Paper 71-25

Scrolling & Downloading Web Results

Ming C. Lee, Trilogy Consulting, Denver, CO

Abstract

Since the inception of the **INTERNET** and Web Browsers, the need for speedy information to make split second business decisions has become part of our daily lives. However, information retrieved from the Web can be overwhelming at times due to the immense size of the data, which is only the beginning of the problem. Web Browsers allow users to access information with ease; however, the number of records a user is able to browse at one time is limited to the amount of memory and processor on their computer. One of the many ways to circumvent the problem is to upgrade their computer with a faster processor and more memory, which is expensive. Therefore, it would be easier to control the number of records the user can view at any one time, and provide an option to allow them to download all of the data if required. This will eliminate the possibility of a frozen keyboard and the need to reboot the computer and would also eliminate any frustrations.

Introduction

There are a couple of ways to control the number of records being viewed by the user. One of them is to provide some search criteria to the users, prompting them to for information. These so call inputted information / parameters are then passed to the program for processing, eliminating unnecessary information. This would be ideal if most of the databases were small in size 20,000 records or less, but in the real world databases are rarely that size. Most of them contain hundreds of thousands and even millions or billions of records. To display all records retrieved at one time can prove to be a challenge even to current high-end computers available in the market today. It is almost impossible to display a million records of retrieved data on a Web Browser without the computer keyboard freezing up and aborting the process altogether. Therefore, the only viable solution is by means of using a scrolling program, and also providing the user the option to download a delimited version of the file to their own desktop when the user is tired of scrolling. **Base/SAS®**, **SAS/MACRO™**, **SAS/IntrNet™** and **HTML** provide ease of coding for both the scrolling and downloading processes within a **SAS®** program.

To successfully invoke the Scrolling / Downloading program, the **HTML** or **htmSQL™** page needs to pass in an initializing variable along with other required information or variables. In this case, a **MACRO** variable contains an initial value, which instructs the program on how to begin. For instance a **MACRO** variable containing the value “**YES**” for first time use will cause the program to initialize all of the necessary variables. With each page displayed, the user will be given several button options to click on including returning to the menu, downloading a delimited file, and scrolling forward and backwards. All of these options (one at a time) can be selected by a click of the button. Each time the scrolling program is being executed a set of pre-calculated scroll values for forward and backward scrolling is passed back to the scrolling program. In the case of downloading a delimited file, a list of parameters is passed to the download program, where it can extract the required information for downloading.

The scrolling and downloading program has been proven to be a very effective tool to circumvent the problem of computer memory shortage, speed of the **INTERNET** lines, and processor speed. By using the scrolling technique, it also optimizes the use of the **INTERNET** and prevents monopolizing the **INTERNET** lines by a single user. This Scrolling and Downloading set of code can be used time and time again, making it recyclable.

Technical Aspects of the Scrolling & Downloading Program

The main core of the scrolling program involves the handling the initial group of record display, the second group of records displayed with the consideration of scrolling back to the initial set of records and the last group of records being displayed. The definition of handling determines the option buttons being displayed to the user. If an initial group of records is displayed, only a Forward or Next button will be shown. Similarly, when the scrolling program reaches the end of the group of records, only a Backward or Previous button will be shown. If the second group of records is displayed, and the user selects Backward or Previous button, the program should be able to display the initial group of records with only the Forward or Next button. Finally, the scrolling program should also be able to handle the case where the number of records received is less than or equal to the scrolling pre-set amount, displaying either no Forward or Backward option scrolling, or just Backward scrolling only.

The downloading program is part of the scrolling program linked by means of a button to invoke the downloading program with the necessary parameters passed to it. The criteria are similar to that of the scrolling program except without the pre-calculated Forward and Backward scrolling ranges. The downloading program uses “**PROC CONTENTS**” to build the delimited file. The information that is kept from the successful execution of “**PROC CONTENTS**” follows:

i. Name
ii. Type
iii. Length
iv. Label
v. Format
vi. Formatl
vii. Formatd
viii. Informat
ix. Informl
x. Informd
xi. Varnum
xii. Nobs

Using the different “FORMATS” and “INFORMATS” variables, a download template is built. With the “NAME” variable, the header is built. A pre-selected delimiter is used, in this case a “**|**” is used, otherwise known as the “PIPE”. The “**|**” is selected to be used as a delimiter mainly because of its rarity of being used in inputting data. Upon invoking the download program, the selected data will be downloaded to the user’s desktop using the template built, where the user can browse or manipulate the data at their own leisure.

The scrolling and downloading program works hand in hand presenting similar data to the user in different formats, thus giving the user the option to see and verify the information selected before retrieving a copy for use.

Program Listing & Description

The program listed below has a pre-set value of 50, that means that only 50 records will be displayed at a time. However, that value can be changed to accommodate the developers choice or it can be changed to pass in a **SAS/MACRO™** variable that contains the user’s preference, which mean minor changes need to be made to the program.

SAS statements that are highlighted below are essential to the successful execution of the scrolling program.

```
* ------------------------------------ *;
*  Program name : test_report.sas      *;
*                                      *;
*  Function : This MACRO call itself   *;
*  over and over again, it pass the    *;
*  scroll values back to itself each   *;
*  time either a ‘Next’ or ‘Previous’  *;
*  button is selected.  Each time when *;
*  the values are passed back to the   *;
*  program the values for ‘Next’ and   *;
*  ‘Previous’ will be calculated each  *;
*  time. It also provides data viewed  *;
*  to be downloaded.                   *;
* ------------------------------------ *;

libname test ‘/data/test’;

%macro viewrec;

%global start stop pstart pstop ftime
userid password status loc;

* ------------------------------------ *;
*  Selecting only required records.    *;
* ------------------------------------ *;

data temp;
   set test.jobs;
   where (compress(upcase(status)) eq
upcase(“&status”));

run;

* ------------------------------------ *;
* Peform Proc Contents to obtain the   *;
* total number of records selected     *;
* from above.  And created a MACRO     *;
* variable to hold the count of the    *;
* records selected.                    *;
* ------------------------------------ *;

proc contents data=temp
out=xxx(keep=nobs);
              noprint;
run;

data _null_;
   set xxx end=eof;

   if eof then do;
      call sympout(‘lastrec’,nobs);
      call
sympout(‘lstrec’,put(nobs,comma10.));
   end;

run;

* ------------------------------------ *;
* Create Start and Stop ranges for     *;
* both forward and backward scrolling  *;
*                                      *;
* Variable start and stop are          *;
* designated for forward scrolling     *;
*                                      *;
* Variable pstart and pstop are        *;
* designated for backward scrolling    *;
*                                      *;
* The “FTIME” variable here refers to  *;
* First Time usage.                    *;
* ------------------------------------ *;

data tempset;
   set temp end=eof;
   format   begin   3.;
   format   ending  3.;
   format   start   3.;
   format   stop    3.;
   format   pstart  3.;
   format   pstop   3.;
   informat begin   3.;
   informat ending  3.;
   informat start   3.;
   informat stop    3.;
   informat pstart  3.;
   informat pstop   3.;

   %if “&ftime” = “YES” %then %do;
      start  = 51;
      stop   = 100;
      pstart = 1;
      pstop  = 50;
      call symput(‘pstart’,pstart);
      call symput(‘pstop’,pstop);
      call symput(‘start’,start);
      call symput(‘stop’,stop);
      if _N_ LT &lastrec then do;
         if (_N_ < 51) then output
tempset;
      end;
   %end;
   %else %do;
      start  = &start;
      stop   = &stop;
      pstart = (start - 50);
      pstop  = (stop  - 50);
      begin  = (start + 50);
      ending = (stop  + 50);
      if _N_ LT &lastrec then do;
         if (start <= _N_ <= stop) then
output tempset;
      end;
      call symput(‘pstart’,pstart);
      call symput(‘pstop’,pstop);
      call symput(‘start’,start);
      call symput(‘stop’,stop);
   %end;

   if "&ftime" NE "YES" then do;
      if stop GT &lastrec then do;
         if eof then output tempset;
      end;
   end;
   else do;
      if start GT &lastrec then do;
         if eof then output tempset;
      end;
   end;

run;

data _null_;
   set tempset end=eof;
   file _webout;

   if _N_ = 1 then do;
      put “Content-type: text/html”;
      put ;
      put “<HTML>”;
      put “<HEAD>”;
      put “</HEAD>”;
      put “<BODY BGCOLOR=’#f5f5cd’>”;
      link header;
   end;
```

← More SAS / HTML codes →

```
   if eof then do;
      link footer;
      put “</BODY>”;
      put “</HTML>”;
   end;

return;

header:

   put “<CENTER>”;
   put “<TABLE BORDER=0 CELLSPACING=0
CELLPADDING=0>”;
   put “<TR>”;
   put “<TD ALIGN=LEFT>”;
   put “<FORM ACTION=’/cgi-
bin/htmSQL/testlib/test_report_main.hsql’
METHOD=POST>”;
   put “<INPUT TYPE=HIDDEN NAME=USERID
VALUE=” “&userid” “>”;
   put “<INPUT TYPE=HIDDEN NAME=PASSWORD
VALUE=” “&password” “>”;
   put “<INPUT TYPE=SUBMIT VALUE=’Go Back
To Report Menu’ ALIGN=LEFT>”;
   put “</FORM>”;
   put “</TD>”;
   put “<TD ALIGN=LEFT>”;
   put “<FORM ACTION=’/cgi-bin/broker’
METHOD=POST>”;
   put “<INPUT TYPE=HIDDEN NAME=USERID
VALUE=” “&userid” “>”;
   put “<INPUT TYPE=HIDDEN NAME=PASSWORD
VALUE=” “&password” “>”;
   put “<INPUT TYPE=HIDDEN NAME=’_debug’
VALUE=’0’>”;
   put “<INPUT TYPE=HIDDEN
NAME=’_service’ VALUE=’newsas’>”;
   put “<INPUT TYPE=HIDDEN
NAME=’_program’
VALUE=’testp.test_download.sas’>”;
   put “<INPUT TYPE=SUBMIT
VALUE=’Download A Delimited File’
ALIGN=LEFT>”;
   put “</FORM>”;
   put “</TD>”;
   put “</TR>”;
   put “</TABLE>”;
   put “</CENTER>”;

return;

footer:

   if “&ftime” = “YES” then do;
      put “<FORM ACTION=’/cgi-bin/broker’
METHOD=POST>”;
      put “<INPUT TYPE=HIDDEN NAME=STATUS
VALUE=” “&status” “>”;
      put “<INPUT TYPE=HIDDEN NAME=LOC
VALUE=” “&loc” “>”;
      put “<INPUT TYPE=HIDDEN NAME=USERID
VALUE=” “&userid” “>”;
      put “<INPUT TYPE=HIDDEN
NAME=PASSWORD  VALUE=” “&password” “>”;
      put “<INPUT TYPE=HIDDEN NAME=_debug
VALUE=’0’>”;
      put “<INPUT TYPE=HIDDEN
NAME=_service  VALUE=’newsas’>”;
      put “<INPUT TYPE=HIDDEN
NAME=_program
VALUE=’testp.test_report.sas’>”;
      put “<INPUT TYPE=HIDDEN NAME=FTIME
VALUE=’NO’>”;
```

```
      put “<INPUT TYPE=HIDDEN NAME=PSTART
VALUE=&pstart>”;
      put “<INPUT TYPE=HIDDEN NAME=PSTOP
VALUE=&pstop>”;
      put “<INPUT TYPE=HIDDEN NAME=START
VALUE=&start>”;
      put “<INPUT TYPE=HIDDEN NAME=STOP
VALUE=&stop>”;
      put “<FONT SIZE=2
COLOR=BLUE><B>View</B></FONT>”;
      put “<INPUT TYPE=SUBMIT VALUE=’
Next ’ ALIGN=LEFT>”;
      put “<FONT SIZE=2 COLOR=BLUE><B>50
Records</B></FONT>”;
      put “</FORM>”;
   end;
   else do;
      if &start GE &lastrec then do;
         put “<FORM ACTION=’/cgi-
bin/broker’ METHOD=POST>”;
         put “<INPUT TYPE=HIDDEN
NAME=STATUS    VALUE=” “&status” “>”;
         put “<INPUT TYPE=HIDDEN NAME=LOC
VALUE=” “&loc” “>”;
         put “<INPUT TYPE=HIDDEN
NAME=USERID    VALUE=” “&userid” “>”;
         put “<INPUT TYPE=HIDDEN
NAME=PASSWORD  VALUE=” “&password” “>”;
         put “<INPUT TYPE=HIDDEN
NAME=_debug    VALUE=’0’>”;
         put “<INPUT TYPE=HIDDEN
NAME=_service  VALUE=’newsas’>”;
         put “<INPUT TYPE=HIDDEN
NAME=_program
VALUE=’testp.test_report.sas’>”;
         put “<INPUT TYPE=HIDDEN
NAME=FTIME     VALUE=’NO’>”;
         put “<INPUT TYPE=HIDDEN
NAME=PSTART    VALUE=&start>”;
         put “<INPUT TYPE=HIDDEN
NAME=PSTOP     VALUE=&stop>”;
         put “<INPUT TYPE=HIDDEN
NAME=START     VALUE=&pstart>”;
         put “<INPUT TYPE=HIDDEN
NAME=STOP      VALUE=&pstop>”;
         put “<FONT SIZE=2
COLOR=BLUE><B>View</B></FONT>”;
         put “<INPUT TYPE=SUBMIT VALUE=’
Previous ’ ALIGN=LEFT>”;
         put “<FONT SIZE=2
COLOR=BLUE><B>50 Records</B></FONT>”;
         put “</FORM>”;
      end;
      else do;
         put “<BR>”;
         put “<CENTER>”;
         put “<TABLE BORDER=0
CELLSPACING=2 CELLPADDING=0>”;
         put “<TR>”;
         put “<TD ALIGN=LEFT>”;
         put “<FORM ACTION=’/cgi-
bin/broker’ METHOD=POST>”;
         put “<INPUT TYPE=HIDDEN
NAME=STATUS    VALUE=” “&status” “>”;
         put “<INPUT TYPE=HIDDEN NAME=LOC
VALUE=” “&loc” “>”;
         put “<INPUT TYPE=HIDDEN
NAME=USERID    VALUE=” “&userid” “>”;
         put “<INPUT TYPE=HIDDEN
NAME=PASSWORD  VALUE=” “&password” “>”;
         put “<INPUT TYPE=HIDDEN
NAME=_debug    VALUE=’0’>”;
         put “<INPUT TYPE=HIDDEN
NAME=_service  VALUE=’newsas’>”;
         put “<INPUT TYPE=HIDDEN
NAME=_program
VALUE=’testp.test_report.sas’>”;

         if &pstart = 1 then do;
            put “<INPUT TYPE=HIDDEN
NAME=FTIME     VALUE=’YES’>”;
         end;
         else do;
            put “<INPUT TYPE=HIDDEN
NAME=FTIME     VALUE=’NO’>”;
         end;

         put “<INPUT TYPE=HIDDEN
NAME=START     VALUE=&pstart>”;
         put “<INPUT TYPE=HIDDEN
NAME=STOP      VALUE=&pstop>”;
         put “<FONT SIZE=2
COLOR=BLUE><B>View</B></FONT>”;
         put “<INPUT TYPE=SUBMIT VALUE=’
Previous ’ ALIGN=LEFT>”;
         put “<FONT SIZE=2
COLOR=BLUE><B>50 Records</B></FONT>”;
         put “</FORM>”;
         put “</TD>”;
         put “<FORM ACTION=’/cgi-
bin/broker’ METHOD=POST>”;
         put “<INPUT TYPE=HIDDEN
NAME=STATUS    VALUE=” “&status” “>”;
         put “<INPUT TYPE=HIDDEN NAME=LOC
VALUE=” “&loc” “>”;
         put “<INPUT TYPE=HIDDEN
NAME=USERID    VALUE=” “&userid” “>”;
         put “<INPUT TYPE=HIDDEN
NAME=PASSWORD  VALUE=” “&password” “>”;
         put “<INPUT TYPE=HIDDEN
NAME=_debug    VALUE=’0’>”;
         put “<INPUT TYPE=HIDDEN
NAME=_service  VALUE=’newsas’>”;
         put “<INPUT TYPE=HIDDEN
NAME=_program
VALUE=’testp.test_report.sas’>”;
         put “<INPUT TYPE=HIDDEN
NAME=FTIME     VALUE=’NO’>”;
         put “<INPUT TYPE=HIDDEN
NAME=START     VALUE=&start>”;
         put “<INPUT TYPE=HIDDEN
NAME=STOP      VALUE=&stop>”;
         put “<FONT SIZE=2
COLOR=BLUE><B>View</B></FONT>”;
         put “<INPUT TYPE=SUBMIT VALUE=’
Next ’ ALIGN=LEFT>”;
         put “<FONT SIZE=2
COLOR=BLUE><B>50 Records</B></FONT>”;
         put “</FORM>”;
         put “</TR>”;
         put “</TABLE>”;
         put “</CENTER>”;
      end;
   end;

return;

run;

%mend viewrec;

%viewrec;

quit;
```

Conclusion

The scrolling and downloading program allows user to conserve precious memory resource on their desktop when interfacing with a web application.

AUTHOR CONTACTS

Ming C. Lee
Trilogy Consulting
700 W. Mineral, Room MN E21.20
Littleton, CO 80120
(303) 707-8452
mxlee5@uswest.com