218-30

The SAS Configuration Wizard: The Essential Final Step in Completing Your SAS® BI Installation and Deployment

Alec Fernandez, SAS Insititue Inc., Cary, NC

ABSTRACT

SAS Business Intelligence (BI) Architecture provides a framework for effectively utilizing many different software packages from a variety of vendors and open-source providers to form an enterprise-class platform. Beginning with SAS products, each software package is highly specialized toward performing a certain set of actions, ranging from running complex statistical models using SAS/ETS software, to running Java Server Pages (JSPs) on a Web server, to displaying interactive graphics on a Web browser. The SAS Configuration Wizard configures all the different parts of the SAS BI Architecture, which can then be deployed across many different hardware platforms and operating systems and function in concert as a cohesive, centrally managed system that enables each application to leverage the strengths of other applications.

INTRODUCTION

The deployment of the SAS BI Architecture involves two separate phases:

- **Installation.** During installation, each separate product or component is moved from the distribution media (CD-ROM, DVD, or Internet) and installed on the machine on which it will run. For the SAS Foundation, installation involves unpacking and installing the files, performing post-installation configuration, and verifying that the installation is functional. On Windows, shortcuts and Windows registry updates are additional installation steps. All other software in the SAS BI Architecture (including SAS components, Java, and the WebDAV server) is installed in a similar manner.

  When installation is complete, most software is totally functional. For example, you can start the SAS Foundation and run SAS procedures and DATA step code, as in the past. The Web server can run Web applications. The Distributed Authoring and Versioning (DAV) server can run as a stand-alone application.

- **The SAS Configuration Wizard.** The SAS Configuration Wizard creates the configuration directory that houses all the scripts and configuration files necessary to run a SAS Metadata Server. It populates the SAS Metadata Server with all the information necessary to create a deployment function (for example, user IDs, permissions, software invocation commands, and library assignments). It configures all the products that allow Web applications, such as the SAS Information Delivery Portal, to talk to the SAS Metadata Server and verify a user’s rights and the location of data sets. It configures the DAV server so that SAS Web Report Studio users can easily store reports from a Web browser. In summary, it configures all the software in a deployment so that each application can connect to all other applications and work collectively to leverage the strengths of all applications and deliver the maximum benefit.

After both phases are complete, you can use your Web browser to connect to a Web server and run a Web application to perform the following steps:

1. Verify your identity and ensure that you have permission to run the report that you requested.
2. Invoke SAS Foundation to run a report that runs against a database on a separate system.
3. Format output that is consistent with your department preferences.
4. Send the report back to the requesting Web browser.
5. Store the report definition and output in a location on your DAV server.

All of these steps can be performed from within the SAS Information Delivery Portal.

PLANNING YOUR SAS DEPLOYMENT

The SAS Configuration Wizard requires a considerable amount of information to perform the configuration, such as which products/components to deploy on each server in an enterprise, machine names, TCP/IP settings, user IDs and passwords, installation directory locations, and user preferences. The information comes primarily from these sources:

- the project directory, specifically the **plan.xml** file in this directory
- information that you enter when prompted by the SAS Configuration Wizard
- files that were placed on your computer during the installation phase (specifically the **Properties** files)
- the operating system itself (specifically environment variable settings and Windows registry settings)

The **plan.xml** file contains a list of all the computers on which the SAS BI Architecture will be deployed and indicates which of the individual components is deployed on each computer.

Conceptually, the **plan.xml** file has the following structure:

```
My Custom Configuration
    My_Unix_Server
            SAS Foundation
            SAS Integration Technologies
            SAS Management Console
            SAS Foundation Services
            SAS OLAP Server
            SAS Metadata Server
            SAS Workspace Server
            SAS Stored Process Server
            SAS Object Spawner
            SAS Connect Spawner
    My_Webserver_Machine
            SAS Foundation Services
            BEA WebLogic Server
            SAS Web Infrastructure Kit
            SAS Web Report Studio
    My_DAVServer_Machine
            Xythos WebFile Server
    My_Windows_Client_Machine
            Open OLAP Cube Studio
            SAS Enterprise Miner
            SAS Enterprise Guide
```

The **plan.xml** file is crucial to a successful SAS BI Architecture deployment because it defines what your deployment will look like, which products it will comprise, and where the products will be deployed.

SAS software can be run on a variety of hardware and software platforms. You can place each component of a SAS deployment on a separate machine, or you can group components on server-class machines. You can do an entire BI deployment on a single PC. This tremendous flexibility in configuring software adds an equal measure of complexity. Because of this complexity, you should carefully consider your computing resources before deciding on a deployment plan.

After you have determined which components to deploy on which machines, the **plan.xml** file should reflect this information accurately. The SAS Configuration Wizard analyzes the **plan.xml** file to determine which products/components to configure on each tier of the deployment and how to deploy the configuration directory. Because installation and configuration are run separately on each machine in the deployment, the **plan.xml** file has a section for each machine. However, the SAS Configuration Wizard has to consider the entire **plan.xml** file, including products or components that are not on the machine or tier that is being configured, when deploying on each machine. For this reason, the **entire plan.xml** file has to be correct for all machines in the deployment before you can begin the installation on the first machine. After you begin deployment on the first machine, you should not alter the contents of the **plan.xml** file.

For example, if your **plan.xml** file has the SAS Metadata Server on a stand-alone server, but you later move the SAS Metadata Server to the server where the SAS Workspace Server and SAS Object Spawner reside, then the SAS

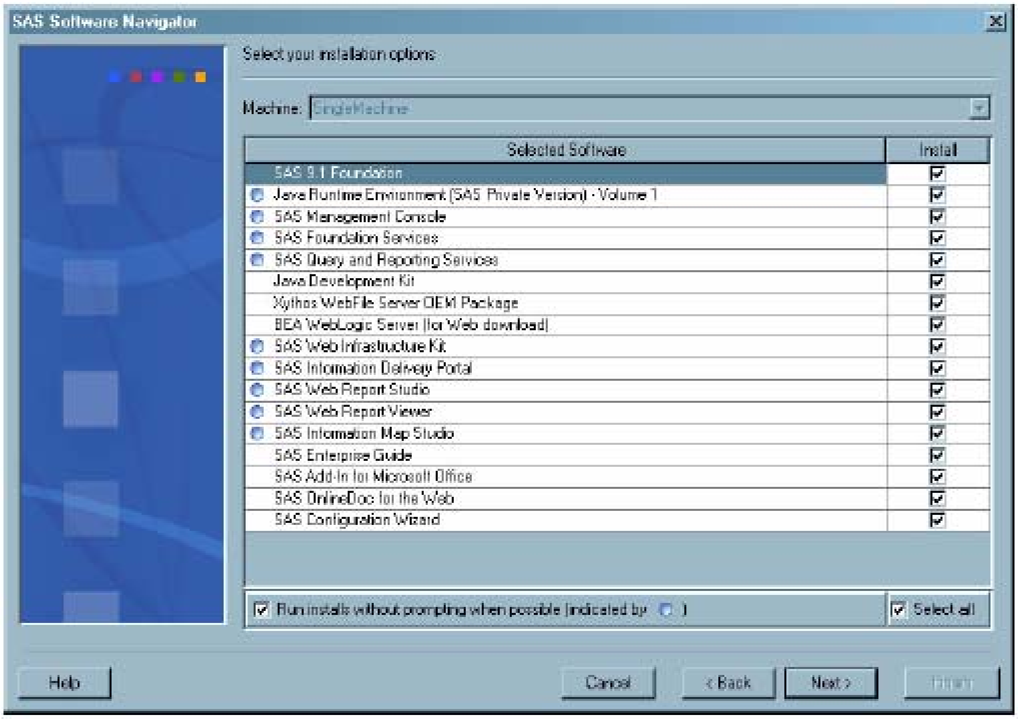
Configuration Wizard configures the deployment incorrectly and the software does not function. Or, if you use an Apache Tomcat server as your Java servlet container on the middle tier in your **plan.xml** file, but later decide to use a BEA WebLogic server, then the SAS Configuration Wizard configures the deployment incorrectly on both the server tier and the middle tier.

The most common causes of difficulty when deploying the SAS BI Architecture in a complex, multi-machine environment are mistakes in the **plan.xml** file. Sample plan files are provided on the installation media that depict typical, best practices deployment scenarios. You should use these sample plan files if they are appropriate. If you cannot use the sample plan files directly, then use them to create a plan file that meets your specific needs. The names of the project directories that contain the sample plan files reflect the types of deployments that they define. For example, the plan file in the **EntETLServer2Machines** project directory could be used to configure a two-tier, SAS Enterprise ETL Server deployment. The plan file in the **EMMin** project directory could be used to configure SAS Enterprise Miner on a single machine.

Each machine and product in a deployment has a variety of settings that configure its behavior. Most of the information that pertains to SAS components is captured during the installation process; however, a SAS deployment can involve different software from a variety of vendors. The SAS Configuration Wizard needs information that pertains to third-party products. You must have all configuration settings available before you begin your SAS deployment. The SAS Pre-Installation Checklists serve as an excellent resource to organize and record all the information that you need while running the SAS Configuration Wizard. Additionally, the checklists provide essential information on tasks that must be completed prior to beginning your SAS installation. Checklists are operating system- and SAS version-dependent and can be found at http://support.sas.com/installcenter. (Choose your operating system and the link for **Planning Installation Edition Kit**.)

RUNNING THE SAS CONFIGURATION WIZARD

SAS deployment begins by invoking the SAS Software Navigator **on the machine that has been chosen to run the SAS Metadata Server**. The SAS Software Navigator prompts you for the physical media (CD-ROM) or SAS Software Depot location, and for the location of your project directory, which as previously noted is the directory that contains your **plan.xml** file. It prompts you for the locations where other SAS components, including the SAS Foundation, will be installed. It lists all software components that are included in the **plan.xml** file, as shown in Figure 1.



Select your installation options

Machine: SingleMachine

Selected Software	Install
SAS 9.1 Foundation	☑
● Java Runtime Environment (SAS Private Version) - Volume 1	☑
● SAS Management Console	☑
● SAS Foundation Services	☑
● SAS Query and Reporting Services	☑
Java Development Kit	☑
Xythos WebFile Server OEM Package	☑
BEA WebLogic Server (for Web download)	☑
● SAS Web Infrastructure Kit	☑
● SAS Information Delivery Portal	☑
● SAS Web Report Studio	☑
● SAS Web Report Viewer	☑
● SAS Information Map Studio	☑
SAS Enterprise Guide	☑
SAS Add-In for Microsoft Office	☑
SAS OnlineDoc for the Web	☑
SAS Configuration Wizard	☑

☑ Run installs without prompting when possible (indicated by ●)

☑ Select all

Figure 1. The SAS Software Navigator **Select your installation options** Window

In most cases, the SAS Configuration Wizard is invoked as the last item in a SAS deployment. If you need to invoke the SAS Configuration Wizard after installing SAS so that you can re-run the configuration or configure a second deployment, the following command is used (all text should be on one line):

```
C:\My_media_location\config1cd\cfgwizard\Disk1\InstData\Windows\setup.exe
-DINPUT_SELECTED_MACHINE="SingleMachine"
-DINPUT_PROJECT_DIRECTORY="C:\\My_Project_Directory "
-DSAS_HOME="C:\\Program Files\\SAS"
```

Note: The double backslashes \\ are needed because Java considers the backslash \ character as special, so the backslash \ must be escaped with a second backslash \.

An alternative method of invoking the SAS Configuration Wizard, which eliminates the possibility of typographical errors, is to invoke the SAS Software Navigator and navigate to the **Select your installation options** panel (shown in Figure 1). Because the SAS Software Navigator records your responses to prompts from earlier invocations, this invocation is simpler because now, you have to select only the **Next** button. When the **Select your installation options** window is displayed, select the **Select all** check box (which toggles the **Install** status of all products/components in the list to ***not be checked***), select the check box next to **SAS Configuration Wizard**, and select the **Finish** button. As a result, products/components are not re-installed and the SAS Configuration Wizard runs.

After the SAS Configuration Wizard is invoked, it prompts you for your language preference, and prompts you for a path where it should create the configuration directory, which is the directory on each server tier or middle tier where configuration files should be written. The tier that runs the SAS Metadata Server is where the SAS Metadata Repository is located. Because data written to the configuration directory is significantly controlled by information in the **plan.xml** file, you should name the configuration directory so that it connotes this close association. By default, the configuration directory has the same name as the project directory.

If you are configuring SAS servers and spawners on a Windows machine in multiple user ID mode (described below), the **SAS Server Configuration Options** panel appears. This panel enables you to specify whether you want to run SAS servers and spawners as Windows services, or whether you want to start SAS servers and spawners using scripts. You should run SAS servers and spawners as Windows services, because running them as services

- ensures that they are restarted in the correct order whenever you restart Windows
- runs SAS servers as the Local System account, which has special security settings
- adds messages to the event log if an error is reported
- registers the dependencies between the servers so that stopping one server stops other dependent servers.

If you do not run SAS servers and spawners as Windows services, then you must add special security settings to the user ID that runs them. In addition, you must remember to restart the SAS servers and spawners in the correct order whenever you restart Windows. (For more information, see the *SAS 9.1.3 Intelligence Platform: Planning and Administration Guide*, specifically the SAS Pre-Installation Checklist and Chapter 7.) For your convenience, Windows shortcuts are added for stopping, starting, and restarting the servers, whether they are installed as scripts or services.

On UNIX machines, scripts are provided to stop, start, and restart services. These scripts facilitate incorporation into UNIX init processing. The UNIX system administrator can decide how to incorporate scripts based on best practices for the particular version of UNIX or Linux. Scripts on a UNIX machine have the same benefits of automatically starting the SAS servers and spawners in the correct order on a Windows machine.

Depending on the components that you are configuring, you are prompted for various additional pieces of information. If you are proceeding through installation and configuration in the documented manner, you will find correct responses to these prompts in the SAS Pre-Installation Checklist. All information must be carefully documented for the deployment to go smoothly. When prompted for user ID information and passwords, you must supply correct passwords for user IDs on the operating system that is running the SAS Metadata Server. Unless you are configuring a single user ID deployment, the SAS Configuration Wizard does not validate the passwords against the operating system. If you enter an incorrect password, you will not receive an error, and the deployment will not function correctly.

SINGLE USER ID DEPLOYMENTS

Some configurations require only a subset of SAS servers. For example, SAS Enterprise Miner requires a SAS Metadata Server and a SAS Workspace Server. SAS Enterprise Guide uses a SAS Metadata Server, SAS Workspace Server, and SAS Stored Processes Server. For these deployments on Windows, you can configure the software to run on a single Windows machine under your usual user ID, instead of under five different SAS user IDs.

If the deployment that is defined in your **plan.xml** file supports single user ID methodology, you are prompted to choose either a single or multiple user ID deployment during configuration. Because of its simplicity, a single user ID deployment can automatically update the current user ID with the necessary security settings and verify the supplied password. A single user ID deployment installs all necessary SAS servers as services and deploys all necessary metadata automatically and silently so that installation is much easier and less time-consuming.

A single user ID deployment does not offer the full range of authorization functionality that is implicit in a multiple user ID deployment. A single user ID deployment is not appropriate for multi-user operating systems, such as UNIX or z/OS, where all users share a common SAS Metadata Server. However, if you are deploying on a Windows PC, and you are the only person who intends to use the software on that particular Windows PC, then the simplicity of the single user ID deployment, coupled with the reduced administration burden, make it the desirable option.

If you are not prompted to choose either a single or multiple user ID deployment during configuration, it is because your deployment is too complex. You must create the necessary user IDs as described in the Pre-Installation Checklist and answer the prompts for the multiple user IDs and passwords.

ADVANCED PROPERTIES EDITOR

When the SAS Configuration Wizard is installed, many files are copied to your system. These files are **Properties** files and contain default values for settings that control your deployment. **Properties** files are grouped by operating system and by the components listed in your **plan.xml** file.

Conceptually, a **Properties** file includes a list of name-value pairs. For example:

```
#Open Metadata Architecture Server Settings
OMAHOST=$HOSTNAME$
OMAPORT=8561
OMAREP=Foundation
OMAAUTH=DefaultAuth
OMAREPDIR=MetadataServer$/$MetadataRepositories$/$Foundation
#Workspace Server Settings
WSPHOST=$IOM_HOST$
WSPPORT=$IOM_PORT$
#Access Control Template Name
DEFAULTACT=Default ACT
#   Shared User Definitions
OMAANAME=SAS Administrator
OMAADMINU=$DNSDOMAIN$\sasadm
#Web Server Settings
WEBSRVPORT=8080
WEBSRVHOST=$FULL_QUAL_LOCAL_HOST$
WEBSRVLOC=http://$WEBSRVHOST$:$WEBSRVPORT$
```

Each property represents one setting for one or more products or components in the deployment. Some properties, such as **WEBSRVLOC** in the previous example, are combinations of previously defined properties.

Some properties get their default values from the operating system and configuration files that were installed during the installation phase. Some default values are supplied by you when you respond to SAS Configuration Wizard prompts.

For the majority of properties, default values are chosen to enable a successful initial deployment on all supported platforms. The **Advanced Properties Editor** window is the last prompt displayed before the configuration directory is generated. The **Advanced Properties Editor** window enables you to override the default values of a subset of the properties that the SAS Configuration Wizard uses while configuring the current machine. There are many properties that control every aspect of a SAS deployment; for example, the port assignments in a complex deployment require dozens of properties. Many of these properties have interdependencies. While the **Advanced Properties Editor** window enables tremendous flexibility in changing settings when conflicts occur, extreme care must be taken when changing properties. The **Advanced Properties Editor** window is intended for highly advanced users who need to make minor modifications to their configurations to accommodate non-standard port assignments and atypical deployments.

Chapter 6 of the *SAS 9.1.3 Intelligence Platform: Planning and Administration Guide* includes some property settings and a complete list of default port assignments. If you have applications that use TCP/IP ports outside of the typical Windows or UNIX standard port definitions, you should review port numbers to check for conflicts with other port numbers that you are using for other applications. If you are using a BEA WebLogic server, the **properties** file contains values for the name and port number of an **Administration Server** and a **Managed Server**. These property values must match the values that you specified when you installed and configured the Web servers. Similarly, you should check the values used by the IBM WebSphere server.

FINISHING THE CONFIGURATION

The last window in the SAS Configuration Wizard checks disk space requirements and begins to generate the actual configuration. This is the section where the configuration is actually deployed. The SAS Configuration Wizard builds a well-defined directory structure under the configuration directory. Inside this directory structure, many customized scripts are created, Web applications are built and deployed, properties are written into files that control the deployment, and other actions are performed. In the single user ID mode, your SAS Metadata Server is started and all the metadata objects are defined and loaded. This is the stage where SAS deployment process culminates; it can take several minutes and should not be interrupted.

CUSTOMIZED INSTRUCTIONS.HTML

After the SAS Configuration Wizard finishes deploying and configuring products or components, it invokes a Web browser and displays a file named **instructions.html**, which is stored in the configuration directory that you specified.

If you chose the single user ID deployment, then you have finished your installation. The instructions in **instructions.html** explain how to start your software and perform optimization and maintenance tasks because all required SAS servers are running and all metadata has been deployed.

If you chose the multiple user ID deployment, you will need to complete the following tasks:

1. Start the required SAS servers in the correct order.
2. Start SAS Management Console so that it can communicate with the SAS Metadata Server.
3. Deploy customized metadata to the SAS Metadata Server.
4. Deploy customized Web applications to the Web application server.
5. Configure the Web application server to run the Web applications (except on Apache Tomcat).

METADATA DEPLOYMENT

The first step in metadata deployment is to start the SAS Metadata Server and SAS Management Console. The instructions in **instructions.html** guide you through starting the SAS Metadata Server and SAS Management Console. On UNIX and Windows, they will have already been started for you. After the SAS Metadata Server is running, you should establish a connection to it using SAS Management Console. Upon connecting to the SAS Metadata Server, SAS Management Console requests information from the SAS Foundation Repository. If this request is the first request from SAS Management Console during configuration, then SAS Management Console will not find a SAS Foundation Repository on the SAS Metadata Server and you will be prompted to follow the instructions necessary to create one.

To facilitate metadata deployment, the SAS Configuration Wizard installed a series of scripts into the configuration directory. Instructions for running these scripts are listed in the **instructions.html** file. Instructions should be followed precisely because skipping a step or performing steps out of order might result in errors and a failed deployment.

Each script takes customized metadata, in the form of an XML file, from the configuration directory and sends it to SAS Management Console. XML files are customized by merging property values from the SAS Configuration

Wizard with template files that were delivered with the SAS Configuration Wizard. SAS Management Console interprets these XML files and relays the resulting metadata to the SAS Metadata Server. Depending on the complexity of the installation, there can be many scripts to run. As noted earlier, you should run scripts in the correct order, do not skip any scripts, and do not run scripts more than once. In addition, manual metadata manipulation via SAS Management Console might be needed after a script runs, but before the subsequent script can run. If this is the case with a script, it is clearly documented in the **instructions.html** file and you should perform the instructions precisely.

After metadata deployment is complete, the SAS Metadata Server is fully configured and ready to control the deployment.

WEB APPLICATION CONFIGURATION AND DEPLOYMENT

During configuration, any selected SAS Web applications, such as SAS Web Report Studio or the SAS Information Delivery Portal, must be built. The SAS Configuration Wizard inserts property values into XML files and other configuration files in the SAS Web application to customize the files for your computing environment. The SAS Configuration Wizard invokes scripts in the SAS **HOME** directory to build the **.war** and **.ear** files that house the SAS Web applications. After these **.war** and **.ear** files are built and ready for deployment, the SAS Configuration Wizard copies them into the **webapps** configuration directory. The configuration directory structure is described in more detail in the following section.

If you have chosen Apache Tomcat as your Web server, the SAS Configuration Wizard copies the **.war** files to the Tomcat **webapps** directory so that the Web application will be deployed when Tomcat is restarted. If you have chosen an alternate Web server, then the location of the **.war** files is provided in the **instructions.html** file but the task of deploying the applications on the Web server is left to the Web administrator.

TROUBLESHOOTING

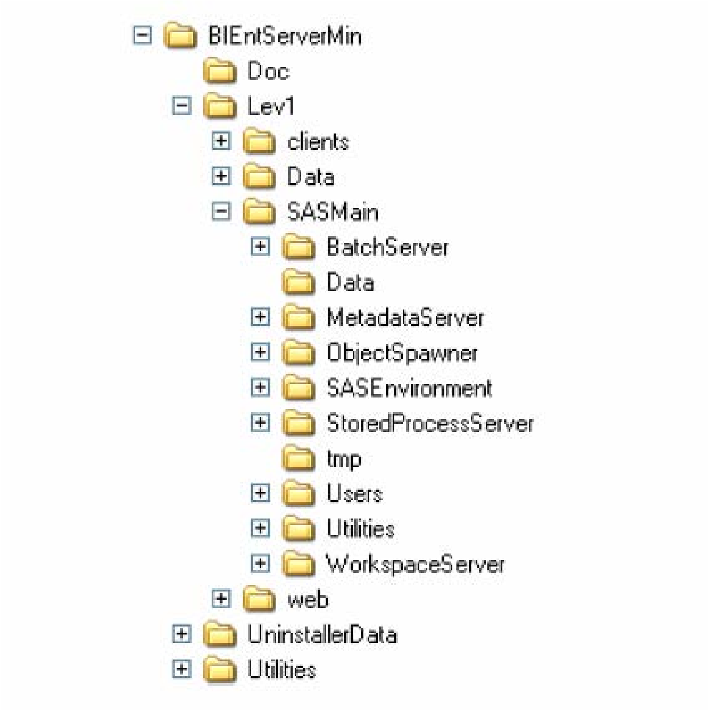
At the end of the **instructions.html** file, URLs are included that you can use to test the Web applications that you configured. If the Web applications are not functioning correctly, or if server connections do not work within SAS Management Console, see Chapter 8 of the *SAS 9.1.3 Intelligence Platform: Planning and Administration Guide* for debugging information.

Error log files can be reviewed to determine the cause of problems. The location of error log files, root causes of errors, and circumventions are listed in the *SAS 9.1.3 Intelligence Platform: Planning and Administration Guide*.

CONFIGURATION DIRECTORY

When you run the SAS Configuration Wizard on a server tier or middle-tier host, it prompts you for the name of a configuration directory, which contains important files for running your system. The contents of the configuration directory are documented in Chapter 9 and Appendix 1 of the *SAS 9.1.3 Intelligence Platform: Planning and Administration Guide*.

Figure 2 shows the structure of the configuration directory for a typical deployment that contains both server components and middle-tier components on the same machine.



```
⊟ BIEntServerMin
    Doc
  ⊟ Lev1
    ⊞ clients
    ⊞ Data
    ⊟ SASMain
      ⊞ BatchServer
        Data
      ⊞ MetadataServer
      ⊞ ObjectSpawner
      ⊞ SASEnvironment
      ⊞ StoredProcessServer
        tmp
      ⊞ Users
      ⊞ Utilities
      ⊞ WorkspaceServer
    ⊞ web
  ⊞ UninstallerData
  ⊞ Utilities
```

Figure 2. Directory Structure for a Typical Deployment

The following directories are at the top level of the configuration directory structure:

Doc
provides a location for you to save your own documents.

Lev1
identifies information that is at production status. Beneath a level, you can create directories that are unique to the level in which they are contained. For example, you could create **Lev2** and **Lev3** directories to support a **development/test/production** environment. However, additional levels are currently supported only by SAS ETL Studio.

UninstallerData
contains utilities to uninstall the configuration.

Utilities
contains utilities that are used at the environment level. For example, this directory contains the facility used to deploy metadata to the SAS Metadata Server.

For more information about configuration directory structures for other types of deployments, see the *SAS 9.1.3 Intelligence Platform: Planning and Administration Guide*.

SAS APPLICATION SERVERS

The **SASMain** directory houses the SAS Application Servers and the **web** directory houses SAS Web applications. Many files in the configuration directory have been customized for your computing environment with property values from the SAS Configuration Wizard. For this reason, the configuration directory cannot be moved from one machine to another machine without extensive modifications. In addition, the configuration directory tree should be treated as a unit when performing backups.

A SAS application server is a logical concept that is used to group SAS sessions that are similarly configured to perform certain tasks. For instance, a SAS Workspace Server is similar to a traditional SAS session being used to run traditional SAS code. The key difference is that the SAS Program Editor, output, and logs can be separated from the running SAS session. SAS Enterprise Guide and SAS/CONNECT software users are familiar with this logical concept because they run client applications on one machine and the SAS session on another machine. This concept has been expanded on and made more flexible by the SAS application server concept. SAS application servers can be remotely administered via SAS Management Console and grouped by their intended use.

For example, consider a SAS application server named **SASAccounting**. The intended use of all SAS sessions invoked using the **SASAccounting** server would be to support accounting activities. Keep the **SASAccounting** server example in mind as the following paragraphs describe how the SAS Configuration Wizard arranges the configuration directory.

The root of the SAS application server tree is the **SASMain** directory. The **SASMain** directory contains an **autoexec.sas** file (actually named **appserver_autoexec.sas**) and a **sasv9.cfg** file. These files are global to all SAS sessions that are invoked in that particular SAS application server environment, which enables you to have special memsize settings, format search criteria, and other SAS option settings that are applicable to a specific task or activity, such as accounting in the **SASAccounting** server example. In addition, you can have library assignments and settings that facilitate connection to the database servers that contain accounting transaction data, following the **SASAccounting** server example. Because many users will be starting SAS sessions sharing this server definition, **rsasuser** is in the **sasv9.cfg** file by default.

The configuration for each application server or Web application server that you define on a machine is contained in a subdirectory of **SASMain**. For example, there will probably be a **MetadataServer** subdirectory and an **ObjectSpawner** subdirectory. On UNIX and Windows machines, each subdirectory contains a script named **<*server-type*>.[sh/bat]** that is used to manage the servers. On Windows machines, you can call the scripts directly or you can use the **Start** menu. Each subdirectory contains a **logs** directory, in which the servers write log files. If you experience difficulty with a server, the **logs** directory is a good place to troubleshoot. For example, if you cannot connect to the SAS Metadata Server using the SAS Management Console, an error is likely to be recorded in the **SASMain/MetadataServer/logs** directory. In the **MetadataServer** directory, you will find the **rposmgr** directory where the SAS Foundation Repository is located, and the **MetadataRepositories** directory where other metadata repositories are located.

SAS Stored Processes Servers are another example of the SAS application server concept. They are similar to SAS Workspace Servers. One critical difference is that SAS Stored Processes servers are intended to run many short-lived jobs for many users who connect to SAS Stored Processes servers through a GUI (either via the Web or another client, such as SAS Enterprise Guide or Microsoft Excel). Another critical difference is that SAS Stored Processes servers are threaded to allow for load-balancing separate requests on the same machine. The SAS Configuration Wizard deploys a different **sasv9.cfg** file in the **StoredProcessServer** directory of the configuration. You could alter this specific file to reduce the memsize setting to enable more of these short-lived jobs to run simultaneously. However, this **sasv9.cfg file** specifically includes settings from the **SASMain sasv9.cfg** file, so changes made in the **SASMain sasv9.cfg** are still included (for example, changes made to support the accounting database servers in the **SASAccounting** server example are included in the configuration file) and will be used unless they are specifically overridden.

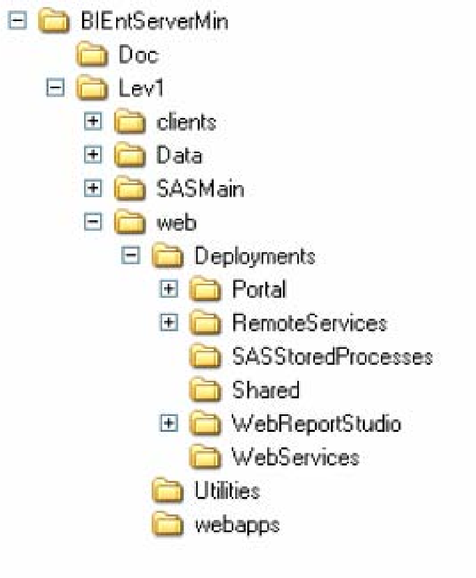
All application servers deployed by the SAS Configuration Wizard are easily scaled to handle more workload by simply adding more connections to your logical server and because of the characteristics of an application server, any SAS job that runs on one application server is guaranteed to run in the same fashion as any other application server.

SAS WEB APPLICATIONS ON MIDDLE-TIER MACHINES

The SAS Configuration Wizard creates a special directory structure in the configuration directory on middle-tier machines. The configuration directory contains customized configuration files for SAS Web applications. The customized configuration files for Java-based application servers are split between the **web** subdirectory of the configuration directory and the equivalent subdirectories of the SAS **HOME** directory. The interrelationship of these two locations is described in the following section.

Because SAS Web applications need to communicate with other components of the SAS infrastructure, they have to be configured with the same information that was used on the server tier. To communicate in a secure manner, some of these customized configuration files have to contain the user IDs and (obfuscated) passwords that are used to authenticate these SAS Web applications when they need to communicate with the server tier.

Figure 3 shows a typical configuration directory in a middle-tier deployment.



```
⊟ BIEntServerMin
    Doc
  ⊟ Lev1
    ⊞ clients
    ⊞ Data
    ⊞ SASMain
    ⊟ web
      ⊟ Deployments
        ⊞ Portal
        ⊞ RemoteServices
          SASStoredProcesses
          Shared
        ⊞ WebReportStudio
          WebServices
        Utilities
        webapps
```

Figure 3. Typical Directory Structure for a Middle-Tier Deployment

If you installed Apache Tomcat as your servlet container, one of the files in the **web** directory is a script named **startServletContainer.[bat/sh]** that starts Tomcat. On UNIX machines, you can call this script directly to start Tomcat. On Windows machines, a shortcut is provided. Note that on Windows machines, do not start Tomcat by selecting the **Start Programs-> Apache Tomcat 4.1 -> Start Tomcat** selection path that was installed with Tomcat. This selection path starts Tomcat, but not with the options that were inserted by the SAS Configuration Wizard. These options are required by SAS Web applications.

THE DEPLOYMENTS DIRECTORY

The **Deployments** directory contains a subdirectory for each SAS Web application that is deployed on a machine. Note that the **Portal**, **RemoteServices**, **WebReportStudio**, and **WebReportViewer** directories contain **logs** directories. These **logs** directories contain log files for the different SAS Web Applications and you should consult log files any time you experience a problem with a SAS Web application.

THE WEBAPPS DIRECTORY

The **webapps** directory contains the .**war** files for SAS Web applications, such as the SAS Information Delivery Portal. These **.war** files contain all files that make up the SAS Web application, such as .jars, .jsps, and .html documents.

SAS Web applications are built using scripts that are located in the SAS **HOME** and **web** directories. The SAS Configuration Wizard inserts property values into the configuration files in the SAS **HOME** directory and invokes the scripts that assemble the **.war** and **.ear** files. Then, these SAS Web applications are copied into the **webapps** subdirectory, which makes it easier to deploy all the SAS Web applications because they all reside in one subdirectory.

If you are using Tomcat as your servlet container to execute your SAS Web applications, the SAS Configuration Wizard copies **.war** files to the Tomcat **webapps** directory. If you are using BEA WebLogic or IBM WebSphere as your servlet container, SAS Web applications are deployed through the server's administration console.

CONCLUSION

The SAS Configuration Wizard is the essential final step in bringing together the different software components that comprise the SAS BI Architecture into one cohesive application. Although long-time SAS users will notice the complexity associated with the SAS BI Architecture, with the aid of the SAS Configuration Wizard, the SAS BI Architecture can be fully configured. Then, the sum of benefits that come from the SAS BI Architecture can be experienced.

REFERENCES

SAS Institute Inc. 2004. *SAS® 9.1.3 Intelligence Platform: Planning and Administration Guide*. Cary, NC: SAS Institute Inc. Available http://supportexp.unx.sas.com/documentation/configuration/iaplanning913.pdf.

CONTACT INFORMATION

Alec Fernandez
SAS Institute Inc.
R3316 SAS Campus Drive
Cary, NC 27513
alec.fernandez@sas.com