Managing Your SAS® Web Applications by Using Web Page Counters Created Through SAS/IntrNet™ Software

Heather Tavel - Trilogy Consulting, Denver, CO

ABSTRACT

U S West analysts use counters on many of their web pages to monitor the use of internal reporting tools. The information from these counters indicates whether a tool is meeting the needs of the customer, and possibly whether the analyst should even continue maintaining the tool. In the move to use dynamic web reports created with SAS/IntrNet software as opposed to static HTML files and Perl scripts, the assumption was that counters based on Perl scripts would still be used. However, counters can be created using simple techniques through SAS/IntrNet tools and base SAS software, keeping the programming consistent, and giving the analyst easy access to data that can be instrumental in determining work priorities. Data retrieved from counter data sets can easily be plotted with SAS/GRAPH® software to find out whether usage is increasing or decreasing, and to also find out patterns of fluctuation.

INTRODUCTION

There are many counters available for use on the Internet. The standard one being used in the Capacity Provisioning group at US West was a Perl script that incremented a value stored in a small file. This counter was also an HTML link to the marketing web page of the individual who created it, which is a link that is not necessary for the business practice of those using the web tool. Because the web tools were being redesigned or newly created using SAS software, we decided to forgo the Perl scripts and just create counters within SAS itself. It turned out to be a very simple task.

Two examples follow: The first uses htmSQL™, ideal when the only SAS processing needed on the HTML page is for the counter, or for pages that contain reports that can be created easily using htmSQL. The other counter is created by an Application Dispatcher program, which can be used for a counter on a page where you need to use more of the SAS software capabilities besides those available through htmSQL. Both are simple pieces of code that simply increment a value in a data set variable.

CREATING A COUNTER DATA SET

The first step in creating a SAS web counter is to create a SAS data set to hold the counts. Because our tools are large and can contain many reports (which sometimes means many programs), in one Dispatcher Application, we store each Application's data set in its own data library. Because of this, each application has its own counter data set. Each data set has only one row, and contains one variable for each report in the Application that requires a counter. The variables are named similarly to the function or name of the report. When creating the data set, the value for each variable is initialized to 0. Using the counters, the values will be incremented by one each time the report page is loaded.

Figure 1: ABCTOOL.COUNTER data set

SMRY1	SMRY2	ADHOC1	DETAIL1
1023	567	223	1357

INCREMENTING AND VIEWING A COUNTER WITH htmSQL™

The htmSQL product is a very quick way to update and access SAS data sets straight from an HTML document, without having to write a SAS program and using the Application Dispatcher. The htmSQL query is placed on the page where you would like to keep the count, often on a tool's main selection page. When that page is loaded, the count must be incremented, and the new value can optionally be displayed. If you choose not to display it for the customer, you can simply reference the counter data set to monitor usage.

```
<HTML>
<HEAD>
<TITLE> Selections for Summary Report
</TITLE>
</HEAD>
<BODY>
…html code for your page…
<!--update the counter variable -- >
{UPDATE datasrc="mysource"}
  {SQL}
   update abctool.counter
     set summary1=summary+1;
  {/SQL}
{/UPDATE}
<!--query the counter value -- >
{QUERY datasrc="mysource"}
  {SQL}
```

```
    select summary1 from
abctool.counter;
    {/SQL}
  <!-- display the value in a small
  table -- >
  <TABLE border=1><TR><TD> Counter:
  {&summary1} </TD></TR></TABLE>
  {/QUERY}
  </BODY>
  </HTML>
```

INCREMENTING AND VIEWING A COUNTER USING THE APPLICATION DISPATCHER

Counters can also be used on report pages created with the Application Dispatcher. We typically use a data step to update the value. This update can be done in a separate data step than the step that creates the actual report portion of the web page, or it can all be combined together into one data step for efficiency.

```
data _null_;
set abctool.counter;
file _webout;
if _n_=1 then do;
 /* increment counter */
 smry1+1;
 put "Content-type: text/html";
 put;
 put "<HTML>";
 put "<BODY>";
 …put statements with html code…
/*  output counter value  */
 put "<TABLE BORDER=1>";
 put  "<TR><TD> Counter: "
      summary1
      "</TD></TR>" ;
 put "</TABLE>";
 put "</BODY>";
 put "</HTML>";
end;
run;
```

Using the Application Dispatcher also makes it easy to add extra macro processing geared toward conditional execution of the counter increment. This may be necessary if the page is hit multiple times during the course of report creation and only the count pertaining to the initial hit is desired. This can be achieved by setting up Boolean flags to be passed along through the URL as name-value pairs, which are then converted into macro variables by the Application Dispatcher's broker. These macro variables can then be tested with %if-%then processing to determine whether the counter should be incremented. Additional features with the Application Dispatcher allows the remote address and/or userid of the requestor to be captured, if you are trying to determine use by certain users. Using counters created with the Application Dispatcher allows for much flexibility in the type of information you are trying to quantify.

CREATING THE MASTER DATA SET

Once the counter has been established, data is captured over the long term by creating a batch program to append the daily results to a master archive data set, which contains an additional variable for the date and/or time stamp. This program can be run every night at the same time.

```
proc sql
    select *
      from abctool.cnt_trnd
    outer union corr
    select * , today() as date
      from abctool.counter;
quit;
```

or

```
data abctool.cnt_trnd
 set abctool.cnt_trnd abctool.counter;
 if date=. then date=today();
run;
```

USE OF THE COUNTER DATA SET OUTSIDE OF THE WEB PAGE

Having a counter on a web page is useful, but it is the equivalent to "eyeballing" your data. You may find that there have been 3000 hits on your tool, but when did those happen? Is the usage growing at a constant rate, or is it declining? Are the users hitting the web tool at the end of the month only, or on a daily basis? Going onto the web page periodically to check the counter number may work, but it's hard to see trends that way. Also, by going into the web page just to check the number, you are creating false hits which increment the counter.

The master archive data set can be used with SAS/GRAPH software to plot usage over time and check any trends that may indicate the following, for example:

1) The tool usage has declined, meaning that meetings should be held with the customer to determine if the tool still meets their needs. If not, it can be revised or removed.
2) Usage is primarily at the end of the month, beginning of the week, etc. (Many of our tools have data preparation programs that are run daily in batch mode. For tools that require a large amount of data set preparation, it is possible that these preparation programs can be run only at the end of the month or weekly. This can free up quite a bit of processing time).
3) Looking at several tool counters at a time can help determine overall web traffic, which may be useful information for "load-balancing".