CHAPTER 1

Overview of SAS BI Web Services

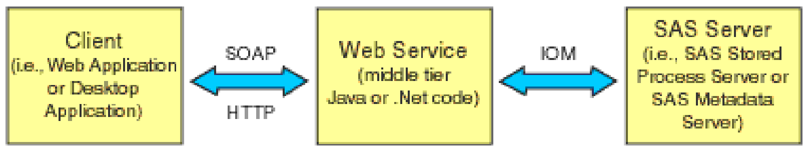
What Are SAS BI Web Services?

A Web service is an interface that enables communication between distributed applications. Web services enable cross-platform integration by enabling applications that are written in various programming languages to communicate by using a standard Web-based protocol, typically the Simple Object Access Protocol (SOAP). This functionality makes it possible for businesses to bridge the gaps between different applications and systems.

There are two implementations of SAS BI Web Services: one written in Java that requires a servlet container, and another written in C# that uses the .NET framework. For information about the differences between SAS BI Web Services for .NET and SAS BI Web Services for Java, see "Deciding Between .NET and Java" on page 2.

The following figure shows how Web services work.

Figure 1.1 Web Services Communications


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)

A client, such as a Web application or desktop application, obtains the Web Service Description Language (WSDL) from the Web service. The WSDL describes the methods that are available, the endpoint (where to call the Web service), and the format of the XML that is required to call the Web service.

Web service clients and servers transport XML data by using a data envelope that is called a SOAP envelope. Any client that can send and receive SOAP messages can access Web services. SAS supports SOAP bindings over HTTP. The client sends XML requests and parameters in a SOAP envelope to the Web service, which tells the Web service to either discover or execute stored processes.

Discover and Execute are the two methods that are specified for XMLA. The Discover method consists of middle-tier code that calls the SAS Metadata Server to get the requested metadata. The Execute method consists of middle-tier code that calls the SAS Stored Process Server to invoke stored processes.

The stored process that is requested is executed by a SAS Stored Process Server. Usually any number of simple string parameters are passed to the stored process, and a stream of XML data is returned. The input parameters to the stored process can be nothing or a combination of simple string parameters and any number of XML streams.

Before SAS 9.2, you could write XML for Analysis (XMLA) Web services only. Starting with SAS 9.2, you can use SAS Management Console to deploy a set of stored processes as a generated Web service. For more information, see "What Are Generated Web Services?" on page 23.

Deciding Between .NET and Java

Installation and Administration Differences

- *What operating environment are you using?*
  - SAS BI Web Services for .NET can be installed only in the following operating environments:
    - Windows XP
    - Windows 2003
    - Windows Vista Business
    - Windows Vista Ultimate
    - Windows Vista Enterprise
    - Windows Server 2008
  - SAS BI Web Services for Java can be installed in any operating environment that is supported by SAS middle tier components.
- *How do you want to administer the Web services?*
  - SAS BI Web Services for .NET are administered by using IIS.
  - SAS BI Web Services for Java are administered by using JBoss, BEA WebLogic, or IBM WebSphere. If you install SAS BI Web Services for Java, then you also need to have a Java Virtual Machine for an application server.
- *Which Web service stack do you want to use?* (A stack is the engine that performs SOAP processing.)
  - SAS BI Web Services for .NET use the .NET Web service stack (Web Services Enhancements 3.0). The .NET stack supports Message Transmission Optimization Mechanism (MTOM) attachments.
  - SAS BI Web Services for Java use the Apache Axis2 Web service stack. This stack supports both MTOM attachments and SOAP Messages with Attachments (SwA). If your service uses attachments and your client is .NET, then use MTOM, which is enabled by default.

SAS BI Web Services for .NET and SAS BI Web Services for Java also differ in the way they handle logging and configuration. SAS BI Web Services for .NET use .NET logging, and SAS BI Web Services for Java use SAS Foundation Services for logging. For more information about configuring Web services, see the *SAS Intelligence Platform: Web Application Administration Guide*.

Client Differences

Web service clients cannot generally identify differences between SAS BI Web Services for .NET and SAS BI Web Services for Java. If the client conforms to the usage rules for Web services, then the client should be able to use either platform.

One difference for the Web service client is how the path is returned for the name of the stored process (DataSourceName). SAS BI Web Services for Java return a fully qualified path. SAS BI Web Services for .NET return a simple path. In both cases, you get the name of the stored process to invoke by using the Discover method of the same Web service.

Creating SAS BI Web Services

Use Web Services: Prerequisite

Before you can use Web services, you need to perform the following steps:

1 Decide whether you want to use SAS BI Web Services for .NET or SAS BI Web Services for Java. Install SAS BI Web Services and the SAS Metadata Server.

   Note: When you install SAS 9.2 BI Web Services, you are actually installing two Web services: the XMLA Web services that were available with SAS 9.1, and the new WebServiceMaker Web service that is used create generated Web services. △

2 Write a SAS program to use as a stored process with Web services. See "Writing SAS Programs for XMLA Web Services" on page 9.

3 Define a stored process server, if one is not already defined.

4 Define a stored process by using SAS Management Console. Use **`XMLA Web Service`** as a keyword and **`Stream`** as the stored process output type when defining a stored process to be called by an XMLA Web service.

Use XMLA Web Services

On the client side, perform the following steps to use XMLA Web services:

1 Locate the Web Service Description Language File (WSDL). You can access the WSDL for a SAS BI Web Service by appending a '?WSDL' onto the service endpoint.

2 Write the code for the client application that uses either the Discover method or the Execute method to call the Web service.

3 Run the code.

For XMLA Web services, the SAS code that implements the Web service, the metadata, and the client code that calls the Web service must all be synchronized. The following table shows how to synchronize these items:

Table 1.1 Items to Synchronize

Item	**SAS Program**	**Metadata**	**Client Code**
Name	The name of the file that contains the SAS code.	Associates the name of the SAS Stored Process with the name of the file.	`<StoredProcess name= 'MyStoredProcess'>`
Input Data	Reads XML from the fileref. `libname instream xml;`	The name of the fileref, which must match the name of the data source.	`<Stream name='instream'> <XMLDataFromClientHere... </Stream>`
Input Parameters	Macros. `&tablename`	The parameter name is specified in the metadata. Parameters are treated as strings, regardless of the type that is specified in the metadata.	`<Parameter name='tablename'> myParam</Parameter>`
Output Data	Writes output to the _WEBOUT fileref as XML. `libname _WEBOUT xml xmlmeta= &_XMLSCHEMA;`	Designates the output as 'Streaming'.	Uses the XML that is returned.

Use Generated Web Services

Follow these steps to use generated Web services:

1 Generate a new Web service:

   a Determine URL of the Web Service Maker endpoint.

   b In SAS Management Console, select a set of stored processes and then select **Actions ▶ Deploy As Web Service** to generate a new Web service that can be used to call the selected stored processes.

   In the **`Web Service Maker URL`** field of the Deploy as Web Service wizard, type the endpoint URL or select an existing URL. The user who performs this action should belong to the SAS BI Web Services Users metadata group so that the new Web service can be stored in metadata. The metadata is created only if the service is successfully deployed in the target container. The new Web service will contain one operation for each stored process that you selected.

   Upon successful deployment, a message displays that tells you the endpoint URL for the newly deployed Web service.

2 Create clients to call the Web service.

A Web service can be created with multiple operations in it. Each operation corresponds to a stored process, and has the same name as the stored process, unless there is a naming conflict. If the name of the stored process conflicts with another name, then a new operation name is created.

Overview of Security for Web Services

A default installation of SAS BI Web Services for Java or .NET is not highly secure. The default security mechanism is SAS authentication. All requests and responses are sent as clear-Text. If users want to authenticate as a specific user, then they can send a user name and password as clear-Text as part of the WS-Security headers. Authentication is performed by authenticating client credentials at the SAS Metadata Server. Whenever user names and passwords must be sent as clear-Text, SSL should be enabled to provide transport layer security.

If you are using XMLA Web services or generated Web services, an anonymous user can be configured. Anonymous users cannot use the Web Service Maker; credentials must always be provided to use the Web Service Maker. If you are using XMLA Web services, you can pass user credentials as XMLA properties in the payload.

SAS BI Web Services can be secured by using Web authentication. This provides a way for SAS BI Web Services to identify the calling user by using basic Web authentication. The following two types of Web authentication can be configured:

- WS-Security message-level security
- HTTP transport-level security

Note: Web authentication can be used with both XMLA Web services and generated Web services but cannot be used with the Web Service Maker Web service. The Web Service Maker must be able to authenticate one-time-passwords that are generated by SAS Management Console clients. △

SAS BI Web Services for .NET can be secured by using Web Services Enhancements 3.0 for Microsoft .NET, which enables support for the latest security and interoperability standards for Web services. For detailed information about using Web Services Enhancements 3.0, WS-Security, and WS-Policy to secure SAS BI Web Services, see the *SAS Intelligence Platform: Web Application Administration Guide*. The current release of SAS BI Web Services for Java does not support WS-Policy.

Consult with your administrator to determine how Web services are configured at your site and how you can invoke them. For more information about setting up Web service security, see the *SAS Intelligence Platform: Web Application Administration Guide*.

Understanding Error Codes

Errors generally fall into one of five categories, and are assigned the appropriate error code for that category. The following table describes these error codes:

Table 1.2 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 AcceptableSyssc configuration option. An additional attribute is added to indicate that 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 timeout error. This error occurs if the user configures SAS BI Web Services with a stored process timeout and the execution of a stored process exceeds this timeout.

Note: Before SAS 9.2, XMLA returned an error code of `99` for almost all errors. △

The following code is an example of a generated SOAP fault that has an error code of **4000**:

```
<SOAP-ENV:Fault>
   <faultcode>Server</faultcode>
   <faultstring>The XML for Analysis provider encountered an error</faultstring>
   <faultactor>XML for Analysis Provider>XML for Analysis Provider</faultactor>
   <detail>
      <sas:Fault code="4000">
         <sas:Exception message="The configured credentials are invalid.">
            <sas:Exception message="The config file contains invalid metadata
credentials."/>
            <sas:Exception message="The user 'anon' is unknown.">
               <sas:Exception message="'anon' is not defined in metadata."/>
            </sas:Exception>
         </sas:Exception>
      </sas:Fault>
   </detail>
</SOAP-ENV:Fault>
```