.html” (title=”Sires sorted
          by Net Merit &omonth., 20&nyear)
     PAGE=”nmtoc.html” (title=”Net Merit TOC”)
     FRAME=”nmfrm.html” (title=”Top bulls for
          Net Merit, 20&yr”)
     PATH=”/external/web/bul/current” (url=
          “”)
     STYLE=webdeflt
     NEWFILE=PROC;
```

This creates the frame set file, nmfrm.html that will be accessed by browsers first. The TOP file (nmtoc.html) will be displayed in the frame down the right side of the page, and the file nmtab1.html will be displayed in the main frame. NEWFILE=PROC causes a new .html file to be written for each PROC step (for instance, multiple PROC REPORT steps executed within a macro). The TOP file includes links to display each of the reports in the main frame.

One drawback to this arrangement is that the macro variables specific to the data are not available in the ODS HTML statement for file names or titles. Another is that the ODS generated table of contents or pages is not always the desired format and is much less flexible than other ODS created items.

A work around for this involves creating an alternate TOP. First, two file references are established; one for the ‘real’ TOP, and one for the ‘dummy’ file. Both include the mod parameter.

```
   filename toc “nmtoc.html” mod;*real file;
   filename tocd “dump.html” mod;*dummy file;
```

The ODS destination is opened assigning the ‘real’ filename as the CONTENTS file. This creates all the files with the proper structure and relationship (the body file assigned at this point is irrelevant, as this will be reset later). Then the ODS destination is closed. The ‘real’ TOP file is modified using data _null_ and the PUT statement.

```
   ODS HTML
   BODY=”dump2.html
   PAGE=toc (title=”NM$ Table of Pages” nobot)
   url=”nmtoc.html”)anchor=t1
   FRAME=”nmfrm.html” ((title=”Top bulls for Net
        Merit, 20&yr”)
   PATH=”/external/web/bul/current” (url= “”)
   STYLE=webdeflt
   NEWFILE=PROC;
   run;

   ODS HTML CLOSE;

   data _null_ ;
        file toc;
        put ‘<P><FONT SIZE=”+1”>Net Merit</P>’;
   run;
```

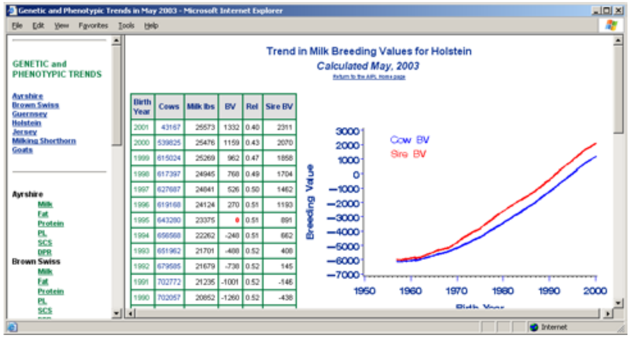
Notice the nobot parameter on the PAGE= option. This causes the file to be left with no closing html tags. The text string in the PUT statement can be any valid HTML.

The ODS destination is then opened and closed for each output procedure that is to have a separate html file. In these steps, however, the dummy TOP file (later deleted) is specified. The ‘real’ TOP file is continued with additional data _null_ steps.

```
   %macro breed (brd, breed, tb);
        data _null_ ;
        file toc;
           put “<A HREF=””&brd.nm.html#&brd””
                target=””body””>&breed</A><br>”;
        run;

        data &tb (where=(brdevl=”&brd” and status
             in (‘A’,’F’));
             set lib.pta&run;
        run;

        ODS HTML
        BODY=”&tb.nm.html” (title=”&Breed Net
           Merit List &omonth., 20&nyear)
        PAGE=tocd notop nobot anchor=”&brd”
        PATH=”/external/web/bul/current” (url=
           “”)
        STYLE=webdeflt;
        NEWFILE=NONE;
        run;

        PROC REPORT data=&tb;
        quit;
        ODS HTML CLOSE;
        ODS LISTING;
   %mend breed;
   %breed (hol,Holstein,HO);
   %breed (jer,Jersey,JE);
```

Once all the files are created, a final ODS statement, using the notop parameter on the PAGE= option closes all the files properly. The BODY= is set to another dummy file (which will also be deleted).

```
   ODS HTML
   BODY=”dump2.html
   PAGE=toc (title=”NM$ Table of Pages” notop)
   url=“nmtoc.html”)anchor=’t3’;
   FRAME=”nmfrm.html” ((title=”Top bulls for Net
        Merit, 20&yr”)
   PATH=”/external/web/bul/current” (url= ’’)
   STYLE=webdeflt
   run;

   ODS HTML CLOSE;
   ODS LISTING;
   run;
```



Trend in Milk Breeding Values for Holstein

Calculated May, 2003

Return to the AIPL Home page

Birth Year	Cows	Milk lbs	BV	Rel	Sire BV
2001	43167	25573	1332	0.40	2311
2000	539825	25476	1159	0.43	2070
1999	615024	25269	962	0.47	1858
1998	617397	24945	768	0.49	1704
1997	627687	24841	526	0.50	1462
1996	619168	24124	270	0.51	1193
1995	643280	23375	[illegible]	0.51	891
1994	656568	22262	-248	0.51	662
1993	651962	21701	-488	0.52	408
1992	679585	21679	-738	0.52	145
1991	702772	21235	-1001	0.52	-146
1990	702057	20852	-1260	0.52	-438

Use system commands to clean-up extra files (dump.html and dump2.html) and change the reference in the frame set file (nmfrm.html) for the body file from dump2.html to whatever table is to be displayed when the page first loads (e.g. HOnm.html). On most UNIX systems this can be done from SAS using:

```
   X sed “s/dump2.html/Honm.html/g”
        /external/web/bul/current/nmfrm.html >
        /external/web/bul/current/nm.html ;
   X rm /external/web/bul/current/dump*.html
   X rm /external/web/bul/current/nmfrm.html
```

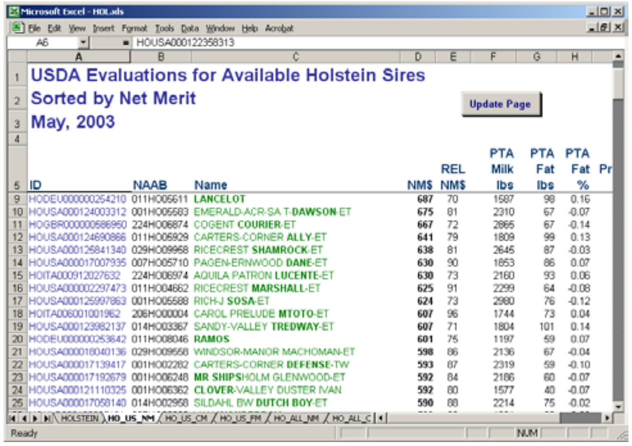
The file that should be linked to in order to access this set of files is now “nm.html”

ALTERNATE OUTPUT FORMATS

EXCEL$^{®}$ or SPREADSHEET

ODS CSV has remained an experimental destination. The files created with this option usually can be as effectively created without ODS and if the intended end use is in a spreadsheet, they ignore most of the available features now found in spreadsheet software. An alternative use ODS HTML, as demonstrated above, and give the files created the .xls extension. Using PROC TEMPLATE, a style definition can be created that produces the desired effect in the spreadsheet created.

Current versions of Excel allow Internet queries to be embedded in a spreadsheet notebook. These scripts go out and retrieve information from an .html page via an Internet connection. It is also possible to add formatting and additional data processing within the spreadsheet. As long as the report format is unchanged, Excel queries can be run at any time to automatically updated a notebook with the latest data.



USDA Evaluations for Available Holstein Sires
Sorted by Net Merit
May, 2003

ID	NAAB	Name	NM$	REL NM$	PTA Milk lbs	PTA Fat lbs	PTA Fat %	Pr
HODEU000000254210	011HO05611	LANCELOT	687	70	1587	98	0.16	
HOUSA000124003312	001HO05583	EMERALD-ACR-SA T-DAWSON-ET	675	81	2310	67	-0.07	
HOGBR000000586950	224HO06874	COGENT COURIER-ET	667	72	2865	67	-0.14	
HOUSA000124690866	011HO05929	CARTERS-CORNER ALLY-ET	641	79	1809	99	0.13	
HOUSA000125841340	029HO09958	RICECREST SHAMROCK-ET	638	81	2645	87	-0.03	
HOUSA000017007935	007HO05710	PAGEN-ERNWOOD DANE-ET	630	90	1853	86	0.07	
HOITA000912027632	224HO06974	AQUILA PATRON LUCENTE-ET	630	73	2160	93	0.06	
HOUSA000002297473	011HO04662	RICECREST MARSHALL-ET	625	91	2299	64	-0.08	
HOUSA000125997863	001HO05588	RICH-J SOSA-ET	624	73	2980	76	-0.12	
HOITA006001001962	206HO00004	CAROL PRELUDE MTOTO-ET	607	96	1744	73	0.04	
HOUSA000123982137	014HO03367	SANDY-VALLEY TREDWAY-ET	607	71	1804	101	0.14	
HODEU000000253642	011HO08046	RAMOS	601	75	1197	59	0.07	
HOUSA000018040136	029HO09558	WINDSOR-MANOR MACHOMAN-ET	598	86	2136	67	-0.04	
HOUSA000017139417	001HO02282	CARTERS-CORNER DEFENSE-TW	593	87	2319	59	-0.10	
HOUSA000017192679	001HO06248	MR SHIPSHOLM GLENWOOD-ET	592	84	2186	60	-0.07	
HOUSA000121110325	001HO06362	CLOVER-VALLEY DUSTER IVAN	592	80	1577	40	-0.07	
HOUSA000017058140	014HO02958	SILDAHL BW DUTCH BOY-ET	590	88	2214	75	-0.02	

This notebook contains data retrieved from a SAS ODS output page. An complete example of this type of interactive notebook can be downloaded from the USDA – AIPL Internet site http://aipl.arsusda.gov/dynamic/sortnew/current/index.html.

RTF or WORDPROCESSOR, and PDF

Another very useful ODS destination if RTF. Rich text format documents can be read by most current word processing software. They provide the benefit of page formatting, which is usually more suitable for printing than HTML, and they are easily edited with a word processor. ODS PDF creates a file using a similar template to ODS RTF that is readable using Adobe Acrobat Reader®.

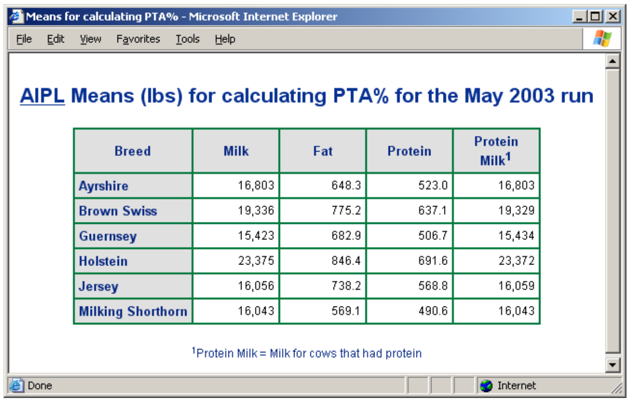
```
        ODS RTF
           FILE=”nm.rtf”
        run;
```

It is possible to open multiple ODS destinations at the same time, creating multiple files from the same procedures. The destinations must be nested (the ODS <destination> CLOSE statements must be in the reverse order from the ODS <destination> OPEN statements). Also, while RTF and PDF have similar attributes, many formatting techniques are specific to the HTML destination. HTML code that is included in variables, titles, or procedures will usually be output as text in the RTF or PDF destination, so you made need separate these procedures.

For HTML documents, it is possible to define an alternative document for printing. Use ODS RTF to create the alternative document, then add a <LINK> tag to the HEADTEXT= option in the ODS HTML statement:

```
  ODS HTML
        BODY=”nm.html”
           HEADTEXT=’ <LINK REF="nm.rtf"
              MEDIA="PRINT" REL="alternate">'
           PATH=”/external/web/bul/current” (url=
              “”)
           STYLE=webdeflt;
   run;
```

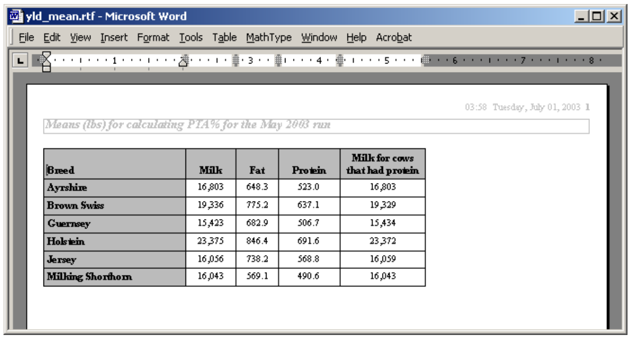
Here is an example of the HTML and RTF documents created by the same PROC PRINT statement. When print is requested while using Internet Explorer to view the top document, whatever software is associated with the RTF format will be used to interpret and print the alternate document format, below.


Means for calculating PTA% - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Done
Internet

AIPL Means (lbs) for calculating PTA% for the May 2003 run

Breed	Milk	Fat	Protein	Protein Milk[1]
Ayrshire	16,803	648.3	523.0	16,803
Brown Swiss	19,336	775.2	637.1	19,329
Guernsey	15,423	682.9	506.7	15,434
Holstein	23,375	846.4	691.6	23,372
Jersey	16,056	738.2	568.8	16,059
Milking Shorthorn	16,043	569.1	490.6	16,043

[1] Protein Milk = Milk for cows that had protein


yld_mean.rtf - Microsoft Word
File Edit View Insert Format Tools Table Window Help Acrobat MathType

03:58 Tuesday, July 01, 2003 1

Means (lbs) for calculating PTA% for the May 2003 run

Breed	Milk	Fat	Protein	Milk for cows that had protein
Ayrshire	16,803	648.3	523.0	16,803
Brown Swiss	19,336	775.2	637.1	19,329
Guernsey	15,423	682.9	506.7	15,434
Holstein	23,375	846.4	691.6	23,372
Jersey	16,056	738.2	568.8	16,059
Milking Shorthorn	16,043	569.1	490.6	16,043

ODS TRACE and ODS SELECT/EXCLUDE

ODS TRACE is a tool for gathering information about SAS output objects. Once the individual objects associated with a procedure are identified, ODS SELECT or EXCLUDE can be used to specify the objects that are included in the output file. Consider PROC REG which has more than one standard output object. By submitting the PROC nested between ODS TRACE ON and ODS TRACE OFF statements, the following information is written to the log:

```
Output Added:
-------------
Name:       ANOVA
Label:      Analysis of Variance
Template:   Stat.REG.ANOVA
Path:       Reg.MODEL1.Fit.PTA_MLK_QTY.ANOVA
-------------

Output Added:
-------------
Name:       FitStatistics
Label:      Fit Statistics
Template:   Stat.REG.FitStatistics
Path:
Reg.MODEL1.Fit.PTA_MLK_QTY.FitStatistics
-------------

Output Added:
-------------
Name:       ParameterEstimates
Label:      Parameter Estimates
Template:   Stat.REG.ParameterEstimates
Path:
Reg.MODEL1.Fit.PTA_MLK_QTY.ParameterEstimates
-------------
```

Each object can be identified by name or label. If a BY statement is included, the Output Added references will be included in the log for each BY Group processed. The name and label will be the same in each group for any object, allowing like objects to be referred to as a group (for example, all the parameter estimates).

The by group name will become part of the path ID, which will be listed for each object. The Path, including the Name is used to identify any individual object.

```
Output Added:
-------------
Name:       ParameterEstimates
Label:      Parameter Estimates
Template:   Stat.REG.ParameterEstimates
Path:
 Reg.ByGroup1.MODEL1.Fit.PTA_MLK_QTY.Parameter
                                     Estimates
-------------
```

To limit output to specific objects, include the ODS SELECT statement before the PROC:

```
ODS SELECT ParameterEstimates;
```

If a BY statement is specified, the above ODS SELECT statement will result in parameter estimates for each BY Group being include in the output file.

```
ODS SELECT
Reg.ByGroup2.MODEL1.Fit.PTA_MLK_QTY.Parameter
Estimates;
```

will result in parameter estimates for only the second BY Group being include in the output file.

ODS EXCLUDE has similar syntax. Using both ODS SELECT and ODS EXCLUDE in the same program can create unexpected results. ODS SELECT ALL resets output selection to the default objects.

SAS/IntrNet

The natural extension to creating scheduled release reports with ODS is to provide dynamic queries that generate reports on demand. This can be done using SAS/IntrNet. Used together, the programming steps described here can be adapted to produce an unlimited variety of results when programs are accessed using the SAS/IntrNet broker.

The basic components to this system are: an HTML form, and a SAS program that generates HTML output, based on the information provided in the form. The HTML form allows the user to submit a customized query through their browser. The information from the form is passed to the SAS system via the SAS broker so that each token becomes a global macro variable. The SAS program to be executed is also embedded in the form. HTML output created by the SAS program is passed back to the user’s browser.

In a simple example, an HTML form allows users to specify columns they would like included in an output report. Submitting the form (using a submit button on the web form) requests execution of an ODS program like the ones described above. The macro variables created from the form can be used in the VAR option of a PRINT procedure. The report created includes only the columns requested. The HTML created is usually not saved as a file. Instead, it is simply passed back to the user’s browser for display.

Providing internet users with more direct access to data and programs can be more efficient and effective than producing static web pages, but creates some additional security issues. For more information on using SAS/IntrNet, consult the SAS OnlineDoc, and the SAS website IntrNet homepage: http://support.sas.com/rnd/web/intrnet/index.html

CONCLUSION

The Internet is the information resource and warehouse of today. Whether reports are provided to the public, the company, or the boss, SAS with ODS HTML will create high quality results with a minimum effort. Further enhancements can be made with tag sets and style sheets. These can greatly reduce the size of output files that contain large tables, reducing download time, but they are not supported by all browsers. This paper provides a sampling of basic tools for producing Internet reports and managing regularly occurring output. The same tools can be used to produce output from statistical procedures for quick review by researchers in different locations by providing them with a web address (URL) for the results. For more information on ODS procedures, visit the ODS community on the SAS website: http://support.sas.com/rnd/base/index-ods-resources.html.

REFERENCES

Haworth, L.E. 2001. Output Delivery System: The Basics. SAS Inst., Cary, NC.

Pass, R. 2001. I’ll have the tabulate a la ODS please, with a table of contents on the side. Page 105 *in* Proc. Northeast SAS Users Group, Baltimore, MD.

CONTACT INFORMATION

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

Ashley H. Sanders
Animal Improvement Programs Lab - USDA
Rm 306, Bldg 005, BARC-West
10300 Baltimore Ave
Beltsville, MD 20705-2350
Work Phone: (301) 504-8667
Fax: (301) 504-8092
Email: asanders@aipl.arsusda.gov
Web: http://aipl.arsusda.gov