

Using SAS®/IntrNet and the WWW to Develop an Executive Information System - University Application

Michelle Smith, College of Charleston, Charleston, SC

ABSTRACT

With the development of SAS®/IntrNet, the World Wide Web (WWW) is becoming a tool for information retrieval and data dissemination. Graphs, charts, and downloadable spreadsheets are available from a user-friendly HTML form. At the College of Charleston, a need for a usable Executive Information System has been in existence since SAS/EIS® was first demonstrated, but with limited users familiar with SAS® and SAS code, an easier interface was requested. With the new capabilities of SAS/IntrNet®, SAS programmers can now use their programs to report information via the web and users can follow an easier request method.

To take this one step further, the author has also developed a user and password authentication to ensure security to on-campus users through the existing VAX system, and to also allow information retrieval from VAX data files. This was developed using SAS/CONNECT® scripts and requires SAS/CONNECT for VAX/VMS and SAS/CONNECT for SAS 6.12.

This paper is intended for users with intermediate SAS programming experience, some SAS/CONNECT script writing experience, using both SAS 6.09 for VAX/VMS and SAS 6.12 for Windows 95.

INTRODUCTION

With the ease of setting up a personal web server on any machine, Hyper-Text Markup Language (HTML) pages can be written to link with SAS/IntrNet to execute SAS code and return various output. In our university system, we saw this as an opportunity to make available frequently recurring requests, as well as meet the needs of upper-level administrators to make enrollment projections and executive decisions. It was developed in place of an executive information system, since the SAS/EIS applications require each user to install portions of the SAS system on their PC's, and also required that each user had the hardware necessary to run these applications. Contrary to this approach, using SAS/IntrNet and the WWW does not require any special hardware requirements (users can use any existing network connection software to get to a web browser), and made a simple to complete form page for most of the applications.

One added component of these applications is password validation added by using the College's existing VAX users and passwords. This provided the author with some obstacles, however, because not all of the College's users have access to use SAS or have access to our stored data files. Therefore, their original logins were maintained by setting up a dummy account to be used for executing SAS code only. This dummy account is accessed if the correct user and password are entered on the HTML form. Verification of this user and pass combination was obtained by writing a simple script file that submits the user and pass only up until log in success is achieved and is then aborted without opening the SAS system (shown in script1.scr - appendix A). The login.sas macro (appendix B) then determines if the log in messages from the initial script are valid. If so, it proceeds to open the dummy account and execute SAS code with script3.scr (appendix C).

If not, the process is aborted.

Some applications developed by the author and used by College administrators follow.

APPLICATIONS

Example #1: Table retrieved from term-based ASCII data files stored on College's VAX

To show an example of the information distributed over the web, here is one application developed to deliver information on graduate student headcount and FTE.

Web address: <http://irp.cofc.edu/gradoff.htm> (figure 1)

Select term and enter valid username and password for the College's VAX on node "Wando."

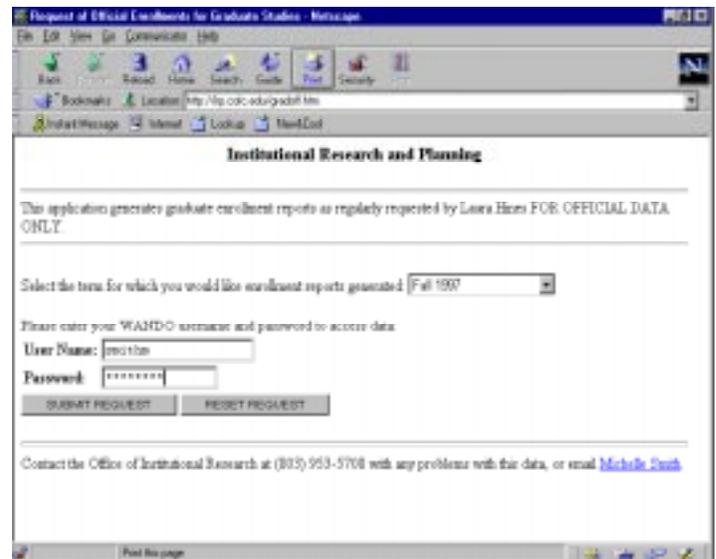


Fig. 1 HTML request page for example #1

This program executes the gradoff.sas (Appendix D) SAS program in PC SAS which in turn, uses SAS/CONNECT to log in to the VAX system, retrieve data, logs out of the VAX, and returns information to PC SAS. The HTML tabulate formatter supplied by SAS Institute turns the SAS tabulate output into HTML code and writes the resulting report to the web browser in an HTML table (figure 2).

College of Charleston
Total Number of Graduate Students
by Degree Type
OFFICIAL 977 (As of 24OCT97)

TYPE/Program

MAJOR DEGREE CLASS	White		Black		Other		Total	
	Men	Women	Men	Women	Men	Women	Men	Women
	EDUC	73	620	15	130	1	5	87
GM	8	60		4	1	2	7	66
MG		1	1					1
ENGL	3	1					3	1
ENVS	3	1	1				4	1
HIST	3	7	1			1	4	8
MATH		1						1
PHED	1	1					1	1

Fig. 2 Results from HTML request in example #1

The resulting information can be written to an HTML file, or displayed with options to download the dataset by clicking on the "Download Data" at the bottom of the procedure output. This dataset is best saved as a comma-delimited text file and then opened as such in a spreadsheet or data base software package.

Example #2: Table displaying data from active student database system

Another aspect of retrieving data from our VAX system is to retrieve dynamic data from the existing student and course databases. The College's database for student and course information is called Student Information Systems© (SIS). To extract data from SIS, a report writing tool called FOCUS© is used to write code and create external reports and data files. Therefore, the SAS script used by the SAS/CONNECT to log in to the VAX is modified to run a FOCUS program through the VAX prompt first, then execute a SAS program using the data file created by FOCUS to produce the resulting HTML output. This is seen as a leap in the direction of making our data more useful and timely, as FOCUS is unable to produce HTML output at this time.

Web address: <http://irp.cofc.edu/seatsav.htm> (figure 3)

Institutional Research and Planning

This application generates state available reports as regularly requested by Undergraduate Studies FOR LIVE DATA ONLY.

Select the term for which you would like the state available reports generated:

Spring 1998

Please enter your WAMCO username and password to access data.

User Name:

Password:

Contact the Office of Institutional Research at (803) 953-5708 with any problems with this data, or email Michelle.Smith

Fig. 3 HTML request page for example #2

AVAILABLE SEATS REPORT
EXCLUDES EVENING AND NORTH AREA COURSES
Undergraduate Courses Scheduled for 981

School of The Arts

Department	Subject	Division	Location/Time		Total Available Seats	Engrpt Max Capacity	Actual Enrollment	Percent Full
			Day	Start Crs				
			Available Seats	Available Seats				
ARTH	ARTH	Entry Level			3	361	338	99.17
		Online			23	295	364	94.10
		Online			29	31	344	113.78
ARTS	ARTS	Entry Level			18	99	81	81.82
		Online			38	345	325	94.81
MUSC	MUSC	Entry Level			88	490	412	83.74
		Online			154	433	437	109.74

Fig. 4 Results from HTML request in example #2

The execution process is essentially the same as in Example #1, except that through SAS/CONNECT script writing, any VAX prompt commands can be executed also (i.e. sending email, file deletions, renaming files).

This VAX command capability through script writing in SAS/CONNECT is also useful in another application by sending an email to notify users that jobs are completed (if dynamic run-time is unacceptable), or in deleting unnecessary files to save disk space.

Example #3: Table displaying data from active student database system with delayed notification

The combination of SAS/CONNECT scripts and VAX command files allow programmers to start the executing of SAS code, and then return the user to the a message page which tells them that their SAS program has been submitted, and they will be sent a message upon completion of the program. This example writes the output to an HTML file, and the user can view that page after they have received the email notification of completion. This solves one of the disadvantages of the SAS/IntrNet and SIS/FOCUS interface of long processing times and does not tie the user's web browser while processing.

Web address: <http://irp.cofc.edu/demoreq.htm> (figure 5)

Institutional Research and Planning

This application generates an updated report of the demographics of all currently enrolled students FOR LIVE DATA ONLY using current term.

Please enter your WAMCO username and password to access data.

User Name:

Password:

Contact the Office of Institutional Research at (803) 953-5708 with any problems with this data, or email Michelle.Smith

Fig. 5 HTML request page for example #3

The user is returned to a verification page (figure 6) and through execution of the command file on the VAX, an email notification is sent to the author and the user.



Fig. 6 Verification of submission of example #3

The email message contains a link to the results page, located at:

<ftp://irp.cofc.edu/currdemo.htm> (figure 7)

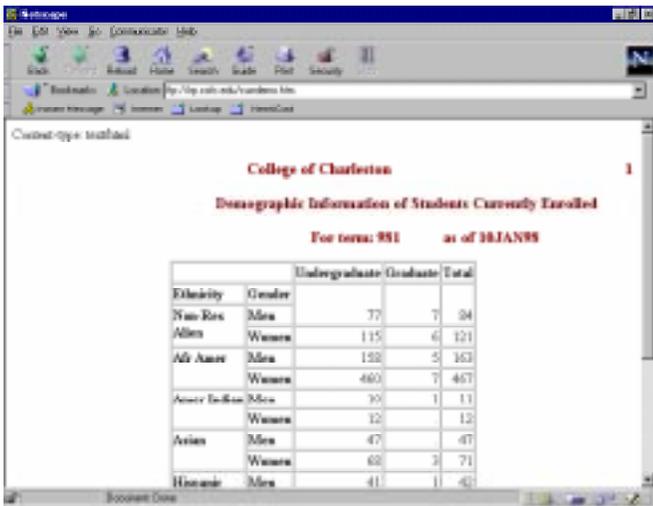


Fig. 7 Results of delayed request from example #3

This HTML file is created on the VAX version of SAS 6.09 using the SAS/IntrNet tabulate formatter and then sent via FTP to the irp.cofc.edu FTP site. This is written in the VAX command file along with the email notification (appendix E).

EXAMPLE #4: Tables generated from a SAS summary data set with drill-down options
 With the purchase of SAS/IntrNet, SAS Institute supplies various samples to be used with web-based applications. One of these is a drill-down demonstration which can be viewed at the SAS Institute's web location (<http://www.sas.com/>).

The request page format was modified so that users are not

limited to default drill-down approaches and can develop their own methods and variables to be used. Since no modifications were made to the Institute-supplied macros, the HTML code is the only code documented here (appendix F).

Web address: <http://irp.cofc.edu/drilldem.htm> (figure 8)

Select which drill-down variables in the desired order. Submit or clear form.

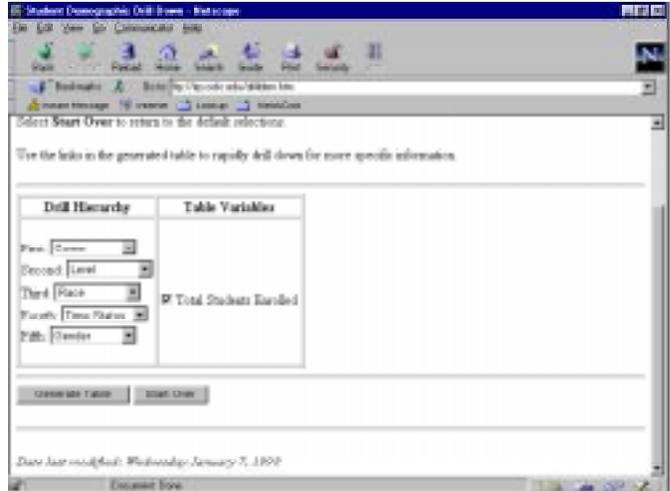


Fig. 8 HTML request page for example #4

Following the default links and selecting new links on each generated page, will result in the following table being finally displayed:

This is one of the most useful tools in management decision making,

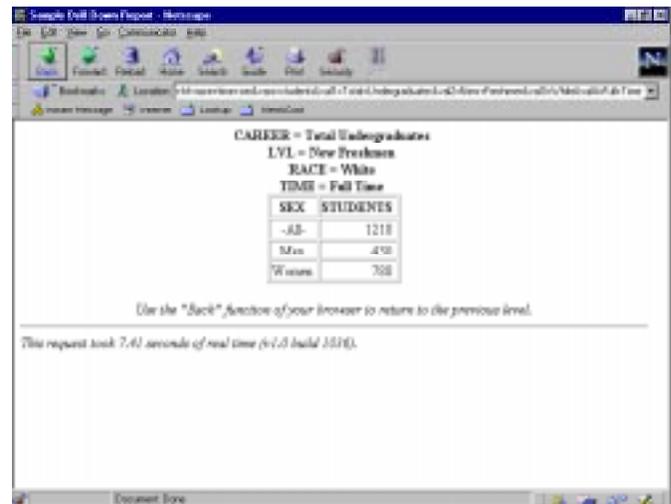


Fig. 9 Results from following drills in example #4

since tables are dynamically selected by the user and customized to their needs.

CONCLUSION

Using this information system as compared to an EIS using SAS/EIS can be summarized below:

ADVANTAGES:

- Security is provided using existing VAX logins.
- Information can be manipulated by the user on the back-end. Reports can be customized so that data re-generation (i.e. retyping data into a spreadsheet) is eliminated.
- Changes to code can be made on the front-end, without user's having to download new programs (as with an EIS) every time modifications are made.
- No additional hardware requirements for those already connected to Internet connections.

DISADVANTAGES:

- Processing time (on some detailed applications, run time exceeds 4-minute limit but can be submitted in batch mode to avoid waiting).
- Non-VAX users cannot access data (could be a security advantage but could also be restrictive).

APPENDIX A - Login Verification Script

```

trace on;
echo on;
/*-----*/
/*-- name:   script1.scr           --*/

/*-- purpose:  SAS/CONNECT SIGNON/SIGNOFF
/**          script for checking for correct WANDO login.
/*-----*/
log "NOTE: Script file 'script1.scr' entered.";
if not tcp then goto notcp;
if signoff then goto signoff;
/*-----VMS LOGON-----*/
**** Send uname and pword from HTML page to WANDO
login;
waitfor 'Username:', 120 seconds : noinit;
type "&uname" CR;
waitfor 'Password:', 60 seconds : nouser;
type "&pword" CR;
scanfor 'Welcome to OpenVMS':chk_fig,
'authorization fail' : nouser,
120 seconds : nostrt;
** Checks to see if uname and pword combination is valid;
** and then logs out without executing SAS;
chk_fig:
waitfor 'VAX V6.2';
waitfor 5 seconds;
log 'MESSAGE: Login is valid...logging out';
abort;
goto signoff;
stop;
/*-----VMS LOGOFF-----*/
signoff:
type 'logout';
stop;

/*----- ERROR ROUTINES -----*/
nouser:
log 'MESSAGE: Invalid USERNAME/PASSWORD
combination.';
waitfor 'failure';
goto nostrt;
abort;
notcp:

```

```

log 'ERROR: Incorrect communications access method.';
log 'NOTE: You must set "OPTIONS COMAMID=TCP;" before using
this';
log ' script file.';
abort;
noinit:
log 'ERROR: Did not understand remote session banner.';
abort;
nostrt:
log 'ERROR: Did not get VMS startup messages after logon.';
abort;

```

APPENDIX B - Login Macro

```

*MACRO - login.sas;
*Author - Michelle Smith;
%macro login(uname,pword,term);
*Macro that checks for users login in Wando, then logs
back out and runs remote SAS account if valid;
filename newlog 'c:\sas\logtest.dat';
*Writes log messages to newlog;
proc printto log=newlog new;
run;
options remote=wando comamid=tcp;
*First script which logs user into wando send login messages
to newlog;
signon 'c:\sas\connect\saslink\script1.scr';
proc printto;
run;
data check;
infile newlog misover;
input word1 $1-8 @;
if word1='MESSAGE:' then do;
input message $10-16;
keep word1 message;
output;
end;
data _null_;
set check;
call symput('errchk',message);
stop;
run;
*If login is not invalid then proceed to login and
execute SAS using dummy account;
%if &errchk ne Invalid %then %do;
signon 'c:\sas\connect\saslink\script3.scr';
%end;
%mend login;

```

APPENDIX C - Login Script for Dummy Account

```

/* trace on;
/* echo on;
/*-----*/
/*-- name:   script3.scr           --*/
/*--          --*/
/*-- purpose:  This script logs into WANDO into the remote
SAS system through a dummy account. --*/
/*-----*/
log "NOTE: Script file 'script3.scr' entered.";
if not tcp then goto notcp;
if signoff then goto signoff;
/*-----VMS LOGON-----*/
**** Logging into RMTSAS account - password protected;
waitfor 'Username:', 120 seconds : noinit;
type "rmtsas" CR;
waitfor 'Password:', 60 seconds : nopass;
type "*****" CR;
scanfor '>' : strt_sas,

```

```

    'Please hit Return or Enter' : mess_log,
    'following disconnected' : recon,
    'authorization failure' : nouser,
    120 seconds : nostrt;
mess_log:
type CR;
waitfor '>' : strt_sas;
stop;
strt_sas:
log 'NOTE: Logged on to VMS.... Starting remote SAS now.';
type
'SAS/DMR/COMAMID=TCP/DEVICE=GRLINK/NOTERMINA
L/NO$SYNTAXCHECK' CR;
waitfor 'SESSION ESTABLISHED', 120 seconds : nosas;
log 'NOTE: SAS/CONNECT conversation established';
stop;
/*-----VMS LOGOFF-----*/
signoff:
type 'logout' CR;
waitfor 'logged out at' , 120 seconds : noterm;
log 'NOTE: SAS/CONNECT conversation terminated.';
stop;

/*----- SUBROUTINES-----*/
recon:
log 'NOTE: Reconnected to previous session. Old SAS
session lost.';
type cr;
waitfor 5 seconds;
type em;
waitfor '>': strt_sas, 10 seconds: nostrt;
/*----- ERROR ROUTINES -----*/
notcp:
log 'ERROR: Incorrect communications access method.';
log 'NOTE: You must set "OPTIONS COMAMID=TCP;"
before using this';
log ' script file.';
abort;
nouser:
log 'ERROR: Invalid USERNAME/PASSWORD
combination.';
abort;
noinit:
log 'ERROR: Did not understand remote session banner.';
abort;
nopass:
log 'ERROR: Did not get password prompt.';
abort;
nostrt:
log 'ERROR: Did not get VMS startup messages after
logon.';
abort;
nosas:
log 'ERROR: Did not get SAS software startup messages.';
abort;
noterm:
log 'WARNING: Did not get messages confirming logoff.';
abort;

```

APPENDIX D - SAS Program from HTML Request

```

/* Program Name: gradoff.sas
   Produces Output as Specified by Graduate Studies
   Written by: Michelle Smith
               Institutional Research and Planning
               College of Charleston
*/;

```

```
libname programs 'c:\sas\intrnet\programs';
```

```

libname sample 'c:\sas\intrnet\sample';
*Log in to WANDO using input of username and password from HTML;
*request page located at http://irp.cofc.edu/gradoff.htm;
%login(&uname,&pwd,&term);
*Pass term entered in HTML form to remote SAS session;
%syslput(newterm,&term);
*Remote submit to VAX version of SAS;
rsubmit;
*VAX commands to list files and their creation dates for report header;
%let source=si$perm;
%let filename=&source:csr028d1_off&newterm..dat;
x dir &filename /brief /noheading /notrailing /output=sicc$fx:filedate.dat;
data datechk;
infile 'sicc$fx:filedate.dat' missover;
input @1 check $1. @;
if check=' ' ;
input date $31-41 @31 fulldate date11.;
if date ne ' ' ;
newdate=put(fulldate,date8.);
call symput('date',newdate);
proc sort; by fulldate;
%sysrput crdate=&date;
*Read in ASCII data file-file layouts changed after 956;
data off;
infile "&source:csr028d1_off&newterm..dat" missover;
input ssn 1-9 sx $50 appterm 271-273 admtype $282-283 satv 290-293
satm 294-297 @;
if &newterm ge 957 then do;
input degree $463-466 class $482-484
entry $485-486 termhrs 454-458 2 career $487 resident $459
rce $51 citizen $52-53 name $10-41 major1 $467-470 major2 $471-474;
end;
else if &newterm lt 957 then do;
input degree $455-458 class $474-476
entry $477-478 termhrs 446-450 2 career $479 resident $451
rce $51 citizen $52-53 name $10-41 major1 $459-462 major2 $463-466;
end;
if career='G';
*Download dataset off to local SAS session to produce report;
proc download data=work.off out=work.enr; run;
*End remote SAS;
endrsubmit;
*Sign off of VAX;
signoff;
*Submit commands to PC SAS;
%let note=OFFICIAL;
options ls=150 ps=50 nocenter nodate nonumber;
proc format;
value typef 1='Nondegree' 2='Deg-Seeking';
value racef 1='White' 2='Black' 3='Other';
value sexf 1='Men' 2='Women';
data enr;
set enr; by ssn;
if first.ssn;
if major1='EDUC' then degree=class;
if degree in ('NDG','UNG') then type=1; else type=2;
if major1='EDUC' then type=1;
if rce='W' then race=1;
else if rce='B' then race=2;
else race=3;
if sx='M' then sex=1;
else sex=2;
fte=termhrs/12;
proc sort; by type race sex;
*Use SAS Institute-supplied tab2htm macro to produce HTML output;
%tab2htm (capture=on);
proc tabulate format=6. missing;
class major1 race sex degree;
tables major1='MAJOR'*degree='DEG or CLASS' all, (race=' ' all)*sex=' ' /
rts=20 misstext=' ' ;

```

```

by type;
format race racef. type typef. sex sexf.;
keylabel n= ' all=Total';
title1 'College of Charleston';
title2 'Total Number of Graduate Students';
title3 'by Degree Type';
title4 "&note &term (As of &crdate)";
footnote1 'Reporting maintained by Institutional Research and
Planning';
footnote2 'Please contact (803) 953-5708 with questions.';
run;
%tab2htm (capture=off, runmode=s, openmode=replace,
htmlhref=_WEBOUT,
    bgtype=color, bg=white, tcolor="#800005", fsize=-1,
center=Y, ftag=BIG TEXT,
    proploc=sashelp.htmlgen.tabprop.slist, ttag=BIG TEXT,
talign=CENTER);
*Create dataset so that user can download data;
data enr;
set enr;
race2=put(race,racef.);
sex2=put(sex,sexf.);
proc summary nway;
class major1 race2 sex2 degree;
var fte termhrs;
output out=sample.gradoff sum=fte hours;
run;
data sample.gradoff(drop=_type_);
set sample.gradoff;
run;
*SAS Institute-supplied websamp catalog used when
downloading sample.gradoff dataset;
    data _null_;
        file _webout;
put '<FORM ACTION="http://153.9.44.11/scripts/broker.exe"
METHOD="POST">';
put '<INPUT TYPE="HIDDEN" NAME="_PROGRAM"
VALUE="sample.websamp.sprdsht.scl">';
put '<INPUT TYPE="HIDDEN" NAME="_SERVICE"
VALUE="newsas">';
put '<INPUT TYPE="HIDDEN" NAME="dataset"
VALUE="sample.gradoff">';
put '<INPUT TYPE="SUBMIT" VALUE="Download Data">';
put '</form>';
run;

```

APPENDIX E - VAX Command File for FTP and Email Notification

```

$! demoftp.com
$!
$! Written by Michelle Smith
$! to generate HTML file and send FTP to irp.cofc.edu
$!
$ to prod
$ SAY= "WRITE SYS$OUTPUT"
$ SAY "Getting the most recent data file....please wait"
$!
$! SAS code that looks for most recent data file
$!
$ sas
    %let term=981;
    %include 'DISKH:[PROD.SISCOG.FEX]getnightlydate.sas';
$!*****
$! Now that SAS is done execute FOCUS
$!
$ DEFINE/NOLOG FOC$DIR1 SI$DATA, SICC$DATA,
HRCC$DATA
$ DEFINE/NOLOG FOC$DIR2 SI$FEX, SICC$FEX,

```

```

HRCC$FEX
$ DEFINE/NOLOG SI$DATA:ITR002.DAT ITRNNN
$ FOCUS/SUBDIR=DELETE
EXEC DISKH:[PROD.SISCOG.FEX]CURRENT2.FEX
FINISH
$ SAY "*****"
$ SAY "Focus exit messages above are normal!!!"
$ SAY "Now creating report and sending email"
$!
$! Now run SAS program that reads FOCUS output and
$! creates HTML file
$!
$ sas
%include 'DISKH:[PROD.SISCOG.FEX]currdemo.sas';
$! Reads users name to send email
$ OPEN/READ MAILADDR SICC$DATA:EMLADDR.DAT
$ READ MAILADDR SENDTO
$ CLOSE MAILADDR
$!
$! Sends output HTML file to irp FTP site
$ ftp irp.cofc.edu /username=anonymous /password=&sendto
    $ send DISKM:[PROD.PRT]currdemo.htm currdemo.htm
$!
$ mail siscc$data:currdemo.res smithm, &SENDTO /subject="Demo
Done"
$ EXIT

```

APPENDIX F - HTML Code for Drill-Down with Variable Choices

```

<HTML>
<HEAD>
<TITLE>Student Demographic Drill Down</TITLE>
</HEAD>
<BODY bgcolor="#ffffff">
<P>To run the drill:
<OL>
<LI>Choose the drill hierarchy you'd like. The drill hierarchy
specifies the order in which the data is displayed.
<LI>Select the variables that you want displayed in the table.
All analysis variables for the table are selected by default.
<LI>Select <b>Generate Table</b> from the bottom of the page.
</OL>
<p>Select <b>Start Over</b> to return to the default selections.
<p>Use the links in the generated table to rapidly drill down
for more specific information.
<FORM ACTION="http://irp.cofc.edu/scripts/broker.exe">
<HR>
<TABLE BORDER>
<TR><TH>Drill Hierarchy</TH>
    <TH>Table Variables</TH>
<TR>
<TD ALIGN=top><BR>
First:
<SELECT NAME="class1" SIZE=1>
<OPTION SELECTED VALUE="career"> Career
<OPTION VALUE="time"> Time Status
<OPTION VALUE="lvl"> Level
<OPTION VALUE="sex"> Gender
<OPTION VALUE="race"> Race
</select><BR>
Second:
<SELECT NAME="class2" SIZE=1>
<OPTION SELECTED VALUE="lvl"> Level
<OPTION VALUE="career"> Career
<OPTION VALUE="time"> Time Status
<OPTION VALUE="sex"> Gender
<OPTION VALUE="race"> Race
</select><BR>
Third:

```

```

<SELECT NAME="class3" SIZE=1>
<OPTION SELECTED VALUE="race"> Race
<OPTION VALUE="lv"> Level
<OPTION VALUE="career"> Career
<OPTION VALUE="time"> Time Status
<OPTION VALUE="sex"> Gender
</select><BR>

```

Fourth:

```

<SELECT NAME="class4" SIZE=1>
<OPTION SELECTED VALUE="time"> Time Status
<OPTION VALUE="lv"> Level
<OPTION VALUE="career"> Career
<OPTION VALUE="race"> Race
<OPTION VALUE="sex"> Gender
</select><BR>

```

Fifth:

```

<SELECT NAME="class5" SIZE=1>
<OPTION SELECTED VALUE="sex"> Gender
<OPTION VALUE="lv"> Level
<OPTION VALUE="career"> Career
<OPTION VALUE="time"> Time Status
<OPTION VALUE="race"> Race
</select><BR>

```

```
<BR>
```

```
<TD align=top>
```

```
<BR><INPUT TYPE="checkbox" NAME="vars"
value="students" CHECKED>Total Students
```

```
Enrolled
```

```
</TD>
```

```
</TABLE>
```

```
<HR>
```

```
<INPUT TYPE="hidden" NAME="dataset"
VALUE="sample.demosum">
```

```
<INPUT TYPE="hidden" NAME="_program"
value="sample.webmacro.drill.macro">
```

```
<INPUT TYPE="hidden" NAME="_service"
VALUE="newsas">
```

```
<INPUT TYPE="hidden" NAME="char" VALUE=" ">
```

```
<INPUT TYPE="hidden" NAME="classlst"
VALUE=&class1&char&class2&char&class3&char&class4&c
har&class5>
```

```
<INPUT TYPE="hidden" NAME="bgtype" VALUE="color">
```

```
<INPUT TYPE="hidden" NAME="bg" VALUE="white">
```

```
<INPUT TYPE="hidden" NAME="title" VALUE="College of
Charleston Student Demographics">
```

```
<INPUT TYPE="submit" VALUE="Generate Table">
```

```
<INPUT TYPE="reset" VALUE="Start Over">
```

```
</FORM>
```

```
</BODY>
```

```
</HTML>
```

ACKNOWLEDGMENTS

SAS, SAS/CONNECT, SAS/EIS, and SAS/IntrNet are registered trademarks or trademarks of SAS Institute Inc. In the USA and other countries. ® indicates a USA registration.

FOCUS, © 1994, Information Builders, Inc.

Netscape Communications, Netscape, Netscape Navigator and the Netscape Communications logo are trademarks of Netscape Communications Corporation.

Student Information Systems, © 1991, 1992, 1993, 1994 Systems and Computer Technology

Michelle L. Eleazer Smith
Institutional Research and Planning
College of Charleston

66 George Street
Charleston, SC 29424
(803) 953-5708
smithm@cofc.edu
http://irp.cofc.edu