

A Case Example of SAS/IntrNet with JavaScript

Desheng(Ben) Xu, Northern Illinois University, De Kalb, IL

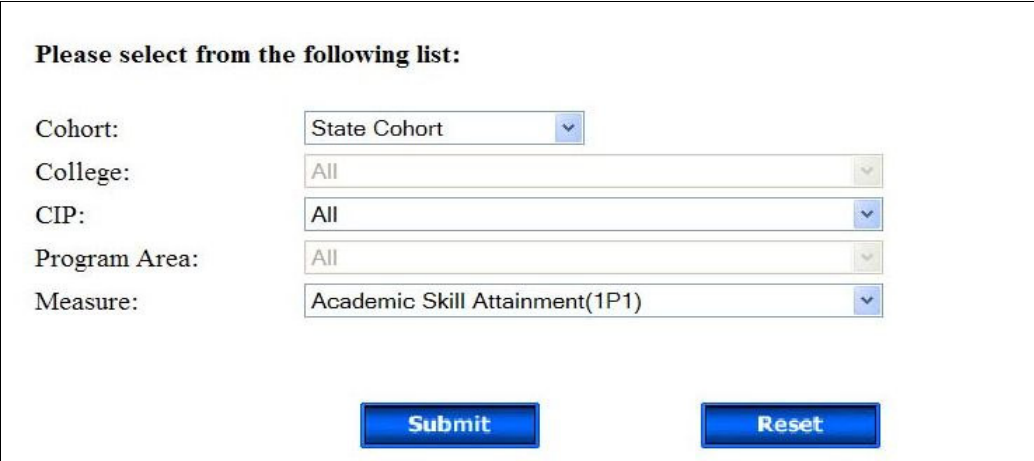
ABSTRACT

In SAS web applications, the parameters are passed to the Application Dispatcher to process. By using JavaScript, the parameters can be dynamically passed to the CGI broker. Also this paper addresses how to build index for SAS data set to speed up the application. In addition, the session concept is applied in the application. This paper is appropriate for intermediate programmers who know SAS/IntrNet as well as JavaScript language.

INTRODUCTION

This is a web application that the author built for a state agency. By using this web application, the reports of postsecondary core performance measures, such as Academic Skill Attainment, Program Completion and Employment Retention etc., can be produced from different hierarchical levels including state, peer group and individual community college. Totally there are seven peer groups in the state with each group having a few colleges. Within a peer group, the overall outcome for the group or the outcome for each college can be generated by choosing different values from the select boxes. The purpose of the Classification of Instructional Program (CIP) is to provide a taxonomic scheme to track, assess and report the program completion activity.

As shown in figure1, when the cohort is chosen as statewide level, the college select box and program area select box are disabled and they are grayed out. This prevents the user from selecting any other values from the select box. On this condition, only the statewide level report is allowed. Obviously, you can view the statewide report by all CIP codes or each individual CIP code.



Please select from the following list:

Cohort: State Cohort

College: All

CIP: All

Program Area: All

Measure: Academic Skill Attainment(1P1)

Submit Reset

Figure 1. Web application interface



Please select from the following list:

Cohort: Peer Group III

College: McHenry County

CIP: Agricultural Business And Production

Program Area: All

Measure: Employment and Continuing Education(3P1)

Submit Reset

Figure 2. College and Program Area select boxes are activated

If a particular peer group is chosen (Figure 2), the college select box is activated. Only the community colleges within this group will be available from the college select box. The program area select box also becomes active because the programs are provided by each college. Then you can select all the programs or just one program for this college.

After the user makes selections and clicks the submit button, the report is generated on the fly (Figure 3). Since the author wants to keep him away from the trouble of data confidentiality issues, the fake data are used in this paper. The table is generated by combining PROC REPORT and ODS skills. The user-selected criteria are listed as titles. Moreover, the report is downloadable as a PDF file.

The work is done with SAS 8.2 on Windows 2000.

Measure - Academic Skill Attainment(1P1)								
Peer Group III								
College - All								
CIP - Agricultural Business And Production								
Program Area - All								
Year	Performance	Total	Special Populations		Gender		Racial/Ethnic	
	Data	Cohort	Spec Pop	Nonspec Pop	Female	Male	White	Other
1900	Number of Students	100	100	100	100	100	100	100
	Performance	0 - 100%	0 - 100%	0 - 100.0%	0 - 100%	0 - 100%	0 - 100%	0 - 100%
1901	Number of Students	100	100	100	100	100	100	100
	Performance	0 - 100%	0 - 100%	0 - 100.0%	0 - 100%	0 - 100%	0 - 100%	0 - 100%
1902	Number of Students	100	100	100	100	100	100	100
	Performance	0 - 100%	0 - 100%	0 - 100.0%	0 - 100%	0 - 100%	0 - 100%	0 - 100%
1903	Number of Students	100	100	100	100	100	100	100
	Performance	0 - 100%	0 - 100%	0 - 100.0%	0 - 100%	0 - 100%	0 - 100%	0 - 100%

Figure 3. A generated report on the web

USE JAVASCRIPT TO ESTABLISH DYNAMIC WEB INTERFACE

Behind the scene, the JavaScript is used to make the web interface more interactive. It also provides a lot of possibility of parameter combinations to be sent to the application broker. Richard Phillips has provided an excellent paper (paper 61 in SUGI 28) on how to take advantage of SAS to "write" the JavaScript codes.

Apparently, the critical step is that you must decide what JavaScript functions are needed for the web application before you pull out the data from SAS data set. In the following HTML codes it shows selecting a different cohort will trigger the onChange event, which in turn, calls three JavaScript functions loadCollege(), loadCIP() and loadProgram() to populate the option items for college, CIP and program area select boxes.

```

<select name="cohort" onChange="loadCollege(); loadCIP(); loadProgram();"
  <option label="State Cohort" value="00000" selected>State Cohort</option>
  <optgroup label="Peer Chort">
    <option value="00001">Peer Group I </option>
    <option value="00002">Peer Group II </option>
    <option value="00003">Peer Group III</option>
    <option value="00004">Peer Group IV </option>
    <option value="00005">Peer Group V </option>
    <option value="00006">Peer Group VI </option>
    <option value="00007">Peer Group VII</option>
  </optgroup>
</select>

```



```

put '      {
put '      if(i==0)
put '      {cselect = true;}
put '      else
put '      {cselect = false;}
put '      document.forms[0].college.options[i] = new Option(peercollabel' i +(-
1)'[i], peercolvalue' i +(-1) '[i], false, cselect);';
put '      }
put '    }
end;
put '}'
put '
run;

```

The Boolean DISABLED attribute is used to control the availability of the select values. By setting its value to be true, it prevents the user from changing the value from the select box. The user is unable to edit the disabled selection and no value is submitted with the form. By setting its value to be false, the select box is enabled, that allows the user to select an option and send the value to the application broker.

The cohort value of “00000” means statewide level is selected. It is the default selection state when the web page is loaded. Under this condition, the college select box is disabled and only the statewide report can be generated. When another cohort is selected, the college select box is enabled and the corresponding colleges for that peer group are populated in the college select box. As a result, the report for the whole group or each individual college can be created by submitting the form.

BUILD INDEX FOR SAS DATA SET

SAS index is quite useful when a very small subset of observations is retrieved from a large data set. By using the index, you can access the particular observations directly and efficiently. In this case example, the index is created for the report data set In order to output the report contents to the web faster. There are a few ways to create index, including DATA STEP, PROC DATASETS and PROC SQL. A composite index called cohort is created by the procedure PROC DATASETS. The key variable School_number is put in front of the variable PeerGroup since the former is more discriminating than the latter, which means the School_number will break the data set into more subsets than the PeerGroup.

```

proc datasets lib=demodata nolist;
  modify m1p1;
  index create cohort = (School_number PeerGroup CompCIP2dig curriculumcode);
  ...
quit;

```

After index is created, SAS will automatically decide if it will use index in the WHERE expression. To force SAS to use INDEX, use IDXNAME= option.

```

%if &measure = 1P1 %then %do;
  %let measure_title=Academic Skill Attainment(1P1);
  data temp1;
    set demodata.M1P1(IDXNAME=cohort where=(&wherecond));
    if school_number ne '' then call symput('college_title', put(school_number,
      $collfmt.));
    .....
  run;
  .....
%end;

```

APPLY SESSION CONCEPT

Typically, the web is in stateless environment since the HTTP is a stateless protocol, which means the second request knows nothing of the first request. Each click to the web site is treated as a new request by the web server. But under some circumstance, certain information is necessary to be kept from the previous request to the following requests. That's how the session concept fits in this situation. The session can be created by DATA STEP or MACRO.

```

Data Step:
  rc=appsrv_session('create');

Macro:
  %let rc=%sysfunc(appsrv_session(create));

```

After the session is created, some automatic macro variables, such as _SESSIONID, _THISSESSION, _REPLAY are set with values to reflect the current session ID. The session ID returns a unique session identifier for the current user and is not easily to be guessed. For instance, a _SESSIONID has the value of “e4jeHkqzI52” and _THISSESSION has the value of “/cgi-bin/broker.exe?_service=Demosrv&_server=localhost&_port=6105&_sessionid=e4jeHkqzI52”.

In addition, a SAS library called SAVE is created automatically after creating a session. If the data set is created or copied to this special library, it is available for the following requests. Similarly, to make the global macro variable existent between requests, the name of the macro variable must start with the keyword SAVE_.

In this web application, when preparing the data set for the HTML report, it is created in the session library. Afterwards, when saving the report into PDF format, there is no need to process the data set again. Therefore, the effort of checking the passed parameters is not necessary any more and a lot of programming codes can be avoided.

```
proc report data=SAVE.reportdata nowd
      style(REPORT)={font_size=1}
      style(HEADER)={font_weight=bold font_size=1}
      style(COLUMN)={font_size=1}
      ;
column ('Year ' Year) ('Performance' item)
      ('Total' All) ('Special Populations' SP NP)
      ('Gender' Female Male) ('Racial/Ethnic' White Other);
define Year/order center;
.....
run;
```

Also, session concept can be applied to engage strong security policies. With using session, the confidential user authentication information like user ID, password is not transferred across the network at each request.

USE ODS TO GENERATE WEB REPORTS

The Output Delivery System (ODS) can be used to send the report to the web. The special file reference _WEBOUT is used for both HTML and PDF destinations. After the report is displayed as the HTML table, at the bottom of the page (Figure 3), a button is provided to save the report into PDF format. In order to view the PDF file, Adobe Acrobat Reader 5.0 or higher version must be installed on the local computer. For more information, go to web site www.adobe.com for details.

All output created by Application Dispatcher programs have an HTTP header. The most basic HTTP header is the content-type header, such as: Content-type: text/html. It informs the browser what type of output should be produced. When the Content-Disposition header field with type "attachment" is used, a file download dialog is forced to pop up (Figure 4).

```
%LET RC=%SYSFUNC (APPSRV_HEADER (CONTENT-DISPOSITION, %STR (ATTACHMENT;
      FILENAME=Measure.pdf) ));

ods pdf body=_webout NOTOC;
      SAS procedure...
ods pdf close;
```

The parameter FILENAME specifies the generated PDF file name. The file download dialog provides the choice to view the file immediately or save it to hard drive for later reference. If the "Open" button is clicked, the PDF file is opened directly in Acrobat Reader. If the "Save" button is clicked, the save as dialog appears and the file can be saved with the default name Measure.pdf.

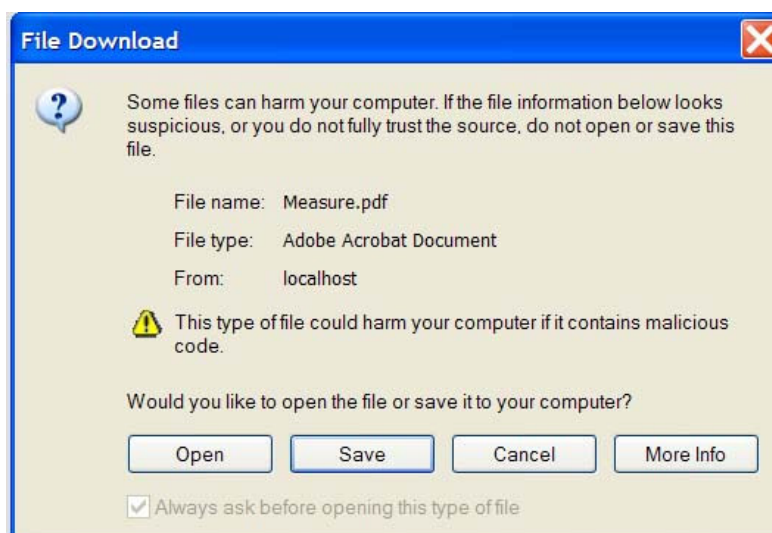


Figure 4. File download dialog

CONCLUSION

Combination of SAS/IntrNet and JavaScript makes the web application very powerful and interactive. Once you decide what functions are needed for the application, SAS can be used to generate the relevant JavaScript functions. By doing so, you can save a lot of efforts especially when the data set is updated. By switching DISABLE property of the select box on/off, the parameter is dynamically passed to the application dispatcher. The skills demonstrated in this paper can be widely used in the report that involves hierarchical variables.

REFERENCES

SAS Institute Inc. (1999), *SAS Online Doc, Version 8*, Cary, NC: SAS Institute Inc.
Phillips, R. *DHTML – GUI on the Cheap*, SUGI 28, Seattle, WA, 2003.

CONTACT INFORMATION

Please send your comments and questions to:

Ben Xu
Regional Development Institute
Northern Illinois University
148 N 3rd St
De Kalb, IL 60115
Work Phone: (815)753-4790
Email: dxu@niu.edu

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are trademarks of their respective companies.