Paper 186

SAS® SOFTWARE AND THE WORLD WIDE WEB: SIMPLIFIED

Faith R. Sloan, Intranet Architect
FRS Associates, LLC
San Francisco, CA 94114-1987

Introduction

Corporate use of the Internet has grown rapidly in the past few years. With that growth has come a demand for applications and databases that both internal and external intranet users can access easily and securely.

It is now quite possible to develop a functional Web site, complete with database applications, on a desktop computer using the SAS System in a matter of hours or a few days -- and often with only a small financial investment.

The World Wide Web allows you to not only access textual information but to also access pictures, sounds, or videos from across the world.

Here's how you can leverage The SAS Software Internet tools to save time, labor, and material costs. This is the true meaning of Rapid Applications Development (RAD).

Hypertext Markup Language(HTML)

This section briefly introduces you to the art of creating web pages the old-fashioned way -- by hand. There are many software "tools" that allow you create web pages without touching any HTML. You have to still correct these HTML editors assumptions so you still have to truly become familiar with HTML. Also, if you are serious about doing more than a simple page or two, and want to further your knowledge as well as your market value, a solid footing in the basics will greatly accelerate what you can do.

Simply stated, HTML, or HyperText Markup Language, is a page layout language that tells a web browser how to display a web page. It doesn't even come close to being a programming language. It's rather simple for the most part. The documents themselves are plain text files with special "tags" or codes that a web browser knows how to interpret and display on your screen.

Now that you know what HTML is, let's start using it.

An HTML document contains two distinct parts: the head and the body. The head contains information about the document
that is not displayed on the screen. The body then contains everything else that is displayed as part of the web page.

The basic structure then of any HTML page is:

```
<HTML>
  <HEAD>
  <!-- header info used to contain extra information
about
       this document, not displayed on the page -->
  </HEAD>
  <BODY>
   <!-- all the HTML for display -->
          :     :
          :     :
          :     :
  </BODY>
  </HTML>
```

With this in mind, let's create our first web page using HTML.

```
<HTML>
  <HEAD>
    <TITLE>SAS TIPS and Techniques</TITLE>
  </HEAD>
  <!-- Faith R. Sloan 15July1998 SAS TIPS and
Techniques-->
  <!—Copyright by FRS Associates, LLC 1998 All rights
Reserved-->
  <BODY>
        "SAS Tips are in abundance on the web. You just
        have to find them. Check out <a href=" http:
        //www.frsa.com/sasmass</a> for weekly tips,
        techniques, and more.
  </BODY>
  </HTML>
```

Notice where the <title>...</title> tag is located. It is in the <head>...</head> section and thus will not be visible on the screen. The <title> tag is used to uniquely identify each document and is also displayed in the title bar of the browser window. This is what the web-based search engines will usually pick up to describe the document, so make it self-explanatory.

Also note that there is a comment tag <!-- ....--> that lists the name of the author and the date the document was created. You could write anything in between the comment tags but it is only visible when you view the actual source of the web page.

Now that you've created your first web page, how can you view it?

1. Save the file as index.html
2. Open your web browser
3. Select Open File... from the File menu.
4. Use the dialog box to find and open the file you created, "index.html"
5. You should now see in the title bar of the workspace window the text "SAS Tips and Techniques" and in the web page below, the one sentence of <body> text you wrote, 'SAS Tips are in abundance on the web. You just have to find them. Check out SAS.Masses for weekly tips, techniques, and more.'

You have created your first web page! The use of an HTML anchor <a href= ...> was used in order to create the clickable link attached to SAS.Masses.

There are many web-based tutorials and primers online. One that I highly recommend is The NCSA's Beginner's Guide to HTML at http://www.ncsa.uiuc.edu/General/Internet/WWW/HTMLPrimer.html.

Now, let's proceed with what the SAS System brings to the web-enabled table.

The Web-Enabled SAS System

What exactly is all of this hoopla surrounding the Web-Enabled SAS System? It actually consists of the Web Publishing Tools and the SAS/IntrNet product. The SAS Software System enables us to publish content on the web; distribute reports on the web and to develop dynamic web-based applications.

Web Publishing Tools

The SAS Institute offers many free, downloadable Web Publishing Tools from their web site at http://www.sas.com/web. First there are *HTML Formatting Tools*. They are a collection of SAS macros that enable you to display your SAS data sets and procedure output via the web without you having to learn HTML. There are three HTML Formatting Tools or SAS macros:

- **Data Set Formatter - %ds2htm**
- **Output Formatter - %out2htm**
- **Tabulate Formatter - %tab2htm**

The next three sections will go into more detail as to how these three HTML formatting tools are utilized.

Dataset Formatter - %ds2htm

The dataset formatter macro generates an HTML-formatted document from any SAS dataset.

This macro is powerful in that it supports WHERE clauses, BY-group processing, and other data presentation capabilities.

The following example demonstrates the simplicity in which this macro works:

```
LIBNAME MYLIB "D:\WEBSITE\HTDOCS\SUGI24";
FILENAME OUTIT "D:\WEBSITE\HTDOCS\SUGI24\dataset.html";
OPTIONS LS=80;
%DS2HTM(htmlfref=outit,
        openmode=replace,
        data=MYLIB.resource,
        bgtype=color,
        bg=white,
        brtitle=Subsetted Resource Dataset,
        ctext=red,
        tbbgcolr=yellow,
        clbgcolr=pink,
        csize=+2,
        obsnum=n,
        where=categcde eq ”AA”,
        var=name htmlfile synopsis,
        caption=Resources where Business
Category=”Analyze and Assess);
```

The above code calls the DS2HTM macro and passes a cornucopia of parameters.

The HTML output generated from the macro is written to the filename referenced by OUTIT, D:\WEBSITE\HTDOCS\SUGI24\dataset.html.

The OPENMODE parameter is set to ‘replace’ which ensures that a new file is created anytime the macro is executed. Another option is to use ‘append’ which comes in handy if:

1. You have multiple HTML formatting macros at various stages within your SAS source code and you want to append to the html output file.
2. You want to append to an existing html output file

The SAS dataset that will be formatted is MYLIB.resource.

BGTYPE specifies the type of background for Web page. Since I used COLOR as the value, I must also use the BG argument that I set to white. The text color is specified by CTEXT to be red; the table background color is defined by TBBGCOLR which is yellow; the column header background color is pink; the size of the text is increased relatively by a factor of 2 as specified by CSIZE; and OBSNUM=n ensures that the observation numbers are not displayed. The WHERE parameter allows for subsetting; the VAR attribute specifies the variables values and headers which will be outputted; and finally the CAPTION will be placed at the top of the table.

The resulting output is displayed in the browser as follow:

The SAS System

Resources where Business Category=Analyze and Assess

Name	HTML Link	Synopsis
Analysis for Improving Performance	book1.html	How can you better improve performance within your business unit? Check out Richard A. Swanson’s book in this regard.
The Memory Jogger Plus +	book2.html	Cool book by Michael Brassard as to how to get the cobwebs out of that old cogger of yours!
Need a Laugh??	laugh.html	Need to relieve a bit of stresS? This resource will truly have you in stitches!! Try it out!
Resource Name c da da da	develop2.pdf	yet another stinkin’ model

The HTML generated and written to dataset.html is:

```
<HTML>
<HEAD>
  <META    NAME="GENERATOR"
       CONTENT="SAS Institute Inc. HTML Formatting
Tools, http://www.sas.com/">
  <TITLE>Subsetted Resource Dataset</TITLE>
</HEAD>
<BODY BGCOLOR=white TEXT=red>
<PRE><H3>The SAS System</H3></PRE>
<P>
<TABLE BORDER=1 WIDTH=100% ALIGN=CENTER CELLPADDING=1
CELLSPACING=1 BGCOLOR=yellow>
  <CAPTION ALIGN=CENTER VALIGN=TOP><FONT
SIZE=+2>Resources where Business Category=Analyze and
Assess</FONT></CAPTION>
  <TR>
    <TH BGCOLOR=pink ALIGN=CENTER VALIGN=MIDDLE>Name
</TH>
    <TH BGCOLOR=pink ALIGN=CENTER VALIGN=MIDDLE>HTML
Link    </TH>
    <TH BGCOLOR=pink ALIGN=CENTER
VALIGN=MIDDLE>Synopsis    </TH>
```

```
  </TR>
  <TR>
    <TD ALIGN=CENTER VALIGN=MIDDLE>Analysis for
Improving Performance    </TD>
    <TD ALIGN=CENTER VALIGN=MIDDLE><a
href="../sugi22/book1.html">book1.html</a>    </TD>
    <TD ALIGN=CENTER VALIGN=MIDDLE>How can you better
improve performance within your business unit? Check
out Richard A. Swanson's book in this regard.    </TD>
  </TR>
  <TR>
    <TD ALIGN=CENTER VALIGN=MIDDLE>The Memory Jogger
Plus +    </TD>
    <TD ALIGN=CENTER VALIGN=MIDDLE><a
href="../sugi22/book2.html">book2.html</a>    </TD>
    <TD ALIGN=CENTER VALIGN=MIDDLE>Cool book by Michael
Brassard as to how to get the cobwebs out of that old
cogger of yours!    </TD>
  </TR>
  <TR>
    <TD ALIGN=CENTER VALIGN=MIDDLE>Need a Laugh??
</TD>
    <TD ALIGN=CENTER VALIGN=MIDDLE><a
href="../sugi22/laugh.html">laugh.html</a>    </TD>
    <TD ALIGN=CENTER VALIGN=MIDDLE>Need to relieve a
bit of stresS? This resource will truly have you in
stitches!! Try it out!    </TD>
  </TR>
  <TR>
    <TD ALIGN=CENTER VALIGN=MIDDLE>Resource Name c da
da da    </TD>
    <TD ALIGN=CENTER VALIGN=MIDDLE><a
href="../sugi22/develop2.pdf">develop2.pdf</a>    </TD>
    <TD ALIGN=CENTER VALIGN=MIDDLE>yet another stinkin'
model    </TD>
  </TR>
</TABLE>
<P>
<HR>
</BODY>
</HTML>
```

Output Formatter - %out2htm

The output formatter macro saves output from any SAS procedure to an HTML file. It converts SAS procedure output to HTML tables.

The following example demonstrates how to capture the output from the CONTENTS procedure:

```
LIBNAME MYLIB "D:\WEBSITE\HTDOCS\SUGI24";
FILENAME OUTIT
"D:\WEBSITE\HTDOCS\SUGI24\contents.html";
OPTIONS LS=80;
%OUT2HTM(capture=on,
         window=output,
         runmode=b);
  PROC CONTENTS DATA=MYLIB.RESOURCE;
  RUN;
%OUT2HTM(htmlfref=outit,
         capture=off,
         window=output,
         openmode=replace,
         runmode=b,
         tcolor=red,
         tface="Arial,Helvetica",
         hcolor=green);
RUN;
```

The above code calls the OUT2HTM macro and turns capture mode to 'on' for the output window. What this means is capture the output from the output window of all procedures until we encounter another call to the OUT2HTM macro which has capture mode=off.

The output from the CONTENTS procedure is captured and is output to the filename referenced by OUTIT, D:\WEBSITE\HTDOCS\SUGI24\contents.html.

The OPENMODE parameter is set to 'replace' which operates just as it does with the DS2HTM macro.

The RUNMODE parameter is set to 'b' which means that this macro is being executed in batch mode.

TCOLOR here will render all title lines using the color red while HCOLOR will render all header lines using the color green. TFACE allows us to choose a font face for the title lines. Our first choice is "Arial" and if the user doesn't have that particular font face on their computer, then "Helvetica" is used. If neither of these font faces exists on the user's computer, then the default specified within their browser is used instead.

The resulting output is displayed in the browser as follow:

```
The SAS System                                1
                                                  23:47
Wednesday, July 29, 1998

                              CONTENTS PROCEDURE

Data Set Name: MYLIB.RESOURCE
Observations:         49
Member Type:   DATA
Variables:            5
Engine:        V612
Indexes:              0
Created:       2:04 Monday, January 6, 1997
Observation Length:   380
Last Modified: 0:29 Tuesday, January 7, 1997
Deleted Observations: 0
Protection:
Compressed:           NO
Data Set Type:
Sorted:               NO
Label:

                  -----Engine/Host Dependent
Information-----

                       Data Set Page Size:       11776
                       Number of Data Set Pages: 2
                       File Format:              607
                       First Data Page:          1
                       Max Obs per Page:         30
                       Obs in First Data Page:   28

              -----Alphabetic List of Variables and
Attributes-----

              #    Variable    Type    Len    Pos
Label

ƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒ

              3    CATEGCDE    Char      5     75
Category Code
              5    HTMLFILE    Char    100    280
HTML Link
              1    NAME        Char     60      0
Name
              4    SYNOPSIS    Char    200     80
Synopsis
              2    TYPE        Char     15     60
Type
```

where
The SAS System **1**
23:47
Wednesday, July 29, 1998
is red and the headers such as **CONTENTS PROCEDURE, -----Engine/Host Dependent Information-----**, etc are all in green.

The resulting HTML is as follow:

```
<HTML>
<HEAD>
  <META    NAME="GENERATOR"
        CONTENT="SAS Institute Inc. HTML Formatting
Tools, http://www.sas.com/">
</HEAD>
<BODY>
<PRE><H3><FONT FACE="Arial,Helvetica" COLOR=red>
The SAS System                                1
                                                  23:47
Wednesday, July 29, 1998
</FONT></H3></PRE>
<PRE><STRONG><FONT COLOR=green>
CONTENTS PROCEDURE
</FONT></STRONG></PRE>
<PRE>Data Set Name: MYLIB.RESOURCE
Observations:         49
Member Type:   DATA
Variables:            5
Engine:        V612
Indexes:              0
Created:       2:04 Monday, January 6, 1997
Observation Length:   380
Last Modified: 0:29 Tuesday, January 7, 1997
Deleted Observations: 0
Protection:
Compressed:           NO
Data Set Type:
Sorted:               NO
Label:
</PRE>
<PRE><STRONG><FONT COLOR=green>                    -----
Engine/Host Dependent Information-----
</FONT></STRONG></PRE>
<PRE>
                       Data Set Page Size:       11776
                       Number of Data Set Pages: 2
                       File Format:              607
                       First Data Page:          1
                       Max Obs per Page:         30
                       Obs in First Data Page:   28
</PRE>
<PRE><STRONG><FONT COLOR=green>               -----
Alphabetic List of Variables and Attributes-----

              #    Variable    Type    Len    Pos
Label

ƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒ</F
ONT></STRONG></PRE>
<PRE>         3    CATEGCDE    Char      5     75
Category Code
              5    HTMLFILE    Char    100    280
HTML Link
              1    NAME        Char     60      0
Name
              4    SYNOPSIS    Char    200     80
Synopsis
              2    TYPE        Char     15     60
Type</PRE>
<HR>
</BODY>
</HTML>
```

Notice the font color and font face attributes within the HTML document above.

Tabulate Formatter - %tab2htm

The tabulate formatter macro saves output from the SAS TABULATE procedure to an HTML file.

The following example demonstrates how to capture the output from the TABULATE procedure:

```
LIBNAME MYLIB "D:\WEBSITE\HTDOCS\SUGI24";
FILENAME OUTIT
"D:\WEBSITE\HTDOCS\SUGI24\tabulate.html";
OPTIONS LS=80 NODATE NONUMBER;
PROC SORT DATA=MYLIB.RESOURCE OUT=RES;
  BY CATEGCDE;
RUN;
PROC SORT DATA=MYLIB.CATEGORY OUT=CAT;
  BY CATEGCDE;
RUN;

DATA NEWRES;
  MERGE RES(in=r) CAT; /* Decodes for categories */
  BY CATEGCDE;
  IF R;
RUN;
%TAB2HTM(capture=on,
         runmode=b);
  PROC TABULATE data=NEWRES format=comma10.
formchar='82838485868788898a8b8c'x;
   CLASS categdsc type;
   TABLE type*categdsc;
   LABEL categdsc='Category';
   TITLE "TAB2HTM Example - WUSS 98";
  RUN;
%TAB2HTM(htmlfref=outit,
         capture=off,
         openmode=replace,
         runmode=b,
         clcolor=red,
         twidth=80);
RUN;
```

The above code calls the TAB2HTM macro and turns capture mode to 'on' in the same manner as the OUT2HTM macro works with the exception of the WINDOW parameter.

The output from the TABULATE procedure is captured and is output to the filename referenced by OUTIT, D:\WEBSITE\HTDOCS\SUGI24\tabulate.html.

The OPENMODE and RUNMODE parameters operate in the same manner here as in the OUT2HTM macro as well. The CLCOLOR parameter defines the column color to be red. Finally, TWIDTH defines the width of the HTML table as 80.

The resulting output from the TABULATE procedure is displayed in the browser and the actual generated HTML document, tabulate.html, contains the following:

```
<HTML>
<HEAD>
  <META    NAME="GENERATOR"
        CONTENT="SAS Institute Inc. HTML Formatting
Tools, http://www.sas.com/">
</HEAD>
<BODY>
<PRE><H3>                         TAB2HTM Example -
WUSS 98</H3></PRE>
<PRE><H3></H3></PRE>
<P>
<TABLE BORDER=1 WIDTH=80% ALIGN=CENTER CELLPADDING=1
CELLSPACING=1>
  <TR>
    <TH ALIGN=CENTER VALIGN=BOTTOM COLSPAN=7
NOWRAP><FONT COLOR=red>Type</FONT></TH>
  </TR>
  <TR>
    <TH ALIGN=CENTER VALIGN=BOTTOM COLSPAN=7
NOWRAP><FONT COLOR=red>Books</FONT></TH>
  </TR>
  <TR>
    <TH ALIGN=CENTER VALIGN=BOTTOM COLSPAN=7
NOWRAP><FONT COLOR=red>Category</FONT></TH>
  </TR>
  <TR>
    <TH ALIGN=CENTER VALIGN=BOTTOM NOWRAP><FONT
COLOR=red>Accelerat-<BR>ing<BR>Learning,<BR>Developme-
<BR>nt, &<BR>Performan-<BR>ce</FONT></TH>

    <TH ALIGN=CENTER VALIGN=BOTTOM NOWRAP><FONT
COLOR=red>Achieving<BR>Business<BR>Results</FONT></TH>
```

```
    <TH ALIGN=CENTER VALIGN=BOTTOM NOWRAP><FONT
COLOR=red>Alternati-
<BR>ve<BR>Learning<BR>Solutions</FONT></TH>
    <TH ALIGN=CENTER VALIGN=BOTTOM NOWRAP><FONT
COLOR=red>Analyze/A-<BR>ssess</FONT></TH>
    <TH ALIGN=CENTER VALIGN=BOTTOM NOWRAP><FONT
COLOR=red>Change<BR>Management</FONT></TH>
    <TH ALIGN=CENTER VALIGN=BOTTOM NOWRAP><FONT
COLOR=red>Consult</FONT></TH>
    <TH ALIGN=CENTER VALIGN=BOTTOM NOWRAP><FONT
COLOR=red>Consulting</FONT></TH>
  </TR>
  <TR>
    <TH ALIGN=CENTER VALIGN=BOTTOM NOWRAP><FONT
COLOR=red>N</FONT></TH>
    <TH ALIGN=CENTER VALIGN=BOTTOM NOWRAP><FONT
COLOR=red>N</FONT></TH>
    <TH ALIGN=CENTER VALIGN=BOTTOM NOWRAP><FONT
COLOR=red>N</FONT></TH>
    <TH ALIGN=CENTER VALIGN=BOTTOM NOWRAP><FONT
COLOR=red>N</FONT></TH>
    <TH ALIGN=CENTER VALIGN=BOTTOM NOWRAP><FONT
COLOR=red>N</FONT></TH>
    <TH ALIGN=CENTER VALIGN=BOTTOM NOWRAP><FONT
COLOR=red>N</FONT></TH>
    <TH ALIGN=CENTER VALIGN=BOTTOM NOWRAP><FONT
COLOR=red>N</FONT></TH>
  </TR>
  <TR>
    <TD ALIGN=RIGHT VALIGN=BOTTOM NOWRAP>1</TD>
    <TD ALIGN=RIGHT VALIGN=BOTTOM NOWRAP>1</TD>
    <TD ALIGN=RIGHT VALIGN=BOTTOM NOWRAP>5</TD>
    <TD ALIGN=RIGHT VALIGN=BOTTOM NOWRAP>2</TD>
    <TD ALIGN=RIGHT VALIGN=BOTTOM NOWRAP>1</TD>
    <TD ALIGN=RIGHT VALIGN=BOTTOM NOWRAP>4</TD>
    <TD ALIGN=RIGHT VALIGN=BOTTOM NOWRAP>4</TD>
  </TR>
</TABLE>
<P>
<B><I> (CONTINUED)</I></B><P>
<HR><P>
<PRE><H3>                           TAB2HTM Example -
WUSS 98</H3></PRE>
<PRE><H3></H3></PRE>
<P>
<TABLE BORDER=1 WIDTH=80% ALIGN=CENTER CELLPADDING=1
CELLSPACING=1>
.
.
.
</TABLE>
<P>
<HR><P>
</BODY>
</HTML>
```

HTML Formatting Tools Summary

As you can see, the SAS Institute has created some very powerful but easy to use tools which does not require knowledge of HTML.

In addition, there are many parameters or arguments associated with the HTML formatting macros. These range from general page formatting, color specifications, font specifications to much more. See http://www.sas.com/web for more detail.

More Web Publishing Tools

There is also the *HTML Method for the Table Editor* and the *VRML Browser Tools.*

The HTML method for the Table Editor enables you to transform SAS/AF tables into tables that you can view through a Web browser. It is invoked from SAS/AF compiled source using the CALL NOTIFY statement.

SAS 6.12 also provides tools which will allow you to create static images as well as animated images for the web. There are four drivers, GIF, IMGGIF, and IMGJPEG, GIFANIM which give you the means to create not only single images using but animated gifs by combining two or more images generated by the SAS/GRAPH Software! See http://www.sas.com/web/ while we proceed with a discussion of the SAS/IntrNet product.

SAS/IntrNet

The SAS/IntrNet software allows developers to create interactive web-based applications which gives users the ability to not only access data but to also analyze and display data dynamically. One of the advantages of SAS/IntrNet is that these web-enabled applications allow us to globally distribute complex applications without the need to install the SAS Software on every user's machine!

SAS/IntrNet consists of many components. The rest of this presentation will focus upon one very powerful and practical component – **The Application Dispatcher.**

Make sure you visit http://www.sas.com/web/ for more detailed information on all of the SAS/IntrNet components.

First we will provide some insight into the Common Gateway Interface or CGI.

Common Gateway Interface (CGI)

CGI is ***NOT*** a programming language. It a protocol or way developers can integrate external gateway scripts and programs with the Web. Before CGI came along, we were only able to render plain, *static* HTML documents via the browser. However, a CGI script or program is executed in real-time, so that it can output dynamic information.

The most common languages used by Web developers are Perl and C++. CGI Scripts are also written in UNIX shell, Tcl, VBScript, Visual Basic and a host of other programming and scripting languages.

The Application Dispatcher and htmSQL, SAS/IntrNet components, are Web gateways written by SAS Institute using the Common Gateway Interface (CGI). The power of SAS/IntrNet is that you do not need CGI programming experience in order to use the Application Dispatcher. All you need to do is create the Web user interface and develop the back-end SAS programs! This allows Web developers to minimize development time by leveraging their existing SAS programming experience.

Application Dispatcher

The Application Dispatcher is the component of SAS/IntrNet software that lets you send information from a web browser to a SAS session for processing and returns the results to your browser. The Application Dispatcher has two components:

1. the Application Broker (SAS Institute-supplied CGI program)
2. the Application Server (SAS session)

Here's how they work together.

First, a user submits a request through a Web browser. The Web server receives the request and invokes the Application Broker.

The Application Broker is a CGI program written by the SAS Institute which resides on your web server. It interprets the user request and sends a message to the Application Server via TCP/IP (the Internet Protocol) passing parameters such as which program to execute, which data to use, etc.

A Dispatcher *program* is the actual SAS, SCL, or macro code you write that processes the data. A Dispatcher *application* is the combination of a Dispatcher program (or programs) and the HTML pages that serve as the user interface.

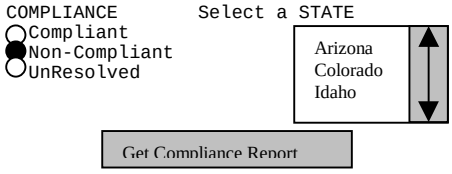
The Application Server is a SAS session that calls the appropriate Dispatcher program from the Application Library defined by the developer. The Dispatcher program may be a SAS program, a macro, or a compiled SCL program. The Dispatcher program runs, and then sends data to the Web server. The data may be an HTML file, plain text, graphics, some other format, or some combination of formats. The Web server sends this data to the user's Web browser, and the transaction is complete.

The following Dispatcher application has 1 HTML page and 2 SAS programs:

Index.html is the main web user interface. It allows the users to select compliance and state parameters which are then passed to the Application Broker, /cgi-bin/broker, which in turn communicates with the Application Server to call a Dispatcher program, GetRpt.sas, which resides in an Application Library pre-defined on the server as *complian*. GetRpt.sas sends the HTML results back to the Web server, which renders the results to the user's Web browser.

```
<HEAD>
<META name="SECURITY" content="CONFIDENTIAL">
<META name="CUID" content="fsloan">
<TITLE>Choose Top-Level Parameters</TITLE>
</HEAD>
<BODY bgcolor="cccfff" TEXT="BLACK" LINK="BLUE"
VLINK="RED">
<FONT FACE="Arial,Helvetica" size="4">
<FORM ACTION="/cgi-bin/broker" METHOD="POST">
<INPUT TYPE="HIDDEN" NAME="_service" value="newsas">
<INPUT TYPE="HIDDEN" NAME="_program"
value="complian.GetRpt.sas">
<INPUT TYPE="HIDDEN" NAME="_debug" value="0">
<CENTER>
<TABLE size="500" border="0" cellpadding="0">
  <tr>
    <td width="100" valign="top">
      <b><font color="red">COMPLIANCE</b></font>
    </td>
<td width="200" valign="top">
       <font color="red"><b>Select a
STATE</b></font><br>
    </td>
  </tr>
  <tr>
    <td valign="top">
      <INPUT type="radio" name="compl"
value="1">Compliant
        <br>
      <INPUT type="radio" name="compl" value="0"
CHECKED>Non-Compliant
        <br>
      <INPUT type="radio" name="compl"
value="2">UnResolved
    </td>
<td width="200" valign="top">
    <center>
  <SELECT name="state" size="3">
   <OPTION value="AZ ">Arizona
   <OPTION value="CO ">Colorado
   <OPTION value="ID ">Idaho
</SELECT>
  </center>
</td>
</tr>
<tr>
  <td colspan="3" align="middle">
    <INPUT type="submit" value="Get Compliance Report">
  </td>
</tr>
</TABLE>
</CENTER>
</FORM>
</FONT>
</BODY>
</HTML>
```

Index.html is presented to the user via the browser close to the following:



When the user selects a compliance level of Non-Compliant, Arizona as the state, and finally clicks on the 'Get Compliance Report' submit button, the HTML Form tag, <FORM ACTION="/cgi-bin/broker" METHOD="POST"> instructs the web server to call the Application Broker, /cgi-bin/broker while passing all of the HTML FORM fields (variables) to this CGI program.

The location of the Application Broker is defined in the ACTION attribute of the HTML FORM tag as /cgi-bin/broker. The data contained in the required field, _PROGRAM, identifies the name of the Dispatcher program, GetRpt.sas. The parameters and their associated values are contained in INPUT and SELECT fields. These fields include compl and state from the <INPUT type="radio" name="compl"...> and <SELECT name="state"...> HTML tags.

The radio button variable compl=0 and the state variable is "AZ". These are not the only 2 parameters being passed to the broker. There are 'hidden' HTML form fields such as _prognam that was mentioned before and _debug. _prognam is used by /cgi-bin/broker to determine which Application Dispatcher program should be executed. In this case, it is GetRpt.sas in the complian Application Library.

Here is GetRpt.sas and an explanation of how it works:

```
%macro GetRpt;
LIBNAME MYLIB "D:\WEBSITE\HTDOCS\SUGI24";
%LET WHERE= (compl=&compl AND state in (&state));
PROC SORT DATA=MYLIB.FINAL OUT=FINAL;
   BY CAT STATE COMPL;
   WHERE &WHERE;
```

```
RUN;
/*=========================================*/
/* Get Counts by measure category and  state */
/*=========================================*/
PROC MEANS NOPRINT DATA=FINAL;
   var compl;
   by cat state;
   class state prod_cd; /* _TYPE_ = 2 is at the prod_cd
or product level */
  output out=stats (Where=(_type_=2)) n=numcompl;
  RUN;
DATA STATS;
  SET STATS;
   LABEL
     NUMCOMPL='TOTAL'
   ;
RUN;
DATA _NULL_;
  CALL symput('c',put(&compl,compl.));
RUN;
TITLE1 "Top Level Summary for &c Measures";
TITLE2 "by Measure Category and State";
%DS2HTM(data=stats,
        var=state,
        id=cat,
        sum=numcompl,
        openmode=replace,
        htmlfref=_webout,
        runmode=s,
        bgtype=color,
        bg=white,center=y,
        twidth=50,
        tcolor=#003333,
        tface=arial,
        tsize=+2,
        clbgcolr=yellow,
        clcolor=brown,
        clface=arial,
        clsize=+1,
        ibgcolr=#ccccff);
%MEND;
%GetRpt
```

First we access the dataset, MYLIB.FINAL and subset it by the passed parameters compl and state, where compl=0 and state="AZ". Then we invoke the MEANS procedure in order to get summary counts.

The **CALL symput('c',put(&compl,compl.));** statement creates a macro variable, &c which will be "Non-Compliant" in this case since &compl=0 and it's format in the permanent SAS format library is "Non-Compliant". It will be used simply to render a self-explanatory title to the user.

Here you will notice that we are using the Dataset Formatting macro of the Web Publishing Tools. Here's where the integration of the Web Publishing Tools and SAS/IntrNet comes into play.

HTMLFREF is set to _webout since all text output produced within an Application Dispatcher program should be written to the Web browser via the fileref _webout.

That's it! When the user clicks on the 'Get Compliance Report' submit button from index.html, the HTML FORM request is sent to the web server with all of its parameters which then causes the broker CGI program, /cgi-bin/broker, to execute. The broker communicates with the Application Server that knows to execute _PROGNAM, GetRpt.sas. The results of GetRpt.sas is finally presented back to the user browser as a summary of how many Arizona entries are Non-Compliant in the MYLIB.final dataset as follows.

```
Top Level Summary for Non-Compliant Measures

       by Measure Category and State

Report Category    State       Total
   Resale         Arizona       26
                  TOTAL         26
```

CONCLUSION

The Web Publishing Tools in conjunction with SAS/IntrNet is a powerful force to contend with. It allows SAS developers to quickly create dynamic, interactive applications throughout an organization or to the public at large without requiring that the viewers have SAS Software installed on their machines.

Check out http://www.sas.com/web to further investigate the many available components with which you can not only generate Web pages that publish your SAS data and output, but can also provide dynamic interaction with the SAS Software from the web.

CONTACT:

FRS Associates, LLC
2750 Market Street, Suite 101
San Francisco, CA 94114-1987
415.626.9796
http://www.frsa.com/, faith@frsa.com

References

SAS Institute Inc,(1998), *"SAS/IntrNet™ Software: Delivering Web Solutions", CD-ROM*

SAS Institute's Web Tools section of their web site - *http://www.sas.com/web/*

Sloan,F.R., (1997), "Developing a PC-SAS® World Wide Web Database System," *Proceedings of the Twenty Second Annual SAS Users Group International Conference.*