

Methods of Reporting via the Web Using SAS/IntrNet™ Software

Kevin Davidson
FSD Data Services, Inc.

ABSTRACT

This paper will discuss various methods of presenting and formatting data using SAS/IntrNet software. Versions 6, 7 and 8 of SAS® software provide an increasing array of options for creating output. Topics to be covered include the traditional DATA step, the use of the Output Delivery System (ODS) with standard SAS software procedures, SAS/GRAPH® software output, the Xplore application, and the SAS/MDDDB® Report Viewer. Examples are presented which show how to extend your normal SAS software reporting code into dynamic web applications that can be run by anyone with a web browser.

INTRODUCTION

Most if not all SAS developers create reports of one or many types. In the age of the internet it is becoming more and more imperative that information be available on-line. To that end, the purpose of this paper is twofold. First is to show some of the new web enablement tools available with SAS/IntrNet software. Second is to show how to migrate from existing SAS software code to take advantage of these web tools.

In order to show how some of these tools might be used, operations typically performed with SAS/BASE software will be presented. It will then be shown how simple it is to transfer the code and functionality to SAS/IntrNet software in order to view your data dynamically.

The following commonly used reporting tools will be reviewed:

1) Data Step

The PUT statement within a data step allows for a very flexible report to be written to either the log window, output window, or to an external file. There are a number of options available which make it an attractive choice for creating a report, including line and column pointer controls, support for various styles of output (column, list, named, and formatted), and conditional execution capability.

2) Proc Report using ODS

The REPORT procedure is a report-writing tool that combines some of the best features of other SAS tools. Reports can be developed either interactively using menu-driven commands or written from scratch. Variables can be presented in a variety of ways (e.g. all observations displayed, values summarized across common values, distinct values presented across columns, etc.). Computed variables are also supported.

The Output Delivery System first appeared in Version 7. Among the enhancements it brings to reporting are: vastly improved font control, the ability to output procedural results to tables, and support for writing data to external files in specified formats including .rtf and .html files. Prior to version 7, in order to get customized procedural output it was often necessary to use PROC PRINTTO to write to an external file and then read the results back in via a data step. ODS has vastly improved the ability to customize output by allowing for selection of which components to display as well as allowing for control over the display itself. A variety of templates are provided with the software and users can write their own as well.

3) SAS/GRAPH

SAS/GRAPH software can be used to create an endless variety of graphical and/or text based reports. The procedures included with the software can create scatter plots, maps, bar charts, pie charts, etc. In

conjunction with ODS, output can be directed to a web browser and Java or Activex can also be utilized with the inclusion of a couple of simple options.

4) Xplore

The Xplore application is a tool that is included with SAS/IntrNet software. It provides some of the same functionality that the EXPLORER window does in an interactive SAS session plus some other features. You can browse data sets, create reports, and perform drill-down analysis.

5) MDDB Report Viewer

The MDDB Report Viewer is also provided with SAS/IntrNet software. It provides a way to view reports and graphs of summarized data stored in a multidimensional database.

SAS/IntrNet software

SAS/IntrNet software is composed of a number of components including Application Dispatcher, HTMSQL, and the MDDB Report Viewer. An Application Dispatcher server is a SAS session which serves out requests from thin clients and returns results via the browser. Parameters are passed to the Application Dispatcher and can be accessed in the form of macro variables. One of the parameters is '_program' which tells the Application Dispatcher which SAS program to run (usually either a standard .sas program or a compiled macro).

The examples displayed below were developed using SAS/IntrNet software, version 8.1. Note that speed and efficiency were sometimes sacrificed for the sake of clarity.

The data sets utilized in this paper are the sample data sets which in version 8 are installed by default to the \core\sample directory (Windows operating system). Within this paper, assume that a libref of SAMPLE has been assigned to that directory. The sample data sets contain fictitious data about a group of employees for that company.

CONVERTING EXISTING CODE TO SAS/INTRNET CODE

1) Data Step

Presume that on a frequent basis, one of the reports that you find yourself running is a simple report that you create in the log window that shows employees who are in poor health (perhaps you want to send them a get well card). A program that you submit might look like this:

```
Data _null_;
  Set sample.patrec(where=(health='Poor'));
  n+1;
  if n=1 then do;
    put "Poor Health Report for &sysdate";
    Put;
  end;
  put patname;
  put @5 'Send card to: ' address;
  put @5 'Patient # is ' patno;
  put;
```

```
run;
```

The results shown in the SAS log look like this:

Poor Health Report for 05SEP00

Barnes, Art

Send card to: 101 Anywhere St. Cary N.C 27511

Patient # is 1

Daniels, Donna

Send card to: 101 Anywhere St. Cary N.C 27511

Patient # is 3

Jones, Mary

Send card to: 500 Dead Head Rd. Cary, N. C. 27511

Patient # is 18

To submit and display this code using SAS/IntrNet software's Application Dispatcher, you'll need to only add a single line along with some basic HTML tags. A small amount of HTML knowledge is sufficient to produce the report in a similar format. The shaded areas below show changes made to the program.

```
Data _null_;  
  Set sample.patrec(where=(health='Poor'));  
  file _webout; *required;  
  n+1;  
  if n=1 then do;  
    put "Poor Health Report for &sysdate";  
    Put <p>;  
  end;  
  put patname <br>;  
  put @5 'Send card to: ' address <br>;  
  put @5 'Patient # is ' patno <br>;  
  put <p>;  
run;
```

Suppose the above program were saved as "poor_health.sas". There are a number of options available for running the program. For simplicity sake we will show but one option. The following HTML code produces a page with a single button.

```
<html>  
<body>  
<body>  
<form action="http://www.yourcompany.com/bin/broker.exe" method="post">  
<input type="hidden" name="_program" value="weblib.poor_health.sas">  
<input type="hidden" name="_service" value="default">  
<input type="hidden" name="_debug" value="0">  
<input type="submit" value="Continue">  
</form>  
</body>  
</html>
```

The page can of course be expanded to include titles, selection lists, etc. When the 'Continue' button is pressed, the report is Figure 1 is produced.

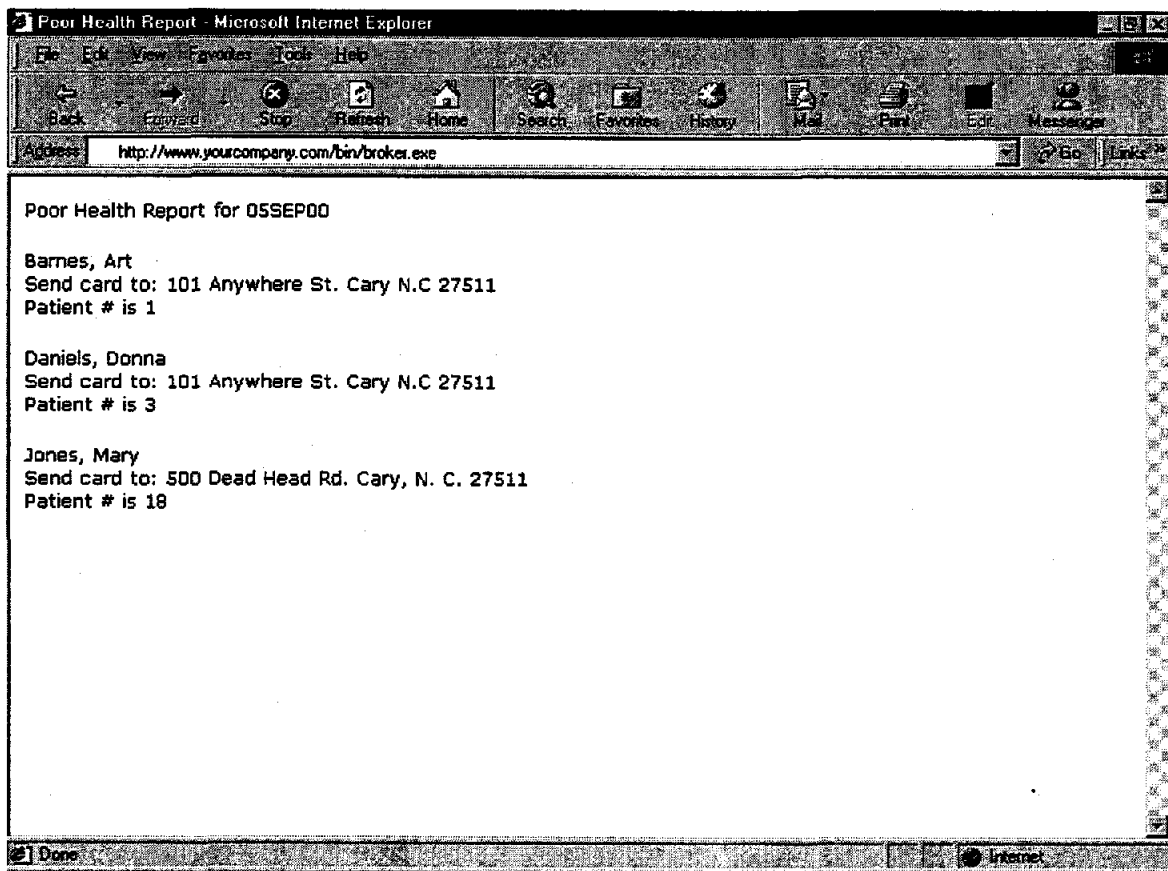


Figure 1. Simple data step output from Application Dispatcher server

2) ODS

Beginning in Version 7, the Output Delivery System allows a much greater control of SAS output. One of the new output destinations is HTML. In addition to being able to create static HTML files, it is also possible to write requested reports dynamically. This example will focus on using ODS with PROC REPORT. The report below produces a report on unpaid invoices. It could be run interactively without the highlighted sections.

```
ods html body= webout;
title;
proc report data=sample.invoice ls=97 ps=500 split="/" center ;
  column ('Outstanding Invoices' ' custname invnum invprice );
  define custname / order format=$10. width=10 spacing=2 left "Customer";
  define invnum / order format=best3. width=9 spacing=2 center "Invoice #";
  define invprice / display format=dollar6. width=8 spacing=2 right "Amount";
  break after custname / skip ;
  compute invprice;
    if invprice > 1000 then call define('invprice','style',style=(background=red));
  endcomp;
run;
ods html close;
```

The highlighted areas show two additions:

- 1) The ODS statements which are required
- 2) There are extensions to ODS which are available in Proc Tabulate and Proc Report. This is but one example of that.

The above code produces the following report:

Customer	Invoice #	Amount
Beach Land	280	\$14
	290	\$19
	300	\$7
Coast Shop	310	
	320	\$6
	330	\$15
	340	\$19
	350	\$6
	360	\$15
	370	\$7
	380	

Figure 2. Proc Report results using ODS and Application Dispatcher server

3) SAS/GRAPH output

The following example simulates a graphical report which shows the average salary within divisions in the company. The Vice President of Human Resources in your company likes to see this report on a monthly basis. In the past you have produced a hard copy report and run it over to her on your lunch hour. You have made the decision to web enable the report, which has the advantage of making it always available and allowing you to spend your lunch hour actually eating lunch.

The following code produces a data set that contains salary information for each employee in 5 divisions of interest. The data set is then used to create a bar chart. Utilizing ODS and Java, you can create an interactive report, including the capability to change the report type on the fly.

```
proc sort data=sample.empinfo out=empinfo(keep=idnum division divcode);
```

```

by idnum;
where divcode in ('CAR' 'CON' 'EDU' 'FAC' 'FIN');
run;

data salary(keep=division divcode salary);
merge empinfo(in=in_empinfo)
      sample.salary(keep=idnum salary enddate where=(enddate=.) in=in_salary);
by idnum;
if in_salary and in_empinfo;
run;

goptions reset=all device=JAVA;
ods html body=_webout parameters=("DISABLEDRILLDOWN"="YES");

axis1 minor=none label=("Division");
axis2 minor=none label=("Salary (Average)") order=(0 to 100000 by 20000) ;
title 'ABC Company - Average Salary by Division';
proc gchart data=salary;
  vbar DIVCODE /
    type=MEAN sumvar=SALARY maxis=axis1
    discrete frame cframe=CXFFFFFFF
    woutline=1 coutline=CX000000 caxis=CX000000
    raxis=axis2 ;
run;
quit;
title;

ods html close;
goptions reset=all;

```

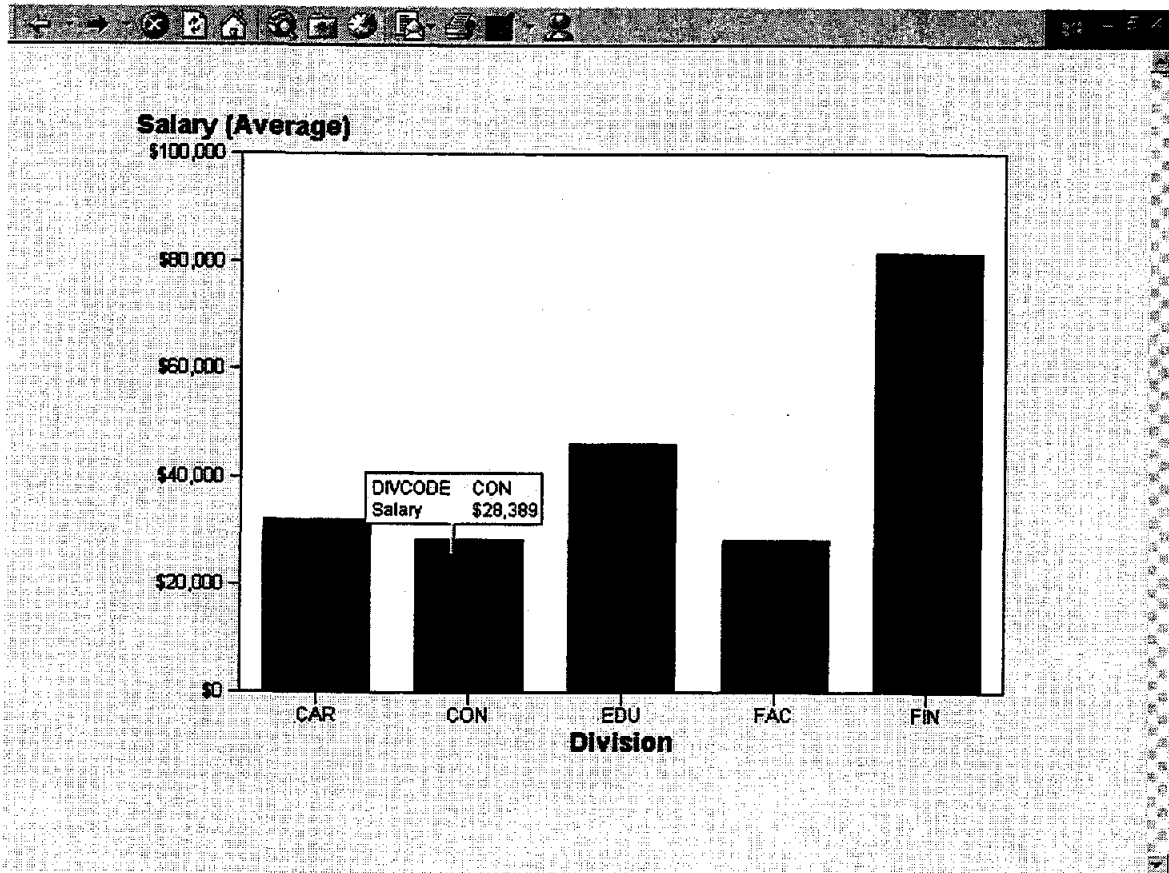


Figure 3. Browser display of PROC GCHART output.

The output in Figure 3 shows results of submitting the program presented above. Note that the popup window showing the average salary of \$28,389 for DIVCODE=CON is produced with a simple pass of the cursor over the bar.

4) Xplore application

As a SAS user, you probably find that browsing your data sets using ViewTable or the SAS Viewer is a necessity. One of the tools built into SAS/IntrNet software is Xplore, which allows you to review your data and catalogs using your web browser. The Xplore application is extremely easy to set up and is even customizable if you are willing to dig into the source code.

The Xplore application setup for this example involved making two small changes to a well documented javascript in the yourserver.yourdomain.com/sasweb/IntrNet8/xplore/webxplib.html file. Namely, the following two options needed to be specified:

- a) `brokername = 'http://www.yourcompany.com/bin/broker.exe';`
- b) `include = new Array('sample');`

The 2nd option limits the libraries displayed to those explicitly listed. You can just as easily exclude certain libraries if that is simpler. There is also an option for defining which member types you want to include ('data', 'catalog', 'view', etc.)

The invocation of the application is done with a URL such as this:

<http://www.yourcompany.com/sasweb/IntrNet8/xplore/webxplore.html>

Going to the above URL presents all the members of the SAMPLE library. The following report shows the results of selecting the EMPINFO data set, a few variables of interest, and choosing to only view the first 50 observations.

The screenshot shows a web browser interface for the Xplore application. On the left, a 'Libraries' pane lists various data sets under the 'SAMPLE' library, including EMPINFO.DATA, EMPLOYEE.DATA, FORMATS.CATALOG, IMAGEDMO.CATALOG, INVOICE.DATA, IPATINT.DATA, JOBCODES.DATA, LEAVE.DATA, LOCKSAMP.CATALOG, MLEAUTO2.CATALOG, PATREC.DATA, PCSAMP.CATALOG, PRODUCT.DATA, PROGRAM.CATALOG, SALARY.DATA, SAMPIMAG.CATALOG, SCREEN.CATALOG, and SLIDE1.DATA. Below this list, a message states: 'This request took 0.94 seconds of real time (v8.1 build 1366)'. On the right, a 'Download As Spreadsheet' button is visible. Below it, a table titled 'Records 1-50 of 309' displays employee data. The table has three columns: 'Identification Number', 'DIVISION', and 'NAME'. The data is as follows:

Identification Number	DIVISION	NAME
333-88-1850	FACILITIES	Alexander, Richard A.
333-88-7366	TECHNICAL SUPPORT	Alford, Betsy S.
301-87-8681	SALES & MARKETING	Amato, Robert G.
333-44-5555	FINANCE	Anaya, John K.
333-78-0101	VIDEO	Armstrong, Armistead W.
566-78-4241	SOFTWARE DEVELOPMENT	Aschenbrenner, Edward T.
739-79-6389	CONTRACTS	Atalla, Ahmed
333-36-6830	SOFTWARE DEVELOPMENT	Autrey, Chris N.
736-15-7096	SOFTWARE DEVELOPMENT	Barker, Louisa A.
265-35-3525	FACILITIES	Barry, Roxanne F.
111-88-7330	TEXAS REGIONAL	Basper, Andrew N.
111-88-7176	TECHNICAL SUPPORT	Beane, Bailey E.
214-01-1720	SOFTWARE DEVELOPMENT	Berg, Stephen M.
737-13-5377	FINANCE	Beaufort, Mary R.
333-88-1981	CONTRACTS	Beekman, Roberta N.
506-09-3698	SOFTWARE DEVELOPMENT	Bessell, Patricia T.
068-30-9977	QUALITY ASSURANCE	Besser, Steven R.
333-88-7063	TEXAS REGIONAL	Best, George L.
769-38-6061	CORPORATE COMMUNICATIONS	Bigson, Debbie R.
730-11-3557	DOCUMENTATION DEVELOPMENT	Billieux, Thomas J.
730-63-8987	HOST SYSTEMS DEVELOPMENT	Bitar, David R.
733-90-1903	SALES & MARKETING	Blumberg, William L.
763-53-1766	SOFTWARE DEVELOPMENT	Boaz, Frederica R.
000-00-1637	QUALITY ASSURANCE	Blue, Kenneth N.

Figure 4. Browser display from the Xplore application.

5) MDDb Report Viewer

If you are a SAS/MDDb user, the SAS MDDb Report Viewer provides you with browser access to your summarized data. The following code uses the same work.salary data set which was produced above to create a very simple MDDb.

```
Libname libmddb 'c:\my_mddb_directory';

proc mddb data=work.salary out=libmddb.salary;
  class division /ascending;
  class begdate /ascformatted;
  var salary / n sum;
  format begdate year4.;
run;
```

There are a couple of ways to view the MDDb data in a web browser. One way is to type the following URL with the attached parameters (the last parameter specifies the MDDb).

http://www.yourcompany.com/bin/broker.exe?_program=sashelp.webeis.rptseld.scl&_service=default&_metabase=libmddb.salary

The above link leads you to the following page:

The screenshot shows a web browser window displaying the SAS MDDb Report Viewer options specification page. The page is divided into two main columns. The left column contains sections for Dimensions (Down, Across), Analysis, Statistics, and Filter Columns. The right column contains sections for Table, Graph, and Filter Listboxes. A 'View Report' button is at the bottom. Below the form, a status message reads: "This request took 0.83 seconds of real time (v8.1 build 1366)".

Section	Option	Value
Dimensions	Down	Division-Year Hierarchy (hier)
	BEGDATE	
	DMSION	
	Across	Division-Year Hierarchy (hier)
Analysis	Salary	
Statistics	Sum	
	Percent of Sum	
	Average	
Filter Columns	BEGDATE	
	DMSION	
Table	Display Table	☒ Yes ☐ No
	Default Title	
Graph	Show Drillpath	☒ Yes ☐ No
	Show Totals	☒ Down ☒ Across
Graph Source	Graph Source	☒ 3D Clickable Graph ☐ Standard GIF Graph
	Location	Bottom
Type	Type	None
	Width	600
Height	Height	450
	Filter Listboxes	
Location	Location	Right
	Width	15
Height	Height	3

View Report

This request took 0.83 seconds of real time (v8.1 build 1366).

Figure 5. SAS MDDb Report Viewer options specification page

A few simple mouse clicks then lead you to your report:



MDDDB Report Viewer: Drill-Down example

Down
Division-Year Hierarchy (hier)
BEGDATE
DMSION

View Report

	Salary
DIVISION	Average
CALIFORNIA REGIONAL	\$32,167
CONTRACTS	\$28,389
EDUCATION	\$46,088
FACILITIES	\$28,405
FINANCE	\$81,500
TOTAL	\$36,722

This request took 1.11 seconds of real time (v8.1 build 1366).

Figure 6. Example of report from SAS MDDDB Report Viewer

As you can see in the above figure customizations to the variables selected can be made even within the report itself, allowing quick investigation of the data. In addition, the data can be downloaded to a spreadsheet with a simple mouse click.

CONCLUSION

SAS/Intrnet software contains a host of features that allow you to extend your working environment past your desktop. For a number of already developed reporting tools created either with a data step or procedural output, you can rapidly extend your existing code for display on the web. In many cases, you can save on distribution costs and time by allowing end users to access up-to-date information when they want it. There is also a wide array of options available which allow the end-user to customize their own report via the browser, thus saving on development time.

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.