

Working Environment in SAS Suites with Metadata Flow

Author: Gaurav K Agrawal

Audience:

The target audiences are individuals who have a basic knowledge of SAS Suite. They may or may not be familiar with the Internet technologies. The people, who are interested to know more about the BI metadata flow in SAS 9 web tools.

Purpose:

This paper talks about the different components of SAS Suite, mainly focuses on the SAS 9 Web tools/terminologies and metadata flow in SAS Suite. It will also explain the web environment capabilities of this Suite and provides an overall idea of metadata flow. As you may already be knowing that SAS Suite is empowered with the capabilities of Java to provide Web-Based and Desktop applications based on the business needs. The SAS suite is empowered with all the power of Java using Java client.

Note: To understand the Web Terminology used for SAS suite, you can refer the References section (Web Terminology Used for SAS Suite) of this document.

Components of SAS Suite

Components of SAS Suite can be divided into four broad categories; Data Source, SAS Servers, Middle Tier, and Client. The following figure 1 provides an overview of these categories:

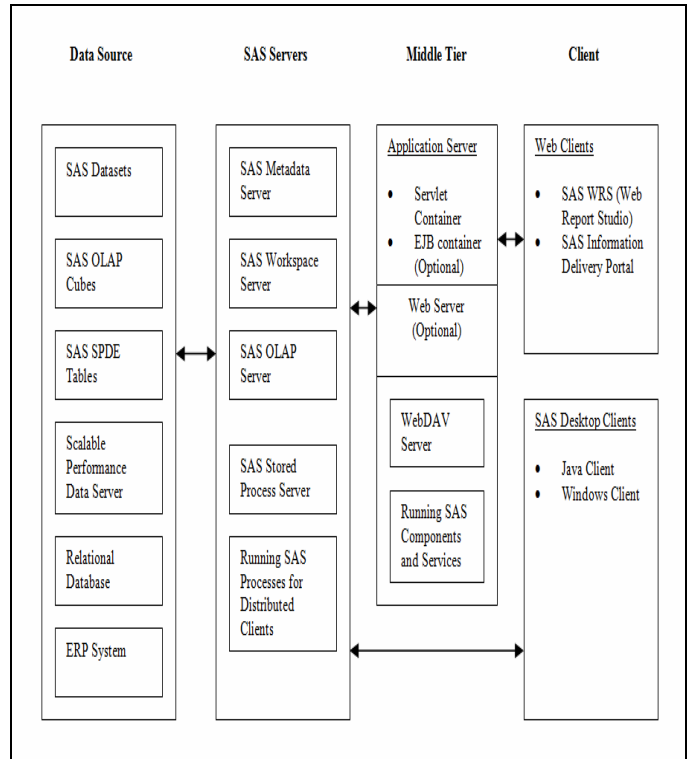


Figure: 1

Web tools used in SAS 9 suite are being explained below.

Web Report Studio (WRS): WRS is a web-based tool for providing the reporting capabilities on the Web. This tool helps to create reports easily as per the business requirements, which empowers the capability of fast decision making. The business users can create reports without having any knowledge of the complex queries. WRS also provides the functionality of creating folders and segregating reports as per different categories decided by the business users. Various types of graphs can also be generated by this tool and can be exported to Excel.

Information Delivery Portal (IDP): Information Delivery Portal is a single access point for aggregated information through an easy-to-use Web-based interface.

IDP provides a common front-end for SAS intelligence, Web-based SAS solutions and other digital content in a role-based, secure, customizable and extensible environment. Users can easily personalize their portal environments, reducing both information overload and IT workload.

Note: SAS suite provides a user friendly desktop client called Management Console, to manage the metadata from desktop.

BI Metadata Flow Diagram in SAS Suite

The Figure 2 given here depicts the complete flow of the metadata/data in web component of SAS Suite. Let us consider an example where WRS opens a page to view all available Information Map(s).

Following are the steps mentioned, which need to be performed to fetch the required information:

1. The Web User requests for all available Information through WRS page.
2. The Web Server accepts the request on the defined port and passes it to the Application Server to be processed.
3. The Application Server checks with the WebDAV server for the available Information Maps.
4. The WebDAV Server checks for all Information Maps and folders in the Map Folder of root.
5. Also, the Application Server checks for the Metadata Entry in Foundation Repository through “SAS Component and Services” for the Information Maps/Folders exist in the Root Folder of WebDAV server.
6. All the Information Maps, for which entry is there in the Root Folder of the WebDAV server and in the Foundation Metadata Repository, will be visible on WRS
7. Using these Information Maps, user would be able to create the reports as per the specific business requirements. Once the metadata of the report is created, this report would be executed and then data will be fetched from the back-end

Note: BI contents (Information Map/Stored Process etc.) existing in the WebDAV root folder should have corresponding entry in the Foundation Repository. The mapping can be directed from Custom repository content to the root of WebDAV server, but metadata entry will not be present in the Foundation repository and the Information maps created in Custom will not be visible on WRS because of this reason.

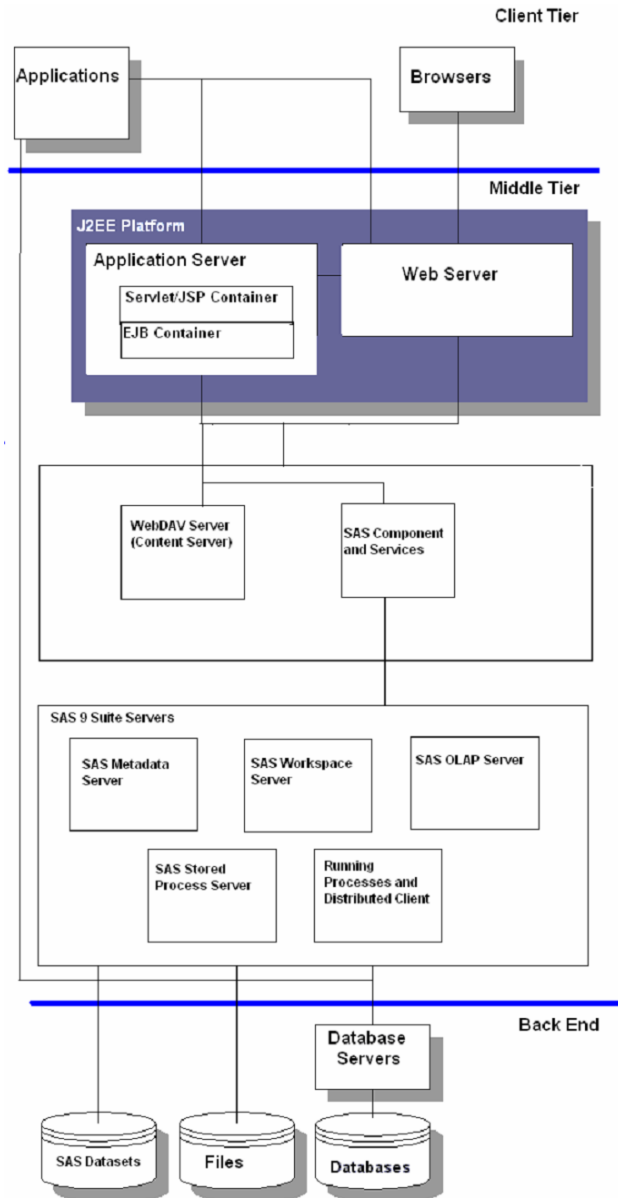


Figure: 2

Repository Used for BI and DI Metadata

SAS provides a Centralized Repository concept as the Foundation Repository for BI (Business Intelligence) metadata. All BI metadata information should exist in the Foundation Repository to get it accessed through the SAS web tools. If the BI information is developed in the Custom Repository, then it should be exported to the Foundation Repository to be made visible on the web tools. For the convenient maintenance of the BI data in Foundation Repository, it is always suggested to create sub folders in the BIP Tree directory and place BI information segregated according to the requirement.

DI (Data Integration) metadata can reside either in Foundation or Custom Repositories as per the created design. It is not mandatory to move DI metadata in the Foundation Repositories.

I have used two types of projects as being described below to explain the flow of BI and DI metadata in more details.

Project Structure One:

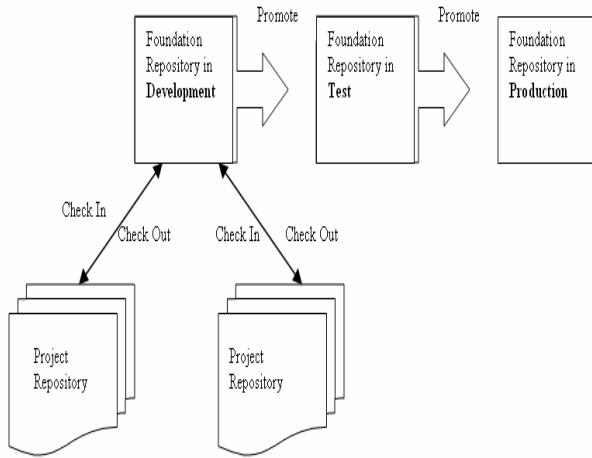


Figure: 3

Project Design: Multiple Project repositories exist under the Foundation repository as shown in the Figure 3. A Project repository will directly be dependent on the Foundation repository.

Working Process: In this scenario, the developer will check out the component to modify or would create a new component in Project repository. The component will be checked in to Foundation repository after making the required changes/development. This process will ensure that all the metadata (BI and DI) will exist in the Foundation repository only.

After completing the development work, the metadata will be migrated from Development to Test environment and then to Production environment. This scenario is very straight forward and does not require any additional responsibility to deal with BI metadata in comparison to DI metadata as all metadata exists in

the Foundation repository and it can be directly accessed by the SAS 9 web tools.

Project Structure Two:

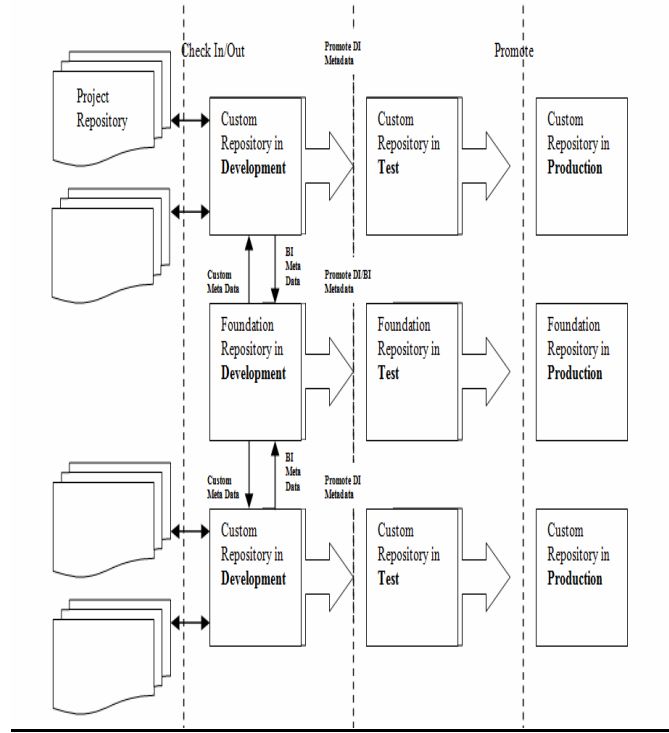


Figure: 4

Project Design: As shown in Figure 4, there are multiple child Custom repositories of Foundation. Further, on the basis of Custom repository, multiple Project repositories are created.

Working Process: In this scenario, developer will check out the components from custom repository to edit or will create new components in the Project repository. The components will be checked in the dependent Custom repository after making the required changes/development. Custom repository will have all BI/DI metadata.

As discussed earlier, BI metadata of the Custom repository will not be available to the SAS 9 web tools and to access BI metadata of custom repository from web tools, it should be moved to the Foundation repository first. It is not mandatory to move the DI metadata from custom repository to Foundation to get its functionality.

After following above mentioned steps, metadata will be promoted from Development to Test to Production environment as per the landscape design.

1. REFERENCES

Web Terminologies Used for SAS Suite

For better understanding of SAS 9 BI tools, it is important to understand the web terminologies of this Suite.

Web Server: Web Server is based on client-server model and uses HTTP (Hypertext Transfer Protocol) to execute the user request to display in the form of Web pages.

Application Server: An application server is a component-based product that resides in the middle-tier of a server centric architecture. It provides middleware services for security and state maintenance, along with data access and persistence.

Hostname: Hostname is the unique name by which a computer is known on a network.

Port: Port is a logical connection place which specifically uses the TCP/IP (Transmission Control Protocol/Internet Protocol) to listen the request from the remote user.

Higher-level applications that use TCP/IP such as the Web protocol, HTTP, have ports with pre-assigned numbers. These are known as "well-known ports" that have been assigned by the Internet Assigned Numbers Authority (IANA). Other application processes are given port numbers dynamically for each connection. When a service (server program) is started, it is said to bind to its designated port number. As any client program wants to use that server, it also must request to bind to the designated port number.

Port numbers are from 0 to 65536. Ports 0 to 1024 are reserved for use by certain privileged services. For the HTTP service, port 80 is defined as a default and it does not have to be specified in the Uniform Resource Locator (URL).

Socket: Normally, a server/application runs on a specific computer and has a socket that is bound to a specific port number. The server just waits, listening to the socket for a client to make a connection request.

The client must know the hostname of the machine on which the server/application is running and the port number on which the Server is listening. To make a connection client invokes the

socket connection by specifying the hostname and port number. The client also needs to identify itself to the server so it binds to a local port number that it will use during this connection. This is usually assigned by the system.

If the connection is accepted on the client and server side, a socket is successfully created and the client can use the socket to communicate with the server.

Servlet: Servlet is a Java application that runs on a Web server or Application server and provides server-side processing such as accessing a database and e-commerce transactions. It is widely used for Web processing and are designed to handle HTTP requests such as Get and Post.

Java Server Page (JSP): JSP is an extension to the Java Servlet technology from Sun that allows HTML to be combined with Java on the same page. Java provides the logic processing and the HTML provides the page layout that will be rendered in the Web browser. At time of compilation JSP is also converted into Servlet internally and then further compiled accordingly.

Extensible Markup Language (XML): XML is a flexible way to create common information formats and share both the format and the data on the World Wide Web, intranets, and elsewhere.

For example, computer makers might agree on a standard or common way to describe the information about a computer product (processor speed, memory size, and so forth) and then describe the product information format with XML. Such a standard way of describing data would enable a user to send an intelligent agent (a program) to each computer maker's Web site, gather data, and then make a valid comparison. XML can be used by any individual or group of individuals or companies that want to share information in a consistent way.

Other SAS Material References

- Online material available on SAS web site
- Working experience on SAS Suite

2. ACKNOWLEDGMENTS

I would like to give many thanks to Mr. Subodh Kumar Agrawal for reviewing this document and to making it more useful and readable to the readers. subodh_agrawal@rediffmail.com

I would also like to thank all my colleagues, who have helped me in gathering the information, which is the base to write this document.