Paper AD07

A Quick and Dirty Query System using HTML, Proc SQL, and SAS IntrNet®

Kevin McGowan, Constella Group, Inc. , Durham , NC

ABSTRACT

Every SAS programmer needs ad hoc queries to help with debugging or to give results to others. Usually programmers will either write a new program or take an existing program and modify it to generate the report or data they need. This is time consuming and can lead to errors. The system described in this paper allows any user to query a database or SAS dataset without writing a SAS program. The system uses basic HTML, proc sql, SAS Macros, and SAS IntrNet to provide programmers and end users with an interactive web page that allows them to enter various parameters to query a database or dataset. The output format can be raw data, data listings (by using proc print) or frequency counts (by using proc freq.)

INTRODUCTION

It is not unusual for a SAS user or programmer or a scientist to need to query a dataset or database to get a small amount of data or simple statistics. Frequently this need is similar to previous requests or uses the same data source as a previous request. This paper describes a system that will allow users to get the data or results they need without having to write a new SAS program or even modify an existing SAS program. There are several advantages to using this system:

- The end user does not need to have any knowledge of SAS programming or the data structure. This is especially important in cases where the data is stored in a complex relational database.

- The interface is simple HTML that can be run on any computer on any platform with an Internet browser. (Windows, Macintosh, Linux, etc.) It also doesn't require the use of any special HTML design system; a simple text editor will work fine.

- Since the SAS program is pre-written the possibility of errors is greatly reduced.

- The use of macro code makes the program easier to modify and adapt to other programs or tasks.

- The system can be made as flexible or as simple as the programmer wants.

- The use of SAS IntrNet means that users of the system do not have to install SAS on their local machine.

- The system frees up programmer time to work on more difficult tasks rather than spend time on routine tasks that end users can do themselves.

The HTML code

As mentioned above the HTML code used is not complicated – it uses basic HTML forms to get input from the user and pass it along to the SAS program that resides on the server system.

The HTML code that calls the SAS program is :

<FORM METHOD="POST" ACTION="http://servername/cgi-bin/broker?&_service=pool1&_PROGRAM=asif.smallq.sas&_debug=131">

This is the standard way to call a SAS program from HTML using SAS IntrNet The parts of this code are defined below:

Method – The method the browser sends the data to SAS IntrNet , in this case the post method is used.

Service - The method that SAS IntrNet uses to run the SAS code. Pool1 was created as a pool service when SAS IntrNet was installed. For systems with many users more than 1 pool service can be set up at installation time.

Program – The SAS program on the server that is run.

Debug – An optional parameter that is passed to SAS IntrNet to help with debugging of the SAS code. Using a value of 131 shows the SAS log to the user so it's probably not needed once the code is debugged and working correctly. It is possible to setup the SAS IntrNet server to only accept certain values for the debug parameter. The reason for that would be a case where the programmer does not want the end user to see the log or other information that SAS IntrNet uses to run the program.

The HTML code that lets the user enter the study number and test number is:

Enter the study number <INPUT NAME="ntpstudy" TYPE="text" size=5>
Enter the test number <INPUT NAME="ntptest" TYPE="text" size=2>

(These are standard HTML text input boxes)

The HTML code that lets the user choose which data to query is:

<Select name=wquery multiple size=2>
<option value=Sac>Sacrifice data

<option value=Tumor>Tumor data
</select>

(This is a standard HTML list selection box.)

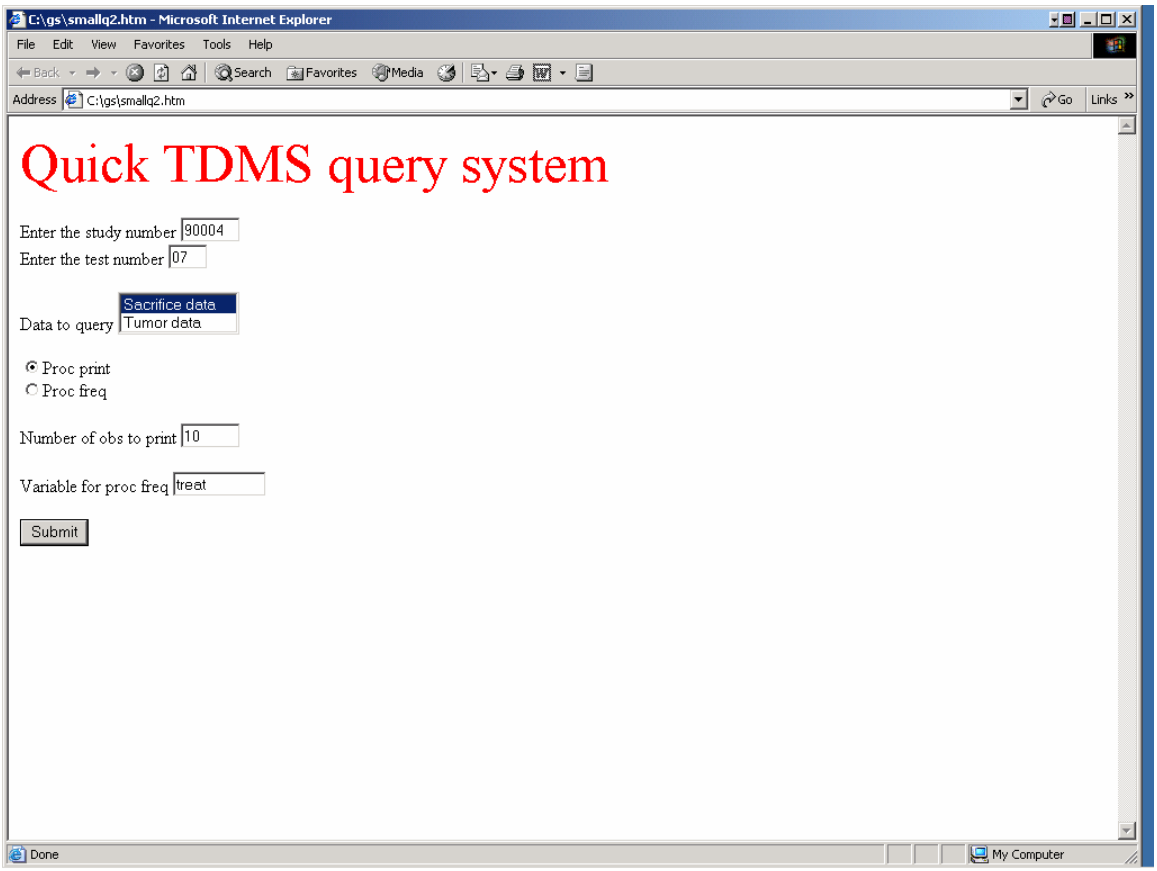
The HTML code that lets the user choose either a proc freq or proc print is:

<INPUT NAME="rtype" TYPE="radio" value="P" checked>Proc print
<INPUT NAME="rtype" TYPE="radio" value="F">Proc freq

(This is a standard HTML radio button Radio buttons are designed so that only 1 of the buttons can be chosen.)

One of the big advantages of using HTML combined with SAS IntrNet is that all the variables that are created using the above input methods are automatically passed to the SAS program as global macro variables. There is no need to use external files or other methods to pass the data to the SAS program. The SAS macro variables have the same name as the HTML variables – in the above example the macro variable for the type of proc to use will be called rtype.

Here is what the HTML code looks like in a browser:



The SAS code

Once the end user has made their choices using the HTML web page SAS IntrNet runs the SAS program that is called using the post method. The macro variables that are created by SAS IntrNet are used to control the flow of execution of the SAS program. The macro variables are also used to pick which data to use and also to subset the data by the study and test number. In order to use the macro variables, the sections of code that are controlled by the user need to be turned into SAS macros.

An example of a proc sql statement that uses the macro variables to subset the data is shown below:

```
proc sql;
        connect to oracle(user=&tdmsuser.  orapw=&tdmspasswd. path=&ntpsid);
        create table wtsex  as
        select * from connection to oracle
         ( select
               animal_num as an_no ,
               r_reference_groups.display_text as insex,
               p_treatments.treat_num as treat
        from
           &ntpdbschema..c_animals,
           &ntpdbschema..r_reference_groups,
           &ntpdbschema..p_experiments,
           &ntpdbschema..p_treatments
         where
             c_animals.RREFG_RAC_ID_SEX=r_reference_groups.rac_id and
             c_animals.pexp_pac_id=p_experiments.pac_id and
             p_treatments.pexp_pac_id=p_experiments.pac_id and
             c_animals.ptre_pac_id=p_treatments.pac_id and
             p_experiments.study_num=&ntpstudy and p_experiments.test_num=&ntptest

            );
        disconnect from oracle;
```

Once this proc sql statement is done executing a SAS dataset named wtsex is created containing the variables an_no, insex, and treat.

(in the above example there are some additional macro variables such as tdmsuser and ntpsid that are not passed into the program from the end user but are stored in an external SAS program that is called at the start of this program.)

The following macro is used to control the flow of the program:

```
%macro print_freq;

  %if  &rtype=P  %then %do;
       proc print;
   %end;

    %else %do;
```

```
      proc freq;
  %end;

    run;

 %mend print_freq;
```

The macro print_freq can be inserted into the program at any point.

By default running a program using SAS IntrNet will not return the standard SAS output to the user. In order to return the output to the user's browser the following commands must be added to the SAS code:

```
      ods html body=_webout(dynamic) rs=none;
```

is needed in the code before any proceduress that produce listing output such as print, freq, and so on.

```
     ods html close;
```

is needed at the end of the procedures that produce output.

One enhancement that can be added to the macro code is to check for the presence of the parameter being passed to the macro. This is a good practice because unlike cases where the macro variables are part of the SAS program, in this case they are entered by the user and therefore can be blank if the user skips that field on the HTML page. This example provides a default value for the program to use to avoid having the program crash in case the user fails to enter the parameter. In this example the proc freq has a tables statement that requires at least 1 variable. If the macro variable fvar is blank a default value of I is used.

```
%macro en_print_freq;

%if &fvar= %then %do;

      %let fvar=i;

    %end;

  %if &rtype=P %then %do;
       proc print;
   %end;

    %else %do;
       proc freq;
         tables &fvar;
```

```
  %end;

   run;

%mend en_print_freq;
```

Alternately, the macro can be designed not to run the proc freq at all if the there is no value entered for the variable to run the freq on:

```
%macro xprint_freq;

  %if &rtype=P %then %do;
     proc print;
  %end;

  %else %do;

     %if &fvar ne  %then %do;
        proc freq;
       tables &fvar;
     %end;

   %end;
   run;

  %mend xprint_freq;
```

It's also possible to check for a blank parameter value in the HTML code rather than in the SAS code, but that would make the HTML code more complex than it needs to be.

CONCLUSION

This system shows that it doesn't take a lot of skill in HTML or SAS to create a user friendly program to allow end users to run SAS programs to perform routine tasks such as data listings or frequency counts. The HTML used here can be learned in just a few hours. This kind of system can provide quick feedback to end users while freeing up programmers to work on more complex programs. Organizations without SAS IntrNet can also use HTML to develop a similar system that uses a regular desktop version of SAS to run the programs. In that case more programming will be needed to pass the macro variables to SAS.

ACKNOWLEDGEMENTS

I would like to thank Pat Crockett of Constella Group for advice on this project and I would also like to thank my wife Lib for help with editing.

This research was supported by the National Institute of Environmental Health Sciences under contract number GS-10F-0351K

CONTACT INFORMATION

Kevin McGowan
Constella Group
2605 Meridian Parkway Durham, NC 27713
Phone: (919) 313-7554
Email: kmcgowan@constellagroup.com
Web: www.constellagroup.com