ACHIEVING DYNAMIC RESULTS WITH SAS/INTRNET™ SOFTWARE

Mary Singelais, Bell Atlantic, Merrimack, NH

ABSTRACT

Using SAS/Intrnet™ technology, members of the Performance Assurance Organization at Bell Atlantic have developed a dynamic Intranet reporting application, easily accessible to our clients from the World Wide Web. The Intranet application provides Bell Atlantic executives the ability to rapidly access real-time, mission critical data directly from their desktop, and create dynamically driven reports via text hyperlinks or from customized web forms. The reports display graphics, detailed or summarized data, or a combination of these formats for a visually powerful effect. This paper focuses on the continuing development efforts for the dynamic Intranet reporting application at Bell Atlantic, and details many of the SAS elements and key techniques behind the development team’s work.

INTRODUCTION

SAS Institute released SAS/Intrnet™ software two years ago, back in 1997. The new technology provided an alternative solution for creating and distributing business reports dynamically from the web. It gave our project the edge we were looking for, to provide Bell Atlantic managers rapid access to the data they needed most, to manage and analyze the day to day activities impacting the organization.

Graphics, business reports, as well as visually enhanced drill down methods can be produced automatically via the web in the time it takes to point and click on a few buttons. You can even click on a hyperlink and download your data selections to an Excel spreadsheet if you’d like for further analysis. With the introduction of SAS/Intrnet™ software, the ability to dynamically produce visually interesting graphs and other customized reports via the web is a ‘can do’ operation with a win-win message for SAS developers and the clients they support.

PROJECT BACKGROUND

Our dynamic web reporting application at Bell Atlantic has been up and running now for two years, as long as SAS/Intrnet™ software has been available. As the new SAS® technology was being introduced, our development team was actively searching for a new methodology to replace the many manual efforts that were needed to create the variety of detailed business reports prepared every day. The timing fit perfectly. The team recognized the potential impact of the new SAS® technology, and immediately set about acquiring it for our development efforts within the Performance Assurance Organization at Bell Atlantic.

The SAS/Intrnet™ technology provided the platform for delivering a new mechanized reporting application. The new process has eliminated the need for staff to spend hours every day manually creating and distributing reports, often with EXCEL and other labor intensive products. Using SAS/Intrnet™ software, combined with SAS publishing tools for the web, HTML web pages, SAS macro logic, SAS/GRAPH software, SAS reporting procedures, as well as basic SAS software, we have developed a successful dynamic web reporting application. The project has replaced many former manual efforts and has been a welcome change for Bell Atlantic managers.

HOW DOES SAS/INTRNET™ SOFTWARE WORK?

With SAS/Intrnet™ software installed, you have the ability to run SAS programs, or dispatcher applications, directly from your personal web browser on your desktop PC. The report output is routed back to your browser and displayed as professionally as any other web document would look.

The user begins by making selection criteria choices from a web page or HTML form, which has been developed as part of the dynamic web application. The web page or HTML form acts as the starting point to invoke the Application Dispatcher. Either by clicking on a submit button or on a hyperlink, the user submits their
request, and the information is passed to an Application Broker component. The Application Broker software accepts the user’s request and passes it to the Application Server, which executes the named Dispatcher Application program included in the software instructions. Results are routed back through these components to your browser. The whole process works so quickly, that in literally seconds, your results can be displayed back through your browser for your personal review and analysis.

For the SAS programmer, there is virtually no maintenance effort needed for these components beyond installing them. The work effort is mainly behind creating the HTML forms or web pages and the SAS programs they interact with.

As for the users, they may not even know that they’re executing SAS at all. Having a dynamic web application gives the users a point and click environment that needs little technical expertise or training to use. Another benefit for the clients is that SAS/Intrnet™ software operates on a thin client platform. Nothing extra needs to be installed on the client’s own computer. They only need a web browser for access to the application’s web site. This saves the clients the expense of any additional software licenses.

Our experience has been that SAS/Intrnet™ technology works effortlessly and seamlessly to the eye of the end user. Clients have had high praises for the new dynamic web reporting process since its implementation two years ago. In fact, we continue to add many new reports to the web site, as our clients have grown to understand and rely on the resourcefulness, quality, and dependability of the SAS driven intranet reporting application designed for them by our team at Bell Atlantic.

USING WEB PUBLISHING TOOLS

There are three SAS® macro web publishing tools, which can be applied to your SAS report procedures, to enable your program results to be displayed back to your browser. The %OUT2HTM macro is used in conjunction with report procedures, such as PROC PRINT, PROC REPORT, or PROC SQL. The %TAB2HTM macro is used specifically with the PROC TABULATE procedure. And the %DS2HTM macro can be used with any SAS data set to display any of the variables on the data set in a table format in your browser.

All three macros are HTML formatting tools. That is, they generate the HTML code that is needed in order for the results, or output, from your SAS reporting procedures, to be correctly interpreted and displayed by your browser. Often with very little code, you can easily convert a typical piece of SAS code into a web enabled report program.

Each of the HTML formatting tools includes an array of optional parameters that you can use, to add further detail or formatting characteristics to your programs. It’s interesting to see the results of a program after adding a few of these parameters to enhance the overall appearance of the report. You can display data rows in one color and a heading in another color. You can add headings or footnotes, change background colors, fonts, and font sizes for many table elements. Center a report or show it left aligned in the browser. A broad variety of parameters exist that are simple to use and offer a host of dynamic effects. The parameters provide professional looking results to your web output that will definitely be noticed.

In addition to using the macro tools, you can write your own HTML code inside a SAS DATA step. We’ve used this approach a number of times now, to develop more customized and complex reports, with special color coding considerations and formatting requirements. Our results have shown us that HTML code, embedded in a SAS DATA step, provides a creative, flexible, and powerful software combination that can achieve the most demanding report requirements we’ve encountered.

MACRO PROCESSING FOR WEB APPLICATIONS

Macro processing is an integral part for building a successful, dynamic web application, using SAS/Intrnet™ software. Macro processing provides the power and flexibility to process a single SAS program for any number of specific data values that could exist in any variable from an input file or created specifically as part of the application by the developer. To get the most benefit from macro processing, incorporate macro variables, macro functions, and macro statements, into your macro programs, to process and control the desired programming results.

Combining these macro processing elements together allows for an extremely flexible, dynamic, and low maintenance development effort. Developers and managers alike will appreciate the advantage to using macro processing functionality, that can quickly produce
quality, data driven results from streamlined, efficiently written processes.

SAMPLE PROGRAMS

The following programs were developed specifically for execution from a web browser. Key features are highlighted for each program to demonstrate the development efforts behind each of the reports. The reports are part of the Performance Assurance Organization’s Intranet reporting application developed at Bell Atlantic.

The Traffic Light Report program demonstrates:

- Point and click execution from a link
- Embedded HTML inside a SAS DATA step
- Coding techniques for color coding output
- Macro processing functionality

Executing a SAS program from a hyperlink is easily accomplished with SAS/IntrNet™ technology. The developer starts by developing the HTML code for the web page. In that code, the developer defines a dynamic hyperlink with an HREF tag. The HREF tag contains all the information needed to execute the appropriate SAS dispatcher program and any program parameters. Here’s part of the HREF tag for the NY region’s current month Traffic Light Report.

```
<A HREF="../cgi-bin/broker.exe?_service=newsas&_program=programs.mcritxtract.sas&report=CURMONTH&reg_ind=NY">North NY</A>
```

Figure 1. Illustrates the web page, as you would see it in your browser. None of the HTML instructions are obvious to the end user. The browser interprets the developer’s HTML web code.

You can select from two time periods, either Current Month on the left side or Previous Month on the right side of the web page. This report can also be produced for three different regions. Each region has its own hyperlink coded for either time period.

By clicking on the hyperlink for North NY under Current Month, in this example, the user would invoke the Application Dispatcher, and the request to process the named SAS program in the HREF tag is activated. The instructions in the HREF tag also include a parameter named report and another parameter named reg_ind. These parameters were created by the program developer. Report is set to CURMONTH. Reg_ind equals NY. Both are passed through to the SAS program as global macro variables. CURMONTH is used in macro processing statements to select current month records from the input data set. Report can also be set to PREMONTH, which is what is coded behind the links under the Previous Month heading. Reg_ind instructs the SAS program to process only NY records.

The report generated by the SAS program needed to include both monthly and year-to-date totals. To accomplish this, I developed a process to store all the summed data in a permanent SAS data set, using array logic to set up twelve areas for each data field, one for each month. The SAS data set also has fields for holding the year-to-date calculations. Array logic is used for the year-to-date totals also.

Another process was then developed to read the SAS data set that stores all the monthly and year-to-date data. For any given month within the year, the report program can extract the appropriate monthly data and year-to-date totals and display the report in the desired format.

Monthly totals are shown on the left side of the report. Three year-to-date columns make up the right side of the report. I used a SAS DATA step with embedded HTML.
Using embedded HTML gives you a tremendous amount of flexibility to write out your data. The following code from the report program demonstrates SAS DATA step logic with embedded HTML, as well as some color coding techniques, and macro logic to control program output.

```sas
Data _null_;  
set work.mmm yyyy  work.mmxtract end=eof;  
file _webout;  
if _n_ =1 then do;  
        put 'Content-type: Text/html';  
        put;  
        put '<html>';  
        put '<title>Mission Critical Report</title>';  
        put '<body bgcolor=#FFFFFF base="http://yourserver.base.com">';  
/*more SAS code*/

Change the heading in the report using macro logic.

if "NY" in("&reg_ind") then do;  
        put '<h2>Bell Atlantic North / New York<br>';  
        end;  
else if "NE" in("&reg_ind") then do;  
        put '<h2>Bell Atlantic North / New England<br>';  
        end;
```

This section of the SAS DATA step code handles the report and column headings. I've specified a different background color for the column headings to enhance the report appearance. You can control font sizes, colors, text, and other details with embedded HTML. The macro variable named ‘procmon’ will correctly display the requested month selected automatically.

```sas
Put '<table border=1 width="100%" ;
Put '<tr><td align="center" bgcolor=#6699CC colspan=14><strong>'procmon' Monthly Totals</strong></td>
Put '<td align="center" bgcolor=#6699EE colspan=3><strong>YTD Totals</strong></td></tr>
put '<th bgcolor=#99CCCC><font size=-2>Monthly Objective Indicator</font></th>
put '<th bgcolor=#99CCCC><font size=-2>Application</font></th>
put '<th bgcolor=#99CCCC><font size=-2>Appl ID</font></th>
put '<th width=110 align="center" nowrap bgcolor=#99CCCC><font size=-2>Scheduled Hours</font></th>
put '<th bgcolor=#99CCCC><font size=-2>Scheduled Minutes</font></th>
put '<th bgcolor=#99CCCC><font size=-2># Seats</font></th>
put '<th bgcolor=#99CCCC><font size=-2>Scheduled Seat Minutes</font></th>
put '<th bgcolor=#99CCCC><font size=-2># of Outage Incidents</font></th>
/*More headings here*/
```

This section of the SAS DATA step controls row coloring field by field. A test on the data field named obj checks for the value of obj. If obj equals a ‘G’, the report row is color coded green.

```sas
else if "SO" in("&reg_ind") then do;  
        put '<h2>Bell Atlantic South<br>';  
        end;
```

```sas
Put '<table border=1 width="100%" ;
```

/*More headings here*/

```sas
This section of the SAS DATA step controls row coloring field by field. A test on the data field named obj checks for the value of obj. If obj equals a ‘G’, the report row is color coded green.

```sas
if obj = 'G' and not eof then do;
```

```sas
/*green*/
```

```sas
Figure 2. Output from the Traffic Light Report.
```

---

**Figure 2. Output from the Traffic Light Report.**

![Traffic Light Report Output](image-url)
<table>
<thead>
<tr>
<th><strong>obj</strong></th>
<th><strong>appn</strong></th>
<th><strong>applid</strong></th>
<th><strong>port</strong></th>
<th><strong>schours</strong></th>
<th><strong>smin</strong></th>
<th><strong>numseats</strong></th>
<th><strong>sent</strong></th>
<th><strong>ypct%</strong></th>
<th><strong>ysdtm</strong></th>
<th><strong>yseatm</strong></th>
<th><strong>&lt;r&gt;</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

If `obj` equals a ‘Y’, then the row color is changed to yellow.

```sas
/*More SAS code*/
```

```
/*More SAS code*/
```
Using HTML code inside a SAS DATA step, along with macro processing techniques, were keys to developing the Traffic Light Report program. This was once a labor intensive, manually produced and distributed report. With SAS/Intrnet™ software, base SAS, embedded HTML, and SAS macro logic, we developed a very professional looking report that runs dynamically in just a few seconds.

Here’s another report that was developed for the web project.

The Problem Management Dynamic Graphs Report demonstrates:

- Point and Click execution from an HTML form
- Macro processing techniques for enhancing program output
- Creating graphs and reports together in one web page
- Using SAS web publishing tools

You can create an HTML form to allow your clients the flexibility to create unique reports based on their own selection criteria. The HTML form can be designed with radio buttons, drop down lists, data entry fields, and more. These options provide different methods for collecting data choices from your user.

Figure 3. represents a portion of the Problem Management Graphs HTML web page. It uses radio buttons to collect the user’s input request.

In this report, the user requests a time frame, either a bar or a pie chart, and then the locations they want the criteria applied to.

Once the user has made their selections, they click on a submit button provided at the end of the form, and the choices they made are passed as parameters through the SAS/Intrnet™ components. The parameters are then processed as global macro variables by the named SAS program.

This report, like the others that have been developed, evolves around creating output based on specific data criteria entered by the user. Most times there are observations that meet the user’s request, but at times, you may find that there are no observations to construct a report from, and that can cause additional processing needs.

You can create and use a macro variable to check whether there are any observations that met the user’s input requirements, and avoid unwanted results, or error messages, from your SAS processing. Testing for observations will add value to and enhance your reporting process. With a simple SAS DATA step, you can issue a user-friendly message to your clients, if their request produces no observations from the input data set.

Figure 4. is an example of providing an informative message when there is no data to construct a report from.
The Dynamic PCMS Graphs Report produces two outputs combined together into one web page. The graphical output is displayed first, followed by the report output, the result of a SAS PROC TABULATE. By combining graphical and report output in one page, you can achieve a highly visual and detailed presentation of your data. To accomplish this presentation style, we used the following methodology.

First, generate the Content-type header inside a SAS DATA step. The first and second PUT statements are required. Then use a third PUT statement, to code the HTML to point to your hyperlink to the SAS/GRAPH program. The third PUT statement processes the separate graph program and displays the results back to your browser.

```sas
%else %if &graph='B' and &num ne 0 %then %do;

data _null_;  
  file _webout;  
  put 'Content-type: text/html';  
  put;  
  put '<center><img src="http://your.system.com/cgi-bin/broker.exe?_service=newsas&_program=programs.latest-bar.sas&locat= "&locat" "'></center>';  
run;

%tab2htm(capture=on);
proc tabulate data=work.yourfile format=12. missing formchar='82838485868788898a8b8c'x;  
  CLASS locc cause;  
  VAR audt inci_ct;  
  TABLE locc=''*(inci_ct*sum='' audt*sum=''*f=9.2),  
    cause='' / printmiss ;  
  LABEL inci_ct='INCIDENTS'  
    audt='Adj. User Downtime';  
run;

%tab2htm(center=y,capture=off,rlsize=-1,clsize=-1,htmlfref=_webout,encode=n,runmode=s,openmode=append);
%end;
```

The PROC TABULATE code simply follows the RUN statement after the SAS DATA step. The result is one web report with two outputs appended together for a streamlined report presentation.

Figure 5 has highlights from part of the web report created from the Problem Management Dynamic Graphs Report. The bar chart option for New England All Centers was selected. The figure is scrolled to capture the bottom of the graph and the beginning of the PROC TABULATE report.
To display the results of the PROC TABULATE back to the browser, we used the SAS web publishing macro %TAB2HTM. The SAS web publishing tools enable SAS output to be displayed back to your browser and provide options to help format the data.

This statement, %tab2htm(capture=on), begins the HTML Tabulate Formatter. The statement, %tab2htm(capture=off), ends the HTML Tabulate Formatter. Many arguments are available to designate formatting characteristics. For example, we used rlsize=-1 to reduce the size of the row label text by 1 and clsize=-1 to reduce the size of the caption text by 1. These options improve the overall appearance of the report as it appears in the browser.

CONCLUSION

In today’s global market, the growth of the World Wide Web and Internet/Intranet access has created a new mode of communication for corporations and their executives. With the development of SAS/Intrnet™ software, complex business information that once took hours or longer to prepare into useful reports can now be retrieved in seconds.

A key component of any successful business is the ability to make information readily available to sales, marketing, and production managers. Using SAS/Intrnet™ software with other web technology helps provide businesses with new ways to communicate important data in a real-time format for better decision making and business planning.

REFERENCES

SAS is a registered trademark or trademark of SAS Institute Inc. in the USA and other countries.

Other brand and product names are registered trademarks or trademarks of their respective companies.


ACKNOWLEDGEMENTS

This past year was a year of steady growth for the web reporting project. Many people collaborated overall to help make this project very successful. I’d like to thank everyone for his or her support, especially Rob Watters, Bob Deloury, Steve Falconer, Noreen Clyne, Don Blaszka, Jennifer Sylvia, Joan Desmarais, Tim Quill, and Nicole Thompson. Thanks to our Director Deirdre Stevenson and our Executive Director Tom Scurrah for continuing to support a strong partnership with SAS Institute in the workplace.

AUTHOR CONTACT INFORMATION

Mary Singelais
Bell Atlantic
237 Daniel Webster Hwy.
Merrimack, NH 03054
(603) 595-1074
Mary.E.Singelais@BellAtlantic.com