

Knit Perl and SAS[®] Software for DIY Web Applications

Philip R Holland, Consultant, Holland Numerics Limited, UK

Abstract

If your organisation develops a web-based SAS[®] application for 30+ users, then the logical choice is to use SAS/IntrNet[™] software for speed, reliability and cost-effectiveness. However, your whole organisation may have fewer than 30 employees, so is there a cost-effective alternative to SAS/IntrNet software which still allows access to SAS-based data?

The answer is yes, and there are a number of possible solutions. For example, by combining a client licence for Base SAS software, a web server application, a simple application dispatcher written in Perl, and a knowledge of HTML, it is possible to write SAS code and HTML to provide drill-down reports on SAS-based data to users with access only to a web browser.

This paper demonstrates some of the techniques available to SAS programmers to generate drill-down applications for web browsers.

HTML Programming

The core of this simple drill-down web application is the HTML FORM, which allows the user to select the data hierarchies to use across the top and down the side of the generated report. The drop-down lists used to subset the data, which will later be automatically generated from the available data, should be hard-coded, or omitted, in the initial HTML code, depending on individual needs.

The 2 hidden INPUT tags, `_program` and `_graphics`, are required by the Perl server application to tell it where the SAS code is located, and whether to generate a text web page, or a single graphics image, respectively. All other INPUT and SELECT tags create SAS macro variables in the generated SAS code.

The initial HTML code will be regenerated by the SAS code in the left-hand column of the 2-column HTML TABLE, with the report placed in the right-hand column.

SAS Programming

The SAS code uses the SAS macro variables, passed to it by the Perl-generated SAS program, to subset and summarise the SAS data, and to regenerate the HTML web page for the next selection.

```
*-----*;  
* Program : Drill8.sas *;  
* Author  : Philip R Holland, Holland Numerics Ltd *;  
* Date    : 13Apr2000 *;  
* Comments: Generate web page with drill-down facility for the *;  
*          SASHELP.PRDSALE SAS v8 dataset: *;  
*          Analysis=ACTUAL *;  
*          Geographic=COUNTRY - U.S.A./GERMANY/CANADA *;  
*          REGION - EAST/WEST *;  
*          Product=DIVISION - CONSUMER/EDUCATION *;  
*          PRODTYPE - OFFICE/FURNITURE *;  
*          PRODUCT - SOFA/BED/TABLE/DESK/CHAIR *;  
*          Date=YEAR - 1993/1994 *;  
*          QUARTER - 1/2/3/4 *;  
*          MONTH - Jan/Feb/.../Nov/Dec *;  
*-----*;  
  
/* Make sure all the expected macro variables exist */  
%global country region division prodtype product year quarter month down  
across title;  
%let analysis=actual;  
  
/* Set the output filename to use and the table colours */  
%let fileref=_webout;  
%let leftcol=ffffc0; /* yellow */  
%let rightcol=c0c0ff; /* blue */  
%let cellcol=ffffff; /* white */  
  
/* Find the path and filename of the physical _WEBOUT file using the  
DICTIONARY views in SASHELP */  
data _null_;  
  set sashelp.vextfl (where=(fileref=upcase("&fileref")));  
  put 'FILENAME=' fileref ', XPATH=' xpath;  
  call symput('extpath',trim(left(xpath)));  
  stop;  
run;  
  
/* Start the "drill" macro definition */  
%macro drill;  
  
/* Assign report title */  
title1 "&title";
```

```

/* Set default values to hierarchies and selections */
%do i=1 %to 8;
  %let c&i=;
%end;
%let flag=0;
%let class1=country;
%let class2=division;
%let class3=year;

/* Update country hierarchy and selection */
%if %substr(&country,1,1) ne %then %do;
  %let c1=country=%str('%&country%');
  %let flag=1;
  %let class1=region;
%end;
%if %substr(&region,1,1) ne %then %do;
  %if &flag=1 %then %do;
    %let c2=%str('%&region%');
  %end;
  %else %do;
    %let c2=%str(region=%'&region%');
  %end;
  %let flag=1;
  %let class1=region;
%end;

/* Update product hierarchy and selection */
%if %substr(&division,1,1) ne %then %do;
  %if &flag=1 %then %do;
    %let c3=%str(and division=%'&division%');
  %end;
  %else %do;
    %let c3=%str(division=%'&division%');
  %end;
  %let flag=1;
  %let class2=prodtype;
%end;
%if %substr(&prodtype,1,1) ne %then %do;
  %if &flag=1 %then %do;
    %let c4=%str(and prodtype=%'&prodtype%');
  %end;
  %else %do;
    %let c4=%str(prodtype=%'&prodtype%');
  %end;
  %let flag=1;
  %let class2=product;
%end;
%if %substr(&product,1,1) ne %then %do;
  %if &flag=1 %then %do;
    %let c5=%str(and product=%'&product%');
  %end;
  %else %do;
    %let c5=%str(product=%'&product%');
  %end;
  %let flag=1;
  %let class2=product;
%end;

```

```

/* Update date hierarchy and selection */
%if %substr(&year,1,1) ne %then %do;
  %if &flag=1 %then %do;
    %let c6=and year=&year;
  %end;
  %else %do;
    %let c6=year=&year;
  %end;
  %let flag=1;
  %let class3=quarter;
%end;
%if %substr(&quarter,1,1) ne %then %do;
  %if &flag=1 %then %do;
    %let c7=and quarter=&quarter;
  %end;
  %else %do;
    %let c7=quarter=&quarter;
  %end;
  %let flag=1;
  %let class3=month;
%end;
%if %substr(&month,1,1) ne %then %do;
  %if &flag=1 %then %do;
    %let c8=%str(and month='%&month%');
  %end;
  %else %do;
    %let c8=%str(month='%&month%');
  %end;
  %let flag=1;
  %let class3=month;
%end;

/* Amalgamate selections and create subset of data */
%let c0=%unquote(&c1 &c2 &c3 &c4 &c5 &c6 &c7 &c8) ;

data selected;
  set sashelp.prdsale
  %if &flag=1 %then %do;
    (where=(
      &c0
    ))
  %end;
;
run;

/* Summarise subset by across and down hierarchies */
proc summary data=selected nway;
  class &&class&down &&class&across;
  var &analysis;
  output out=prdsumm sum=;
run;

```

```

/* Create list of hierarchy values for drop-down selections
on web page and count the values in each hierarchy */
%do i=1 %to 3;
  proc sql;
    create table values&i as
      select distinct
        &&class&i
      from   selected
      order by
        &&class&i
    ;
  quit;

  proc sql;
    reset noprint;
    select count(*) into :nvalues&i from values&i;
    reset print;
  quit;
%end;

/* Write HTML for top of web page */
data _null_;
  file &fileref;
  put '<HTML>';
  put '<HEAD><TITLE>' "&title" '</TITLE></HEAD>';
  put '<BODY>';
  stop;
run;

/* Reassign _WEBOUT to allow text to be appended */
filename &fileref "&extpath" mod;

/* Start outer table and submission form in left table cell */
data _null_;
  file &fileref;
  put '<CENTER><TABLE CELLPADDING=5><TR><TD BGCOLOR="" "&leftcol" "">';
  put '<FORM ACTION="../scripts/new-sas8.pl" METHOD=POST>';
  put '<INPUT TYPE=HIDDEN NAME="_program" VALUE="code.drill8.sas">';
  %if %substr(&country,1,1) ne %then %do;
    put '<INPUT TYPE=HIDDEN NAME=country VALUE="" "&country" "">';
  %end;
  %if %substr(&division,1,1) ne %then %do;
    put '<INPUT TYPE=HIDDEN NAME=division VALUE="" "&division" "">';
  %end;
  %if %substr(&prodtype,1,1) ne %then %do;
    put '<INPUT TYPE=HIDDEN NAME=prodtype VALUE="" "&prodtype" "">';
  %end;
  %if %substr(&year,1,1) ne %then %do;
    put '<INPUT TYPE=HIDDEN NAME=year VALUE="" "&year" "">';
  %end;
  %if %substr(&quarter,1,1) ne %then %do;
    put '<INPUT TYPE=HIDDEN NAME=quarter VALUE="" "&quarter" "">';
  %end;
  put '<BR>Report title?<BR>';
  put '<INPUT TYPE=TEXT NAME=title VALUE="" "&title" "" SIZE=25><BR>';
  put '<HR>';
  stop;
run;

```



```
/* Complete the web page HTML by closing the TABLE and BODY sections */
data _null_;
  file &fileref;
  put '</TD></TR></TABLE></CENTER>';
  put '</BODY>';
  put '</HTML>';
  stop;
run;

/* End the 'drill' macro definition, execute it and clear the output
filename */
%mend drill;

%drill;

filename &fileref clear;
```

Perl Programming and Operational Details

```
print "HTTP/1.0 200 OK\n";
#! d:\perl\bin\perl.exe -w
# File name: c:\httpd\cgi-bin\...
# ...new-sas8.pl
#
# USER Modification begins here
#
# If running on Window, set 1; UNIX, set 0.
$OS_WIN= 1; ## If Windows, set 1.
# Directory for SAS to use for temporary files
$HOME= 'g:\\web_server\\temp';
# File containing libref & directory name pairs
$CONF= 'g:\\web_server\\scripts\\new-sas8-perl.cfg';
# This is the full path name of the SAS System.
$SAS_EXE= 'f:\\progra~1\\sasins~1\\sas\\v8\\sas.exe -nologo';
# SAS options
$OPTIONS= '';
#
# End of USER Modification
#
.....
```

Figure 2. Sample user modifications to the header of the Perl server application

The Perl server application, which was written by Michael Yu, needs to be customised before it can be called from a web page (see Figure 2.):

- `$OS_WIN` is used indicate whether the web server is on a Windows or UNIX platform, as they have different file naming conventions.
- `$HOME` is the server directory which will hold the temporary files created by each web request.
- `$CONF` is the parameter file (see Figure 3.) containing the pointers to directories on the server required by the processing.
- `$SAS_EXE` is the command line used to execute the SAS System on the server.

```

#
# File name: new-sas8-perl.cfg
#
# A line starting with pound sign, '#', or blank is a comment line.
# Typically place this file in the web server machine's CGI-BIN directory
#
# Syntax:
# a-SAS-libref!physical-directory-existing-on-the-Web-server-machine
# Note the use of exclamation mark, '!', as separator

code!g:\web_server\code
web!g:\web_server\wwwroot

```

Figure 3. Sample Perl server application parameter file

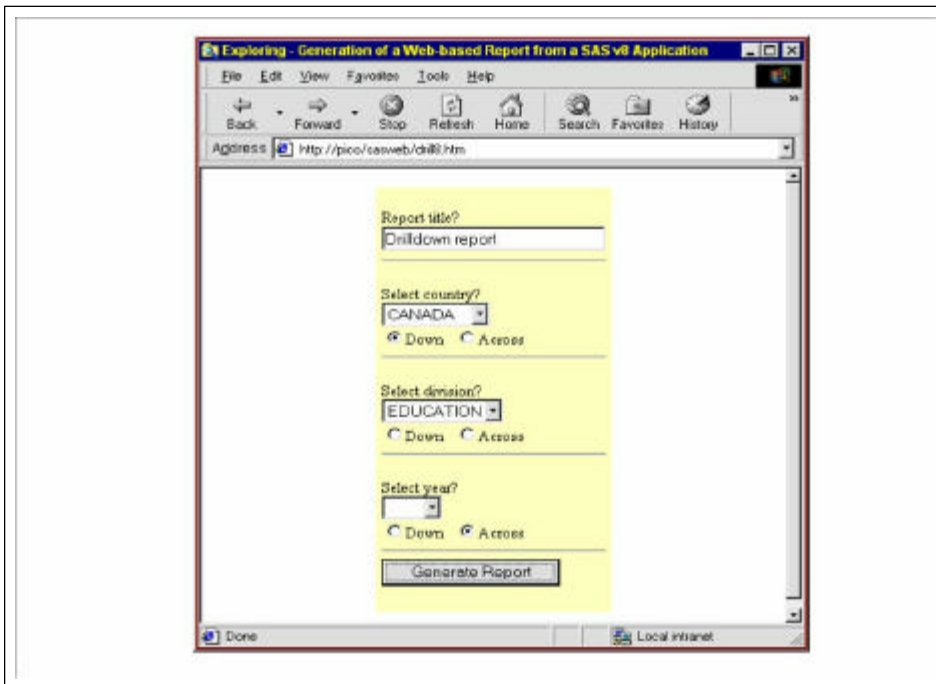


Figure 4. Web page used to generate the initial web request

The initial HTML web page calls the Perl server application used to generate a short SAS program (see Figure 5.), which contains the following:

- SAS macro variables corresponding to the FORM variables from the web page.
- FILENAME statement pointing the `_WEBOUT` file reference to the HTML output file.
- %INCLUDE statement pointing to the SAS program on the server to be executed.

```
%let year= ;
%let title=Drilldown report;
%let country=CANADA;
%let division=EDUCATION;
%let down=1;
%let across=3;
title ; footnote ;
filename _WEBOUT 'g:\web_server\temp\p-2074705.out';
%inc 'g:\web_server\code\drill18.sas';
```

Figure 5. Sample SAS program generated by Perl server application

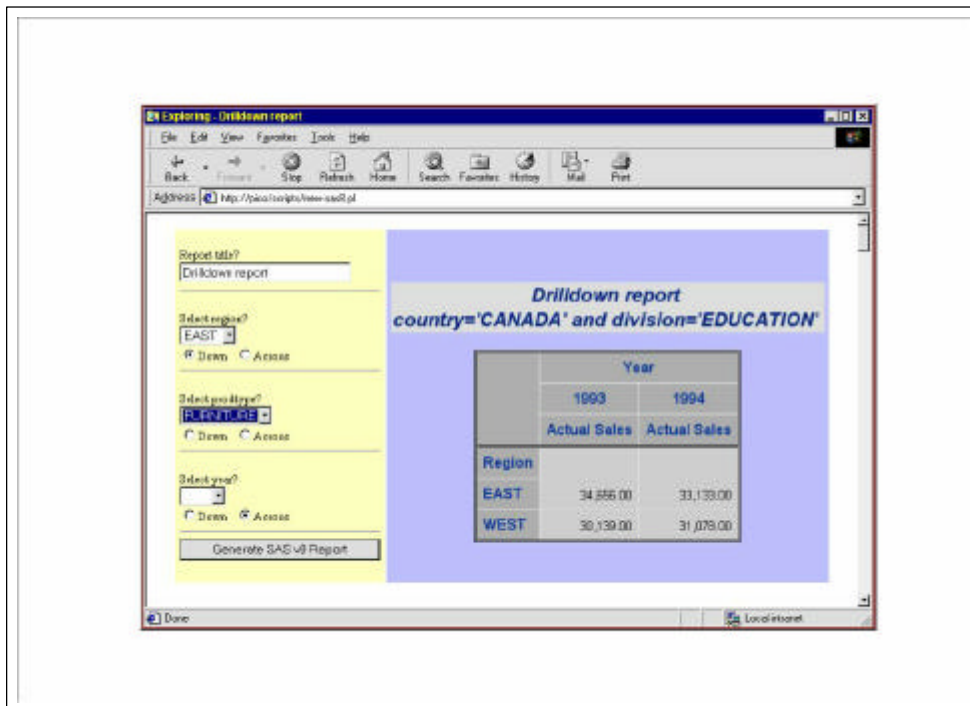


Figure 6. Web page generated using SAS version 8 software

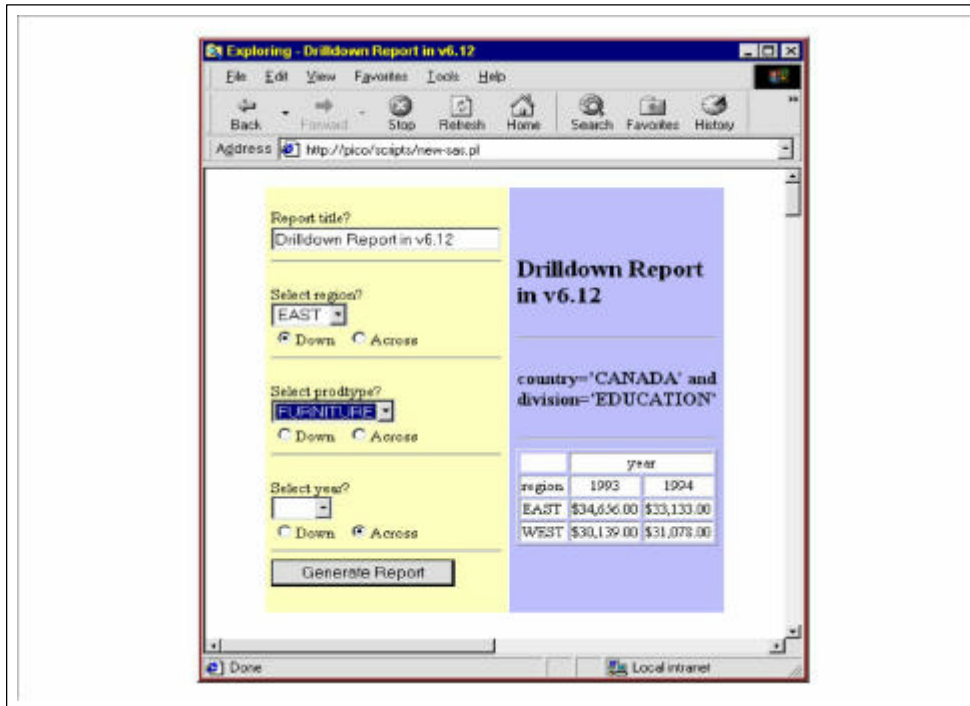


Figure 7. Web page generated using SAS version 6.12 software

System Requirements

Server Requirements

- Windows NT, or UNIX, operating system software running on a server platform with sufficient processing power, memory and disk space to run multiple SAS software sessions.
- Web server software.
- Base SAS version 6.12 or 8 software installed on the web server. Note that SAS version 6.12 code cannot use the ODS statements, so alternative coding will be required, e.g. Web Publishing Tools (macros supplied by the SAS Institute).
- Perl software version 5.002, or above, installed on the web server.

Client Requirements

- Web browser software, e.g. Netscape Navigator 3 or above, Internet Explorer 3.0 or above, or similar software.

References

- 'The Complete Guide to the SAS Output Delivery System', SAS OnlineDoc V8
- Michael Yu, 'Perl to SAS', SUGICD.ZIP (from the SUGI 24 CD-ROM Proceedings)

Contact Information

The author is a consultant for Holland Numerics Ltd and can be contacted at the following address:

Address: Philip R Holland
Holland Numerics Ltd
94 Green Drift
Royston
Herts. SG8 5BT
UK
E-mail: <phil.holland@bcs.org.uk>
Web: <http://www.hollandnumerics.com/>
tel. (mobile): +44-(0)7714-279085

Copies of the code used in the paper can be found on the web site given above.

SAS and SAS/IntrNet are a registered trademarks of SAS Institute Inc., Cary, NC, USA.