Using SAS/IntrNet® to Track the Impact of Welfare Reform in North Carolina

Dean F. Duncan, Ph.D.
Kimberly A. Flair
Rosemary D. Hallberg
Sally S. Muller
Hye-Chung (Monica) Kum
Jordan Institute for Families,
University of North Carolina at Chapel Hill, Chapel Hill, NC

ABSTRACT

In the fall of 1998, the Jordan Institute for Families at the School of Social Work at the University of North Carolina at Chapel Hill (UNC-CH) under contract with the North Carolina Division of Social Services (NCDSS), began developing a web site to disseminate information contained in multiple longitudinal databases of the Work First program in North Carolina. The databases, created in SAS on a UNIX based system, contain data on close to 340,000 families and 800,000 individuals. This paper describes some of the challenges of creating a dynamic and low maintenance web site using the power and versatility of the SAS system. In addition, the authors discuss the transition from SAS 6.12 with SAS/IntrNet® Version 1.2 to SAS 7 with SAS/IntrNet® Version 2.0.

This paper is intended for beginning SAS application developers to help them avoid some problems and build a customized dynamic SAS web site.

INTRODUCTION

Since 1993, the Jordan Institute for Families has collected data on Work First recipients from several state data systems. Each month, data are extracted from state databases and moved by tape to a Unix server at UNC-CH. The data extracts are woven into a longitudinal file that tracks the experiences of families and individuals that have received payments through the Aid to Families with Dependent Children (AFDC) or Work First programs in North Carolina for any month since January 1995. The data in the longitudinal file are used to explore policy options for the NCDSS and to support an evaluation of welfare reform being conducted by Maximus, Inc. The data are also designed to be used to assist counties in developing and using performance measures in managing the Work First program. In North Carolina, as in many other states, human service programs are supervised by the state but administered by counties.

In the past, after the data on each county's performance were analyzed and project personnel presented the results to county and state staff. North Carolina has 100 counties, and presenting the data to each of them in person is not practical. In addition, the project director wanted the data to be available to users in the counties and state on an ongoing basis. Presenting the data over the Internet seemed like a good solution.

Initially we wanted the web site to provide each county access to its own data as well as to state level data.

We considered a number of products for accomplishing this task. We decided to use SAS/IntrNet® for a number of reasons. First, our data was already in SAS data sets. Second, we needed a program that would make updating the pages more efficient. SAS/IntrNet® allowed us to accomplish this by creating pages dynamically. A third reason was that no client software—other than a browser—would be needed to access the information on the site. Finally, the use of SAS macros afforded the opportunity to update multiple programs by simply editing one program. This paper demonstrates some of the strategies we recommend to develop web-based applications and the challenges we encountered.

TECHNICAL COMPONENTS

The entire web site resides on two servers. The initial html page is on an NT server, and the SAS/IntrNet® pages, as well as the data and the SAS/IntrNet® product, are stored on a UNIX based server.

One of the unique features about this web system is that the first page is the only one in HTML. After the user clicks on the county or state from the main page, the selection is passed to subsequent SAS programs through global macro variables called "county" and "label". Not only do these variables determine which title prints on the second page and on the tables and graphs following, but they also dictate which data to pull from the aggregated data sets, consisting of information on over 775,000 individuals. The use of global macro variables and a "config" file, which is included in all the program files (one for each table or graph), allows quick access to the most current data and makes the system extremely easy to maintain. Each month, the web manager simply replaces the data files with the most current data and edits one file ("config") to change the date variables to reflect the new month or quarter of the data. This sample excerpt from "config" shows the variables that are updated each month:

```
%global county;
%let lstupdt= 13JAN00;
%let curmo = 12;
%let curyr = 99;
%let curmmyy = DEC99;
%let curmono = _60;
%let nomo = 60;
%let lstmo = 12;
%let lstyr = 99;
%let Qstart=952;
%let Qend=994;
%let Latest=9912; * must be a multiple
of intval;
%let First=9501;
%let intval=6; * 6 months;
```

This strategy has advantages. First, the small number of pages limits the burden on the servers. Second, SAS allows real-time demonstration of the data; hence, updates are instantly reflected on the web page.

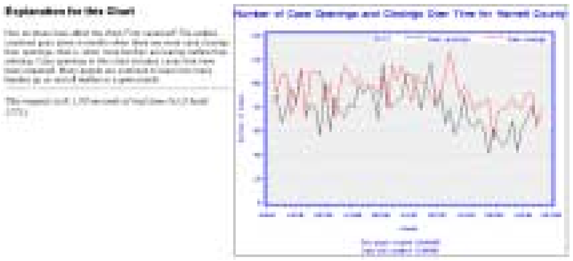
In addition to these efficiency measures, we wanted the web site to be flexible and user-friendly. Other macro variables allow users to create their own graphs and pull specific data rather than limiting them to certain choices. Also, variable names have labels that furnish phrases that are easily understood by the user.

PROBLEMS ENCOUNTERED

Since we present the majority of the data in SAS/GRAPH®, we wanted a way to make the graphs more attractive than the typical sketchy, two-dimensional graphs. Through SAS/GRAPH® options, we were able to change the font, colors, and labeling of the graphs. In the GOPTIONS statement, the device-driver “device=GIF570” allowed us to present the graph in a size easily viewable on a 15-inch monitor and printable on standard size paper.   We also used a macro in the “config” file to help us control the y-axis scaling. The sample code is shown here:

```
%macro setval(fn, maxvar, intval);
  proc means data=&fn max noprint;
  var &maxvar;
  output out=means max=maxnum;
  run;
data _null_;
  set means ;
  n=int(maxnum /(&intval));
  i=0;
  do while (n > 99);
    n=int (n/10);
    i=i+1;
  end;
  if (i=0) and (n>9) then ival =
(int(n/10)+1)*10;
  else ival = (n+1)*(10**i);
  mval=ival*(&intval);
  call symput(’INTVL’,PUT(ival,5.));
  call
symput(’MOSTCASE’,PUT(mval,5.));
%mend setval;
```

Our intended audience consisted of users without statistical backgrounds, so another challenge was to provide information about how to interpret the graphs without forcing the user to click on a Help option. By invoking a text program which calls the graphing SAS program, we were able to present text and graph side by side.


Explanation for the Chart
Number of Case Openings and Closings Over Time for Harnett County

In addition, many of our users wanted to save the tables and graphs to spreadsheet or word processing programs on their PC’s in order to include them in reports or presentations. We originally tested a “Download” button which would allow the user to save the file in an .asp format, but we discovered that this required the user to have write-access to the server, an option that was not acceptable for security reasons. We discovered that simply saving the page as an .xls file allowed the user to create a table in Excel. We included these instructions after the table on the bottom of the page, so the user could see how to save the file. The code that produces this message is contained in the “config” file.

```
%macro excelmsg;
data _null_;
file _webout;
put ’To save this table as an Excel
spreadsheet, click on File from menu
at top of screen,’;
put ’ then click on Save As. Then
choose your directory and’;
put ’ change .html in the file name to
.xls. ’;
put ’Click Save. You can now open your
new file in Excel. Note: This only
works in Office 97.’;
put;
%mend excelmsg;
```

Passing two word county names and the name of the state in the global macros also presented problems. We had to create macros in our “config” file to output these names and select the data correctly. Here is the code to do this:

```
%if "&County"="New Hanover" %then
    %let
Ncnty=New%NRBQuote(%)20Hanover;
  %else  %if "&County"="North
Carolina" %then
    %let
Ncnty=North%NRBQuote(%)20Carolina;
  %else %do;
    %let Ncnty=&County;
  %end;
```

TRANSITION TO SAS/INTRNET® 2.0

Recently we upgraded our SAS products from SAS 6.12 and SAS/IntrNet® 1.2 to SAS 7.0 and SAS/IntrNet® 2.0. This affected our web site, and several adjustments had to be made to maintain the site’s original appearance and functionality. The first included an adjustment of the hpos and vpos options under the GOPTIONS statement. When using device=GIF570 in the GOPTIONS, in Versions 7 and 8, SAS changed the default hpos and vpos to be compatible with UNIX fonts. Our graphs use a SWISSL font, but after the upgrade, the font was out of proportion to the graph. SAS technical support shared their default values of hpos and vpos for the GIF570 device driver in version 6.12 with us.   To restore our original format, we used hpos=83 and vpos=43 in our programs.

Another adjustment came with the variable names and their length. Some data was not printing because SAS 7 accepts long variable names, whereas SAS 6.12 does not. Thus, variables we had recoded were no longer necessary, and we adjusted accordingly.

Our web site is being upgraded to include more data and graphs, and as we add more programs we wanted a way to easily maintain uniformity across pages. By adding macros to our “config” program, which is included in every program, we can define and control the site’s appearance from one file. This will make adding pages much easier in the future, since we will not have to format each one separately.

CONCLUSION

The SAS/IntrNet® product has been an effective way to disseminate our data in a timely fashion, allowing counties to make better informed policy decisions. The ability to dynamically distribute the data through the use of a manageable number of programs has been well received. Just as the users enjoy the timeliness of the information, the developers appreciate the low number of pages and programs that make the web site easy to maintain. We have worked through many technical and data issues and are looking forward to new capabilities.

Our web site is located at: http://ssw.unc.edu/workfirst

CONTACT INFORMATION

We are interested in your comments. You can contact the project's principal investigator at:

Dean Duncan, Ph.D.
Jordan Institute for Families
School of Social Work
University of North Carolina at Chapel Hill
301 Pittsboro St., CB# 3550
Chapel Hill, NC 27599-3550
(919) 962-7897
dfduncan@email.unc.edu