Paper 41-27

SAS® Software-to-Web Configuration: From Scratch to Final Products

Haiping Luo, Dept of Veterans Affairs, Washington, DC

ABSTRACT

This paper walks the reader step by step through the whole process of SAS/IntrNet® configuration and SAS-to-Web application development. The process includes installing and configuring SAS/IntrNet service, preparing SAS programs to interact with various data sources and to deliver data and graphical output to the Web, and developing Web interface to receive interactive SAS output. The walk-through uses a healthcare application as a sample to demonstrate the process. The configuration is applicable to a Windows network environment. The data source formats include text, Excel, and Access database. The reader needs to have basic knowledge of Windows platform, local area networks, base SAS, databases, internet and HTML to implement the process.

INTRODUCTION

SAS/IntrNet allows intranet/internet users to receive dynamic output from SAS programs without the need of installing SAS in their own computer systems. While using IntrNet strongly enhances SAS system's capacity, configuring SAS/IntrNet can be very tricky. The instructions on how to make SAS/IntrNet work are insufficient and scattered in installation instructions, SAS web site, online documents, and course books. The instructions often assume things that the person who is installing IntrNet may not know. This paper tries to assemble all the necessary information together to illustrate the whole process of making SAS/IntrNet deliver information to the Web. This process includes installing IntrNet software, configuring dispatcher service, and developing a SAS-to-Web application. The following sections illustrate the steps in detail and provide troubleshooting tips along the process.

BASIC SCENARIO

The process we are illustrating in this paper assumes an evaluation scenario in a Windows network environment. That is, it is assumed that you have had base SAS software installed in your PC and that now you want to test if SAS/IntrNet is suitable for your information delivery system. You have received an evaluation package from the SAS Institute. The evaluation period is 30 days. The evaluation package includes Base SAS and IntrNet products. You are hesitating to install the evaluation package because you do not want it replaces your current SAS products and forces the whole SAS software expire in 30 days. The installation steps in this paper show how to install only the IntrNet components to your current SAS system. With this installation, when the evaluation period is over, only IntrNet will stop working. For someone who is evaluating the Base SAS and IntrNet, the installation is simpler. He/she can just install the whole package then follow the configuration steps later in this paper.

SAS/IntrNet contains various services that you can use individually or in combination to address specific needs of delivering dynamic information to the web. We assume you will test one of the IntrNet services, the Application Dispatcher – Compute Service. The Application Dispatcher framework allows dynamically delivering SAS programs' output to web users.

SYSTEM SETTINGS

To install SAS/IntrNet and complete the testing, we assume that you have at least 2 PCs connected in a Windows local area network (LAN). These 2 PCs plus a LAN directory will play 4 roles:

1. A Web server
2. A SAS server
3. A PC to store SAS programs
4. A PC to test web browsing

We will call these 4 roles 'Web Server', 'SAS Server', 'Code PC', and 'Internet PC'. In a larger-scale SAS/IntrNet implementation, each of these roles can be taken by a different computer or multiple computers. In our test, we assume only limited resources can be used. We will put two roles in one PC. That is, Web Server and SAS server will be in the same computer. We will call this computer 'Computer One'. Another computer ('Computer Two') should take the roles of an Internet PC to test web delivery of SAS output. The role of Code PC will be played by a LAN server, which has file folders for LAN users to store commonly accessible files. SAS programs will be stored here to test across-network program calling.

Computer One can have Windows 95, 98, NT Workstation, NT server, or Windows 2000 installed as the operation system. We assume it is a NT workstation with Base SAS (version 8.2) installed and without Web server installation. Computer Two can be any Windows system and have no SAS installation.

We assume that you are an administrator for Computer One and have a user account in the LAN. IntrNet requires that the person who installs the software has the administrative privilege at this PC. Also after the installation, IntrNet will only run while a local administrator is logging in the PC.

We assume Computer One as the SAS Server or as the Web Server has a fixed IP address. If this IP address is not fixed, you will have to change IntrNet configuration and your files accordingly whenever the IP address changes. To find out your IP address, go to the command prompt at your PC. Type 'ipconfig'. Your IP is the one listed as 'IP Address'.

Since we assume Computer One does not have Web server installation, you need to install Web server into this PC. You can get a free Web server software from Microsoft. For NT server and Windows 2000 Professional version, Internet Information Server (IIS) is included in the system CD. For Windows 95/98/NT Workstation or NT server, you can download Personal Web Server or IIS from a NT 4.0 Option Pack located at Microsoft's web site:
http://www.microsoft.com/ntserver/nts/downloads/recommended/NT4OptPk/default.asp?FinishURL=%2Fdownloads%2Frelease%2Easp%3FReleaseID%3D28609%26area%3Dsearch%26ordinal%3D2%26redirect%3Dno

You need to save all the disks in this link site to a folder then run the setup to install the Web server into Computer One. During the installation, you will set yourself as this web server's administrator.

INSTALL AND CONFIGURE INTRNET SERVICE

The installation and configuration portion includes six steps. In our test, all six steps are done at Computer One. But in different steps, Computer One plays different roles. We will indicate the actual role of Computer One in that step in a parenthesis in the instruction. If you install IntrNet into physically different computers, a step applies to a particular role should be installed into the particular computer which plays that role in your setting. To test the success of the installation, a seventh step is added to the end of this section.

1. INSTALL INTRNET COMPONENT

This step installs an Application Server to SAS Server in Computer One. This Application Server is one of the two components in the Application Dispatcher framework. The other components in the Application Dispatcher is the Application Broker. Step 4 covers its installation. To install the Application Server:

- At Computer One (as the SAS Server), insert 'Setup' CD.
- Select SAS System setup, custom installation (to avoid replace Base SAS), Add component to SAS system. Then choose the current SAS folder (assume it is in d:\sas8) which contains SASV8.cfg file, install into the current SAS folder, select the default setting for profile: 'c:\documents and settings\all users\sas8\'.
- Select the default temporary folder for SAS use: d:\sas8\temp\
- Let SAS add items to the program box item 'the SAS System'.
- Select the component to install:
  Unselect all other components except 'SAS/IntrNet'. You will receive warning when you uncheck 'Core of SAS system', click 'OK' to close this warning message and still uncheck 'Core of SAS system'.
- Now highlight 'SAS IntrNet' component item (make sure it is still checked), click 'options' button, a sub window will open, make sure all items in this sub window are selected, then click Continue to close this sub window.
- Click next will bring you to the review settings window. The current settings are:
  Custom Setup:
  Installing to The SAS System folder.
  SAS/IntrNet on d:\sas8
- Click next.
- A 'needs next disk' window will ask for 'Solution disk 1'. Put the CD into the CD-ROM drive and click OK.
- A 'needs next disk' window will ask for 'SAS Shared components'. Put the CD into the CD-ROM drive and click OK.
- After this installation, click 'finish'.

2. INSTALL SAS SERVICE CONFIGURATION UTILITY

SAS Service Configuration Utility helps to configure connection services used by the Dispatcher:

- At Computer One (as the SAS Server), insert 'Setup' CD.
- Click 'Additional Setups', select 'SAS service configuration utility' from the drop down list.
- Click 'next' at the next window then choose a folder to store information about specific IntrNet services. Assume you use "D:\home\SASuser\IntrNet".
- Follow the rest windows to finish installing this utility.

3. CREATE A SAS/INTRNET SERVICE

Now you can use SAS Service Configuration Utility to configure a SAS session service used by the Dispatcher:

- At Computer One (as the SAS Server), go to Windows' 'Start' button, highlight 'Programs', 'the SAS System', 'IntrNet', click 'Create a New IntrNet Service'.
- An 'IntrNet Config Utility 8.2' window will open. It reads:

  This utility allows you to set up a service definition or configure the Load Manager for the SAS/IntrNet Application Dispatcher. For service definitions, you should know the type of service you wish to create and the TCP port number(s) or name(s) you wish to use (socket services only) before continuing. This utility will create a service directory containing a work area, a startup program and other required files. A batch file and Start Menu shortcut is created to start servers for socket services.

- There are three session services you can choose.
  'Socket service' is a TCP/IP service allowing the same SAS session being used by the web requests repeatedly.
  'Launch service' will launch a new temporary SAS session on the SAS server each time a web request is made.
  'Pool service' starts multiple SAS servers on the same node as the Load Manager.
- For our test, choose a socket service for TCP/IP service.
- Enter service name 'default'. This can be any name, but SAS/IntrNet samples use a service called 'default'. So in order to view those samples, we create a 'default' service. Multiple services can be created using this utility.
- Choose a home directory for this service 'default'. We will use 'D:\home\SASuser\IntrNet'. A folder called 'default' will be created under this path. Files will be copied to this folder. This folder contains: 1) SAS programs and related files which start SAS sessions for IntrNet service and define the basic session settings; 2) a 'logs' sub-folder to keep logs for the 'default' service; 3) a 'temp' folder to serve as the work area and hold temporary files for the SAS session.
- Choose a TCP port for this service. This port is a small memory area in the SAS Server which listens to Web Server's requests. You should choose a port not used by other services. We will choose 5008.
- Leave administrator's password field empty for this service and click 'next'.
- The next window tells you that the Utility is ready to create the service using the indicated settings.
- The next window says that the service 'default' has been created. You need to follow the instruction in the box to continue configuration. The instruction is (words in parentheses are my explanation):

  Complete the following steps to use this service:
  * Install the Application Broker on your Web server machine.
  * Create a service definition similar to the following example in your broker.cfg file (at the Web server) for the 'default' service. If you can edit broker.cfg on this system (i.e., the Web server), you can cut and paste the example directly into the editor now.

```
  SocketService default
  ServiceAdmin "[SAS admin name]"
  ServiceAdminMail "[SAS admin-email]@[your-site]"
  Server       [your (SAS Server's) IP address]
  Port         5008
  FullDuplex  True
```

- You should create a documentation file to save this information and to save other installation information. After creating the documentation file, cut and paste the above instructions into that file then save it.
- Click 'next' then 'finish' to exit the Utility.
- Now go to Windows 'Start' button, Programs, the SAS System, IntrNet, a 'default service' item has been added to the submenu under IntrNet. You can use the Service Directory under 'default' submenu to view the contents of the 'default' service folder.

4. INSTALL APPLICATION BROKER IN WEB SERVER

The Application Broker is a CGI tool installed at Web Server. This CGI tool passes all incoming web users' requests to SAS Server. SAS Server executes requested SAS programs in batch mode and sends the results to the Application Broker, which then sends the results to requesting web users.

- At Computer One (as Web Server), insert 'Client-side Components' CD.
- A browser window for the 'index.html' page in this CD will open. Click 'SAS/IntrNet Software'.
- Click 'Install' icon to install 'SAS/IntrNet software for Version 8'. This will open another web page.

- At this page, go to 'CGI Tools for the Web Server' table, then go down to 'Windows' row. Click and read the instructions for installing CGI tools for Windows.
- Then go back to the table page, click 'websrv.exe' at this ‘Windows’ row to start installation.
- When the ‘Open or Save’ window pops up, select 'Open' websrv.exe instead of saving it.
- A 'SAS IntrNet CGI Tools' window will open, click 'Next'.
- Click 'next' for copyright notes. Click 'next' to leave 'Typical' as the installation type at the next window.
- The next window lets you choose the web server folder to install IntrNet samples and supporting files. These files have to be in a virtual directory called 'sasweb' in a web server folder. We will put this ‘sasweb’ folder under the internet root directory 'c:\inetpub\wwwroot\'. You can reach this folder from the web by calling URL 'http://your-web server-name/sasweb'. If you want to choose another folder, you can browse from this window to find the folder and add a 'sasweb' folder as a sub directory. This folder above the sasweb should be defined by Web Server as a virtual directory.
- The next window asks for the physical path to put IntrNet CGI executables. Choose Web Server’s 'c:\inetpub\wwwroot\scripts' to put the CGI files. ‘Broker.exe’ will be put into this folder. You may choose other folders, but SAS/IntrNet samples use this path to find broker.exe.
- If this folder does not exist, a window will ask if you want to create it. Click 'yes'. Then give the URL address for the new script's directory as ‘[your-web-server-IP]/scripts'.
- The next window lists the current settings for the installation:

```
CGI executable and configuration files will be copied to
  c:\inetpub\wwwroot\scripts
Sample files will be copied to
  c:\inetpub\wwwroot\sasweb\IntrNet8
MDDB Report Viewer files will be copied to
  c:\inetpub\wwwroot\sasweb\IntrNet8\MRV
Utility files will be copied to
  c:\inetpub\wwwroot\sasweb\IntrNet8\tools
Graphics components will be copied to
  c:\inetpub\wwwroot\sasweb\graph
```

- Save this settings to your IntrNet installation documentation file, then click 'next'.
- Check 'open readme file' then click 'finish'.
- The readme file for CGI tools tells you how to test from a web browser if the installation is successful. This same information is also in the ‘Client-side Components’ CD and SAS web site: http://www.sas.com/rnd/web/intrnet/
- We need to do more configurations before we can test the CGI installation.

5. CONFIGURE BROKER.CFG IN WEB SERVER

The Application Broker program, broker.exe, at Web Server relies on a configuration file, broker.cfg, in the same folder to find SAS IntrNet services. In Step 3, we have created one IntrNet service, the socket service ‘default’, at SAS Server. We need to tell broker.exe at Web Server where to contact this ‘default’ service. We will cut and paste the configuration information from Step 3 to broker.cfg file then edit it accordingly.

- At Computer One (as Web Server), open a text editor such as WordPad or NotePad (do not use a word processor like ‘Word’). From the text editor, open the configuration file under Web Server’s root folder: 'c:\inetpub\wwwroot\scripts\broker.cfg'.
- This file has many ready-to-use commands commented out by ‘#’ at the first column. You can simply remove the # sign and put a command in use.
- We will insert the information for ‘default’ service into the ‘socket service section’ as:

```
SocketService default
ServiceAdmin "[SAS admin-name]"
ServiceAdminMail "[SAS admin-email]@[your-site]"
Server       [your (SAS Server’s) IP address]
Port         5008
FullDuplex  True
```

- Then you can put SAS administrator’s name and email in the text you just pasted.
- For Server IP address, put your SAS Server’s IP address rather than its friendly URL address given by the SAS configuration utility. This can prevent unsolved URL when domain name server or file is unavailable.
- Then find, uncomment and edit the following commands:

```
SelfURL http://Web-Server-IP/scripts/broker.ex
Administrator "web admin name"
AdministratorMail web-admin@yourdomain.com
```

- Leave other commands as it and save this broker.cfg file.

6. CUSTOMIZE APPSTART.SAS IN SAS SERVER

A SAS IntrNet service is started by a SAS program called appstart.sas. This program is associated with a specific service. In Step 3, we created a socket service ‘default’. This ‘default’ service has an appstart.sas program located in its service directory. ‘Appstart.sas’ tells the SAS Session where SAS code, data sources, and library will be located. We need to customize this appstart.sas file to point to our test code.

- At Computer One (as SAS Server), open a text editor or SAS program editor. From the editor, open appstart.sas in the home directory of service ‘default’: D:\home\SASuser\IntrNet\default\appstart.sas.
- Add the following statements before the ‘run;’ statement at the end of the program to define the location of our test programs, SAS datasets, and libraries:

```
* defines path for SAS library;
allocate library mylib 'q:\sastest';
* defines path for SAS programs;
allocate file myfile 'q:\sastest';
* defines path for SAS datasets;
allocate library mydata 'a:\test';
* lists program filename - myfile;
* lists SAS library libname – mylib;
proglibs myfile mylib;
* list common SAS datasets libname for all
programs;
datalibs mydata;
```

- The above paths point to a ‘q:\’ drive, which is a directory at a LAN server. This directory is mapped to SAS Server as ‘q:\’ drive. SAS programs will be put in ‘’q:\sastest’ to test across-network processing.
- Save appstart.sas.

7. TEST THE INSTALLATION

Now all installation and configuration steps are done. We can test the Installation from a web browser:

- First at SAS Server (Computer One in our case), go to Windows ‘Start’, Programs, The SAS System, IntrNet, default service. Click ‘Start Interactively’. This starts the SAS IntrNet Session under the ‘default’ service setting. A popup window ‘SAS/Appsrv 5008’ will stay at the desktop. It tells which program is running, where the output goes, and where the service log goes. Now SAS Server is ready to process requests from the Application Broker.
- Second at Web Server, check the Internet Service Manager to make sure the Default Web Site is not stopped.
- Open a web browser at Computer One, type: http://[web-server-ip]/scripts/broker.exe?

- The browser should return Application Dispatcher's version information and SAS IntrNet links.
- Clicking the 'IntrNet Samples' link will bring you to the sample home page installed in Web Server. You can explore sample applications' html files and SAS programs from the Application Dispatcher link there.
- You can go through the same three actions at Computer Two to test reaching the Application Broker over LAN.

If your test runs into troubles, you may try the following tips:

- If you cannot start the 'default' service interactively, it may be because that you did not log in as the local administrator; it may also because the port your defined in Step 3 has been occupied by other services. If you logged in as Computer One's administrator, you may want to delete the 'default' service and create it again following Step 3 with another port number. The instructions to delete a service is in the 'Client-side Components' CD as well as in SAS web site:
  http://www.sas.com/rnd/web/intrnet/dispatch/srvwin.html
  After the new 'default' service is created, go to broker.cfg at Web Server (see Step 5's instructions), change the port number for the new 'default' socket service accordingly. With these changes, the new 'default' should start.
- If you get 'page not found' message when you try to open broker.exe at Computer One, first make sure the Internet service at this Web Server has been started. You can use 'Internet Services Manager' of Web Server to check and start 'Default Web Site'. Also you should make sure that broker.exe has been placed physically under the scripts subfolder of the 'wwwroot' folder as we specified in Step 4. Then you can go to broker.cfg in Web Server (Step 5) and check if:
  Server IP address matches SAS Server IP address; and
  SelfURL address matches Web Server IP address and scripts' virtual path.
  In our case, both IP addresses are the same Computer One IP address.
- If you can browse broker.exe from Computer One but not from Computer Two, check first if you typed in correctly Web Server's IP address and other URL information at Computer Two. You may also use 'ping' to test if Computer Two has a connection to Computer One. Here is how to test:
  Open 'command prompt' at Computer Two.
  Type 'ping [Web-Server-IP-address]'.
  If you receive 'Request timed out', the TCP/IP connection between Computer Two and Web Server is broken. You need to ask the LAN helpdesk to check and fix your connection.
- If you can reach broker.exe but cannot run any IntrNet sample applications, you should check if:
  The running service is the 'default' service; and
  Sample files are in the path as indicated at Step 4.
  It is possible that you configured everything correctly but used a different path for broker.exe or sample folders. In this situation, you can still run your own programs although the given SAS samples will not run.

Once the installation and configuration pass the test, we can start to develop our own applications.

DEVELOP A TEST APPLICATION

In this section, we will develop a simple sample to illustrate the process of IntrNet application development. Our data sources are inpatients admission data from several hospitals. These data are in three formats: text, Excel spreadsheet, and Access database. We will develop an IntrNet application to allow web users to calculate average length of stay by the group(s) they choose. This application has three components: web user interface, data sources, and SAS code. The following subsections demonstrate how to make these three components work together to deliver information to web users. You can get the complete files of this application from the links listed at the end of this paper.

WEB USER INTERFACE

Our web user interface is a HTML page located at Web Server's root directory:
C:\inetpub\wwwroot\ sas-intrnet-test-2002-1-9.htm

In this HTML page, a form is established to collect user's choice on how to group data when calculating average length of hospital stay:

```
<form name="form1" method="post"
action="http://[Web-Server-
IP]/scripts/broker.exe">
<input type="hidden" name="_service"
value="default">
<!-- myfile is a filename defined at
appstart.sas as 'q:\sastest' -->
<input type="hidden" name="_program"
value="myfile.intrnet-test-2002-1-9.sas">
 <table width="75%" border="1">
  <tr>
    <td>
  Select the first level of grouping:</td>
  <td>
  Then select the second level of
grouping:</td>
  <td>
  Then select the third level of
grouping:</td>
  </tr><tr>
  <td>
  <select name="by1">
     <option value="">
     <option value="hosp_id">Hospital
     <option value="admyear">Year
     <option value="dxcode">Diagnosis
  </select></td>
 <td>
  <select name="by2">
     <option value="">
     <option value="hosp_id">Hospital
     <option value="admyear">Year
     <option value="dxcode">Diagnosis
  </select>
  </td>
  <td>
  <select name="by3">
     <option value="">
     <option value="hosp_id">Hospital
     <option value="admyear">Year
     <option value="dxcode">Diagnosis
  </select>
  </td></tr>
  <tr>
  <td colspan=3 align=center>
  <INPUT TYPE="CHECKBOX" NAME="_DEBUG"
VALUE="131">Check Here to Show SAS
Log&nbsp;&nbsp;&nbsp;&nbsp;
  <input type="submit" name="Submit"
value="Submit">
```

In this form, broker.exe is called in the 'action' attribute of the form tag; IntrNet service 'default' is called in the hidden input field named '_service'; SAS code is called in the hidden input field named '_program'; the '_debug' check box allows the user to view SAS log. Note that the program file 'myfile.intrnet-test-2002-1-9.sas' has a three-level name. The first level 'myfile' is the filename for program path defined in 'appstart.sas'. The second and third levels are the actual program file name and extension. The extension must be '.sas'.

There are three macro variables defined in the body of the form, by1, by2,and by3. Each of them may have a value of empty, ‘hosp_id’ (Hospital), ‘admyear’ (Admission Year), or ‘dxcode’ (Diagnosis). The value of these three macro variables will be passed to the SAS program to determine how to group data while calculating length of stay. The user may select the first, second, and third group to do the calculation. For example, if the user selects ‘Hospital’ as by1 and ‘Year’ as by2, the average length of stay (ALS) will be a table of ALS by hospital then by year. If the user submits the form without making any selection, the result will be ALS for all hospitals, all years, and all diagnoses.

DATA SOURCES

Our three data sources are: a hospital ID and name look up file in text format; a diagnosis code and name table in an Access database; and an admission record table in an Excel file. We assume our data are located at different places across the LAN. We will define Data Source Name (DSN) to reach the Access and Excel tables at SAS Server. You may use DSNless connection strings to reach different data sources. The key point on data source connection is that the sources should be reachable by SAS Server no matter where the SAS code is located. This is because that all programs are executed at SAS Server in a batch mode even the code may sit in a different system.

In our case, 2 User DSNs are defined at SAS Server using ODBC Administrator window. One is called ‘sugi27test’, pointing to a ‘data’ range defined in the admission record table in the Excel file. The other is called ‘sugi27testdb’, pointing to the diagnosis databases in Access.

SAS CODE

With the above settings, we can now develop a SAS program accordingly. Our program’s path is:
'q:\sastest\sas-intrnet-test-2002-1-9.sas '
The program has 5 portions: 1) define system options and global macro variables; 2) obtain and process data; 3) calculate average length of stay based on user-submitted macro value; 4) draw horizontal bar charts to illustrate the results; 5) clean up the session.

In the first portion, three macro variables are defined as global macro variables using the same name submitted by the HTML form: by1, by2, by3. This will maintain their values throughout the running of this requested job. Then system options and graphic options are selected. The path and filename for the text lookup file is also defined.

```
** grouping variables selected by user;
%global by1 by2 by3;
options symbolgen ps=100 nocenter nodate
nonumber;
goptions transparency device=gif noborder;
%let path=q:\sastest\;
filename h "&path.intrnet-test-2002-1-9.txt";
```

In the second portion, data in Excel and Access are obtained through ODBC connections. The text file is readed. Then all data are merged.

```
proc sql;
** get hospital stay data;
 connect to odbc(dsn=sugi27test);
     create table stays as select  * from
connection to odbc
    (select * from data);
  disconnect from odbc;
** get diagnosis code and name table;
 connect to odbc(dsn=sugi27testdb);
     create table dxname as select  * from
connection to odbc
    (select * from DiagnosisCode);
  disconnect from odbc;
quit;
```

```
. . .
```

When the data processing is completed, put an ods statement to direct future output to the requesting web browser:

```
ods listing close;
ods html body=_webout path=&_tmpcat
(url=&_replay) rs=none;
```

The ods options send one piece of output directly back to the web browser by writing to fileref _webout; send other pieces of output to a temporary location (&_tmpcat); then include special replay URL so that the body page links back to this server to retrieve the output from the temporary storage location. This setting allows multiple graphic output being replayed at the same output page.

Next, the calculation of average length of stay and the generation of graphics are enclosed in a macro ‘als’. Using a macro enable us to use %if statement to execute statements conditionally. We need to decide if we should calculate by group. If the user did not select any group, then the calculation will be for all data. When the user selected groups in a certain order, we have to check each possibility to determine how to write the statements. The following statements illustrate the logic:

```
* sort only when a by group is chosen;
%if &by1 ne or &by2 ne or &by3 ne %then %do;
    proc sort data=stays2;
            by &by1 &by2 &by3;
%end;
* calculate sum of stay days and total
patients;
proc means sum data=stays2 noprint;
    var ls;
* add by group only when by1, by2, or by3 has
at least one value;
%if &by1 ne or &by2 ne or &by3 ne %then %do;
    by &by1 &by2 &by3;
         id
         %if &by1 ne %then %do;
             &by1.name
         %end;
         %if &by2 ne %then %do;
             &by2.name
         %end;
         %if &by3 ne %then %do;
             &by3.name
         %end;
            ;
 %end;
 output out=lssmfile n=totpat sum=lssum;
```

After the calculation is done, a cleanup portion is put at the end to kill all datasets in work library and comment out possible hanging parenthesis, quotation marks, macro variable values, and like. This is useful because a SAS IntrNet session is a continuous session. A hanging error or value may affect the results of later requests.

```
ods html close;
ods listing;
proc datasets kill library=work;
quit;
run;
*'; *"; *); */;
%let by1=;
%let by2=;
%let by3=;
```

TEST AND TROUBLESHOOT THE APPLICATION

Once the program is completed, we can test it from Computer One and Computer Two. From a web browser at either computer, type:
http://[Web-Server-IP]/ sas-intrnet-test-2002-1-9.htm
The ‘Average Length of Stay’ page should open. You may make

any selection then submit the request. If it returns the results as your expected, the application development is successful. If it returns error or nothing, you may check the ‘Show SAS Log’ box then submit again to see if a log can be returned to the browser.

Usually we cannot expect a success in the first try. Here are some common troubleshooting tips:

- Make sure the default SAS service and the default web site has started.
- Try to run the SAS program in regular session (not in web session) to see if the code can get the expected results. You can set values for the macro variables using %let statement and comment out the ‘ods html’ statements.
- SAS’ program editor sometime generates some ‘□’ in the text. These symbols may prevent a program to run correctly in IntrNet service. You may want to open a text editor such as ‘EditPlus’ to remove all such symbols and not to edit the program in SAS editor.
- If you have random IP for your web server or SAS server, the IP address might have changed since your last run. Use ‘ipconfig’ at the command prompt to check the address. If IP has changed, you need to go through configuration files to correct those IP addresses.
- Recheck to make sure all the paths in program code and in configuration files are correct. SAS log and the log file in the IntrNet service directory can help you diagnosis the source of the problem.

CONCLUSION

Using SAS/IntrNet can bring the versatility of SAS applications to wide range of internet and intranet users. This paper illustrated the basic process of installing, configuring, and utilizing IntrNet’s Application Dispatcher component. SAS/IntrNet has more components other than the Application Dispatcher Compute Service. Hopefully when you are familiar with the installation process, trying other IntrNet services will be easier.

NOTE

Due to the limitation of this publication, the complete application files and any update to this paper can be found in this link: http://chinafoundation1.org/hp/sugipapers.htm

ACKNOWLEDGMENTS

I want to thank Danielle Gao and Jin Dai at the Economics Policy Institute in Washington, DC for their help and support on completing the evaluation process.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Author Name	Haiping Luo
Company	Dept of Veterans Affairs
Email:	haiping.luo@mail.va.gov