4941-2020

SAS® Stored Process Web Services and Dynamic Data Exchange

Greg Dorfner, Prime Therapeutics LLC; Dwight Buffum, SAS Silver Circle Solutions;
Danni Luo, Prime Therapeutics LLC;

ABSTRACT

The growing need for a Web User Interface is becoming more apparent in todays' applications. At Prime Therapeutics LLC we paired a Pega® Web interface with SAS® through Web services and SAS Stored Processes to build an application named Benefit Modeling Tool (BMT). This paper will describe the steps we took in building and supporting the BMT application.

INTRODUCTION

Benefits Impact Calculator (BIC) is a custom-developed SAS application at Prime Therapeutics LLC using SAS Enterprise Guide as the UI. It is built on the SAS 9.4 M3 platform, running on AIX 7.1, by Prime's Information Management team and the Benefit Design Solution & Modeling team. It serves two purposes: Re-pricing modeling to calculate Benefit savings for our clients and identifying patterns of usage for prediction. Clients who have chosen to implement the BIC-generated recommendations have enjoyed significant cost savings.

With BIC usage increasing into more areas of the company, there was pressure to move from the SAS Enterprise Guide based UI to a browser-based UI. Thus, the Pega tool was used to create the browser-based UI and SAS Stored Processes were added to invoke UI and BIC back-end SAS programs dynamically. The result is the new Benefit Modeling Tool (BMT) application. BMT has provided more user-friendly features, and maximum flexibility and availability for all users, including those who are not familiar with SAS Enterprise Guide.

This paper will focus on our major work with SAS Stored Processes and the Pega Web UI interface that make up BMT, and the administration of those pieces.

SAS STORED PROCESS WEB SERVICES

A Web server was initially defined as "a software system designed to support interoperable machine-to-machine interaction over a network" (World Wide Web Consortium, 2004). This simple and direct definition still provides us a quick view today of what the Web server is. "SAS Web Server is used as a load balancer for distributing HTTP requests to SAS Web Application Server instances", per SAS.

The best way we found for Pega to interact with SAS was through web services. A web service allows a PROGRAM to talk to a web page. SAS BI Web Services expose SAS Stored Processes for execution by using web service protocols. The remote clients specify input parameters through the Pega UI that drive execution of SAS code, and obtain results from that execution. Web services make it possible to create clients that perform this act in a myriad of languages and on a variety of operating systems by using HTTP to exchange messages.

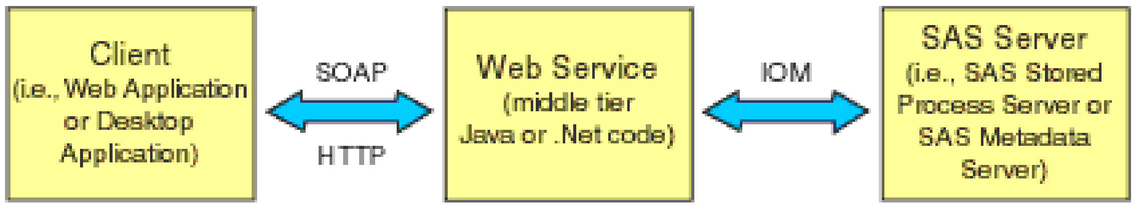

Client
(i.e., Web Application or Desktop Application)
SOAP
HTTP
Web Service
(middle tier Java or .Net code)
IOM
SAS Server
(i.e., SAS Stored Process Server or SAS Metadata Server)

Figure 1. Web Service

SAS BI WEB SERVICES and STRUCTURED WEB SERVICES

For every stored process, you can obtain a description of the structure of input and output web service messages that can be used to invoke the stored process. The document that describes this structure is called a Web Services Description Language (WSDL) file.

SAS BI Web Services automatically expose a WSDL file for every SAS stored process. These WSDL files include detailed information about the inputs and outputs of each stored process using XML schema descriptions.

The WSDL file includes the URLs of endpoints to invoke these stored processes by using the SOAP protocol over HTTP. Typically, you use WSDL files to automatically generate code in your client framework that then invoke the web services.

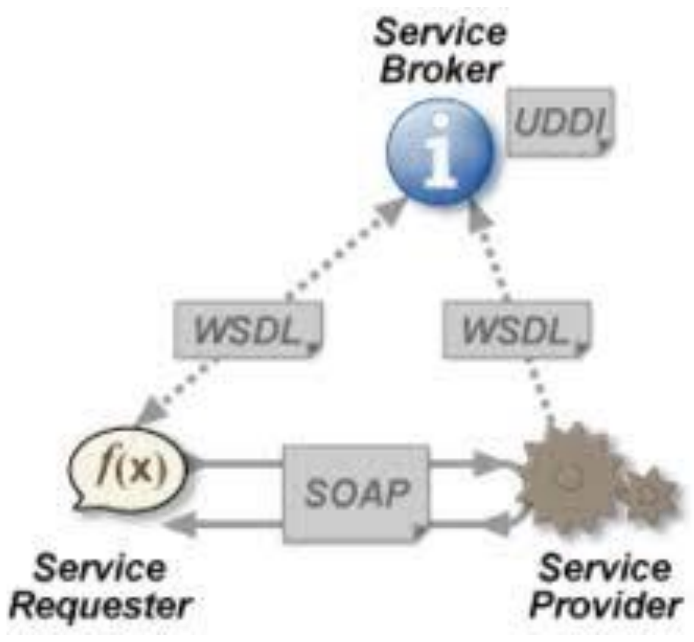


Figure 2. Structured web services

- UDDI: Universal Description, Discovery and Integration
- WSDL: Web Services Description Language
- SOAP: Simple Object Access Protocol

IMPORTANT FEATURES OF STORED PROCESSES (STPs)

A stored process is a SAS program that is stored on a server and defined in metadata. It can then be executed as requested by client applications.

Stored processes can be used for Web reporting, analytics, building web applications, delivering packages to clients or to the middle (web) tier, and publishing results to channels or repositories. Stored processes can also access any SAS data source or external file and create new data sets, files, or other data targets that are supported by SAS.

Storing SAS stored processes in the SAS Metadata Server enhances change control management. Instead of embedding the SAS code into client applications, one can centrally maintain and manage this code from the SAS Metadata Server and thus ensure that every client that invokes a stored process always gets the latest version.

Using stored processes also enhances security and application integrity because the programs that access sensitive data live on the server instead of being widely distributed with the client applications.

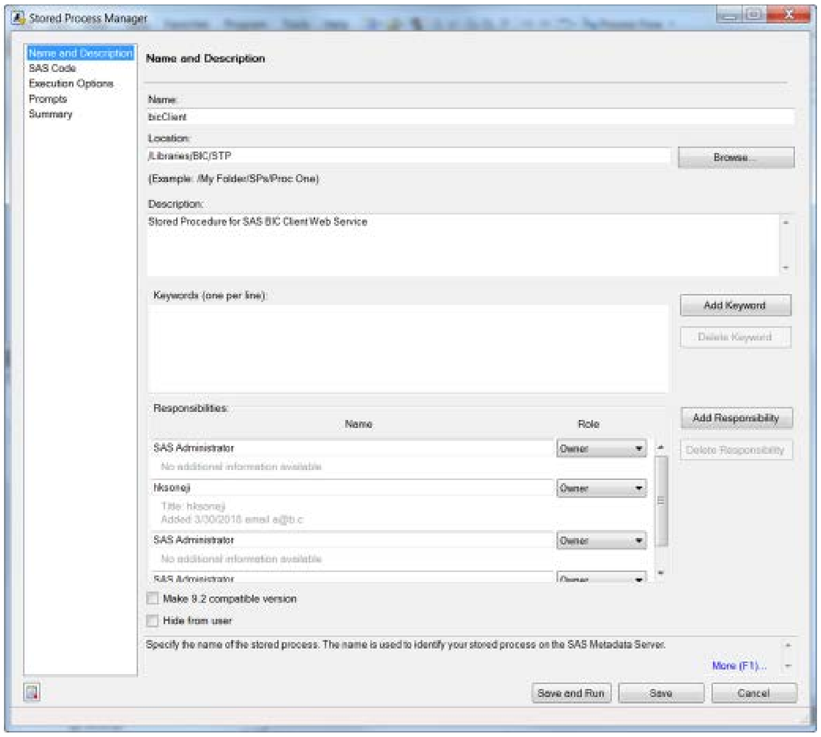
These SAS stored process programs can be invoked from multiple client contexts. For BMT, we have deployed a Pega-based user interface that invokes synchronous and asynchronous SAS stored processes.

BUILDING AND MODIFYING A STORED PROCESS IN SAS ENTERPRISE GUIDE

Almost any SAS program can be a stored process. You can use the SAS program editor, SAS Enterprise Guide, or any text editor to write a stored process. The following is a stored process code sample for Getting Parameter or Data/Files In/Out of Stored Processes.

Stored Process Manager Dialog Boxes

Name and Description


Stored Process Manager

Name and Description

Name:
bicClient

Location:
/Libraries/BIC/STP

(Example: /My Folder/SPs/Proc One)

Description:
Stored Procedure for SAS BIC Client Web Service

Keywords (one per line):

Responsibilities:

Name	Role
SAS Administrator No additional information available	Owner
hksoneji Title: hksoneji Added 3/30/2018 email a@b.c	Owner
SAS Administrator No additional information available	Owner
SAS Administrator	Owner

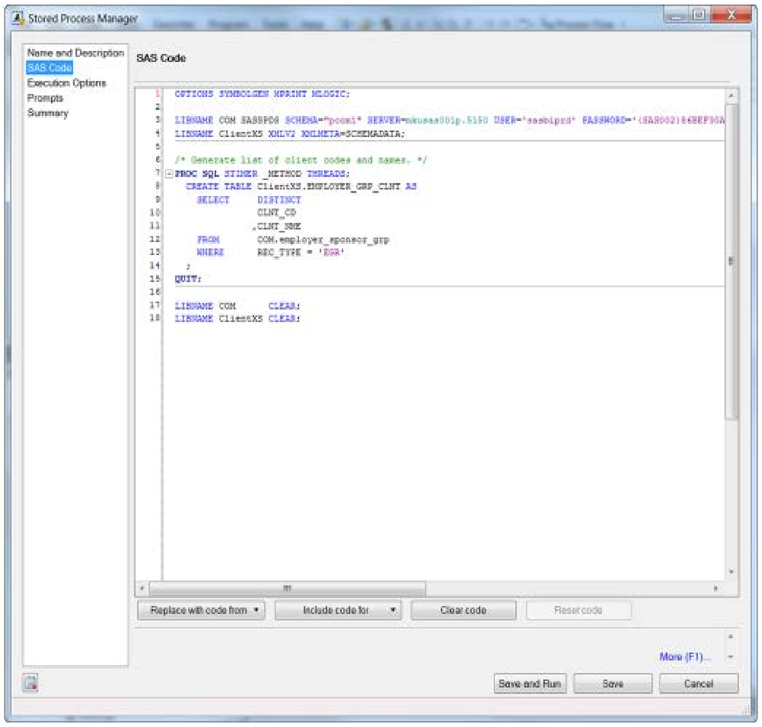
Make 9.2 compatible version

Hide from user

Specify the name of the stored process. The name is used to identify your stored process on the SAS Metadata Server.

Figure 3. Name and Description

Stored Process Manager Dialog Boxes SAS Code



```
OPTIONS SYMBOLGEN MPRINT MLOGIC;

LIBNAME COM SASSPDS SCHEMA="pcomi" SERVER=mkusas00ip.5150 USER='sasbiprd' PASSWORD='{SAS002}86BEF3CA
LIBNAME ClientXS XMLV2 XMLMETA=SCHEMADATA;

/* Generate list of client codes and names. */
PROC SQL STIMER _METHOD THREADS;
  CREATE TABLE ClientXS.EMPLOYER_GRP_CLNT AS
    SELECT     DISTINCT
               CLNT_CD
              ,CLNT_NME
    FROM       COM.employer_sponsor_grp
    WHERE      REC_TYPE = 'EGR'
  ;
QUIT;

LIBNAME COM      CLEAR;
LIBNAME ClientXS CLEAR;
```

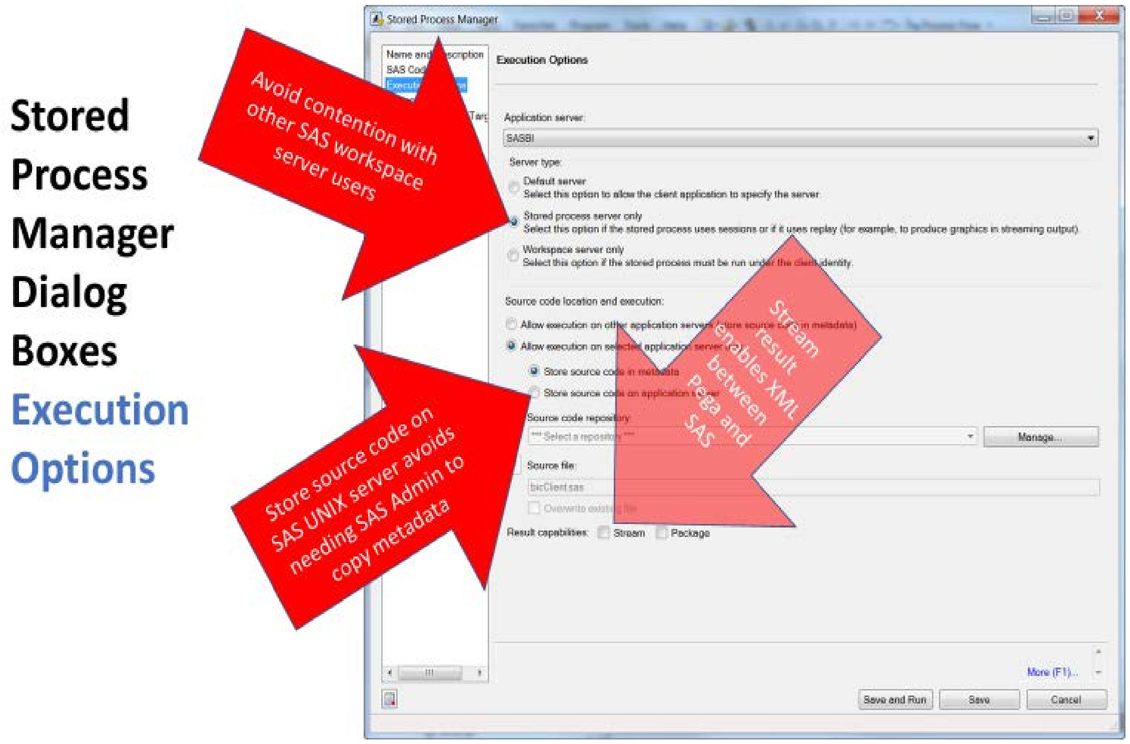
Figure 4. SAS Code

Code sample for getting data in / out of stored process:

```
LIBNAME ClHrcyXS XMLV2  XMLMAP='/path/to/XMLMAP.map';  /* Input XML stream from Pega */

LIBNAME BnftDsXS XMLV2  XMLMETA=SCHEMADATA; /* Output XML stream to PEGA */

        DATA WORK.CAG_PEGA;
          SET ClHrcyXS.EMPLOYER_GRP_CAG;
          IF _N_  =1 THEN CALL SYMPUTX( 'SPON_GRP_NME', SPON_GRP_NME );
        RUN;
```


Stored Process Manager Dialog Boxes
Execution Options
Avoid contention with other SAS workspace server users
Store source code on SAS UNIX server avoids needing SAS Admin to copy metadata
Stream result enables XML between Pega and SAS
Stored Process Manager
Save and Run
Save
Cancel
More (F1)...
Manage...

Execution Options

Application server: SASBI

Server type:

- Default server — Select this option to allow the client application to specify the server.
- [✓ Stored process server only] — Select this option if the stored process uses sessions or if it uses replay (for example, to produce graphics in streaming output).
- Workspace server only — Select this option if the stored process must be run under the client identity.

Source code location and execution:

- Allow execution on other application servers (store source code in metadata)
- [✓ Allow execution on selected application server only]
  - [✓ Store source code in metadata]
  - Store source code on application server

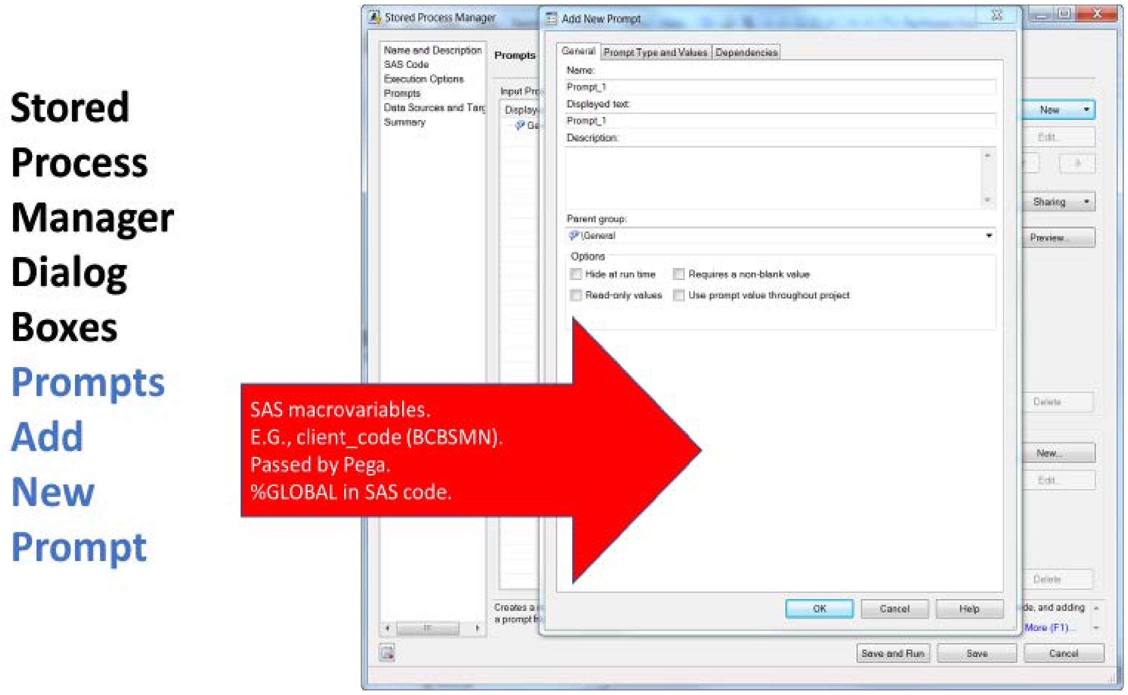
Source code repository: *** Select a repository ***

Source file: bicClient.sas

Overwrite existing file

Result capabilities: Stream Package

Figure 5. Execution Options



Name: Prompt_1

Displayed text: Prompt_1

Description:

Parent group: \General

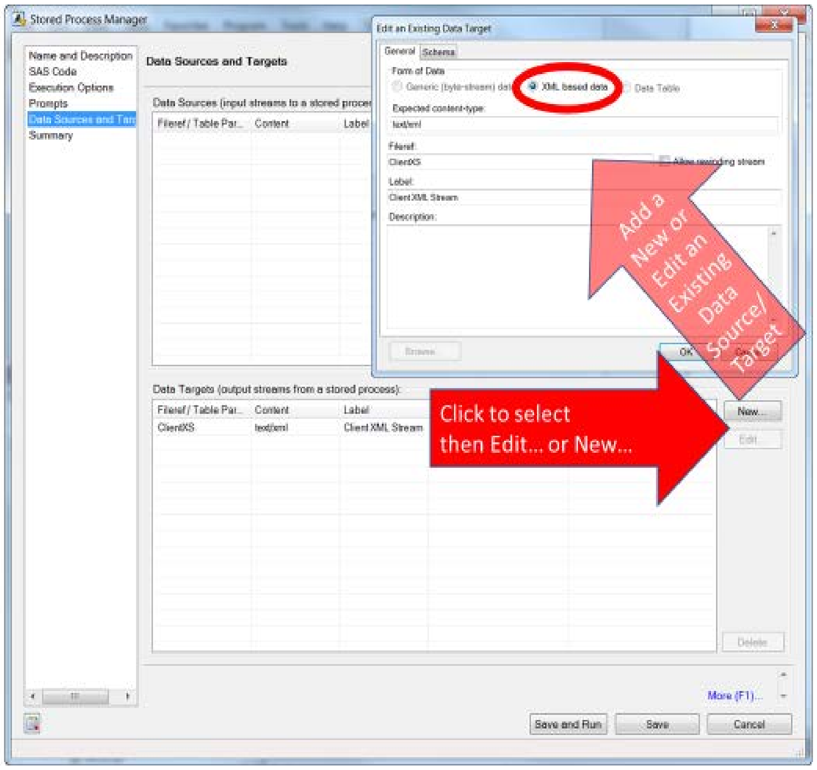
Options

- Hide at run time
- Requires a non-blank value
- Read-only values
- Use prompt value throughout project

Figure 6. Prompts

Stored Process Manager Dialog Boxes

Data Sources and Targets



Data Sources and Targets

Data Sources (input streams to a stored process):

Fileref / Table Par...	Content	Label

Data Targets (output streams from a stored process):

Fileref / Table Par...	Content	Label
ClientXS	text/xml	Client XML Stream

Form of Data: Generic (byte-stream) data [✓ XML based data] Data Table

Expected content-type: text/xml

Fileref: ClientXS

Allow rewinding stream

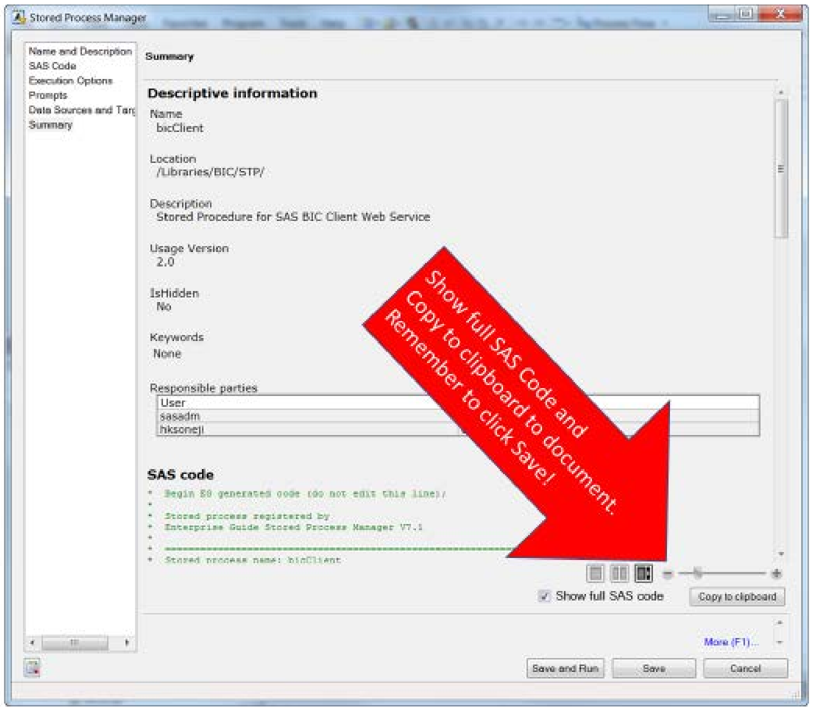
Label: Client XML Stream

Description:

Figure 7. Data Sources and Targets

Stored Process Manager Dialog Boxes

Summary



Summary

Descriptive information

Name
bicClient

Location
/Libraries/BIC/STP/

Description
Stored Procedure for SAS BIC Client Web Service

Usage Version
2.0

IsHidden
No

Keywords
None

Responsible parties

User
sasadm
hksoneji

SAS code

```
*  Begin EG generated code (do not edit this line);
*
*  Stored process registered by
*  Enterprise Guide Stored Process Manager V7.1
*
*  ====================================================
*  Stored process name: bicClient
```

[✓] Show full SAS code

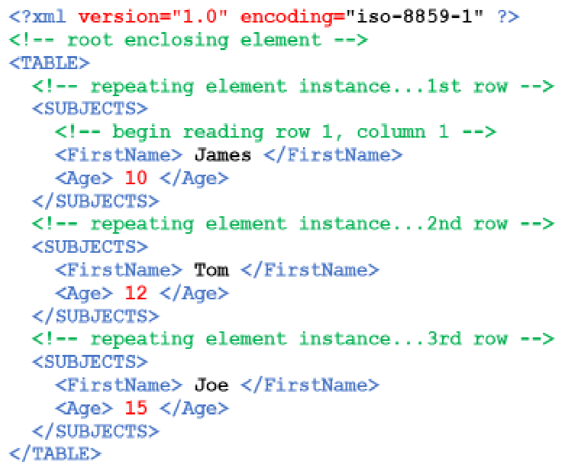
Figure 8. Summary

XML AND SAS

To successfully input an XML document using its XML engine, SAS requires the source XML document to be well-formed with a rectangular structure. More specifically, the XML document must meet the following requirements:

1. The root enclosing element (top-level node) of an XML document is the document container. For SAS, it translates to a library.
2. The nested elements occurring within the container (repeating element instances) begin with the second-level instance tag.
3. The repeating element instances must represent a rectangular organization. For a SAS data set, they become a collection of rows with a constant set of columns.

SAS natively reads GENERIC XML that conforms to a row and column definition

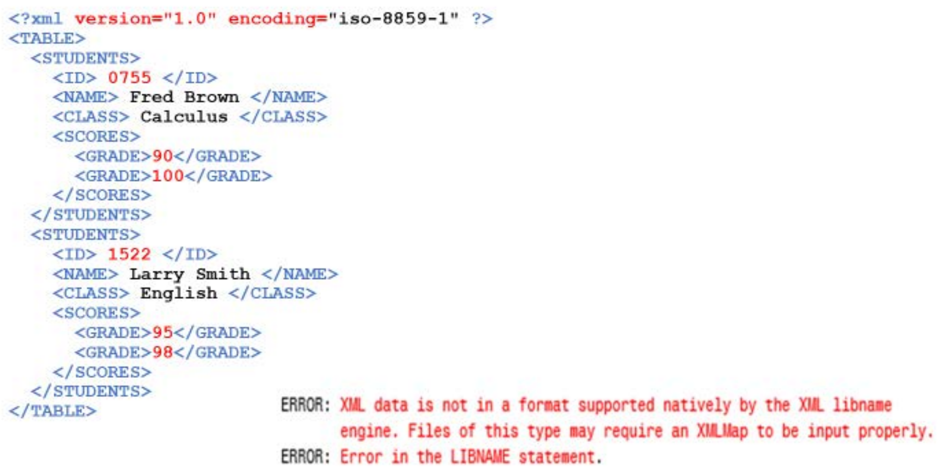


```
<?xml version="1.0" encoding="iso-8859-1" ?>
<!-- root enclosing element -->
<TABLE>
  <!-- repeating element instance...1st row -->
  <SUBJECTS>
    <!-- begin reading row 1, column 1 -->
    <FirstName> James </FirstName>
    <Age> 10 </Age>
  </SUBJECTS>
  <!-- repeating element instance...2nd row -->
  <SUBJECTS>
    <FirstName> Tom </FirstName>
    <Age> 12 </Age>
  </SUBJECTS>
  <!-- repeating element instance...3rd row -->
  <SUBJECTS>
    <FirstName> Joe </FirstName>
    <Age> 15 </Age>
  </SUBJECTS>
</TABLE>
```

Figure 9. Generic XML file

SAS needs an XMLMap to read non-GENERIC XML that violates a row and column definition.

For example below, under the field SCORES, another field GRADE is wrapped in it.



```
<?xml version="1.0" encoding="iso-8859-1" ?>
<TABLE>
  <STUDENTS>
    <ID> 0755 </ID>
    <NAME> Fred Brown </NAME>
    <CLASS> Calculus </CLASS>
    <SCORES>
      <GRADE>90</GRADE>
      <GRADE>100</GRADE>
    </SCORES>
  </STUDENTS>
  <STUDENTS>
    <ID> 1522 </ID>
    <NAME> Larry Smith </NAME>
    <CLASS> English </CLASS>
    <SCORES>
      <GRADE>95</GRADE>
      <GRADE>98</GRADE>
    </SCORES>
  </STUDENTS>
</TABLE>
```

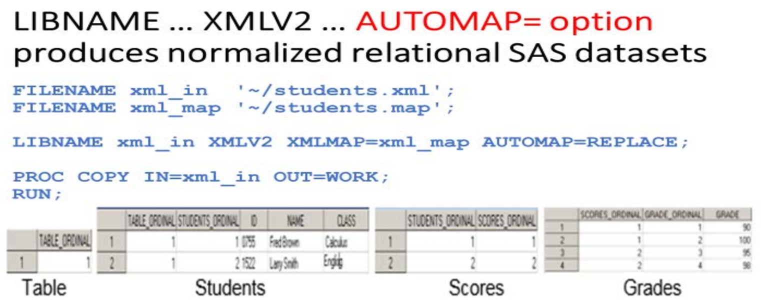
```
ERROR: XML data is not in a format supported natively by the XML libname
       engine. Files of this type may require an XMLMap to be input properly.
ERROR: Error in the LIBNAME statement.
```

Figure 10. Non-Generic XML file

XML MAPS

The XMLMap file referenced in the error message is a SAS-specific file that instructs the XMLV2 LIBNAME engine about how to create rows, columns, and other information from the XML markup. You can use either of the following methods to generate the XMLMap file:

1. SAS 9.4 statement LIBNAME ... XMLV2 ... AUTOMAP= option.



LIBNAME ... XMLV2 ... AUTOMAP= option
produces normalized relational SAS datasets

```
FILENAME xml_in  '~/students.xml';
FILENAME xml_map '~/students.map';

LIBNAME xml_in XMLV2 XMLMAP=xml_map AUTOMAP=REPLACE;

PROC COPY IN=xml_in OUT=WORK;
RUN;
```

	TABLE_ORDINAL
1	1

Table

	TABLE_ORDINAL	STUDENTS_ORDINAL	ID	NAME	CLASS
1	1	1	0755	Fred Brown	Calculus
2	1	2	1522	Larry Smith	English

Students

	STUDENTS_ORDINAL	SCORES_ORDINAL
1	1	1
2	2	2

Scores

	SCORES_ORDINAL	GRADE_ORDINAL	GRADE
1	1	1	90
2	1	2	100
3	2	3	95
4	2	4	98

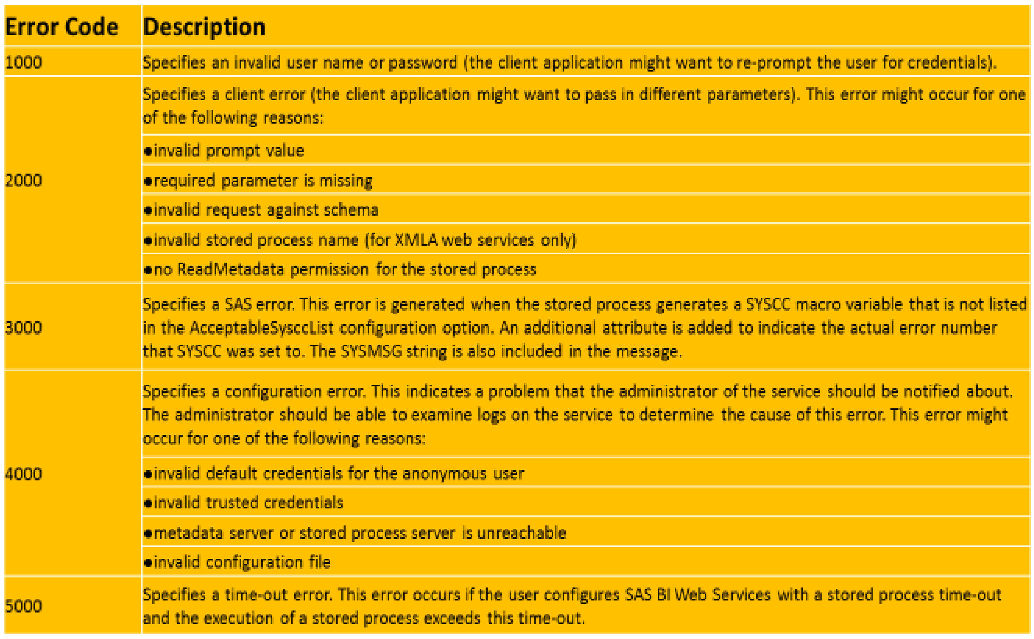
Grades

Figure 11. Libname with AUTOMAP

2. Using SAS XML Mapper.

SAS Website video
[How to Automatically Generate XMLMap Files (video)]
[How to Generate Custom XMLMap Files (video)]

UNDERSTANDING STP WEB SERVICE ERROR CODES



Error Code	Description
1000	Specifies an invalid user name or password (the client application might want to re-prompt the user for credentials).
2000	Specifies a client error (the client application might want to pass in different parameters). This error might occur for one of the following reasons:
	●invalid prompt value
	●required parameter is missing
	●invalid request against schema
	●invalid stored process name (for XMLA web services only)
	●no ReadMetadata permission for the stored process
3000	Specifies a SAS error. This error is generated when the stored process generates a SYSCC macro variable that is not listed in the AcceptableSysccList configuration option. An additional attribute is added to indicate the actual error number that SYSCC was set to. The SYSMSG string is also included in the message.
4000	Specifies a configuration error. This indicates a problem that the administrator of the service should be notified about. The administrator should be able to examine logs on the service to determine the cause of this error. This error might occur for one of the following reasons:
	●invalid default credentials for the anonymous user
	●invalid trusted credentials
	●metadata server or stored process server is unreachable
	●invalid configuration file
5000	Specifies a time-out error. This error occurs if the user configures SAS BI Web Services with a stored process time-out and the execution of a stored process exceeds this time-out.

Figure 12. STP Web Service Error Code samples

SAS ADMINISTRATION REQUIRED FOR SUPPORTING STORED PROCESSES AND WEB SERVICES

IMPLEMENTATION AT PRIME

A standard SAS installation includes all the components needed to run SAS Stored Processes from a web based front-end. However, it may take some additional configuration and understanding on the part of the SAS Administrator to properly support it.

The first thing that became important to this architecture in our implementation was the SAS mid-tier components. This tier includes the SAS Web Server, SAS Web Application Server, and SAS Web Application Infrastructure Platform—components that interact with other more familiar components to deliver the BMT application, such as the Metadata Server and Stored Process Server in the SAS Servers tier and a Web browser in the Clients tier.

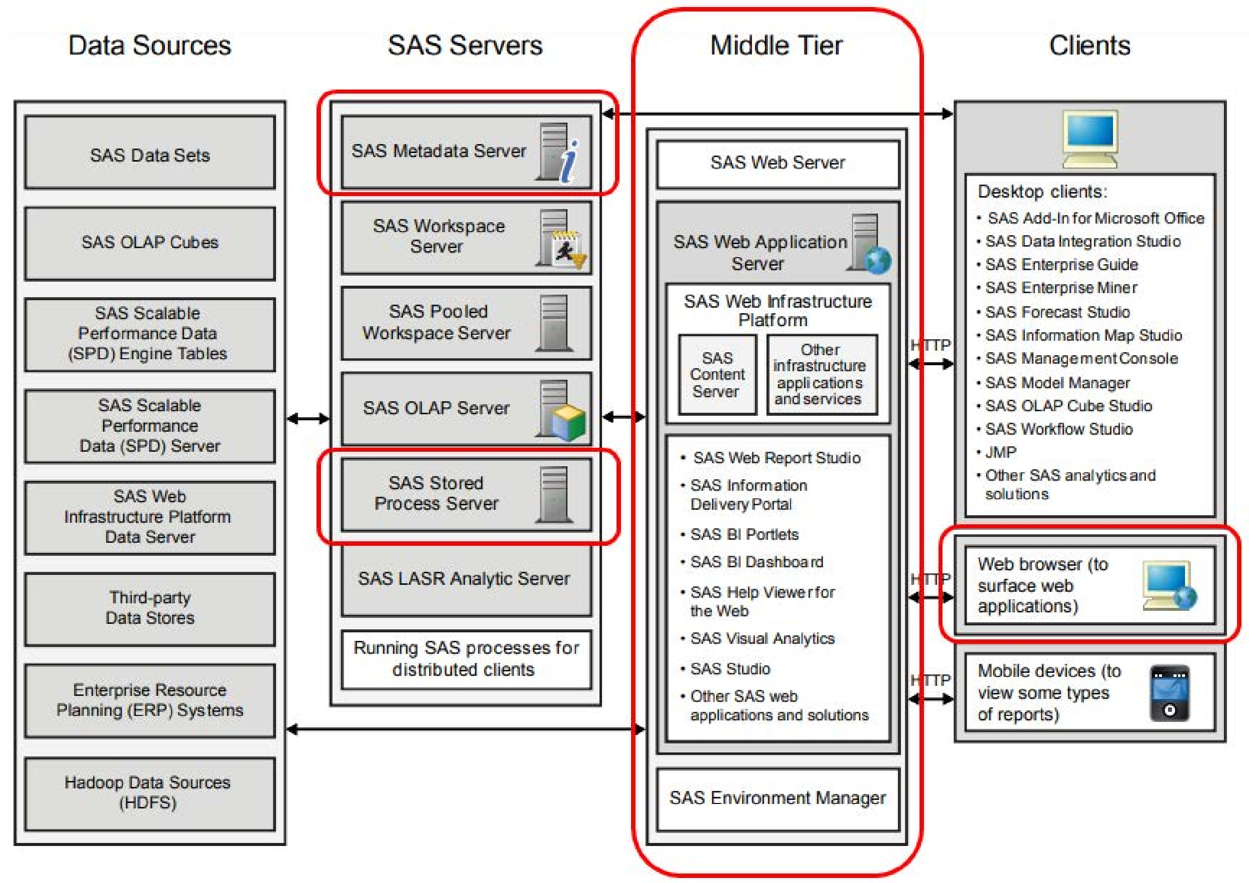

Data Sources
SAS Servers
Middle Tier
Clients
SAS Data Sets
SAS OLAP Cubes
SAS Scalable Performance Data (SPD) Engine Tables
SAS Scalable Performance Data (SPD) Server
SAS Web Infrastructure Platform Data Server
Third-party Data Stores
Enterprise Resource Planning (ERP) Systems
Hadoop Data Sources (HDFS)
SAS Metadata Server
SAS Workspace Server
SAS Pooled Workspace Server
SAS OLAP Server
SAS Stored Process Server
SAS LASR Analytic Server
Running SAS processes for distributed clients
SAS Web Server
SAS Web Application Server
SAS Web Infrastructure Platform
SAS Content Server
Other infrastructure applications and services
• SAS Web Report Studio
• SAS Information Delivery Portal
• SAS BI Portlets
• SAS BI Dashboard
• SAS Help Viewer for the Web
• SAS Visual Analytics
• SAS Studio
• Other SAS web applications and solutions
SAS Environment Manager
HTTP
HTTP
HTTP
Desktop clients:
• SAS Add-In for Microsoft Office
• SAS Data Integration Studio
• SAS Enterprise Guide
• SAS Enterprise Miner
• SAS Forecast Studio
• SAS Information Map Studio
• SAS Management Console
• SAS Model Manager
• SAS OLAP Cube Studio
• SAS Workflow Studio
• JMP
• Other SAS analytics and solutions
Web browser (to surface web applications)
Mobile devices (to view some types of reports)

Figure 13. Architecture of the SAS Intelligence Platform

In a nutshell, the SAS Stored Process Web Application is the heart of the mid-tier portion of the application. It allows web clients to run SAS Stored Processes and return the results to a web browser. Here is how the SAS Stored Process Web Application processes a BMT request:

1. A BMT user enters information in the Pega web browser UI and clicks the Run button. The information is sent to a web server, which invokes the SAS Stored Process Web Application.

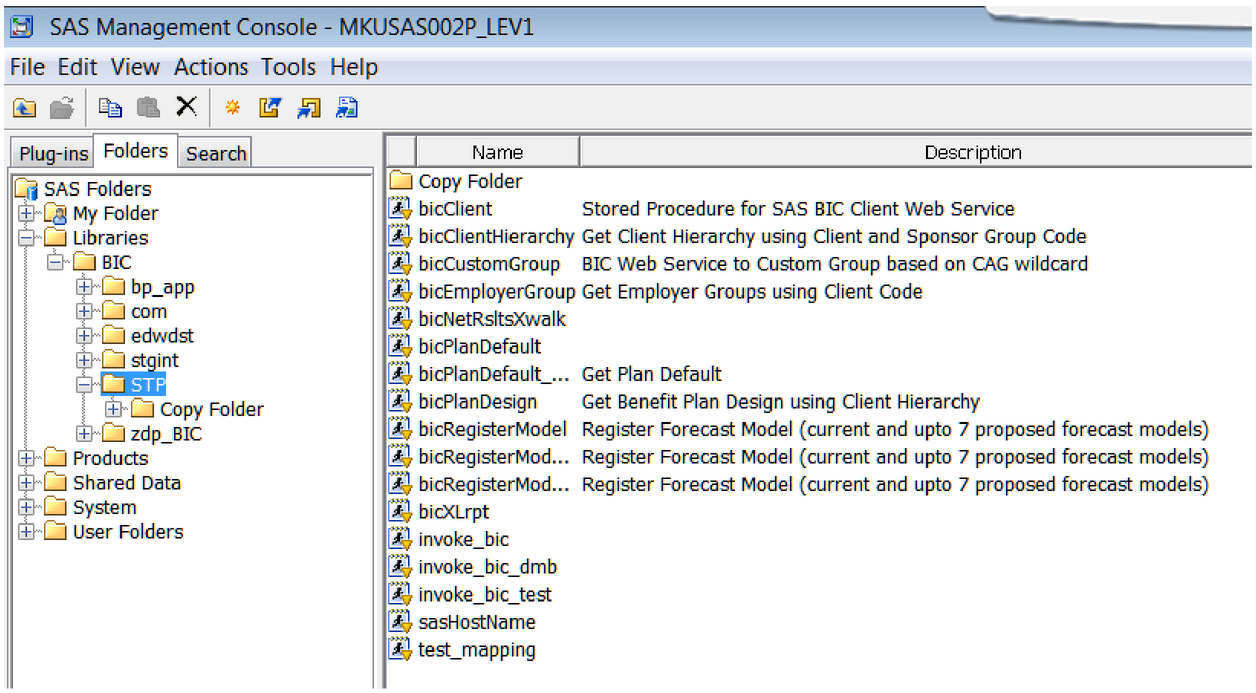
2. The Stored Process Web Application contacts the SAS Metadata Server to retrieve the stored process data.

3. The stored process data is then sent by the Stored Process Web Application to a stored process server via the object spawner.

4. The stored process server runs the SAS stored process program that processes the information and sends the results back through the web application and web server to the web browser.

To get all this to work smoothly requires some additional set-up in the Metadata Repository (MR).

For our implementation a folder STP was created to hold the SAS Stored Processes.


SAS Management Console - MKUSAS002P_LEV1
File Edit View Actions Tools Help
Plug-ins
Folders
Search

```
SAS Folders
  My Folder
  Libraries
    BIC
      bp_app
      com
      edwdst
      stgint
      STP
        Copy Folder
      zdp_BIC
  Products
  Shared Data
  System
  User Folders
```

Name	Description
Copy Folder	
bicClient	Stored Procedure for SAS BIC Client Web Service
bicClientHierarchy	Get Client Hierarchy using Client and Sponsor Group Code
bicCustomGroup	BIC Web Service to Custom Group based on CAG wildcard
bicEmployerGroup	Get Employer Groups using Client Code
bicNetRsltsXwalk	
bicPlanDefault	
bicPlanDefault_...	Get Plan Default
bicPlanDesign	Get Benefit Plan Design using Client Hierarchy
bicRegisterModel	Register Forecast Model (current and upto 7 proposed forecast models)
bicRegisterMod...	Register Forecast Model (current and upto 7 proposed forecast models)
bicRegisterMod...	Register Forecast Model (current and upto 7 proposed forecast models)
bicXLrpt	
invoke_bic	
invoke_bic_dmb	
invoke_bic_test	
sasHostName	
test_mapping	

Figure 14. Creating a STP Folder in MR

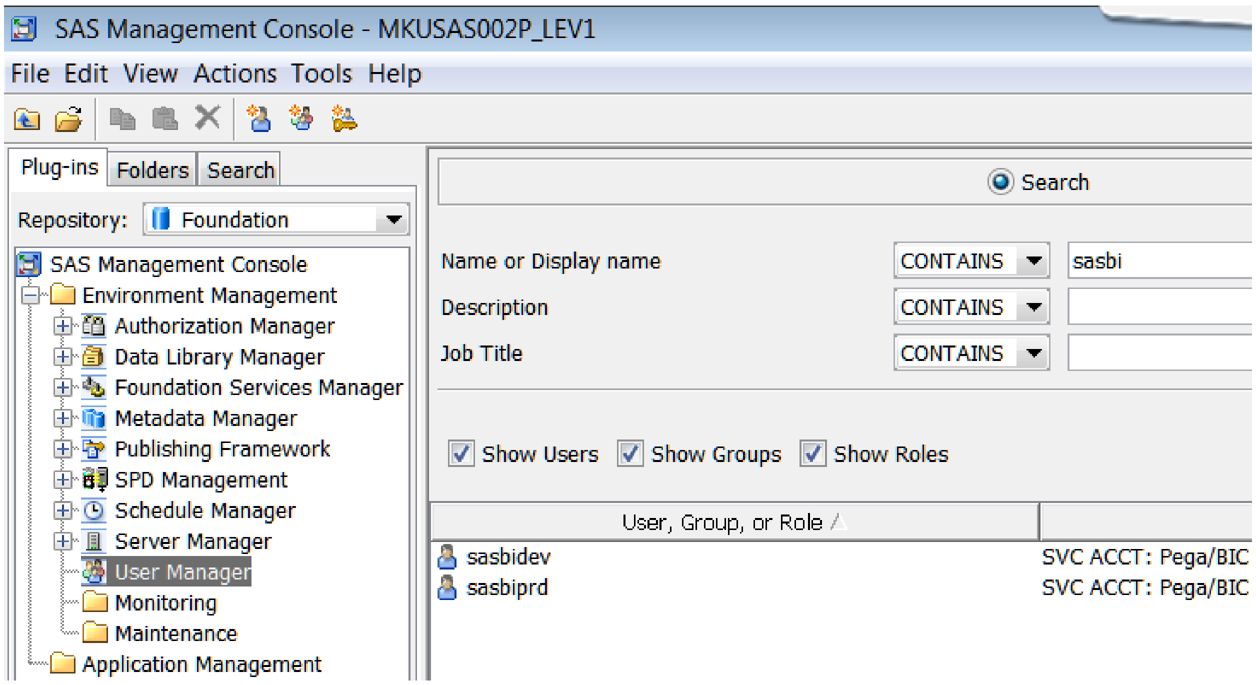
Stored Process Server Configuration

The BMT front-end requires individual User authentication but then switches to a shared Service Account for connecting to the Metadata Server and running the Stored Process. Thus, a Service Account was created (sasbiprd) and had to be added to MR Users. In addition, this account was also created in Active Directory in order to interact with SAS for PC Files to create Excel output.

By default, the Stored Process Server (SPS), as defined in the Metadata Repository, runs under the SAS installer account which is sas at Prime. But for the aforementioned reasons it needed to be switched to run under sasbiprd.

To make the switch the following had to be done in the Management Console:

- Server Manager > SASBI > SASBI - Stored Process Server ... Options tab: Change Multi-User Credentials value from sas to sasbiprd
- Edit the SAS Trusted User Account
- Users > SAS Trusted User account .... Accounts tab: Add sasbiprd (defaultauth)



Repository: Foundation

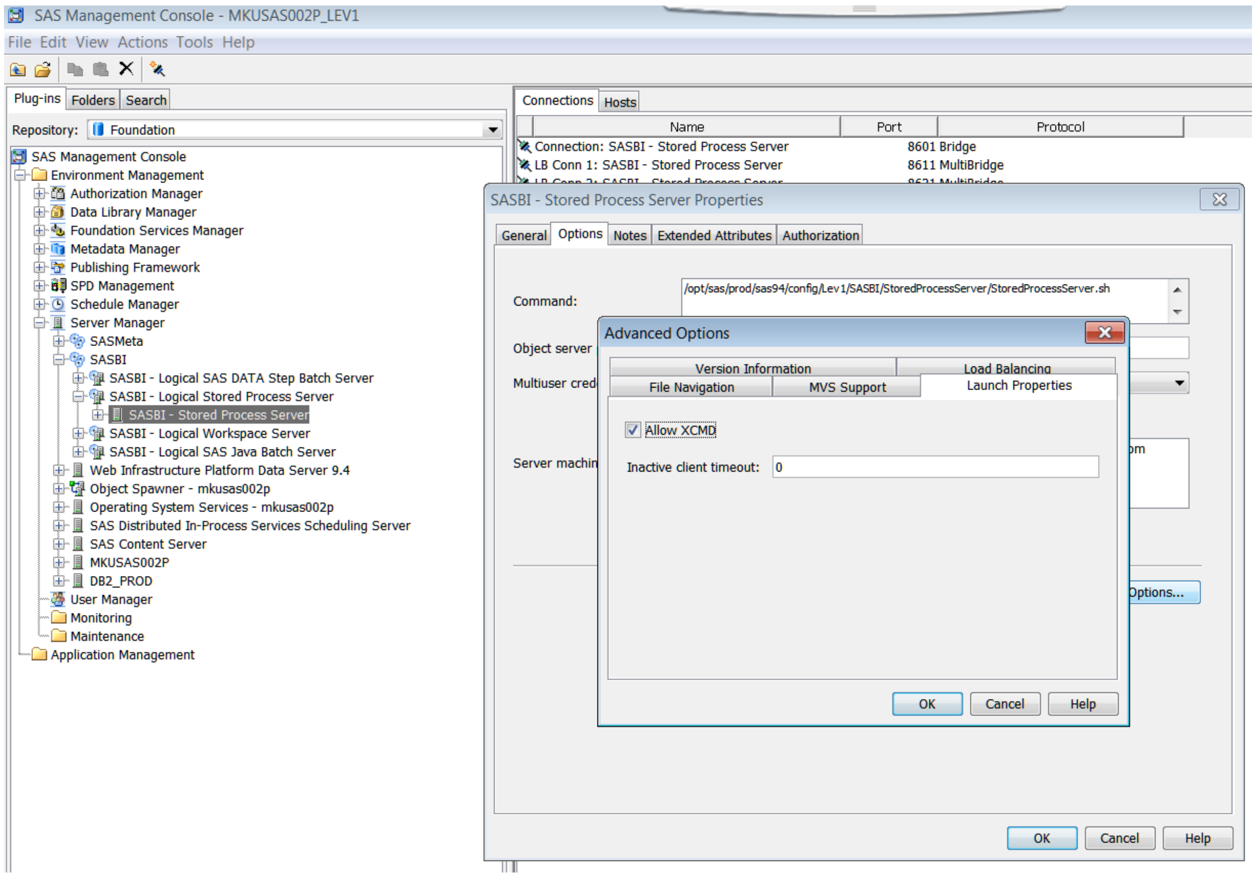
Name or Display name	CONTAINS	sasbi
Description	CONTAINS	
Job Title	CONTAINS	

[✓] Show Users [✓] Show Groups [✓] Show Roles

User, Group, or Role	
sasbidev	SVC ACCT: Pega/BIC
sasbiprd	SVC ACCT: Pega/BIC

Figure 14. Creating a STP Service Account in MR

Also, some of the STPs need to launch Unix shell scripts so "Allow xcmd" had to be enabled.


SAS Management Console - MKUSAS002P_LEV1

SASBI - Stored Process Server Properties

Command: /opt/sas/prod/sas94/config/Lev1/SASBI/StoredProcessServer/StoredProcessServer.sh

Advanced Options — Launch Properties

☑ Allow XCMD

Inactive client timeout: 0

Figure 15. Enabling the Option - Allow XCMD

SUPPORTING THE IMPLEMENTATION

Monitoring STP Logs

A very important piece of this application is being able to support it. The mid-tier has a lot of logs in various locations that together can paint a picture of the health of the application.

To monitor STPs that have run you would check this log:

```
cd /opt/sas/prod/sas94/config/Lev1/SASBI/StoredProcessServer/Logs
```

This log can be very verbose so use an egrep command to just find out how many STPs have run in each month:

```
egrep "USER_ID=|_PROGRAM=/" SASBI_STPServer_yyyy_mm*
```

The above command yields this:

```
SASBI_STPServer_2019-02-20_mkusas002p_3146590.log:2019-02-
20T14:52:24,326 INFO  [02895372] 4062:sasbiprd -  USER_ID=xxxxxxxx

SASBI_STPServer_2019-02-20_mkusas002p_3146590.log:2019-02-
20T14:52:24,326 INFO  [02895372] 4062:sasbiprd -
_PROGRAM=/Libraries/BIC/STP/bicRegisterModel
```

Stopping and Starting the Mid-Tier

In addition, the mid-tier components can be stopped and started without the overall SAS environment being stopped and started. This is occasionally necessary if an issue arises with just the SAS mid-tier.

When stopping and starting the mid-tier it is good practice to clean-up old logs periodically. See the Appendices for a list of logs and log cleanup detail.

TROUBLESHOOTING

The BMT front-end is served up through an F5 load balancer and thus requires a JVM to be running that connects to the SAS mid-tier. The SAS Stored Processes can be reached outside the BMT front-end for smoke testing purposes via the familiar ports that SAS configures as part of a standard installation, i.e., http://MKUSAS002P.primetherapeutics.com:7980/SASStoredProcess/do

If you can run a STP with the above URL but not the BMT front-end then you know the issue is not with the SAS mid-tier but with the F5 load balancer or JVM.

If the above URL doesn't respond, then you can restart the SAS mid-tier and not have to do a full restart of all the SAS servers. This example shows the "status" command, but you would use "stop" and "start" to recycle the mid-tier:

```
sas@mkusas002p:/opt/sas/prod/sas94/config/Lev1 # ./sas.servers.mid status

SAS JMS Broker is UP
SAS Cache Locator Service ins_41415 is UP
SAS Web Server is UP
SAS Web App Server SASServer1_1 is UP
SAS Environment Manager is UP
SAS Environment Manager Agent is UP
```

CONCLUSION

Today's business applications often require a Web user interface. A web service is the best way to interact with SAS Stored Processes to fulfill that need. SAS BI Web Services expose SAS Stored Processes for execution by using web service protocols. It allows the remote clients to specify input parameters, drive execution of SAS code, and obtain results from that execution. The information covered in this paper will help you understand the approach we took in building BMT using SAS Stored processes to work with Web Services.

The success of the BMT project required a close collaboration between the SAS Developers and the Administrator. A SAS Administrator supporting such an application requires skills that may be unfamiliar when compared to traditional SAS execution environment skills. This paper highlights some key things the SAS Administrator needs to know to configure and support a SAS web-based application like BMT in the SAS operational environment.

ACKNOWLEDGMENTS

Thanks to the Benefit Design Solution & Modeling team at Prime Therapeutics LLC for their contribution of business knowledge, their work in Web interface design and their coordination.

Thanks to the IT Data Warehouse and Reporting team and the BI Tools team from Prime Therapeutics LLC for their expertise, skills and support.

CONTACT INFORMATION

Greg Dorfner
gdorfner@primetherapeutics.com
Danni Luo
dluo@primetherapeutics.com
Dwight Buffum
dmbuffum@gmail.com

RECOMMENDED READING

Web Services with SAS® by Anton Fuchs (SAS Institute)
SAS® 9.4 BI Web Services: Developer's Guide
SAS® 9.4 XMLV2 LIBNAME Engine Tip Sheet
SAS® 9.4 XML LIBNAME ENGINE: User's Guide
SAS® 9.4 Intelligence Platform: Middle-Tier Administration Guide, Fourth Edition
Frederick Pratter, Eastern Oregon University, La Grande, OR

APPENDIX 1 KEY MID-TIER LOGS

When the mid-tier is healthy you will see few if any WARN or ERRORs in the following logs

/opt/sas/prod/sas94/config/Lev1/Web/WebAppServer/SASServer1_1/logs/...

CATALINA.OUT
GEMFIRE.LOG
SERVER.LOG

localhost_access_log.yyyy-mm-dd.txt LOG contains HTTP commands (GET, POST, etc.)

If things start going awry you may see these kinds of errors : HTTP/1.1" 500

Stop looks like this in catalina.out:

```
10:33:10,539 | INFO  | [StandardService] | Stopping service Catalina
```

Startup looks like these below lines in catalina.out. Look for "Initialization complete"

```
2018-06-04 10:36:39,042 WARN  (main)
[com.springsource.tcserver.serviceability.rmi.JmxSocketListener] Detected LDAP
configuration
'/opt/sas/prod/sas94/config/Lev1/Web/WebAppServer/SASServer1_1/conf/jaas.config', but
there is no ldap configuration entry specified.
10:36:39,123 | INFO  | [Catalina] | Initialization processed in 2520 ms
10:38:12,889 | INFO  | [StandardService] | Starting service Catalina
        ...more lines....
21:46:09,574 | INFO  | [Catalina] | Server startup in 253902 ms
        ...more lines....
2018-06-04 10:36:39,123 INFO  (main) [org.apache.catalina.startup.Catalina] Initialization
processed in 2520 ms
        ...more lines....
2019-10-24 21:46:54,931 [Thread-186] INFO  [unknown]
com.sas.svcs.content.relationship.dao.scheduling.RelationshipSchedulingHandler -
Initialization complete
```

APPENDIX 2 MID-TIER LOG FILE CLEANUP

Once the mid-tier is stopped remove these files (your path and Lev will be different than what is shown in red below):

```
/opt/sas/prod/sas94/config/Lev1/Web/Logs/SASServer1_1
        *.log
        *.log.*

/opt/sas/prod/sas94/config/Lev1/Web/WebAppServer/SASServer1_1/logs
        *.log
        *.out
        *.txt
        *.txt.*
        *.log.*

/opt/sas/prod/sas94/config/Lev1
        Snap.*
        heapdump.*
        javacore.*
```

/opt/sas/prod/sas94/config/Lev1/Web/gemfire/instances/ins_41415

    *.log
    *.dat
    .log.

/opt/sas/prod/sas94/config/Lev1/Web/WebServer/logs

    *.log

/opt/sas/prod/sas94/config/Lev1/WebInfrastructurePlatformDataServer/Logs

    *.log

/opt/sas/prod/sas94/config/Lev1/Web/WebAppServer/SASServer1_1
Remove the entire temp directory

/opt/sas/prod/sas94/config/Lev1/Web/activemq/data
Remove all the files within and the entire kahadb directory

/opt/sas/prod/sas94/config/Lev1/Web/WebAppServer/SASServer1_1/work/Catalina
Remove the entire localhost directory