

Collecting data via the Internet with SAS/INTRNET® and SAS/SHARE® Software

Stacy Buffington, Texas Education Agency, Austin, Texas

Doyle McDonald, SAS Institute Inc., Austin, Texas

ABSTRACT

The Texas Education Agency (TEA) has had much success with the use of SAS/INTRNET software, primarily utilizing the web publishing tools in order to dynamically publish reports to the World-Wide Web. Another exciting component of this product is the data collection capability. With the added use of SAS/SHARE, we can collect data from the client via the Internet and update SAS data sets directly.

This software combination definitely delivers in many aspects when it comes to collecting data via the Internet. SAS programmers need not rely on CGI scripting for web application programming, and furthermore, the transformation of the flat files created by CGI into a SAS data set is eliminated.

In the scope of this paper, we will briefly discuss the business problems that TEA identified with regards to our specific need for Internet commerce, the reasons for choosing SAS software to solve those problems and lastly, specifics on how we developed and implemented our solution using SAS/INTRNET and SAS/SHARE software.

THE BUSINESS PROBLEM

TEA was in need of an improved methodology for collecting and updating data. Our requirements included the capability to automate the updating of SAS data sets, while providing ease of use for our clients.

The existing data collection system for the Texas School Accountability system was in dire need of improvement. Data is collected from district superintendents in order to calculate their accountability rating. The cumbersome and lengthy system crossed platforms and software, while relying on input from paper documents received from the client, in order to update a SAS data set. Furthermore, inaccuracies were prevalent due to the process of updating the data "by hand".

CHOOSING SAS SOFTWARE AS THE SOLUTION

Having successfully built our web site for Performance Reporting with SAS/INTRNET, the obvious choice was to use SAS/INTRNET again for the data collection projects. SAS/INTRNET was an easy choice for our division to make with regards to our reporting needs via the internet because we already had in-house many SAS programmers with many years of SAS programming experience. When it became evident that we would need to not only report data to our clientele but to collect data from them as well, we felt the task of collection may not be as simple. Would SAS/INTRNET software be appropriate for this purpose? Would we need to explore other alternatives such as CGI?

SAS/SHARE software provided the answer to these problems. Our SAS representative assured us that SAS/SHARE would integrate perfectly with SAS/INTRNET, that it would provide the necessary multi-user, concurrent data access that internet commerce demands and that it would be quite simple to implement and deploy within the framework of our SAS/INTRNET environment.

Technical representatives with the SAS Institute came to our office, reviewed and discussed our project plans and helped us develop a model by which we could proceed. We were also given a product demonstration using SAS/INTRNET and SAS/SHARE software, as well as an in-depth look at and walk-through of the source code used in the demonstration. After this meeting, we met and discussed the

DEMONSTRATION OF SURVEY EXAMPLE

SAS solution (as well as other alternatives) and found that, again, SAS software was the most logical choice.

DEVELOPMENT AND IMPLEMENTATION

Working with the model developed jointly by SAS representatives and our staff, we identified three basic steps to successful development and implementation.

Starting a SAS/SHARE server

Two features of SAS/SHARE software are being used for managing concurrent update of SAS data libraries at TEA; the SERVER procedure and the REMOTE engine.

PROC SERVER controls and executes input and output requests to SAS files on behalf of other SAS sessions. We refer to an execution of this procedure as a SHARE server.

The REMOTE engine enables a user's SAS session to access shared data by communicating with a SHARE server.

To start our SHARE server, we pass the SAS program containing the PROC SERVER code to SAS with the AUTOEXEC parameter.

```
SAS -autoexec /applib/path/share/shrauto.sas
```

The SHRAUTO.SAS file contains the following source code:

```
proc server server=shr1
  msgnumber dtformat=datetime19.2; run;
```

Allocating SAS/SHARE data libraries via SAS/INTRNET

To allocate SHARE server managed data libraries, we simply need to add to the LIBNAME statement the SERVER parameter identifying which SHARE server will be performing the management service.

```
libname survdata
  '/applib/path/data' server=tea.shr1;
```

In some installations, the SERVER= parameter requires a two-level name; first level being the domain name of the physical server, second level being the name of the SHARE service. The name of the SHARE service and its TCP port assignment is defined in the SERVICES file found on the physical server.

Inserting/Updating data with SAS programming

There are, of course, many ways to update SAS data sets. The following code is supplied in simplified form as a familiar example of how easy this process is. The same process is described in more detail later in this document.

```
proc sql;
  insert into survdata.survresp
    (survid,survitem,respname,
     respdate,respctype,respvalu)
  values
    ("01","1","sasdlm@wnt.sas.com",
     "McDONALD","01JAN99'D","A","1");
quit;
```

Identifying the survey and the respondent

SAS/IntrNet®, SAS/SHARE® Survey Example	
Select the survey you wish to respond to from the list below	
Web Site Useability Survey	
Please complete the following information in order that we may process your responses to this survey. Thank you for your participation!	
Enter your name:	
Doyle McDonald	
Enter your E-Mail Address:	
sasdlm@wnt.sas.com	
Reset	Proceed

The web interface at left was generated using htmSQL®, a component of SAS/INTRNET software that is basically a hybrid scripting language combining features of both SQL and HTML. htmSQL® scripts are designed to send SQL queries to a SAS/SHARE server, which returns the data requested and htmSQL® then formats the HTML and returns the finished result to the browser.

In this example, a SAS dataset that contains descriptive information regarding currently available surveys is queried and presented to the survey respondent. Text input fields are provided for identification purposes.

Below is an excerpt from the htmSQL® script that generated this web interface.

```
{query datasrc="demo"}
{sql}
  select survid, survname
    from survey.survmast
   order by survname
{/sql}
<select name=survid>
{eachrow}
  <option value={&survid}>{&survname}
{/eachrow}
</select>
{/query}
```

Entering responses to the survey

Web Site Useability Survey	Response
GROUP 1 QUESTIONS	
Did you find our web site easy to navigate and use?	Sometimes
Did you feel the content was consistent with your expectations?	Positive
GROUP 2 QUESTIONS	
Was response time for your queries acceptable?	Positive
Do you feel our web site was worth the time spent visiting it?	Very Positive
GROUP 3 QUESTIONS	
How would you rate the aesthetics of our site?	Very Positive
Were your objectives in visiting our site accomplished?	Positive
Do you plan to return to our site in the future?	Yes
Once completed, press the button at right to SUBMIT your responses	Submit Reset

The interface at left was built using Base SAS programming in conjunction with SAS/INTRNET software. The HTML that comprises this web page was generated using file and put statements in a data step.

Again, SAS datasets contain the descriptive information that make up the content of this web page.

Below is an excerpt from the SAS program that generated this web interface.

```
if first.itemgrp then do;
  put '<tr>';
  put '<th align=left colspan=2 valign=bottom bgcolor=#ad9482>'
    '<font color=white size=3>' grptext '</font></th>';
  put '</tr>';
end;
```


The following is an excerpt from the Base SAS code used to generate and deliver the previous HTML report.

```

data _null_;
  timestmp=put(time(),time8.);
  unique='c:\temp\ss' || trim(left(compress(timestmp,':')) || '.html';
  call symput('unique',unique);
run;

data _null_;
  set work.survresp end=eof;
  by grptext survitem;
  file "&unique";
  if _n_=1 then do;
    set survdata.survmast(where=(survid="&survid"));
    put '<html><body bgcolor=#cdb4a2><center>';
    (other data step code removed)
  if eof then put '</table></body></html>';
  run;

filename emresp email "&email" subject="Survey Response"
  attach="&unique";

data _null_;
  file emresp;
  put 'Thank you for helping make our survey a success.';
  put 'The attached file is a summary of responses to this survey.';
run;

```

CONCLUSION

SAS/INTRNET and SAS/SHARE software provides the necessary functionality and preferred ease of use that TEA needs for our projects requiring this level of internet commerce: software that is flexible yet easy to use and deploy, software that is quick to develop and easy to maintain and last, but not least, software that integrates with our existing infrastructure for web based applications.

ACKNOWLEDGMENTS

SAS, SAS/INTRNET and SAS/SHARE are registered trademarks of SAS Institute Inc. in the USA and other countries. © indicates USA registration.

REFERENCES

SAS Institute Inc. (1991), *SAS SHARE Software, Version 6, First Edition*, Cary, NC: SAS Institute Inc.

CONTACT INFORMATION

Stacy Buffington
 Texas Education Agency
 1701 N. Congress Ave.
 Austin, TX 78701
 512-305-8587
sbuffing@tmail.tea.state.tx.us

Doyle McDonald
 SAS Institute, Inc.
 11920 Wilson Parke Avenue
 Austin, TX 78720
 512-258-5171
sasdlm@wnt.sas.com