

# Regulatory Overview of Using SAS/IntrNet<sup>®</sup> to Collect Data from Thousands of Users

Xin (Lucy) Luo, Z, Inc, Dallas, TX  
 Russell Denslow, Sodexho Campus Services, Orlando, FL  
 Sandra Archer, Sodexho Campus Services, Orlando, FL

## ABSTRACT

The Sodexho Unit Information Collection web page is a web-based survey system that was developed using SAS/IntrNet technologies combined with Base SAS, SAS/MACRO, HTML, and JavaScript. It enables Sodexho to collect operational site information from campus food service sites across the United States. The information collected includes contact information, operation type, service hours, holidays and semester span dates. The survey system has been running since May 2002, and has proved to be a very useful tool for collecting up-to-date client information. Please note that the SAS programs in this paper are simplified and are intended for readers with advanced knowledge of SAS/Base, SAS/Macro, SAS/IntrNet and some knowledge of HTML. The web site requires Internet Explorer v5.0 to run.

## INTRODUCTION

Sodexho is the leading provider of food and facilities management services in the U.S. and Canada, offering innovative outsourcing solutions in food service, housekeeping, grounds keeping, plant operations and maintenance, asset and materials management, and laundry services to corporations, health care and long term care facilities, retirement centers, schools, military, college campuses and remote sites. The Campus Services Division is responsible for providing campus dining, facility management, concessions and arena management services to universities, colleges and independent schools across the United States. The purpose of this project is to collect the serving times for more than 1,000 food service operation sites, including daily hours of service, holidays and semester dates.

With the development of the Internet, the web has become a convenient tool to collect and view information. SAS can be a powerful tool to produce a visually pleasing, dynamic interface for collecting data from a remote location or user site. This paper provides an overview of collecting data from operational sites using SAS/IntrNet, HTML (Hyper Text Markup Language), JavaScript and SAS/Macro code simultaneously. A user can examine, update and complete the information on the web page without installing SAS software on the user's machine. When the "SAVE" button is clicked, the entered data is stored in a SAS data set located on the web server.

Note that the web pages displayed in this paper are only representations of what the actual web pages look like, for readability purposes.

## DYNAMIC WEB PAGE OVERVIEW

The user starts the survey from the page shown in Figure 1. Every username and access PIN number is unique and obtained from the webmaster.

Figure 1. Home page of Unit Information Collection Web  
 Welcome to our unit information collection web page

Please enter your 10 digit number and password without dashes or spaces

10 Digit Unit Number:

Password:

This page was initially designed as a static page using MS FrontPage 98, and then the HTML source code was modified as needed. A portion of the modified code that produced Figure 1 is shown below.

```
<form method="get" action="http://.../broker.exe"
  onsubmit="return FrontPage_Form1_Validator(this)"
  name="FrontPage_Form1">
  <input type="submit" value="enter" name="LOGIN"> </font>
  <input type="hidden" name="_debug" value="0">
  <input type="hidden" name="_program"
    value="SOD.HOME.SAS">
  <input type="hidden" name="_service" value="sod">
  </form>
```

When the user clicks the "ENTER" button, all the actions will be completed by a SAS program, SOD.HOME.SAS, which is programmed in the Application Dispatcher. The Application Dispatcher, which has been set up on a Windows 2000 Server, is composed of the Application Server and the Application Broker. The Application Broker interprets the parameter values received from the web browser and passes it to the Application Server. The Application Server invokes the SAS program in a directory inaccessible via the web and returns the results to the end user via the Application Broker (SAS Institute Inc., 2001; Klenz, 1998).

If the username and password are valid, the page shown in Figure 2 is opened.

Figure 2. Contact and Unit Information Page

**Contact and Unit Information**

**Note :** Since this information is at the operational site level, you will need to fill out this survey for each of your operations.

Information based on your 10 digit unit number from our database.

10 Digit Unit Number: 8888888888

Unit / location name:

Unit Address:

City:  State:  Zip:

GM / Contact name:

The above information is from the Unit Master. Please correct if necessary.

Your First Name:

Your Last Name:

Click the **Save & Continue** Button to save your data and continue with the survey.

Click the **Save & Exit** Button to save your data, exit and return to the survey at a later time.

The original SAS data set on the server contains contact information for each operational site. By running SOD.HOME.SAS, that information will be retrieved and displayed in the browser. The following code is an excerpt from SOD.HOME.SAS that produces Figure 2.

```
put '<form action=http://...../broker.exe
method=get name=Contact>';
put '<p><font face="Arial black" color="#000080"
size="5">Contact and Unit Information</font></p>';
put '<p><b><font color="#808080" face="Arial Black">
10 Digit Unit Number:;</font></b>';
put '<font face="Arial">'; put location; put '</font></b></p>';
put '<p><b><font color="#808080" face="Arial Black">
Unit / location name:;</font></b>';
put '<font face="Arial">';put client; put ' </font></b></p>';
put street; put city; put state;
.....
put '<p align="left"><b><font color="#800080"> Click the
</font><font color="#FF0000">Edit';
put '</font><font color="#800080">button to return to the prior
page to make changes</font></b></p>';
put '<p align="left"><b><font color="#800080">Click the </font>
<font color="#FF0000">Save & Continue</font><font
color="#800080"> Button to save your data and continue with
the survey.</font></b></p>';
put '<p align="left"><font color="#800080"><b>Click the </b>
</font><b><font color="#FF0000"> Save & Exit</font>
</b><font color="#800080"> <b>Button to save your data,
exit and return to the survey at a later time.</b></font></p>';
put '<p align="center"> <input name=button onclick="edit()"
type=button value="Edit">';
put '<input name=button onclick="save()" type=button
value="Save & Continue">';
put '<input name=button onclick="exit()" type=button
value="Save & Exit">';
put '<input type="hidden" name="_debug" value="0">
<input type="hidden" name="_program"
value="sod.cont_review.sas">
<input type="hidden" name="_service" value="sod">';
put '<input type="hidden" value=&location name='location'>';
put '</form>';
```

If any information is incorrect or incomplete, the user will revise and input the correct data values in the HTML form. By clicking the “Save & Continue” button, the data values are read and the Application Dispatcher passes them to the SAS program as macro variables. The following code will update the data set.

```
data get_info;
set sod.websur;
where location="&location";
state="&st";
city="&ct";
street="&str";
zip="&zp";
first_name="&fstname";
last_name="&lstname";
(..other SAS code...)
run;
```

The code below defines the location of the SAS Application Broker, the type of service (socket, pool, launch) and the name of the SAS program to be run.

```
put '<form action=http://...../broker.exe
method=get name=Contact>
put '<input type="hidden" name="_debug" value="0">
<input type="hidden" name="_program"
value="SOD.CONTACT.SAS">
<input type="hidden" name="_service" value="sod">'
put '<input type='hidden' value=&location name='location'>'
put '</form>';
```

JavaScript is a widely used scripting language with the capability of input validation. In this web survey, JavaScript is used to handle multiple queries under one form and to check the required fields or numeric values before submitting a CGI (Common Gateway Interface) request. For example, when the user clicks the “Save & Continue” button in Figure 2, the “function save ()” in the following code will be activated. First, the code examines whether the user filled in the “First Name”, “Last Name”, and “Phone Number” completely. If any of these three required fields are still blank, a warning window will pop up to prompt the user to fill in the required field. Otherwise, the code “CONT\_REVIEW.SAS” will be executed and the user will be shown the web page of Figure 3. If the user clicks the “Save & Exit” button, “function exit ()” in the following code will be activated and the code “THANK.SAS” will be submitted. Each button has its own “if-then-else” logic to decide the appropriate SAS Code or URL to pass to the Application Broker.

```
put 'function save();
put '{';
put 'if ((document.forms["Contact"].elements
["fstname"].value==0) || (document.forms["Contact"].
elements["lstname"].value==0));
put '{';
put 'window.alert("Please fill out Your First Name and
Your Last Name");';
put '>';
put 'else';
put '{';
put 'if((document.forms["Contact"].elements["phone1"].
==0)||((document.forms["Contact"].elements["phone2"].value==
0)||((document.forms["Contact"].elements["phone3"].value==0)));
put '{';
```

```
(SAS code continued)
put ' window.alert("Please fill out Phone Number Completely");';
put ';';
put ' else ';
put ' {';
put '{document.forms["Contact"].elements
["_program"].value="SOD.CONT_REVIEW.SAS";';
put ' document.forms["Contact"].submit();';
put ';';
put ';';
put ';';
put ';';
put 'function exit()';
put '{document.forms["Contact"].elements
["_program"].value="SOD.THANK.SAS";';
put ' document.forms["Contact"].submit();';
put ' ';
put '</script>';
```

After the program has performed data validation and updated the SAS data set on the server, the review page (Figure 3) opens to provide feedback to the user. The user is given an opportunity to review the input.

Figure 3. Review Unit Information Page

If there is still incorrect information, the user can select the "Edit" button in Figure 3 and the screen will go back to the previous page to allow correct data entry. If the user finds no errors, he may choose to continue the survey or exit the survey and finish at a later time.

If a clickable link on the left side of Figure 3 is selected, the user will be routed to other pages of the survey, including user's operation type, service hours, holidays and semester span dates (Figures 4 through 9). The following code is used to activate the frame navigation on the web pages as demonstrated on the left side in Figure 3:

```
put '<p><a href=http://...../broker.exe?_service=sod&_program=SOD.CONTACT.SAS&_debug=0&location= &location> Operation Type Selection</a></p>';
put '<p><a href=http://...../broker.exe?_service=sod&_program=SOD.RES_FORM.SAS &_debug=0&location=&location>Resident Form</a></p>';
put '<p><a href=http://...../broker.exe?_service=sod&_program=SOD.SESSION.SAS&_debug=0&
```

```
(SAS code continued)
flag=1&location=&location>Semester Selection
</a></p>";
put "<p><a href=http://...../broker.exe?_service=sod&_program=SOD.SODSESRMB.SAS&_debug=0&flag=1&location=&location>Semester Span Dates
</a></p>";
put "<p><a href=http://...../broker.exe?_service=sod&_program=SOD.SODHLDYSEL.SAS&_debug=0&flag=1&location=&location>Holiday Selection
</a></p>";
put "<p><a href=http://...../broker.exe?_service=sod&_program=SOD.SODHLDYRMB.SAS&_debug=0&flag=1&location=&location>Holiday Span Dates
</a></p>";
put "<p><a href=http://...../broker.exe?_service=sod&_program=SOD.SODOUTPUT.SAS&_debug=0&flag=1&location=&location>Output</a></p>"
```

(Note that the left navigation bar is displayed on every webpage during this survey. It is not shown in the figures in this paper in order to make the main part of the page more readable.)

The displays shown in Figures 4 and 5 are for users to select their operation type and fill in the service hours.

Figure 4. Operation Type Page

Figure 5. Operation Services Hours Page

Type of Service	Serving Hours (Mon - Thu)		Serving Hours (Friday)	
	Start	End	Start	End
Breakfast	6:30 AM	9:30 AM	6:30 AM	9:30 AM
Brunch	N/A	N/A	N/A	N/A
Lunch	11:00 AM	2:00 PM	11:00 AM	2:00 PM
Dinner	4:00 PM	8:00 PM	4:00 PM	8:00 PM
Late Night	10:00 PM	1:00 AM	10:00 PM	1:00 AM

  

Type of Service	Serving Hours (Saturday)		Serving Hours (Sunday)	
	Start	End	Start	End
Breakfast	6:30 AM	9:30 AM	N/A	N/A
Brunch	N/A	N/A	8:00 AM	2:00 PM
Lunch	11:00 AM	2:00 PM	N/A	N/A
Dinner	4:00 PM	8:00 PM	4:00 PM	8:00 PM
Late Night	10:00 PM	1:00 AM	N/A	N/A

SAS can generate HTML drop-down menus (Figure 5) for data entry. The following code produces the HTML drop-down menus.

```

put '<td width="20%" height="23" align="center">
  <select size="1" name="Res_Break_Sh14" align="center">;
put '<option selected value="N/A">N/A</option>;
put '<option value="12:00 AM">12:00 AM</option>;
put '<option value="12:30 AM">12:30 AM</option>;
put '<option value="1:00 AM">1:00 AM</option>;
put '<option value="1:30 AM">1:30 AM</option>;
put '<option value="2:00 AM">2:00 AM</option>;
put '<option value="2:30 AM">2:30 AM</option>;
put .....;
put '</select></td>;
    
```

Figure 6. Semester Selection page

In the Semester Selection page of Figure 6, the user inputs the number of sessions in each season for his business. JavaScript is used to control the input numbers, which cannot exceed three for fall and spring semester. If a number greater than three is entered, an error message window will pop up to prompt the user to try again. This is accomplished with the following code:

```

put '<script Language="Javascript">;
put 'function Fall_validation();
put '{';
put 'if ((document.forms["SemsterSel"]
  .elements["Fall"].value!=0 ) &&(document.forms
  ["SemesterSel"].elements["Fall"].value!=1 )
  &&(document.forms["SemesterSel"].elements
  ["Fall"].value!=2 ) &&(document.forms
  ["SemesterSel"].elements["Fall"].value!=3 ));
put '{window.alert("Please input a Number less than 4
  in the Fall field");';
put 'document.forms["SemesterSel"].
  elements["Fall"].focus();';
put '}';
put 'function Spring_validation();
put '{';
put 'if ((document.forms["SemesterSel"]
  .elements["Spring"].value!=0 ) &&(document.forms
  ["SemesterSel"].elements["Spring"].value!=1 )
  &&(document.forms["SemesterSel"].elements
  ["Spring"].value!=2 ) &&(document.forms
  ["SemesterSel"].elements["Spring"].value!=3 ));
put '{window.alert("Please input a number less than 4 in the
  Spring field");'; put ' document.forms ["SemesterSel"].
  elements["Spring"].focus();';
put '}';
put '</script>;
    
```

Once the number for each semester is filled in, the Semester Span Dates Page (Figure 7) opens with fixed rows based on the number of semesters entered by the user on the semester selection form (Figure 6). Drop-

down boxes are provided for the user to select the semester beginning and ending dates.

Figure 7. Semester Span Dates Page

Session	Date Begins	Date Ends
Fall1	August 3 2002	December 30 2002
Spring1	January 3 2002	April 30 2002
Summer1	May 5 2002	July 20 2002

The Holiday Selection page (Figure 8) permits the user to select holidays that apply to his campus. When the holidays are checked and submitted, the Holiday Span Dates page (Figure 9) opens with only selected holidays displayed. The user selects the beginning and ending dates for each holiday. Some holidays are initialized with a default date (e.g. Labor Day, Thanksgiving, and Independence Day in Figure 9). These default dates can be modified by the user.

Figure 8. Holiday Select Page

Figure 9. Holiday Span Dates Page

Session	Date Begins	Date Ends
Labor Day	September 3 2002	September 3 2002
Thanksgiving	November 28 2002	November 29 2002
Indep Day	July 4 2002	July 4 2002

When the user submits the holiday span dates, the final output screen will display all the information entered by the user for final review (Figure 10). After reviewing the

collected information, the user can edit the information for this operational site, return to the login screen to enter data for another operation site, or exit the survey. Upon exiting, the user will see a final "Thank you" page (Figure 11).

Figure 10. User Final Confirm Page

**Thank you for your participation.**  
Please confirm the information you have provided.

**Location/Unit Name:** Test Unit  
**10 Digit Unit Number:** 8888888888  
**Address:** 100 Pine Street  
**City:** Orlando **State:** FL **Zip:** 32826  
**GM Name:** Lucy Luo  
**Updated information that you have provided:**  
**First Name:** Lucy **Last Name:** Luo  
**GM/Contact Phone:** 123-123-1234  
**GM/Contact Fax Number:** Not Available  
**Unit's physical location:** Physical Location on Campus

**Resident Dining Serving Hours**

Type of Service	Serving Hours ( Friday )		Serving Hours ( Mon - Thu )	
	Start	End	Start	End
Breakfast	6:30 AM	9:30 AM	6:30 AM	9:30 AM
Brunch	N/A	N/A	N/A	N/A
Lunch	11:00 AM	2:00 PM	11:00 AM	2:00 PM
Dinner	4:00 PM	8:00 PM	4:00 PM	8:00 PM
Late Night	10:00 PM	1:00 AM	10:00 PM	1:00 AM

  

Type of Service	Serving Hours ( Saturday )		Serving Hours ( Sunday )	
	Start	End	Start	End
Breakfast	6:30 AM	9:30 AM	N/A	N/A
Brunch	N/A	N/A	8:00 AM	2:00 PM
Lunch	11:00 AM	2:00 PM	N/A	N/A
Dinner	4:00 PM	8:00 PM	4:00 PM	8:00 PM
Late Night	10:00 PM	1:00 AM	N/A	N/A

**Retail Operation Type**  
Faculty Dining; Snack Bar; Convenience Store

**Retail Serving Hours**

Serving Hours (Monday)		Serving Hours (Tuesday)		Serving Hours (Wednesday)		Serving Hours (Thursday)	
Start	End	Start	End	Start	End	Start	End
8:00 AM	5:00 PM	8:00 AM	5:00 PM	8:00 AM	5:00 PM	8:00 AM	5:00 PM

  

Serving Hours (Friday)		Serving Hours (Saturday)		Serving Hours (Sunday)	
Start	End	Start	End	Start	End
8:00 AM	5:00 PM	8:00 AM	5:00 PM	8:00 AM	5:00 PM

**Semester Span Dates**

Session	Date Begins	Date Ends
Fall(1)	August 3 2002	December 30 2002
Spring(1)	January 3 2002	April 30 2002
Summer(1)	May 5 2002	July 20 2002

**Holiday Span Dates**

Holiday	Date Begins	Date Ends
Labor Day	September 3 2003	September 3 2003
Thanksgiving	November 28 2002	November 29 2002
Indep Day	July 4 2002	July 4 2002

If you need to make changes to the above information, Click [Here](#)  
If the above information is correct and you still have another subunit to fill out, Click [Here](#)  
If all above information is correct and you have no other subunits to fill out, Click [Here](#)

Figure 11. Thank You Page

**Thank you for your participation!**

You have finished our survey, and all information has been saved in our database.

## CONCLUSION

Created using SAS/IntrNet, JavaScript and HTML, the Unit Information Collection Web Page provides the Sodexho Campus Services team a method of collecting data from multiple sites across the United States. The website users enjoy the flexibility of a web-based survey and data entry conveniences such as drop-down menus. At the same time, the webmaster has the data instantly and is able to produce periodic completion reports to follow up quickly on the no responses. SAS/IntrNet integrates the SAS system and the World Wide Web to form a data entry system that reduces reporting errors and validates data in real time without the need for specialized client-side software. By combining SAS and SAS/IntrNet the data collection possibilities seem endless.

## REFERENCES

SAS Institute Inc. (2001), *SAS/IntrNet<sup>®</sup> Software: A Roadmap*, Cary, NC: SAS Institute Inc.

Klenz, B.W., "Web-Enabling SAS<sup>®</sup> Applications", 1998.  
<<http://www.sas.com/rnd/web/intrnet/papers/sesug/sesug.html>>.

## ACKNOWLEDGMENTS

The Technical Support staff at SAS Institute, as always.

This work was made possible by the unconditional support of Mr. Mohamood Bhatia, Senior Vice President of Finance, Sodexo Campus Services. Additionally, the authors would like to express their deepest gratitude to Dr. Morgan Wang, Director of the Data Mining program at University of Central Florida, for his invaluable technical support and guidance.

## CONTACT INFORMATION:

Your comments and questions are valued and encouraged.

Contact the authors at:

Xin (Lucy) Luo  
Z, Inc.  
1999 Bryan Street  
Suite 1110  
Dallas, TX 75201  
Work Phone: 214-720-6184  
E-mail Address: [lucy.luo@eia.doe.gov](mailto:lucy.luo@eia.doe.gov)

Russell E. Denslow  
Sodexo Campus Services Division  
283 Cranes Roost Blvd.  
Suite 260  
Altamonte Springs, FL 32701  
Work Phone: 407-339-3230  
Email Address: [Russell.Denslow@sodexoUSA.com](mailto:Russell.Denslow@sodexoUSA.com)

Sandra J. Archer  
Sodexo Campus Services Division  
283 Cranes Roost Blvd.  
Suite 260  
Altamonte Springs, FL 32701  
Work Phone: 407-339-3230  
Email Address: [Sandra.Archer@sodexoUSA.com](mailto:Sandra.Archer@sodexoUSA.com)

## TRADEMARK CITATION

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 registered trademarks or trademarks of their respective companies.