

Delivering Dynamic Content with the SAS® Stored Process Web Application

Frederick Pratter, Eastern Oregon University, La Grande, OR

ABSTRACT

Many Web developers have been moving away from the CGI technology that was the basis for SAS/IntrNet® toward Microsoft .NET or cross-platform Java Web applications. The SAS® *Stored Process Web Application*, new in Version 9, can be used by developers with little or no Java programming experience; under some circumstances it can be used as a replacement for SAS/IntrNet. This paper will introduce SAS Stored Processes, show how they may be accessed from another application, such as Enterprise Guide and the Stored Process Web Application, and demonstrate a sample Web Application.

OVERVIEW: JAVA WEB APPLICATIONS AND SAS STORED PROCESSES

The *SAS Stored Process Web Application*, new in Version 9, can be used by developers with little or no Java programming experience; it can be considered a replacement for the SAS/IntrNet product for some sites. To review, a Java Web Application is a collection of HTML forms, JavaServer Pages and servlets. In order to provide Web content, these Java components require a *Web container*. The Apache Tomcat servlet engine is currently the most widely used platform, but compatible servlet engines are also available from BEA (*WebLogic*) and IBM (*WebSphere*). There are also a number of other open source Web container projects, such as Jetty and Winstone, but these have not been validated to work with SAS.

A SAS Stored Process is simply a SAS program that has been registered with the Metadata Server so that it can be accessed from another application, such as Enterprise Guide or the Stored Process Web Application using the SAS Open Metadata Architecture. (See Pratter, 2004 for more information about SAS Integration Technologies and the OMA.)

At present, there are three ways to use SAS Stored Processes:

1. *IOM Direct Interface Stored Processes* were introduced in Version 8 and operate only on a SAS Workspace Server.
2. The *Stored Process Service* application programming interface (API) can be used to run stored processes, either from JavaServer Pages, servlets, custom tagsets and/or Java applications. The Stored Process Service API requires *SAS Foundation Services*.
3. The *Stored Process Web Application* is an alternative to the SAS/IntrNet Application Broker; like the earlier approach, it requires no Java programming experience on the part of the developer.

Information on the first of these is available from the *IT Developer's Guide: Stored Processes* under "IOM Direct Interface Stored Processes". This approach is deprecated in version 9, although it continues to be supported. The Stored Process Service API is used to develop Java applications and is beyond the scope of this presentation. This focuses on the second new component introduced in Version 9, the SAS Stored Process Web Application.

USING THE SAS STORED PROCESS WEB APPLICATION AS A REPLACEMENT FOR SAS/INTRNET

Many Web developers have been moving away from the older Common Gateway Interface (CGI) technology toward Microsoft .NET or cross-platform Java Web applications. The SAS/IntrNet Application Broker uses CGI to pass requests to the Application Server. Consequently, it avoids many of the problems usually associated with process-intensive CGI applications, and few performance related bottlenecks have been reported. Nevertheless, there has been continuing interest both within and outside SAS in developing a more modern approach. The SAS Stored Process Web Application potentially can replace SAS/IntrNet at some sites.

Unfortunately, since the new SAS components are currently available only for Windows-based servers, those sites that run Web services from a UNIX host cannot convert to SAS Stored Processes, at least until the Web Infrastructure Kit is available for more platforms. In addition, even at sites that run Windows-based Web hosting, a further issue is that Microsoft Internet Information Server (IIS) is not compatible with Java applications and does not support servlet containers. You would have to install Tomcat or another servlet engine alongside IIS in order to use the SAS Stored Process Web Application. In this case

you definitely would not want to install the Apache HTTP server, since there can be only one listener on port 80 at a time. Finally, as described in the following section, the new technology requires a substantial effort by the SAS system administrator to set up; sites without the luxury of a full-time administrator may not want to convert yet.

The advantages of the new software, however, are significant enough that sites where it is possible to make the transition should do so, at least for new applications. The first part of this presentation, then, explains the procedures that are necessary to install the new software. Once this has been accomplished, creating Web programs that use SAS Stored Processes is relatively easy, and does provide performance and security enhancements that may be worth the effort. (For more information about converting from SAS/IntrNet to the Stored Process Web application, see the section “Converting SAS/IntrNet Programs to Stored Processes” in the *IT Developer’s Guide: SAS Stored Processes*).

THE SAS STORED PROCESS WEB APPLICATION

INSTALLING THE WEB INFRASTRUCTURE KIT

In order to create a SAS Stored Process Web Application it is necessary first to install two new SAS IT components: the *SAS Web Infrastructure Kit* and *SAS Foundation Services*. (This is documented in the *Developer’s Guide* under “Stored Process Software Requirements”, although curiously it is not mentioned in the *IT SAS Web Infrastructure Kit* documentation.) In Version 9.1 these two products are available from the installation disk labeled *SAS Client-Side Components Volume 2*, although in this context they are not client-side components at all but mid-tier elements that must be installed on the system where the SAS Metadata server is running.

You also should install the *SAS Management Console 9.1* on a client workstation with access to the SAS server. In addition, *SAS Enterprise Guide 3.1* is handy for creating and testing stored procedures, although not essential.

Two other required 3rd-party software components are the correct version of the Java Development Kit (JDK) and a validated servlet container such as Tomcat 4.1. The installation media should include two disks labeled *Third Party Software Components*:

- Volume 1 contains the *Java Software Development Kit Version 1.4.2_04* validated for use with the SAS components. (Note that as of this writing, this release of the JDK is no longer available from Sun. The SAS installation instructions further suggest that you turn off automatic updates for the product.)
- Volume 2 has links to validated releases of two other products: *Tomcat Version 4.1.18* and *Apache Web Server 2.0.45*. These three components (the JDK, the servlet engine, and optionally the Web server) should be installed before the SAS products. Volume 2 also has links to the BEA and IBM servlet container download pages, if these are preferred to Apache Tomcat.

Finally, it is important to note that as of Release 9.1.3 of SAS Integration Technologies, the *Web Infrastructure Kit* is only available for the Windows platform. Since most Web sites that use the Tomcat servlet engine run Linux or some version of UNIX, this is unfortunate. The developers at SAS are aware of this problem and are working to make the WIK available for more platforms. The best advice, as always, is to check with a SAS Marketing Representative, who can help develop a site plan with the most appropriate technologies available. For the moment, in order to use the SAS Stored Process Web application it is necessary to configure a Windows system as the Web server as well as the SAS server.

The following table, based on the “Stored Process Software Requirements” section in the *IT Developer’s Guide*, summarizes the required software configuration:

Software	Windows Server	Windows Client
Servlet Container (e.g., Tomcat 4.1)	Required	
HTTP server (Apache 2)	Optional	
SAS Web Infrastructure Kit (WIK)	Required	
SAS Foundation Services	Required	Optional

Base SAS, SAS/GRAPH, SAS Integration Technologies Version 9.1	Required	
Java Development Kit (JDK 1.4.2)	Required	Optional
SAS Management Console 9.1	Required on either server or client	Required on either server or client
SAS Enterprise Guide 3.1		Optional

Table 1. Stored Process Software Requirements

First, install Tomcat (or one of the other two approved servlet engines) on the Windows host. The Apache Web server is optional, since Tomcat can also serve HTML as well as JSP and servlets.

Next install SAS, the correct JDK (not Java 1.5), Foundation Services and the WIK, in that order, on a Windows server. The Management console can be installed on the server or on a network client. If the server is in a remote location, it is much more convenient to have it on your desktop system. As noted above, you probably want to have EG 3.1 on the client desktop, since it will make the process of developing and testing Stored Processes much simpler.

Finally, if you are going to be developing Java programs to use the Stored Process Service API, you will need the correct JDK and Foundation Services on your desktop. If you have AppDev Studio 3.0 installed on the client, *webAF* can be used to create Java Web applications using the Foundation Services classes (see the second part of this chapter for examples).

The IT installation procedure has been improved considerably in Release 9.1.3 but it is still a laborious and time-consuming job. **Table2** lists the nine SAS IT components on the *Version 9.1 SAS Client-Side Components, Volume 2* installation disk. This table is provided to clarify the somewhat intimidating selection of installation choices. Presumably, this arrangement will change in future releases, but this table is nonetheless useful if only as a guide to the platforms for which these components are currently available and where to find documentation on them.

1. SAS Integration Technologies Documentation 9.1
Platforms: Windows, 64-bit Enabled Solaris, 64-bit Enabled AIX, HP-UX IPF, z/OS, 64-bit Enabled HP-UX, Linux, Tru64 UNIX
Description: Installs a local copy of the on-line documentation. Includes the following volumes, plus a glossary of terms: ▪ Technical Overview ▪ Server Administrator's Guide ▪ Administrator's Guide ▪ Administrator's Guide (LDAP Version) ▪ Developer's Guide ▪ SAS Web Infrastructure Kit: Administrator's Guide ▪ SAS Web Infrastructure Kit: Developer's Guide
2. SAS Web Infrastructure Kit 1.0
Platforms: Windows, Solaris (64-bit), AIX (64-bit), HP-UX IPF
Description: Create Web applications to access SAS data and services; requires a J2EE servlet container such as Apache Tomcat, BEA WebLogic or IBM WebSphere.
Documentation: a) Technical Overview: SAS Stored Processes b) Administrator's Guide: SAS Stored Processes c) Developer's Guide: SAS Stored Processes d) SAS Web Infrastructure Kit: Administrator's Guide e) SAS Web Infrastructure Kit: Developer's Guide

3. SAS Foundation Services 1.1
Platforms: Windows, 64-bit Enabled Solaris, 64-bit Enabled AIX, HP-UX IPF, 64-bit Enabled HP-UX, Linux
Description: A set of Java classes to provide the following services: ▪ metadata and content repository access ▪ stored process execution ▪ IOM client connection service for Java ▪ dynamic service discovery ▪ user authentication ▪ profile management ▪ session context management ▪ activity logging ▪ event management ▪ information publishing
Documentation: a) Technical Overview: Foundation Services b) Administrator's Guide: SAS Foundation Services c) Developer's Guide: SAS Foundation Services d) Developer's Guide: Java Clients e) Developer's Guide: Foundation Services API
4. SAS BI Web Services for .NET 1.3
Platforms: Windows only
Description: Provide Web Services for the .NET framework; used to invoke and list SAS Stored Processes. BI Web Services require a connection to a SAS Metadata Server and a SAS Stored Process Server.
Documentation: a) Technical Overview: SAS BI Web Services b) Developer's Guide: SAS BI Web Services
5. SAS BI Web Services for Java 1.0
Platforms: Windows, 64-bit Enabled Solaris, 64-bit Enabled AIX, HP-UX IPF
Description: Provide cross-platform Web Services for Java; used to invoke and list SAS Stored Processes. BI Web Services require a connection to a SAS Metadata Server and a SAS Stored Process Server.
Documentation: a) Technical Overview: SAS BI Web Services b) Developer's Guide: SAS BI Web Services
6. SAS Integration Technologies Client for Windows 9.1
Platforms: Windows only
Description: Windows application to communicate with a SAS server; installed automatically as part of the BAS SAS or IT client installation, this executable is only required if the IT client is installed by itself.
7. SAS Integration Technologies Administrator 1.6
Platforms: Windows only
Description: Java application to manage Object Servers, Spawners and the publishing framework; requires an installed LDAP server.
Documentation: Administrator's Guide (LDAP Version): Getting Started - Using the Integration Technologies Administrator

8. SAS Package Reader 1.6
Platforms: Windows, 64-bit Enabled Solaris, 64-bit Enabled AIX, HP-UX IPF, 64-bit Enabled HP-UX, Linux, Tru64 UNIX
Description: Client application to retrieve contents of a SAS package as an archive file, usually as an attachment to an e-mail message. An archive is denoted by an .spk file extension, which is an abbreviation for “SAS Package”.
Documentation: Developer’s Guide: Publishing Framework - SAS Package Reader
9. SAS Subscription Manager 1.5
Platforms: Windows, 64-bit Enabled Solaris, 64-bit Enabled AIX,
Description: Java applet that interacts with an LDAP server to provide services to manage subscriptions; the documentation indicates that this applet will not be supported in future releases of IT.
Documentation: Developer’s Guide: Publishing Framework - SAS Subscription Manager

Table2 SAS Integration Technologies Components: Installation

CONFIGURING THE SERVER

The only tricky part of all this is installing the WIK. The pre-installation checklist for setting up required user accounts indicates that six specific local accounts should be created on the Metadata server:

- SAS Administrator (**sasadm**)
- SAS Demo User (**sasdemo**)
- SAS General Server (**sassrv**)
- SAS Guest (**sasguest**)
- SAS Trusted User (**sastrust**)
- SAS Web Administrator (**saswbadm**)

Make sure to create these account on the Windows host system first, before installing the WIK—you will need to specify them as you are installing the software. After the WIK has installed successfully there should be a file called `instructions.html` in the installation directory `\SAS\9.1`, specifying the manual configuration steps required to complete the deployment. A sample set of steps is illustrated in **Figure 1**; the actual steps you will see are based on the specific products licensed for your site and the options provided during the initial installation.



Throughout this document, this image indicates that an automation script has been provided to execute the steps of this section. After executing any of the automated scripts, you may need to refresh the view in SAS Management Console in order to see the updates.

High Level Overview of the Steps

Note: The steps listed below were generated based the software you planned to configure on this machine.

1. Start your Metadata Server
2. Start the SAS Management Console
3. Define your Metadata Repository
4. Define your Metadata Users
5. Defining Default Authorizations
6. Define your SAS Application Server
7. Define your Stored Process Server

8. Define your OLAP Server
9. Define your Data Step Batch Server
10. Define your Object Spawner
11. Define your SAS/CONNECT Server
12. Define your SAS/CONNECT Spawner
13. Define your SAS/SHARE Server
14. Define your Job Scheduler Server
15. Define your HTTP Server
16. Define the SAS Foundation Services to the metadata
17. Load SAS Stored Process samples
18. Load Web Infrastructure Kit "primer" metadata
19. Start your Object Spawner
20. Start your OLAP Server
21. Start your SAS/CONNECT Spawner
22. Start your SAS/SHARE Server
23. Start your SAS Services Application
24. Deploying your Web Applications
25. Start your Tomcat Server
26. Using your Applications
27. Getting More Information

Figure 1. Web Infrastructure Kit Configuration

The biggest change between Version 9.1 and 9.1.3 is the introduction of the batch scripts, indicated by the  symbol. Assuming that you have performed a generic installation and that all of the software has been set up correctly, the automation scripts should simplify the process substantially.

In addition to the configuration instructions, a second information file is supplied under the SAS home directory in Web\Portal2.0.1\wik_readme.html. This document provides additional instructions for installing Version 1.0 of the SAS Web Infrastructure Kit and contains information for setting up security for the different servlet engines.

The most important issue in the WIK readme file relates to environment variables and how Tomcat is installed. On a Windows host running Windows 2000, XP or some other NT descendent, Tomcat can be run as a background service that is loaded automatically when the system boots up. The Tomcat 4.1 installation procedure for Windows includes several optional check boxes. By default, "Install as NT Service" is not checked. If you do check this box (and it is recommended) after the installation you will have to remove the Tomcat service and replace it; the setup instructions are included in the wik_readme.html instructions.

First, to uninstall the service, type the following at a command prompt window:

```
%CATALINA_HOME%/bin/tomcat.exe -uninstall "Apache Tomcat 4.1"
```

Next, reinstall with the SAS specified options; these can be implemented as a batch file as shown in **Example 1** below.

```
set JAVA_HOME=C:\j2sdk1.4.2_05
set CATALINA_HOME=C:\Tomcat4.1
set CATALINA_OPTS=-Xms512m -Xmx1024m -server
    -XX:-UseOnStackReplacement -Djava.awt.headless=true

rem The following command should be on a single line
%CATALINA_HOME%/bin/tomcat.exe
    -install      Apache-Catalina      %JAVA_HOME%/jre/bin/server/jvm.dll    -
    Djava.security.manager
```

```
-Djava.security.policy=%CATALINA_HOME%/conf/catalina.policy
Djava.class.path=%CATALINA_HOME%/bin/bootstrap.jar;
    %JAVA_HOME%/lib/tools.jar
-Dcatalina.home=%CATALINA_HOME% %CATALINA_OPTS% -Xrs
-start org.apache.catalina.startup.BootstrapService
-params start
-stop org.apache.catalina.startup.BootstrapService
-params stop
-out %CATALINA_HOME%/logs/stdout.log
-err %CATALINA_HOME%/logs/stderr.log
```

Example 1. Tomcat Installation Batch File

If you do not install Tomcat as a service, then you will need to restart the Tomcat servlet engine every time the host reboots. Also, you cannot change your mind and run the above batch file later, since the Tomcat 4.1 executable is only loaded if you install it initially as a service.

To start the Web engine manually, be sure to use Start→Programs→SAS→9.1→Start Tomcat rather than Start→Programs→Apache Tomcat 4.1→Start Tomcat in order to start the application, since the SAS installed batch startup file specifies some additional functions that are required for the Stored Process Web Application. In order to stop Tomcat, use the default selection Start→Programs→Apache Tomcat 4.1→Stop Tomcat.

It is to be hoped that in future releases the instructions in the two instructions files will be combined in order to simplify the already miserable life of your SAS system administrator.

In any event, assuming everything is now installed correctly, it is time to write a SAS Stored Process web application.

Using the SAS Stored Process Web Application

In order to illustrate how to convert a SAS/IntrNet program to use the Stored Process Web Application, we shall describe a simple example program that uses PROC REPORT to display data from the SHOES dataset supplied with the default SAS installation. For this example, the program is stored as shoes.sas in the “My Documents” folder for user sas on the remote host.

```
%* Sales report Example - Display Product by Region;
%macro salesrpt;

    %global region;
    proc report data=sashelp.shoes;
        by region;
        %if (&region ne ) %then %do;
            where region="&region";
        %end;
        title "Sales by Product";
        footnote "Data are current as of &sysptime &sysdate9";
        column product sales;
        define product / group;
        define sales / analysis sum;
    quit;

%mend salesrpt;

%salesrpt
```

Example 2. Sample Program to Generate Dynamic Output

The steps required to build and run a Stored Process Web application are documented in the *IT Developer's Guide: Stored Processes*. The following example assumes that you have installed SAS Enterprise Guide 3.1 (EG) to create and test the SAS program and the SAS Management Console to register it as a Stored Process. Note that there is also a separate, different *Stored Process Wizard* in Enterprise Guide. You can use one of the other to register the Stored Process but do not mix them or your Web application may not work correctly.

Creating a Stored Process Repository with the SAS Management Console 9.1

The Web Infrastructure Kit Installation procedure installs a default Stored Process repository called *Foundation* along with a set of sample processes. Unless you want to use `/Stored Process/Sample` as the URL for all of your Web applications, you will need to create a new folder using the *SAS Management Console Stored Process Manager*.

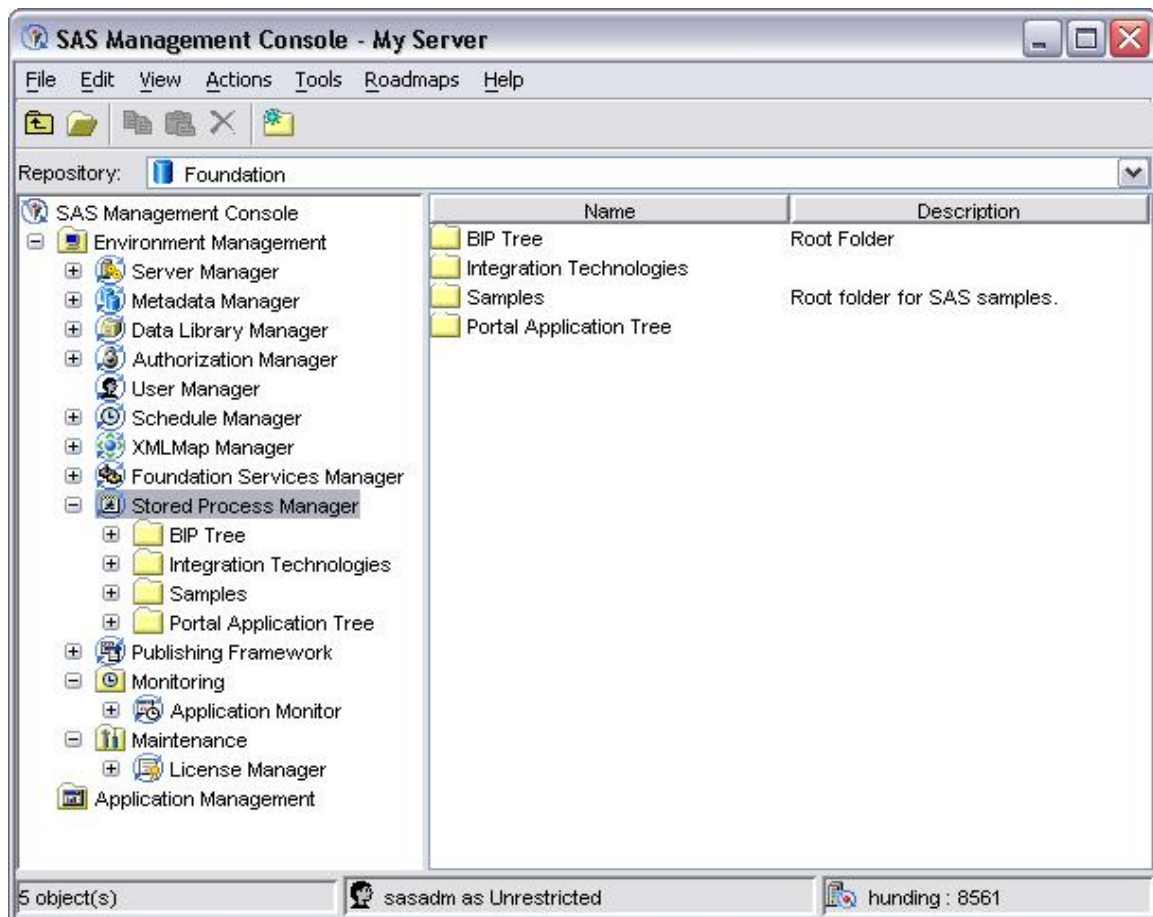


Figure 2. Management Console Stored Process Manager

Figure 2 shows the initial configuration for the Stored Process Manager. To create a new folder, select **Actions** → **New Folder** or just click on the **New Folder** icon on the toolbar. A single screen appears, on which a name for the new repository folder can be supplied-- for this example, the new folder is “BBU Examples”. Note that this is a *logical* folder; it need not exist as a physical directory on the host.

In order to convert the sample program to a Stored Process using the *Management Console*, it is necessary add three lines of code as shown below in **Example 3** in bold. (If you use the *Enterprise Guide* Stored Process Wizard, the additional macros are added to the program automatically.)

The comment at the top is required; *ProcessBody indicates the beginning of the Stored Process. The two macro invocations %stpbegin and %stpend call corresponding built-in macros that supply parameters to the Web Application and the Output Delivery System (ODS). Most of these macro variable values can be overridden to produce different output effects as desired; see the *Developer's Guide: Stored Processes* at "Input Parameters" for details.

```
*ProcessBody;

%global region;

%stpbegin;

%* Sales report Example - Display Product by Region;
%macro salesrpt;

    proc report data=sashelp.shoes;
        by region;
        %if (&region ne ) %then %do;
            where region="&region";
        %end;
        title "Sales by Product";
        footnote "Data are current as of &systime &sysdate9";
        column product sales;
        define product / group;
        define sales / analysis sum;
    quit;

%mend salesrpt;

%salesrpt

%stpend;
```

Example 3. SAS Stored Process Program

To start the Management Console *Stored Process Wizard*, right click on the new folder and select **New Stored Process**.



Figure 3. Create New Stored Process

Supply a name for the process, as illustrated. This name will be used in the URL for the Web application. The description and keywords are optional. Click **Next** and the screen shown in **Figure 7** should appear.



Figure 4. Specify Server, Source File and Result type

In order to register the Stored Process four parameters are needed:

1. **SAS Server** – select either the Stored Process Server or the Workspace Server. In this case, we want to use the former so that the Web application is shareable and the results are streamed back to the client.
2. **Source Repository** – this can be any folder on the host. To add a new location, select the **Manage** button.
3. **Source File** – the SAS program, in this case “shoes.sas”.
4. **Output** – for a Web application, select **Streaming**. The default is **None**, which will result in the program running with no output, which is almost certainly what you do not want to happen.

The input field is used to get XML or other streaming input data and is not needed for this example.



Figure 5. Add Run-time Parameters

The screen shown in **Figure 8** is used to specify the macro parameters that will be supplied at run time. Note in **Figure 5** that a global parameter `region` is defined. Click on **Add** to add this parameter and the screen shown in **Figure 9** should appear.

The image shows a Windows-style dialog box titled "Add Parameter...". It has a standard title bar with a close button (X) in the top right corner. The dialog is divided into several sections. The "Properties" section contains three text input fields: "Label:" with the value "region", "SAS variable name:" with the value "region", and "Description:" which is empty. Below this is a "Boolean properties" section containing four checkboxes: "Modifiable" (checked), "Visible" (checked), "Required" (unchecked), and "Expert" (unchecked). At the bottom of the main area, there is a "Type:" dropdown menu set to "String" and a "Default value:" text input field which is empty. A "Constraints..." button is located to the right of the "Default value:" field. At the very bottom of the dialog are three buttons: "OK", "Cancel", and "Help".

Figure 6. Specify parameters

Typing the word “region” into the `Label` field and pressing the tab key automatically causes the same value to appear in the `SAS variable name` textbox. Click **OK**, then on **Finish** and the new stored process is created and registered.

Testing the Stored Process

In order to check whether the new Stored Process works as desired, you can open SAS Enterprise Guide 3.1 and click on **View → Stored Process List**.

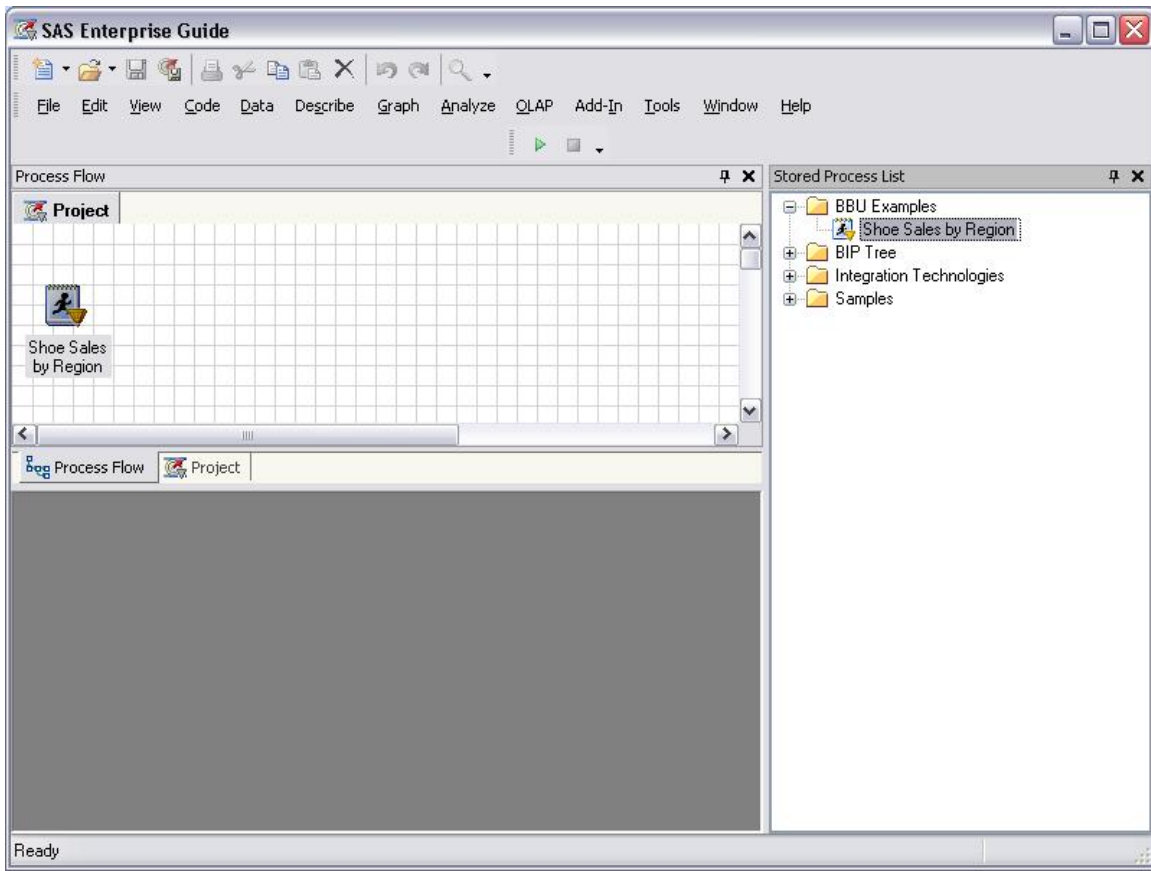


Figure 7. SAS Enterprise Guide Stored Process List

Drag “Shoe Sales by Region” to the Process Flow window or right-click on the icon in the Stored Process list and select Add to Project.

Right click on the icon in the Process Flow window and select Run This Stored Process. Since the sample program requires a `region` parameter, a window will pop up asking for the value of this run-time parameter. Typing “Asia” as shown in **Figure 11** runs the report for this region. Note that the footnote uses the built-in macro variables `&sysdate` and `&systemtime` but there is no need to specify these values.

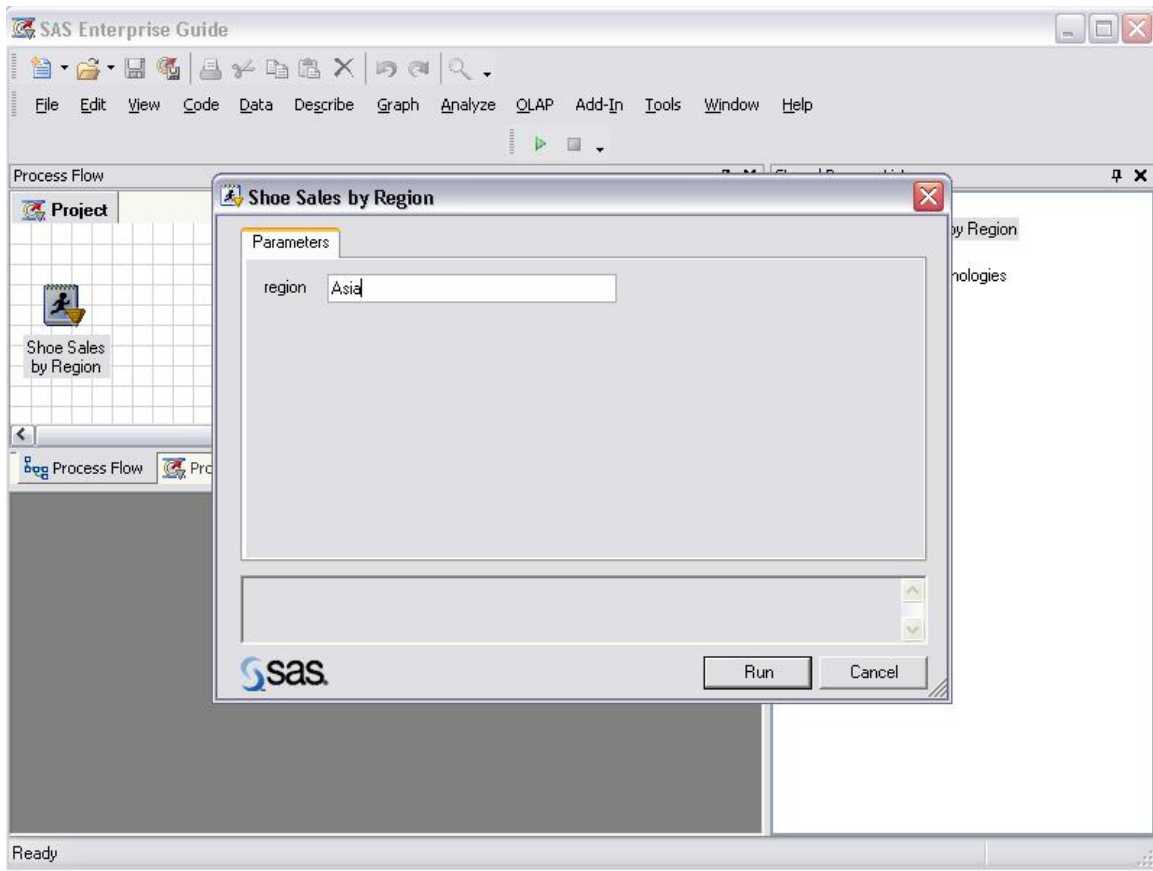


Figure 8. Prompt for Run-time Parameter

At this point a connection is made to the Stored Process Server, which requires a valid SAS user ID to run, as shown in Figure 12.



Figure 9. Enter SAS User ID and Password

Note that this can be any of the user accounts defined in the installation. Once a valid user and password have been supplied, the stored program should run and display the results for the specified region, as **Figure 13** illustrates.

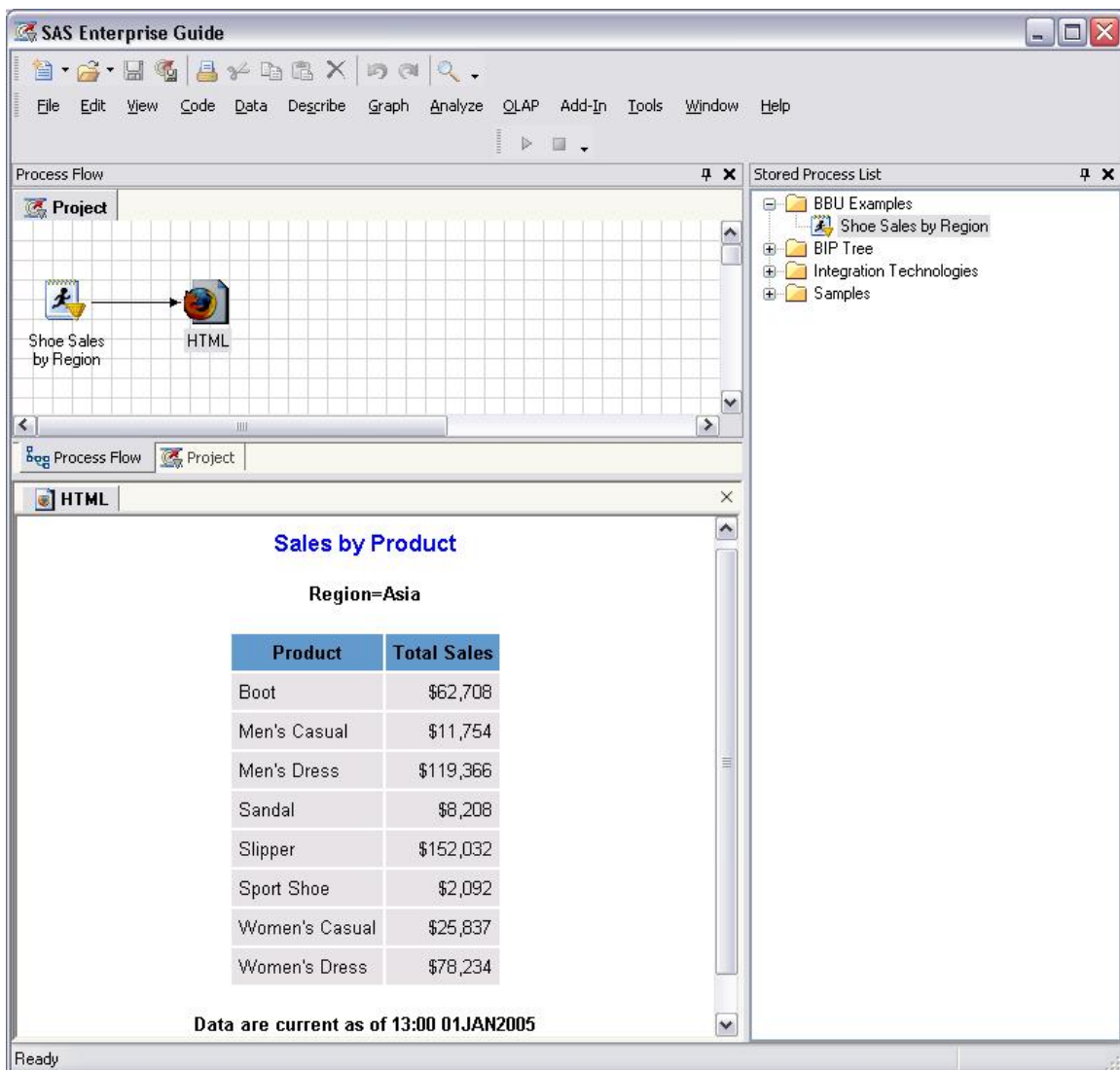


Figure 10. Enterprise Guide: HTML Streaming Output

Running the Web Application

Now that the program has been successfully debugged, running it as a Web application is simply a matter of specifying the correct URL in your favorite Web browser window. Since the repository folder is "BBU Examples" and the name of the stored procedure is "Shoe Sales by Region", the corresponding URL as shown in **Figure 14** is:

```
http://hunding:8080/SASStoredProcess/do?_program=/BBU Examples/
  Shoe Sales by Region&region=Asia
```

In this case, since Tomcat is running as the only Web server, the required server address is <http://hunding:8080>; port 8080 is the default port for this Web container. Note in **Figure 14** that spaces in the URL are replaced by the equivalent ASCII code “%20” which indicates a space character in the URL.

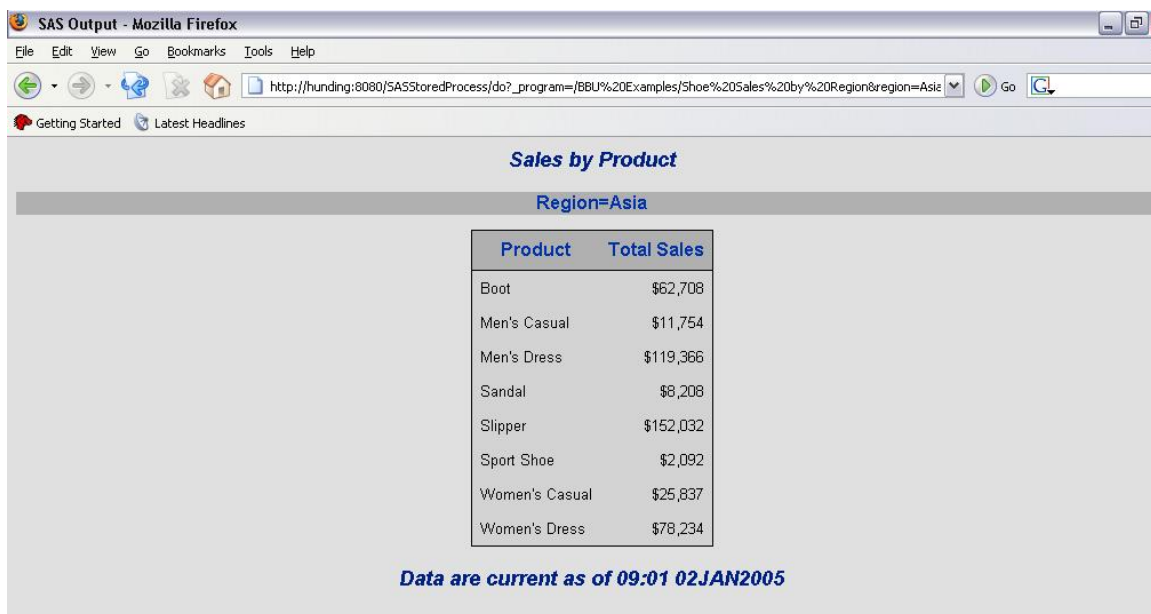


Figure 11. HTML Output from Web Application

The command to run the SAS Stored Process Web Application is very similar to the way SAS/IntrNet performs the same function:

- SAS/Intrnet Application Broker: `/cgi-bin/broker`
- Stored Process Web Application: `/SASStoredProcess/do`

As is usual for HTML, the run-time parameters follow the question mark character “?” and are separated by ampersands “&”. The first parameter `_program` is required and specifies the folder and the procedure name `/BBU Examples/Shoe Sales by Region`. Since the desired report is to be run solely for the region “Asia” the second parameter is `region=Asia`; note that quotes are not allowed nor required around the value of an HTML parameter.

CONCLUSION

In Version 9, SAS has supplied a comprehensive set of tools that can be used to generate dynamic server-side Web pages. SAS Stored Processes can be used to automate repetitive tasks and to supply dynamic Web content. The Stored Process Web Application was designed for SAS developers who are not familiar with Java and who may need to update SAS Intr/Net applications to run in a multi-user load-balancing environment. As this example illustrates, it is not necessary to write any Java code at all in order to use the SAS Stored Process Web Application. The only requirements are some SAS programming skills. As one developer noted at a recent user group conference, the Stored Process Web Application is “the Son of IntrNet”.

ACKNOWLEDGMENTS

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.

The Faculty Scholars Program, Office of Research and Sponsored Programs, Eastern Oregon University generously provided support for this research.

CONTACT INFORMATION

Frederick Pratter
Computer Science/Multimedia Studies Program
Eastern Oregon University
La Grande OR 97850
541-962-3065

REFERENCES

URL references are current as of the date of publication, but please check the Enterprise Integration Community pages at <http://support.sas.com/rnd/itech/intro.html> for the most up-to-date information.

SAS DOCUMENTATION

- The SAS *Integration Technologies: Version 9 Documentation*, available at <http://support.sas.com/rnd/itech/library/library9.html> includes the following on-line manuals:
 - Technical Overview
 - Server Administrator's Guide
 - Administrator's Guide
 - Administrator's Guide (LDAP Version)
 - Developer's Guide
 - SAS Web Infrastructure Kit 1.0: Overview
 - SAS Web Infrastructure Kit 1.0: Administrator's Guide
 - SAS Web Infrastructure Kit 1.0: Developer's Guide
- *SAS Stored Processes — A Roadmap*. http://support.sas.com/rnd/itech/papers/101519_0504.pdf.

USER PRESENTATIONS

- Fecht, Marje and Peter R. Bennett. 2005. "End-to-End Web Reporting Using SAS Enterprise Guide 3.0. Who Does What and When," Proceedings of the Thirtieth Annual SAS® Users Group International Conference. Cary, NC: SAS Institute Inc.
- Pratter, Frederick. 2004. "Access to SAS Data Using the Integrated Object Model (IOM) in Version 9.1," Proceedings of the 12th Annual Western Users of SAS Software, Inc. Users Group Conference (WUSS). Cary, NC: SAS Institute Inc.
- Rossland, Eric and Kari Richardson. 2005. "Creating and Using SAS Stored Processes with SAS Enterprise Guide," Proceedings of the Thirtieth Annual SAS® Users Group International Conference. Cary, NC: SAS Institute Inc.
- Wright, Jeff. 2005. "Catch the Stream: Stored Processes, ODS, and Java," Proceedings of the Thirtieth Annual SAS® Users Group International Conference. Cary, NC: SAS Institute Inc.