SAS Publishing

SAS® 9.1.3 Intelligence Platform

Administration Guide

Fifth Edition

The Power to Know.

The correct bibliographic citation for this manual is as follows: SAS Institute Inc. 2006. *SAS® 9.1.3 Intelligence Platform: Administration Guide, Fifth Edition*. Cary, NC: SAS Institute Inc.

SAS® 9.1.3 Intelligence Platform: Administration Guide, Fifth Edition

SAS Institute Inc., SAS Campus Drive, Cary, North Carolina 27513.

1st printing, July 2006

SAS Publishing provides a complete selection of books and electronic products to help customers use SAS software to its fullest potential. For more information about our e-books, e-learning products, CDs, and hard-copy books, visit the SAS Publishing Web site at **support.sas.com/pubs** or call 1-800-727-3228.

Contents

PART 1

Introduction

CHAPTER

1

Introduction to Administering the SAS Intelligence Platform

SAS Intelligence Platform Documentation

SAS publishes a set of documents about the SAS Intelligence Platform. The principal documents in this set are described in Table 1.1.

Table 1.1 SAS Intelligence Platform Documentation

Title	Description
SAS Intelligence Platform: Overview	□ Explains what the SAS Intelligence Platform is and how it can benefit your business. □ Discusses the architecture of the platform, including the roles of SAS and Java application servers. □ Details your data storage options. □ Explains the purpose of the various client applications that can be part of the system. □ Provides an overview of the SAS 9 security model.
SAS Intelligence Platform: Installation Guide	□ Explains how to install the SAS Intelligence Platform and how to perform the initial configuration of the system. □ Lists additional configuration tasks that you may need to perform and directs you to the appropriate part of the SAS Intelligence Platform document set for information about these tasks.

Title	Description
SAS Intelligence Platform: Administration Guide	□ Explains how to manage your metadata server and your metadata repositories. □ Discusses establishing connectivity to your data sources and optimizing data storage. □ Covers the administration of clients such as SAS Data Integration Studio, SAS Information Map Studio, and SAS Web Report Studio. □ Explains how to tune your SAS servers and Java application server to improve the performance of the system. □ Provides information about system maintenance.
SAS Intelligence Platform: Security Administration Guide	□ Explains how to use external authentication providers (such as the host operating system) to validate user identities. □ Discusses support for single sign-on from a SAS application to other computing resources. □ Explains how to use the platform's metadata-based authorization facility, which determines who can take which actions on which resources.

The document that you are reading now explains how to administer the SAS Intelligence Platform after it has been installed. "What Is Covered in This Document?" on page 4 summarizes the contents.

To get the most out of this document, you should read the following material before you proceed to Part 2 or this book or beyond:

- Read *SAS Intelligence Platform: Overview* for an overview of the architecture of the platform.
- Read Chapter 2, "Understanding the State of Your System," on page 7 for an explanation of the state of your system immediately after installation. The installer will have created users and groups that you need to know about. Also, the wizard used to configure your system will have performed many tasks that you need to know about. For instance, it will have created configuration files, scripts, SAS data sets, and metadata objects.

What Is Covered in This Document?

As mentioned above, this document covers the post-installation administration of your system. It is divided into four main parts:

- Part 2, "Metadata Server Administration"
- Part 3, "Data Administration"
- Part 4, "Application Administration"
- Part 5, "Advanced Topics"

Part 2, "Metadata Server Administration," explains how to complete the configuration of the metadata server after the completion of a planned installation of the SAS Intelligence Platform. It also discusses managing both the server and the repositories that it controls. This management includes tasks such as creating new metadata repositories and backing up the metadata server.

Part 3, "Data Administration," focuses on the products that you need to install and the metadata objects that you need to create in order to establish connectivity to your data sources (and data targets). It also deals with topics such as setting up shared access to SAS data and explains how the use of different data-access engines affects security.

Part 4, "Application Administration," as its name implies, deals with administering individual applications. It contains chapters on administering SAS Data Integration Studio, SAS Information Map Studio, SAS Web Report Studio, SAS Web OLAP Viewer for Java, and SAS Enterprise Miner. (The SAS Information Delivery Portal has its own administration guide, the *SAS Web Infrastructure Kit: Administrator's Guide*.)

Part 5, "Advanced Topics," covers a number of topics. It discusses how to configure SAS servers, such as workspace and stored process servers, to improve the performance of the system. It discusses how to tune a Java application server to improve the performance of the Web applications SAS Web Report Studio, SAS Information Delivery Portal, and SAS Web OLAP Viewer for Java. It discusses how to set up a compute grid to speed up the execution of SAS code that can make use of a parallel algorithm. And it includes information on other administrative tasks, such as promoting metadata from one instance of the platform to another.

CHAPTER

2

Understanding the State of Your System

Overview of Understanding the State of Your System

Before you begin administering the SAS Intelligence Platform, you should understand a few things about the current state of your system, that is, your starting point. This short chapter covers the following subjects:

- When the installer set up your system, he or she created operating system user accounts for a special set of SAS users and groups. In addition, the installer registered a subset of these users in the metadata. The section "Users and Groups Used in the Configuration of the System" on page 8 explains what roles these users play in the configuration and what metadata objects were created to represent them.
- On each server- and middle-tier machine, the SAS Configuration Wizard created a *configuration directory*. This directory contains a set of subdirectories in which are stored configuration files, scripts, and special purpose SAS data sets. The section "Configuration Directories" on page 13 explains more specifically what type of files and directories you will find in your configuration directories.

- If your configuration contains a workspace server and a stored process server, these servers were configured in a particular way. It is important to understand how they were configured and what your options are for reconfiguring them. For information on this topic, see "Workspace and Stored Process Servers" on page 17.
- During installation, the SAS Configuration Wizard created scripts and shortcuts that you use to start and stop the various servers in your system. The section "Starting and Stopping Your SAS Servers" on page 18 explains how to start and stop the servers on all supported platforms and explains the cases in which you must start servers in a particular order.

Users and Groups Used in the Configuration of the System

During installation, a number of operating system user accounts and groups were created. The exact users and groups that were created depend on the software that was installed and the platforms on which that software was installed. For information on exactly what user accounts and groups were defined, see the chapter "Setting Up Required Users and Groups" in the *SAS Intelligence Platform: Installation Guide*. The first two subsections below explain what roles these users and groups play in the configuration, and the third subsection explains what metadata objects were created for these users and groups.

Users

The table below lists the users that were—or may have been—used in the configuration of your system and explains the role that each user plays in the configuration.

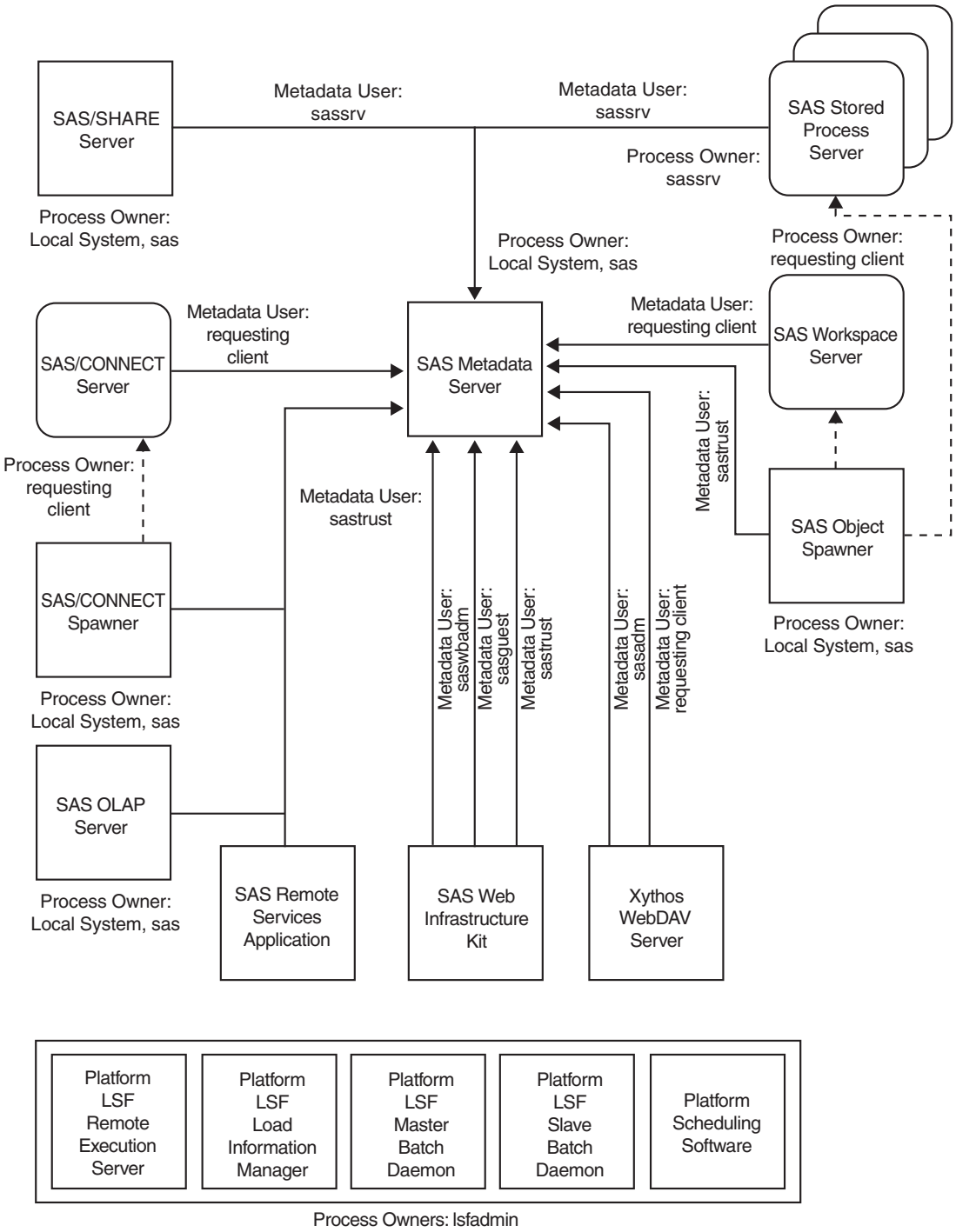
Table 2.1 Special Platform Users

User (Recommended Name)	Role
SAS Administrator (sasadm)	The SAS Administrator account is used in a couple of ways. First, during setup, the SAS Configuration Wizard uses this account to connect to the metadata server from SAS Management Console. Later, you can use this account to administer the metadata and OLAP servers. In addition, because sasadm is an *unrestricted user*, you can use the account to access any metadata in the metadata repository (except for passwords, which an unrestricted user can overwrite, but cannot read). You should not use this account to run applications other than SAS Management Console. Also, you should use it to run SAS Management Console only when you are performing tasks that require special privileges. In particular, do not use this account in cases where passwords need to be acquired, for example, when you are creating metadata for database tables.
SAS Trusted User (sastrust)	The SAS Trusted User account is used by the OLAP server, and Web applications that use Web authentication, to impersonate already authenticated clients on the metadata server. That is, the OLAP server and the Web applications authenticate clients; then, if a client needs to interact with the metadata server, these programs communicate with the metadata server on the client's behalf using the sastrust account. This arrangement prevents clients from having to be authenticated multiple times and from having accounts on multiple back-end servers. The sastrust account is also used by the object spawner. When the spawner receives a request to start a workspace or stored process server, it uses this account to connect to the metadata server in order to read the appropriate server definition.
SAS Demo User (sasdemo)	The SAS Demo User account is an account that you can use to test any of the SAS clients.
SAS General Server User (sassrv)	This user is the process owner for stored process servers. In addition, both stored process servers and SAS/SHARE servers use this account when they communicate with the metadata server.
SAS Web Administrator (saswbadm)	The SAS Web Administrator account has permission to administer the portal Web application. The portal Web application shell uses the SAS Web Administrator to perform specific tasks, such as deploying portlets and creating SAS group permission trees. The SAS Web Administrator also has administrative privileges for all of the portal Web application content. The SAS Web Administrator can access a portal user's pages and share content with any SAS group.
SAS Guest (sasguest)	The SAS Guest account is used to provide general access to your system's metadata. For example, if you have installed the SAS Information Delivery Portal, this user configures the Public Kiosk for the portal Web Application.

User (Recommended Name)	Role
LSF Administrator (lsfadmin)	This account is required if you are installing the Platform Suite for SAS, in support of either scheduling or grid computing. The Platform LSF services run as this user. In addition, this user is the owner of the LSF configuration and log files and has permission to perform cluster-wide operations, edit configuration files, and reconfigure a cluster.
LSF User (lsfuser)	This account is required if you will be scheduling reports. Command line programs like the Batch Report Generation Tool are run as this user.
SAS Installer (sas)	The installer uses the sas account when he or she installs and configures software on UNIX and z/OS systems. In addition, the SAS Installer is the owner of configuration directories and their contents and is the process owner for items such as the metadata server, the OLAP server, and the object spawner.

The following figure summarizes how some of these accounts are used by the SAS servers in a business intelligence system. The figure shows who owns each server process and which account each server uses to communicate with the metadata server.

Figure 2.1 Process Owners



Groups

Two operating system user groups may have been defined during installation.

If you have SAS servers running on Windows systems, most likely a group called SAS Server Users has been created. This group should have been assigned whatever user rights are required to start SAS servers on your Windows machines. You can grant

these same rights to any users you add to the system by making those users members of this group.

If your system includes UNIX machines, a group called sas should have been defined. This group is used to control access to the configuration directories on those machines. Typically, you will not add any users to this group.

Metadata Objects

At the end of the installation process, certain metadata objects must exist in your metadata repository. This section lists the User and Group objects that must have been defined in the metadata in order for your servers and applications to work correctly. You can use the User Manager plug-in to SAS Management Console to verify that these objects have been created properly.

Table 2.2 Summary of Metadata Identities

Metadata Identity	User ID*	Password**	Authentication Domain	Group Membership Information
User: SAS Administrator	sasadm			
User: SAS Trusted User	sastrust			member of: SAS System Services group, SAS General Servers group
User: SAS Guest	sasguest	********	DefaultAuth	
User: SAS Demo User	sasdemo	********	DefaultAuth	member of: Portal Demos
User: SAS Web Administrator***	saswbadm	********	DefaultAuth	member of: Portal Admins group, SAS System Services group
Group: SAS System Services				members: SAS Trusted User, SAS Web Administrator
Group: SAS General Servers	sassrv	********	DefaultAuth	members: SAS Trusted User
Group: Portal Admins***				members: SAS Web Administrator
Group: Portal Demos***				members: SAS Demo User
Group: WRS Administrator****				

Metadata Identity	User ID*	Password**	Authentication Domain	Group Membership Information
Group: WRS Report Author****				
Group: WRS Advanced User****				

* These are the recommended IDs. They should correspond to accounts in your authentication provider. On Windows, the user ID in the login should be fully qualified with a host or domain name, for example, *host-name*`\sasadm`.

** If you are logged on to SAS Management Console as an unrestricted user, you will always see ******** in the password column, even if no password was specified.

***You only need this metadata identity if you are running SAS Information Delivery Portal.

****You only need this metadata identity if you are running SAS Web Report Studio.

For more information about why the SAS General Servers group must be set up this way—and about the problems you will see if it is not set up this way—see "Overview of the Initial Load Balancing Setup for Stored Process Servers" on page 406.

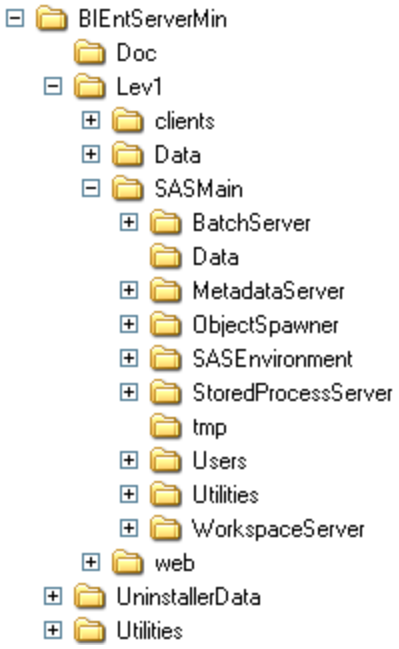
Configuration Directories

The SAS Configuration Wizard created a configuration directory on each server- and middle-tier machine in your system. The subsections below discuss the most frequently used files in these directories.

Configuration Directory: Server-Tier Machines

When you run the SAS Configuration Wizard on a server-tier host, the wizard prompts you for the name of a configuration directory. Then, in this directory, the wizard builds a directory structure that contains important files for managing your system. See the following display.

Display 2.1 Configuration Directory



For a complete discussion of this directory structure, its purpose, and its contents, see Appendix 2, "Understanding the SAS Configuration Environment," on page 519. This section discusses the most frequently used files in the structure.

Note: By default, the configuration directory is located in *drive*`:\SAS` on Windows systems, in *installer's-home-directory*`/SAS` on UNIX systems, and in the directory specified in the `CONFIG_DIR` environment variable on z/OS systems. △

Security-Related Files

If you configured a metadata server on this machine, the `MetadataServer` folder contains three files that affect security:

- `adminUsers.txt`
- `trustedUsers.txt`
- `trustedPeers.xml`

The `adminUsers.txt` file defines your system's *administrative users* and *unrestricted users*. The `trustedUsers.txt` file defines the system's *trusted users*. For explanations of the tasks these users can perform, see "Rights Granted to Administrative, Unrestricted, and Trusted Users" on page 29.

The configuration wizard creates one unrestricted user: the SAS Administrator (sasadm). This user is listed in `adminUsers.txt`, and the user's ID is preceded by an asterisk. The configuration wizard also creates one trusted user: the SAS Trusted User (sastrust). You create this type of user by adding a user ID to the file `trustedUsers.txt`. (For more detailed information about how to create these special users, see "Configure New Privileged Users of the SAS Metadata Server" on page 30.)

You do not normally need to edit the `trustedPeers.xml` file. By default, the metadata server trusts connecting workspace and stored process servers as peers. That is, these clients do not have to supply credentials when they connect to the metadata server. For information about the role of the `trustedPeers.xml` file and how to edit it, see the *SAS Integration Technologies: Server Administrator's Guide* at

http://support.sas.com/rnd/itech/doc9/admin_oma/security/auth/security_imptrust.html.

Configuration Files

The configuration directory also contains a set of configuration files that affect the behavior of the servers in the system. For example, in the directory **SASMain** (or **SASApp**), you will find the following files:

- **sasv9.cfg**
- **omaconfig.xml**

The file **sasv9.cfg** contains SAS system options that are shared by all of the SAS servers defined in the SAS application server. The file **omaconfig.xml** is one of several configuration files that affect the operation of the metadata server.

In addition, most of the subdirectories in **SASMain** that correspond to SAS servers contain a configuration file that affects a particular type of server. For instance, the **MetadataServer** directory contains a file called **sasv9_MetadataServer.cfg**, and the **StoredProcessServer** directory contains a file called **sasv9_StoredProcSrv.cfg**. These files typically supplement the configuration file **sasv9.cfg**.

For now, just be aware that these files exist and, perhaps, take a look at their contents. You will find information about when and how to edit these files throughout the SAS Intelligence Platform document set.

Server Start-Up Scripts and Logs

Each server or spawner that you configure on a machine is represented by a directory inside the **SASMain** directory. For example, you might see a **MetadataServer** folder and an **ObjectSpawner** folder. On UNIX and Windows systems, each such directory for a server that you can start directly contains a script called *server-type*.*extension* that takes a parameter **start**. On UNIX systems, you call these scripts directly to start servers and spawners. On Windows systems, you can call the scripts directly, or you can use the Start menu, for example, **Start ▶ Programs ▶ SAS ▶ *SAS-config-dir* ▶ Start SAS Object Spawner**.

Note: On z/OS systems, the servers run as started tasks, so you start them using a console command of the form:

```
START started-task-name
```

△

Notice also that each server directory contains a **logs** directory. This directory holds log files for a particular server and is the first place you should look for information if problems arise with a server.

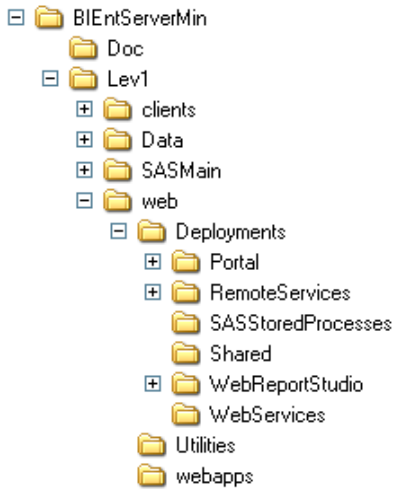
Metadata Repository

In addition to a **logs** directory and a start-up script, a **MetadataServer** directory contains a **MetadataRepositories** directory. It is important to know that this is the location where your foundation repository resides. This is also the location where the administrator for SAS Data Integration Studio creates project repositories for data-integration specialists, as explained in “Setting Up Change Management” on page 248.

Configuration Directory: Middle-Tier Machines

The SAS Configuration Wizard also creates a special directory structure in a configuration directory on middle-tier machines. See the following display.

Display 2.2 Configuration Directory (Middle Tier)



In this structure, the main files of interest are located in the **`web`** directory.

Servlet Container Start-Up Script

If you installed Tomcat as your servlet container, one of the files in the **`web`** directory is a script named **`startServletContainer.`***extension*, which starts Tomcat. On UNIX systems, you call this script directly to start the server. On Windows machines, it is more common to use the Start menu: **Start ▶ Programs ▶ SAS ▶ *configuration-name* ▶ Start Tomcat**. On Windows systems, do not start Tomcat by selecting **Start ▶ Programs ▶ Apache Tomcat 4.1 ▶ Start Tomcat**. This selection path will start Tomcat, but not with the options that are required by the SAS Web applications.

If you are using a J2EE application server such as the BEA WebLogic Server or the IBM WebSphere Application Server, you should start your server from the server's administration console or by using a script supplied with the server.

Note: You must start the SAS Services Application before you start your servlet container or J2EE application server. △

The Deployments Directory

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

The webapps Directory

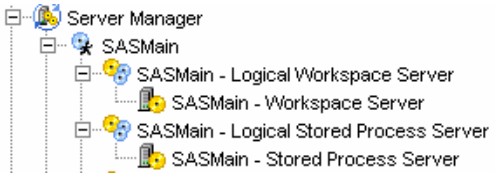
The **`webapps`** directory contains the Web application archive (WAR) files for Web applications such as the SAS Information Delivery Portal. These WAR files are actually JAR files that contain all of the files that make up the Web application, such as servlets, JavaServer pages, and HTML documents.

If you are using the Tomcat servlet container to execute your Web applications, the SAS Configuration Wizard has already copied these WAR files to Tomcat's **`webapps`**

directory. If you are using the BEA WebLogic Platform or the IBM WebSphere Application Server for this purpose, the installer at your site will have deployed your Web applications using the server's administration console.

Workspace and Stored Process Servers

When the installer at your site ran the SAS Configuration Wizard, that person defined metadata for a SAS application server.



This application server can contain components of a number of types. However, the most common components are a workspace server and a stored process server. These servers are also the servers with the most configuration options. This section explains how the SAS Configuration Wizard configures workspace and stored process servers. It also touches on the options that you have for reconfiguring these servers.

The wizard configures your initial workspace server to be a standard workspace server. When using such a server, each client must establish a connection to a single-user server process, use the server, and then disconnect. You can customize a workspace server (or servers) for better performance or for a specific use by using the following methods:

- pooling
- load balancing

When you configure a pooling workspace server, you enable clients to use a connection from a pool of previously created connections to workspace server processes. Using a pooled server is a good idea in cases where clients need to use a connection for a brief period of time, because it enables clients to avoid the overhead of opening connections and starting server processes. For information on how to convert a standard workspace server to a pooled server for use with SAS Web Report Studio, SAS Information Delivery Portal, or SAS Web OLAP Viewer for Java, see "Workspace Server Pooling for SAS Web Report Studio and SAS Information Delivery Portal" on page 384.

If you have created workspace servers on more than one host, you can balance a load across these servers by defining a load balanced logical workspace server. When you load balance a set of workspace servers, you create a cluster. The object spawners responsible for starting the workspace servers in a cluster take care of the load balancing and direct new traffic to the most available server. This type of configuration is most useful when you have a large number of workers (such as data integration specialists) using a workspace server for relatively long-running jobs. For information about how to create a cluster of load-balanced workspace servers, see "Load Balancing Workspace Servers for Desktop Applications" on page 400.

The SAS Configuration Wizard configures your initial stored process server to be load balanced. (With respect to load balancing, there is an important difference between workspace and stored process servers: whereas load-balanced workspace servers must run on different hosts, a workload can be balanced across multiple stored process server processes running on the same host. Each such process can handle requests from multiple clients.) By default, the object spawner balances a workload across three stored process server processes. If you later need to scale the system up, you can either

increase the number of stored process server processes on a machine or add a new host to your system and run an additional stored process server on that machine. For information on how to create a load-balanced cluster of stored process servers, see "Load Balancing Stored Process Servers on Multiple Hosts" on page 407.

Note: For detailed information on pooled and load-balanced servers, see "Pooling and Load Balancing" in the *SAS Integration Technologies: Server Administrator's Guide* at **`http://support.sas.com/rnd/itech/doc9/admin_oma/`**. △

Starting and Stopping Your SAS Servers

The installer of your system started some of your SAS servers when he or she performed the initial configuration of your system. However, as you administer your system, you will need to start, stop, and restart (stop and start) these servers. How you perform these tasks depends on the platform on which your servers are running and on how they were configured. See the following subsections:

- "Windows" on page 18
- "UNIX" on page 18
- "z/OS" on page 19

There are also a few rules about the order in which you must start your servers and service applications. For information on this subject, see "Starting Servers in the Correct Order" on page 19.

Windows

If your metadata server, object spawner, or OLAP server is running on a Windows machine and you have chosen to run your servers as services (highly recommended), your servers will start automatically when you restart your machine. You can then stop, start, or restart your servers in one of two ways.

The easiest way to perform these tasks is to use the **`Start`** menu. For example, you can restart your object spawner by selecting **Start ► Programs ► SAS ►** ***SAS-config-dir*** **► Restart SAS Object Spawner**. The menu on which the **`Restart SAS Object Spawner`** entry appears will also contain the entries **`Start SAS Object Spawner`** and **`Stop SAS Object Spawner`**. In addition, the menu will contain analogous entries for your metadata server and OLAP server, if appropriate.

You can also perform these operations by using scripts that the SAS Configuration Wizard has created in your configuration directory. For example, in the directory *SAS-config-dir***`\Lev1\SASMain\MetadataServer`**, there will be a script named **`MetadataServer.bat`**. If you execute this script by using the appropriate argument—**`start`**, **`stop`**, or **`restart`**—you can start, stop, or restart the metadata server service. The other server directories contain comparable scripts.

If your servers are running on a Windows machine and you have chosen not to run your servers as services, you can still control the servers by using either the **`Start`** menu or the scripts that were created in your configuration directory. Make sure that you are logged on as a member of the **`Administrators`** group. The only difference you will see is that your servers are not started automatically when you restart your machine.

UNIX

On a UNIX system, you start, stop, and restart servers by following these steps:

1. Log on as the SAS User (recommended user name sas).
2. Change directories to *SAS-config-dir*/**Lev1/SASMain/***server-type*.
3. Execute the *server-type*.**sh** script in that directory. This script takes one of three parameters: **start**, **stop**, and **restart**. To stop the metadata server, you would use the command **MetadataServer.sh stop**.

It is also possible to configure your system so that certain servers or spawners run as daemons. For example, to make the metadata server run as a daemon, you can copy the **MetadataServer.sh** script to the boot directory of your platform and add the needed information and links to the host's start and stop commands so that the metadata server is started at boot time and stopped at shutdown time. See your UNIX system administrator or the system administration guide for your platform for more information.

z/OS

On a z/OS system, you can start or stop a server by following these steps:

1. Log on as the SAS User (recommended user name sas).
2. Start or stop the server by using a console command of the form

   ```
   START started-task-name
   ```

   or

   ```
   STOP started-task-name
   ```

   Each server is associated with a different started task.

Note: You can perform the equivalent of a restart by stopping and then starting a server. △

Starting Servers in the Correct Order

There are a few dependencies among the servers that affect the order in which you should start your SAS servers and object spawners, the SAS Services Application, and third-party servers. Follow the guidelines below:

1. Start the metadata server before starting other SAS servers and spawners.
2. Start the metadata server before starting the SAS Services Application.
3. Start the SAS Services Application and your WebDAV server before starting your servlet container or J2EE application server.

Accessibility

For Window-based accessibility technologies to interoperate with SAS Java-based desktop applications, you need to download the Java Access Bridge (JAB) from Sun Microsystems at this url: **http://www.sas.com/govedu/java.html**. The SAS Java-based applications that run on Windows are listed under "Clients" in the "Architecture of the SAS Intelligence Platform" chapter of the *SAS Intelligence Platform: Overview*.

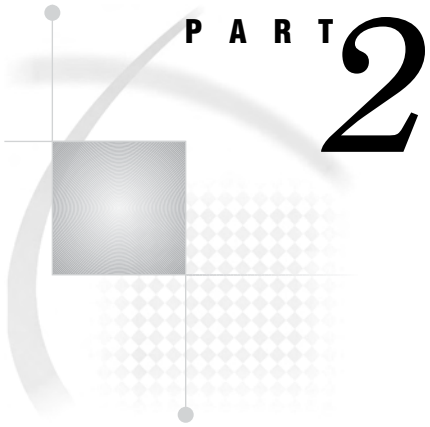
After downloading the JAB, you need to customize the standard installation procedure to facilitate access to it. This setup assumes that Java is already installed on the machine. A known risk is that SAS applications can get polluted by JAR files that

might be installed in the **jre/lib/ext** directory. To solve this problem, SAS has implemented a custom class loader to replace the system class loader. The custom class loader loads only approved extensions that SAS configures based on the JRE the vendor provides. A consequence of hiding the existing directory is that it disables installed accessibility-related applications. To configure SAS Java applications to permit access to the Java Access Bridge:

1. On the client machine, locate all copies of the file **sas.java.ext.config**.
2. In each instance of the file, edit the file in a plain text editor, such as Microsoft Notepad.
3. Add the following two lines to the bottom of the file:

   ```
   access-bridge.jar
   jaccess-1_4.jar
   ```

   Note: The filename of the **jaccess*.jar** file depends on the version of the JRE. Listed here is the filename for Version 1.4. If you are using a different version, your filename will vary.
4. Save the changes and close the file.



PART 2

Metadata Administration

CHAPTER

3

Understanding and Configuring the SAS Metadata Server

Overview of the SAS Metadata Server and Its Initial Configuration

What This Chapter Covers

By now, you are probably somewhat familiar with the SAS Metadata Server. It was introduced in the *SAS Intelligence Platform: Overview*. In addition, if you installed your own SAS software, you have already configured a metadata server and used it to write metadata about users, SAS servers, and other items to your foundation metadata repository.

This chapter presents more detailed information about the metadata server. It then takes a detailed look at how your platform's metadata server has been configured and discusses your options for modifying the default configuration. You should read this section carefully because there are a couple of configuration tasks that you might need to perform to optimize the performance of your metadata server. You might want to change other aspects of the configuration as well, depending on the requirements at your site.

Note: Subsequent chapters will look at the ongoing maintenance of your metadata repositories and metadata server. △

What the Metadata Server Does

The SAS Metadata Server is a multi-user server that serves metadata from one or more SAS Metadata Repositories to SAS Intelligence Platform client applications via the SAS Open Metadata Architecture. The SAS Open Metadata Architecture is a general-purpose metadata management facility that provides common metadata services to applications. The SAS Open Metadata Architecture enables the following:

- exchange of metadata between separate applications, which makes it easier for these applications to work together. This capability also saves development effort because the applications no longer have to implement their own metadata facilities.
- centralized management of your metadata resources. Because there is a common framework for creating, accessing, and updating metadata, it is easier to manage the applications that rely on this metadata.

The goal of the SAS Open Metadata Architecture is to enable centralized, enterprise-wide metadata management: one metadata server provides metadata services to all applications in the enterprise in an environment that supports multiple repositories and hundreds of concurrent users.

Multiple SAS Metadata Servers can be configured in support of a software development life cycle. For example, you can configure a development metadata server, a test metadata server, and a production metadata server. However, the SAS Open Metadata Architecture does not support sharing of metadata between SAS Metadata Servers, but supports only replication services.

How the Metadata Server Works

The SAS Metadata Server is a SAS process that enables clients to read metadata from and write metadata to one or more SAS metadata repositories. The repositories on the metadata server are managed by a repository manager.

The *repository manager* is a library of tables that define the set of repositories that can be accessed by a SAS Metadata Server. A metadata server cannot be started without a repository manager and can have only one repository manager.

A *repository* is a physical location where a collection of related metadata objects, such as the metadata for a set of tables and columns maintained by an application, is stored.

To use the SAS Metadata Server, SAS Intelligence Platform applications connect to the SAS Metadata Server and issue SAS Open Metadata Interface method calls that access SAS metadata repositories.

The Initial Configuration of the Metadata Server

This section summarizes the state of the metadata server immediately after the installation of the platform—that is, at the point where you have finished running the SAS Configuration Wizard. It also refers you to instructions for changing all aspects of this configuration.

The maximum number of threads that the metadata server can use has been set to a value representing the number of processors on the metadata server host machine.

This value will not be the appropriate setting for all metadata server host machines. For information about how to configure the number of threads that the metadata server can use, see "Controlling the Number of Threads Used by the Server" on page 26.

Workunit journaling has not been enabled.

Workunit journaling is a feature that improves the performance of the metadata server on multi-processor hosts. When this feature is enabled, updates to the metadata requested by client applications are recorded in a workunit journal file immediately—enabling clients to continue their work—and written to disk later. For more information about how workunit journaling works and how to configure it, see "Turning on Workunit Journaling" on page 27.

Two special users of the metadata have been set up: the SAS Administrator and the SAS Trusted User.

You may want to configure additional special users, especially *administrative* and *unrestricted users*. For more information about the roles and privileges of these users and about how to configure additional special users, see "Defining Administrative, Unrestricted, and Trusted Users" on page 29.

The metadata server has been configured to write a daily log to the directory *SAS-config-dir*`\Lev1\SASMain\MetadataServer\logs` and to use a particular logging level.

For information about how to change the default logging behavior, see "Logging Events and Errors" on page 31.

The metadata server has been configured to authenticate users with the host operating system's authentication mechanism.

For information about how to use a different authentication provider, see "Using LDAP or Active Directory" in the chapter "Customizing the Authentication Configuration" in the *SAS Intelligence Platform: Security Administration Guide*.

The metadata server has been configured to use SAS proprietary encryption to encrypt user credentials that are sent over the network.

To encrypt more data or to use a different encryption algorithm, see "Enable Encryption" in the chapter "Securing a Deployment" in the *SAS Intelligence Platform: Security Administration Guide*.

The metadata server has been configured to use SAS data sets for the repository manager and for the foundation metadata repository.

We strongly recommend that you do not change this aspect of the configuration for performance reasons. However, we recognize that it is possible that your company may require you to store the repository manager tables and repository tables in a

third-party relational database. For information about what databases are supported and information about how to reconfigure this aspect of your system, see "Creating the Metadata Repository in a Database Management System" on page 36.

Note: If you are going to make this change, you should make it as soon as possible after installation to reduce the amount of metadata that you need to re-create. △

Controlling the Number of Threads Used by the Server

How the Threading Options Are Set by Default

The way in which the metadata server uses threads is determined by the values of the following options:

- The **MAXACTIVETHREADS** server configuration option specifies the maximum number of metadata server threads that are allowed to run concurrently on the metadata server. The number of processors available on the server host machine determines the number of queries that can be active at any time.
- The OBJECTSERVERPARMS options THREADSMIN and THREADSMAX, which are specified in the metadata server invocation command, determine the number of threads that are maintained in the server's thread pool. The thread pool defines the total number of threads that are available to a server, regardless of the number of user requests that are received or accepted.

By default, **MAXACTIVETHREADS** is set to *number-of-processors*, and THREADSMIN and THREADSMAX are set to (*number-of-processors* * 2) + 1.

The settings of THREADSMIN and THREADSMAX optimize performance by ensuring that the number of available threads is compatible with the number of threads that can be accepted concurrently by the metadata server. Setting THREADSMIN and THREADSMAX to the same value prevents the metadata server from creating and destroying threads that might not be used. A THREADSMAX setting that is slightly higher than the number of processors on the metadata server ensures that an active thread is always available to the server. If clients experience significant delays in response from the metadata server and not all of the processors are used at 100 percent capacity, you can increase the THREADSMIN and THREADSMAX settings to try to get better utilization.

The default settings are the recommended starting point for the case in which you want to optimize the performance of the metadata server—perhaps at the expense of other processes running on the metadata server host machine. If you want to limit the number of threads that the metadata server will use, you can change these settings by following the directions in the next section.

Change the Default Number of Threads

To change the settings of **MAXACTIVETHREADS**, THREADSMIN, and THREADSMAX, perform these steps:

1. Open the file **omaconfig.xml** in a text editor. This file is located in the directory *SAS-config-dir***Lev1\SASMain**.
2. Edit the value of the **MAXACTIVETHREADS** attribute of the **<OMA>** element. Change the value of the attribute from *number-of-processors* to a lower value.

3 Save your changes to **omaconfig.xml**.
4 Open the file **sasv9_MetadataServer.cfg** in a text editor. This file is located in the directory *SAS-config-dir***\Lev1\SASMain\MetadataServer**.
5 Change the values of the THREADSMIN and THREADSMAX options in the string that serves as the argument to the OBJECTSERVERPARMS option, as shown below:

```
THREADSMIN=(MAXACTIVETHREADS*2)+1 THREADSMAX=(MAXACTIVETHREADS*2)+1
```

6 Save your changes to **sasv9_MetadataServer.cfg**.
7 Restart the metadata server.

Turning on Workunit Journaling

What Is Workunit Journaling?

The metadata server is an in-memory server. As such, it maintains two copies of the metadata:

- It holds one copy of the metadata in main memory to enable high-speed access by clients.
- The other copy is the permanent copy that the metadata server keeps in metadata repositories on disk.

The default behavior of the metadata server is to update both copies of the metadata before informing clients that a request has been successfully completed. This ensures that both copies contain the same information. In addition, the requesting client receives messages both when repository updates are successful and when they fail.

Beginning in SAS 9.1.3, customers can optionally choose to post updates to the memory copy of repositories immediately, and to queue disk updates to a high-speed workunit journal file until the CPU is available to apply the updates to disk. Workunit journaling is disabled by default. When workunit journaling is enabled, the following actions occur:

- Updated records are written to two additional areas: the workunit journal file and a workunit buffer.
- Metadata that is held in memory is updated when the workunit journal file and the workunit buffer are updated.
- A separate process is started to update repositories on disk from data that is held in the workunit buffer.
- Clients receive a message that their requests have been completed when the workunit journal file and the memory copy of the repositories are updated.

Because the workunit journal file can be updated more quickly than repositories, updated metadata becomes available to clients sooner. If the metadata server fails before the update process has had a chance to apply all updates from the workunit journal buffer to repositories, the metadata server automatically recovers them from the workunit journal file when it is re-started. The server does this before making the memory copy of the metadata available, and before it accepts any new requests. The metadata server does this by comparing all updates that are marked as pending in the workunit journal file to repositories to ensure that all records were applied and that they were applied completely.

The progress of the repository update process is tracked in the metadata server log. If an error occurs when the update process is writing to repositories, an alert e-mail

message is sent to the metadata server administrator so that the administrator can respond to the problem.

Note: The performance benefits of workunit journaling come from separating the memory and disk update processes into separate threads that can be run on separate processors; therefore, workunit journaling is a performance optimization only for metadata servers running on multi-processor hosts. On a single-processor host, the disk update process will compete with new client requests for CPU cycles, thereby nullifying any performance benefits. △

Configure Workunit Journaling

To enable workunit journaling, you must add several attributes to the **<OMA>** element in the XML configuration file **omaconfig.xml**. You must also set several metadata server invocation parameters in the file **sasv9_MetadataServer.cfg**. Perform the following steps to configure workunit journaling:

1. Open the file **omaconfig.xml**, which is located in the directory *SAS-config-dir***\Lev1\SASMain**.
2. Add the following attributes to the **<OMA>** element:

   WORKUNITJOURNALPATH="*filename***"**
   The **WORKUNITJOURNALPATH** attribute activates workunit journaling and specifies the name of the workunit journal file. For ease of maintenance, use a name such as **WorkunitJournal.dat**. If you specify a simple filename, this file will be created in the directory *SAS-config-dir***\Lev1\SASMain\MetadataServer**. You can also specify a relative pathname; the pathname should be relative to the **MetadataServer** directory.

   WORKUNITJOURNALSIZE="*size-in-bytes***"**
   The **WORKUNITJOURNALSIZE** attribute specifies the size of the workunit journal file in bytes. We recommend that you start with a file size of 20MB (20000000). This space is reserved in the file system and is unavailable to other processes.

   ALERTEMAIL="*email-address* | (*email-address-1*, *email-address-2* ...)**"**
   The **ALERTEMAIL** attribute specifies the e-mail address to which the metadata server should send mail if an error occurs while it is updating the workunit journal file. The address that you specify will appear in the **To** field of the e-mail message. To specify more than one address (for example, work and home e-mail addresses) enclose the addresses in parentheses and separate them with a comma.

   ALERTEMAILATTACHMENT="*filename***"**
   The **ALERTEMAILATTACHMENT** attribute specifies the name of a file—such as **omaconfig.xml**—that you would like attached to any alert e-mail sent.

   Once you have made these additions, your **<OMA>** element might look like this:

```
<OMA ADMINUSERS="MetadataServer/adminUsers.txt"
     TRUSTEDUSERS="MetadataServer/trustedUsers.txt"
     GROUPSFORMEMBERASSOCDELETE="DESKTOP_PORTALPAGES_GROUP,
        Portlet Template Group,
        OrderingGroup,
        DESKTOP_PAGEHISTORY_GROUP,
        Portal Collection"
     WORKUNITJOURNALPATH="WorkunitJournal.dat"
     WORKUNITJOURNALSIZE="20000000"
     ALERTEMAIL="sastest@company.com"
```

```
ALERTEMAILATTACHMENT="omaconfig.xml"/>
```

There are two additional attributes that you might need to add to the **<OMA>** element: **WORKUNITJOURNALYIELDCOUNT** and **ALERTEMAILTEST**. The first specifies how many updates the metadata server can make to the workunit journal file before it must yield to another internal task. (You do not usually change the value of this option.) The second is a temporary option that enables you to determine whether the opitons previously discussed for the alert e-mail have valid values. For further information on this subject, see "The omaconfig.xml File: Reference" on page 42. This section also includes more in-depth discussions of the options shown in the preceding code snippet.

3 Save the file **omaconfig.xml**.

4 Open the file **sasv9_MetadataServer.cfg** in the directory *SAS-config-dir***\Lev1\SASMain\MetadataServer**.

5 Add the following invocation parameters to the configuration file. The metadata server needs this information in order to send the alert e-mails mentioned previously.

-emailsys SMTP
The SAS Metadata Server uses the SMTP (Simple Mail Transfer Protocol) e-mail interface to send alert e-mail messages to the metadata server administrator.

-emailhost *mail-server*
Specify the full name of your mail-server host.

-emailid *account-of-sender*
Specify the e-mail address that you want to appear in the **From** field of an alert e-mail. The EMAILID option supports e-mail addresses in two forms: "*server-name<user-ID@domain>*" and "*<user-ID@domain>*". If you have multiple metadata servers (for example, development, test, and production servers), you can include a server name to indicate which server is sending the e-mail.

The lines that you add to the configuration file might look like this:

```
-emailsys SMTP
-emailhost mailhost.company.com
-email "Metadata Server 1<johnd@company1.com>"
```

6 Save your changes to **sasv9_MetadataServer.cfg**.

7 Restart the metadata server.

Defining Administrative, Unrestricted, and Trusted Users

Rights Granted to Administrative, Unrestricted, and Trusted Users

It is possible to define three different types of privileged users of the metadata server: *administrative users*, *unrestricted users*, and *trusted users*. A user who has *administrative user* status on the metadata server can perform the following tasks that other users cannot perform:

- create repositories, delete repositories, and change a repository's registration—for example, to enable or disable auditing or to add or remove a dependency on another repository
- create user definitions and their initial logins
- create user-defined permissions

- pause the SAS Metadata Server or temporarily change the access state of one or more SAS Metadata Repositories or the repository manager—for example, from full access to read-only and back to full access again
- refresh individual repositories or all repositories on the server to recover memory
- enable or disable Applications Response Management (ARM) logging
- stop the SAS Metadata Server

A user who has *unrestricted user* status can perform all of the tasks that an *administrative user* can perform. In addition, the *unrestricted user* has the following privileges:

- unrestricted access to metadata describing application resources (except passwords, which the *unrestricted user* can overwrite but cannot read).
- the ability to modify and delete permission metadata.
- the ability to modify and delete identity, login, and access control metadata. (An *administrative user* can create identity and login metadata, but cannot modify or delete it unless he is specifically assigned ReadMetadata and WriteMetadata permission to the identity.)

A *trusted user* can perform the following tasks that are not allowed for "non-trusted" user connections:

- acquire or release credential handles on behalf of other users
- use the credential handles to make metadata and authorization requests as another user
- generate one-time-use passwords to allow clients to connect to the metadata server directly

No one logs in as a *trusted user* and makes requests on behalf of other users. Instead, the account is used by certain platform servers when they need to make metadata requests on behalf of their users.

Privileged Users Configured by the SAS Configuration Wizard

The SAS Configuration Wizard configures one *unrestricted user* and one *trusted user*. It does not configure any *administrative users*. The SAS Administrator, whose recommended user ID is sasadm, is an *unrestricted user*. The SAS Trusted User, whose recommended user ID is sastrust, is a *trusted user*.

Configure New Privileged Users of the SAS Metadata Server

You can define new privileged users by editing the files **`adminUsers.txt`** and **`trustedUsers.txt`**. The most likely case is that you will want to create one or more *administrative users* for the reasons discussed in "Create a Group of Metadata Administrators" in the chapter "Securing a Deployment" in the *SAS Intelligence Platform: Security Administration Guide*. However, the instructions below explain how to define all three types of special users.

To define an *administrative* or *unrestricted user*, perform the following steps:

1. Open the file **`adminUsers.txt`**, which is located in the directory *SAS-config-dir*`\Lev1\SASMain\MetadataServer`.
2. This file contains an entry for the SAS Administrator (usually sasadm). Add analogous entries for other users whom you want to make *administrative* or *unrestricted users*. If you do not precede a user name with an asterisk (*), the user

will be an *administrative user*. If you do precede the name with an asterisk, the user will be an *unrestricted user*.

On Windows systems, you must qualify domain user IDs with a domain name and local user IDs with a hostname. For example, the entry for a local user might be:

```
D1234\adminUser
```

3 Save your changes to the file.

4 Restart the metadata server.

To define a *trusted user*, perform the following steps:

1 Open the file **trustedUsers.txt** in the directory *SAS-config-dir***\Lev1\SASMain\MetadataServer**.

2 This file contains an entry for the SAS Trusted User (usually sastrust). Add analogous entries for other users whom you want to make *trusted users*. On Windows systems, you must qualify domain user IDs with a domain name and local user IDs with a hostname. For example, the entry for a local user might be:

```
D1234\trustedUser
```

3 Save your changes to the file.

4 Restart the metadata server.

Logging Events and Errors

The SAS Metadata Server creates log entries that record server events, errors, and traces. These entries are directed to the destination specified by the LOG parameter on the SAS Metadata Server start command, and are useful to anyone needing to know the status of the SAS Metadata Server and whether a given request was successful. The log entries also contain information that is useful for diagnosing server startup problems and authorization failures, and that assists in method call debugging.

The SAS Metadata Server can write seven categories of information to the SAS log, depending on the SAS Metadata Server invocation options that you select. These categories include:

1 SAS Metadata Server start and stop information, including the invoking user ID, SAS long version number, SAS Metadata Model version, the directory where the metadata server was started, and configuration file options

2 user connection and disconnection events

3 repository creation/deletion, open/close, pause/resume, and refresh events

4 errors, including task/thread exceptions, memory allocation errors, I/O errors, application logic errors, authentication errors, and authorization failures

5 authentication events

6 XML input and output strings

7 traces invoked by **omaconfig.xml** debugging options

The Default Logging Configuration

The SAS Configuration Wizard enables the first four levels of logging discussed in "Logging Events and Errors" on page 31 by adding the following invocation options to the file *SAS-config-dir***\Lev1\SASMain\MetadataServer\sasv9_MetadataServer.cfg**:

```
-log "SAS-config-dir\Lev1\SASMain\MetadataServer\logs\MetadataServer#d#b#y.log"
-logparm "rollover=auto open=replaceold write=immediate"
-pagesize max
-linesize max
```

Table 3.1 on page 32 explains what each option does.

Table 3.1 Log File Related Options in sasv9_MetadataServer.cfg

Option	Explanation
LOG	Enables the first four levels of logging and specifies the path to the log file. In the default case, the metadata server will create daily logs with names like **MetadataServer01Mar2006**.
LOGPARM	Specifes that information should be written to the log file immediately (instead of being buffered) and that a new log file should be opened each day. (The frequency with which a new log file is created depends on the log file name.)
PAGESIZE	Sets the number of lines per page to 32,767, the maximum number.
LINESIZE	Sets the number of characters per line to 256, the maximum.

For more information about these options, see the *SAS Language Reference: Dictionary*. To change the values of the options, stop the metadata server, make the desired changes to the configuration file, and then restart the server.

For information about adding logging capabilities beyond those configured by default, see the following topics:

- "Enable the Logging of Authentication Events" on page 32
- "Capturing XML in the Log" on page 33
- "Debugging Options" on page 46

Enable the Logging of Authentication Events

An authentication event is a response to a request for a credential handle. This handle is a token representing a user's authorizations on the SAS Metadata Server that can be stored on an interim server to reduce the number of authorization requests that are made to the server on behalf of a given user. The SAS Metadata Server and SAS Metadata Server clients call GetCredentials and FreeCredentials methods to obtain and release credentials on behalf of users.

You turn on the logging of authentication events by specifying APPLEVEL=2 or APPLEVEL=3 as an OBJECTSERVERPARMS option in the SAS Metadata Server start command.

APPLEVEL=2 enables IOM server logging with the default log entries and additionally logs authentication requests by clients.

APPLEVEL=3 enables IOM server logging with all of the information provided at APPLEVEL=2 and additionally logs requests by unregistered user IDs that are mapped as PUBLIC. You can use information about user IDs that are mapped as PUBLIC to identify candidates for registration.

The log entries for GetCredentials and FreeCredentials requests (APPLEVEL=2) look like this:

```
2003063:17.10.15.12: 00000032:  4:GOODSERVER@DOMAIN: Credential obtained for
  SMTTST1@DOMAIN.
2003063:17.11.13.14: 00000093:  4:GOODSERVER@DOMAIN: Credential freed for
  SMTTST1@DOMAIN.
2003063:17.50.16.48: 00000032:  4:ROGUESERVER@DOMAIN: Not trusted to obtain
  Credential for SMTTST4@DOMAIN.
2003063:18.01.10.41: 00000093:  4:ROGUESERVER@DOMAIN: Not trusted to free
  Credential for SMTTST2@DOMAIN.
```

Note: Omitting the APPLEVEL option from the SAS Metadata Server start command has no effect on server logging (authentication events will not be logged, but logging invoked by other log parameters will continue uninterrupted). The default value of APPLEVEL, if omitted, is APPLEVEL=1. The only other supported option (APPLEVEL=0) disables server logging. *Do not specify APPLEVEL=0.* △

To configure the logging of authentication events, perform these steps.

1. Open the file *SAS-config-dir***\Lev1\SASMain\MetadataServer\sasv9_MetadataServer.cfg**. This file contains a line similar to this:

   ```
   -objectserverparms "cel=credentials protocol=bridge port=8561
     classfactory=2887E7D7-4780-11D4-879F-00C04F38F0DB
     trustsaspeer='MetadataServer\trustedPeers.xml'"
   ```

2. Add either **applevel=2** or **applevel=3** to the list of object server parameters (that is, to the string following **-objectserverparms**).
3. Save your changes to **sasv9_MetadataServer.cfg**.
4. Restart the metadata server.

Capturing XML in the Log

When clients of the SAS Metadata Server need to create, update, delete, or query a metadata object, they make their requests using the SAS Open Metadata Interface. For example, to add a metadata object to a repository, a client uses the AddMetadata method. One of the parameters to this method is an XML description of the object to be added, such as the following description of a table:

```
"<PhysicalTable Name="NECust" Desc="All customers in the northeast region"/>";
```

In addition, the metadata server returns an XML string that echoes the input and includes the object ID of the new metadata object.

This section explains how to capture XML of this type in the metadata server log.

- For further information about what type of information will be captured in the log, see "The XML Representation of Metadata Objects" on page 33.
- For information about configuring the metadata server so that it will capture XML in its log files, see "Configure the Logging of XML" on page 35.

The XML Representation of Metadata Objects

As mentioned in "Capturing XML in the Log" on page 33, when clients communicate with the SAS Metadata Server, information about metadata objects is represented in XML. The format of the XML that is written to the log depends on whether the SAS Open Metadata Interface request was made via the standard interface or via the DoRequest method. For more information about the SAS Open Metadata Interface call interfaces, see the *SAS Open Metadata Interface: Reference*.

The following is an example of a log entry for a SAS Open Metadata Interface AddMetadata request that was issued via the DoRequest method:

```
2003043:15.07.35.21: 00000019:    IOM CALL {compRef:3cbbdd0}->OMI::DoRequest():
inString=
<multiple_requests>
  <AddMetadata>
    <Metadata>
      <RepositoryBase Name="scratch2" Desc="Repository 2" Path="scratch2">
        <DependencyUses>
           <RepositoryBase Name="scratch1" Desc="Repository 1" Path="scratch1"/>
        </DependencyUses>
        <DependencyUsedBy>
          <RepositoryBase Name="scratch3" Desc="Repository 3" Path="scratch3"/>
        </DependencyUsedBy>
      </RepositoryBase>
    </Metadata>
    <Reposid>A0000001.A0000001</Reposid>  <!-- Repository manager -->
    <Ns>REPOS</Ns>
    <Flags>268435456</Flags>  <!-- OMI_TRUSTED_CLIENT -->
    <Options/>
  </AddMetadata>
</multiple_requests>

2003043:15.07.35.38: 00000019: 2:SASTST@DOMAIN]: RegisterRepository
Id=A0000001.A5M9N7FJ, Name=scratch2, Path=scratch2, Engine=., Options=.,
AuditPath=..
2003043:15.07.35.56: 0000000019: 2:SASTST@DOMAIN]: RegisterRepository
Id=A0000001.A53X8NYG, Name=scratch1, Path=scratch1, Engine=., Options=.,
AuditPath=..
2003043:15.07.35.77: 00000019: 2:SASTST@DOMAIN]: RegisterRepository
Id=A0000001.A5IOGZ01, Name=scratch3, Path=scratch3, Engine=., Options=.,
AuditPath=..
2003043:15.07.35.90: 00000019:  IOM RETURN 0={compRef:3cbbdd0}->OMI::DoRequest():
outString=<multiple_requests><AddMetadata><Metadata><RepositoryBase
Name="scratch2" Desc="Repository 2" Path="scratch2" Id="A0000001.A5M9N7FJ"
Access="0" <RepositoryType=""><DependencyUses><RepositoryBase Name="scratch1"
Desc="Repository 1" Path="scratch1" Id="A0000001.A53X8NYG" Access="0"
RepositoryType=""/></DependencyUses><DependencyUsedBy><RepositoryBase
Name="scratch3" Desc="Repository 3" Path="scratch3" Id="A0000001.A5IOGZ01"
Access="0" RepositoryType=""/> </DependencyUsedBy></RepositoryBase></Metadata>
<Reposid>A0000001.A0000001</Reposid><Ns>REPOS</Ns><Flags>268435456</Flags>
<Options/></AddMetadata></multiple_requests>  retval=0
```

The following is an example of a log entry for the same request issued via the standard interface:

```
2003043:15.20.04.12: 00000019:    IOM CALL {compRef:3cbbdd0}->OMI::AddMetadata():
inMetadata=
   <RepositoryBase Name="scratch2" Desc="Repository 2" Path="scratch2">
      <DependencyUses>
        <RepositoryBase Name="scratch1" Desc="Repository 1" Path="scratch1"/>
      </DependencyUses>
      <DependencyUsedBy>
        <RepositoryBase Name="scratch3" Desc="Repository 3" Path="scratch3"/>
      </DependencyUsedBy>
```

```
    </RepositoryBase>
    reposid=A0000001.A0000001  ns=REPOS  flags=268435456
options=
2003043:15.20.04.25: 00000019: 2:SASTST@DOMAIN]: RegisterRepository
Id=A0000001.A55LJU3G, Name=scratch2, Path=scratch2, Engine=., Options=.,
AuditPath=..
2003043:15.20.04.37: 00000019: 2:SASTST@DOMAIN]: RegisterRepository
Id=A0000001.A5PE8JLK, Name=scratch1, Path=scratch1, Engine=., Options=.,
AuditPath=..
2003043:15.20.04.53: 00000019: 2:SASTST@DOMAIN]: RegisterRepository
Id=A0000001.A5ZJIVYZ, Name=scratch3, Path=scratch3, Engine=., Options=.,
AuditPath=..
2003043:15.20.04.62: 00000019: IOM RETURN 0={compRef:3cbbdd0}->OMI::AddMetadata():
outMetadata=<RepositoryBase Name="scratch2" Desc="Repository 2" Path="scratch2"
Id="A0000001.A55LJU3G" Access="0" RepositoryType=""><DependencyUses>
<RepositoryBaseName="scratch1" Desc="Repository 1" Path="scratch1"
Id="A0000001.A5PE8JLK" Access="0" RepositoryType=""/></DependencyUses>
<DependencyUsedBy><RepositoryBase Name="scratch3" Desc="Repository 3"
Path="scratch3" Id="A0000001.A5ZJIVYZ" Access="0" RepositoryType=""/>
</DependencyUsedBy></RepositoryBase>  retval=0
```

Configure the Logging of XML

To configure the logging of XML, perform these steps:

1. Open the file
   *SAS-config-dir***Lev1\SASMain\MetadataServer\sasv9_MetadataServer.cfg**.
   This file will contain a line similar to this one:

   ```
   -objectserverparms "cel=credentials protocol=bridge port=8561
     classfactory=2887E7D7-4780-11D4-879F-00C04F38F0DB
     trustsaspeer='MetadataServer\trustedPeers.xml'"
   ```

2. Add the following object server parameters to the existing list of parameters (that is, add them to the string following **-objectserverparms**):

   ```
   iomlevel=1 jnlStrMax=10000 jnlLineMax=10000
   ```

   Your values for the second and third parameters may vary from what is shown above.

   Table 3.2 on page 36 explains what these parameters mean:

Table 3.2 Parameters Required to Capture XML in the Log File

Parameter	Explanation
IOMLEVEL	Traces protocol independent logic, including calls. A value of 0 (the default if IOMLEVEL is omitted) disables the capturing of input and output XML strings. A value of 1 enables the capturing of input and output XML strings.
JNLSTRMAX	Specifies the maximum journal string length. The default is 500 characters. If this parameter is omitted, a maximum of 500 characters of the XML string is logged, and the following message is appended to the XML: **`*truncated to 500 characters (jnlStrMax)`**.
JNLLINEMAX	Specifies the maximum journal line length. The default is 1000 characters. If this parameter is omitted, a maximum of 1000 characters of the XML string is logged, and the following message is appended to the XML: **`*truncated to 1000 characters (jnlLineMax)`**.

Creating the Metadata Repository in a Database Management System

It is possible to modify your system so that your repository manager and metadata repositories are stored in a database management system from another vendor. Two such DBMSs are supported: Oracle and DB2. However, there are some restrictions on their use:

- The Oracle or DB2 software must be installed on a Windows or UNIX system.
- Both the metadata server and the database server must be running on the same operating system.

Prerequisites for Setting Up a DBMS Repository

Before you can create your repository manager and metadata repository in an Oracle or DB2 database, you must meet a few prequisites:

- Obviously, you must have an Oracle or DB2 server that you can access from your metadata server host. A database administrator must have created at least two schemas for you: one for the repository manager tables and one for the foundation repository tables. We recommend that you name the schema for the repository manager SASRMGR. If you plan to create additional repositories (custom or project repositories), you will also need a schema for each such repository. The database administrator must also have supplied you with the information that you will need to connect to the database and access these schemas.
- You must have installed the SAS/ACCESS Interface to Oracle or the SAS/ACCESS Interface to DB2 on your metadata server host machine. If your database server is running on a UNIX machine, you must also have configured the SAS/ACCESS product as explained in the *Configuration Guide for SAS Foundation for UNIX Environments*. This document is included in your SAS Installation Kit.
- You must also have installed the database client software on your metadata server host.

Create the Repository Manager and a Metadata Repository in an Oracle or a DB2 Database

Once you have met the prerequisites explained above, you can create your repository manager and metadata repository by performing these steps:

1. Stop the metadata server.
2. Edit the configuration file *SAS-config-dir***Lev1\SASMain\omaconfig.xml** so that the metadata server will have the information that it needs to connect to the database server and to create the repository manager tables there. By default, the SAS Configuration Wizard writes the following content to this file:

```
<?xml version="1.0" encoding="UTF-8"?>
<OMAconfig>
  <OMA ADMINUSERS="MetadataServer/adminUsers.txt"
  <GROUPSFORMEMBERASSOCDELETE="DESKTOP_PORTALPAGES_GROUP,
      Portlet Template Group,
      OrderingGroup,
      DESKTOP_PAGEHISTORY_GROUP,
      Portal Collection"/>
  <RPOSMGR PATH="MetadataServer/rposmgr"/>
</OMAconfig>
```

If you plan to create your repository manager in an Oracle database, you should edit the **<RPOSMGR>** element in this file so that it contains the attributes shown below:

```
<RPOSMGR LIBREF="RPOSMGR" PATH="" ENGINE="ORACLE"
    OPTIONS="reread_exposure=yes preserve_names=yes
    dbchar_constant_is_spoofed=yes connection=shared readbuff=1
    insertbuff=1 spool=no utilconn_transient=no [dbmax_text=4096]
    PATH=path USER=user PASSWORD=password [SCHEMA=schema]"/>
```

Note: The option **dbmax_text=4096** is optional. Use it only on 64-bit platforms. Also, you can omit the **schema** option if you want to create the repository manager tables in a schema whose name matches the name of the user specified in the options. Generally, however, the repository manager tables are created in a schema called SASRMGR. △

There are four values that you may need to supply in the XML element:

path	an alias (service name) that represents information about the protocol to be used in communicating with the database, the name of the server hosting the database server, and the port on which the server is listening. You can define this alias on the metadata server host by using an Oracle configuration tool, or you can have your database administrator create it.
user	the user ID that you will use to connect to the database server.
password	the corresponding password.
schema	the name of the schema in which you plan to create the repository manager tables.

For further information about the attributes that appear in the **<RPOSMGR>** element when you connect to an Oracle database, see "<RPOSMGR> Attributes Used to Connect to an Oracle Database" on page 39.

If you plan to create your repository manager and metadata repository in a DB2 database, you should edit the **<RPOSMGR>** element in this file so that it contains the attributes shown below:

```
<RPOSMGR LIBREF="RPOSMGR" PATH="" ENGINE="DB2"
    OPTIONS="reread_exposure=yes preserve_names=yes
    dbchar_constant_is_spoofed=yes connection=shared readbuff=0
    insertbuff=1 spool=no utilconn_transient=no [dbmax_text=4096]
    DB=database USER=user PASSWORD=password [SCHEMA=schema]"/>
```

Note: The option **dbmax_text=4096** is optional. Use it only on 64-bit platforms. Also, you can omit the **SCHEMA** option if you want to create the repository manager tables in a schema whose name matches the name of the user specified in the options. Generally, however, the repository manager tables are created in a schema called SASRMGR. △

There are four values that you may need to supply in this XML element:

database	the name of the DB2 data source or database.
user	the user ID that you will use to connect to the database server.
password	the corresponding password.
schema	the name of the schema in which you plan to create the repository manager tables.

For further information about the attributes that appear in the **<RPOSMGR>** element when you connect to a DB2 database, see "<RPOSMGR> Attributes Used to Connect to a DB2 Database" on page 40.

3 Save your changes to **omaconfig.xml**.

4 Restart your metadata server. The metadata server will create a set of repository manager tables in the schema that you specified in your **omaconfig.xml** file.

Note: If you check you metadata server log, you may see the error message "ERROR: Invalid option name DBCONDITION." You can ignore this message. △

At this point, you have created the repository manager. The following steps create the initial (foundation) metadata repository.

5 Start SAS Management Console, and connect to the metadata server as the SAS Administrator (sasadm).

6 You see a dialog that displays the message "No Repositories are present on the requested server. Would you like to add a repository now?" Click **Yes**. A wizard that enable you to define a new repository starts.

7 On the Select Repository Type page, select **Foundation** and click **Next**.

8 On the General Information page, leave **Foundation** in the **Name** field, and click **Next**.

9 On the Definition of Data Source page, perform the following steps:

   a Select **Oracle** or **DB2** from the **Engine** drop-down list. (This will cause the **Path** field to be dimmed and the **Options** text area to be populated.)

   b In the **Options** text area, supply values for the **PATH** (Oracle only), **DB** (DB2 only), **USER**, and **PASSWORD** options. Also, add a **SCHEMA** option if the name of the schema to be used for the foundation repository does not match the value of the **USER** option.

   c Click **Next**.

   d On the Current Settings page, click **Finish**.

   e You see a dialog that displays the message "You must pause the server in order to apply the new authorization inheritance scheme. Pause the server now?" Click **Yes**.

f You see a dialog that display the message "Repository initialization was successful." Click **OK**.

At the end of this procedure, you have created a foundation repository. You can create additional repositories—custom and project repositories—by repeating the steps shown above. To start the wizard that adds a repository, right-click the **Active Server** in SAS Management Console (the **Active Server** is located under the Metadata Manager) and select **Add Repository** from the pop-up menu.

One final—but important—point. Generally, during installation, you create a foundation repository in SAS data sets and register some metadata in that repository. You may have also added metadata to the repository since then. After you create your new foundation repository in an Oracle or a DB2 database, you must copy your existing metadata to the new repository, or re-create it.

<RPOSMGR> Attributes Used to Connect to an Oracle Database

When you edit the file **omaconfig.xml** in order to create a repository manager in an Oracle database, you must specify the attributes shown below for the **<RPOSMGR>** element:

LIBREF
specifies the libref that SAS will use to connect to the repository manager. The usual value of this attribute is **RPOSMGR**.

ENGINE
specifies the SAS/ACCESS engine that you are using to connect to your database. In this case, the value will be **ORACLE**.

PATH
specifies the file-system path to the repository manager directory. Since the repository manager will be created in a database, set the value of this attribute to the empty string (`""`).

OPTIONS
contains a list of the LIBNAME options that SAS/ACCESS should use when it connects to the Oracle database.

The LIBNAME and connection options that make up the value of the **OPTIONS** attribute include the following:

reread_exposure=yes
specifies that the SAS/ACCESS engine will behave like a random access engine for the scope of the connection.

preserve_names=yes
preserves spaces, special characters, and mixed case in DBMS column and table names.

dbchar_constant_is_spoofed=yes
required and is valid only when used for access to SAS Metadata Repositories by the Oracle engine.

connection=shared
specifies that all tables that are opened for reading by the LIBNAME or libref share this connection.

readbuff=1
specifies the number of rows in a single Oracle fetch. A value of 1 is required.

insertbuff=1
: specifies the number of rows in a single Oracle insert operation. A value of 1 is required.

spool=no
: prevents spooling in the Oracle engine. This spooling is unnecessary because the SAS Metadata Server uses the SAS In Memory Database for the repository tables.

utilconn_transient=no
: specifies that a utility connection is maintained for the lifetime of the libref.

dbmax_text=4096
: (optional) on 64-bit platforms, specifies the length of any very long DBMS character data that is read into SAS or written from SAS via the SAS/ACCESS engine.

path=*path*
: specifies an alias representing the database server to which you want to connect. The alias points to information such as a hostname, a port number, and a transmission protocol. If you are not sure what path to use, consult with your database administrator.

user=*user*
: specifies an Oracle user name. If the user name contains blanks or national characters, enclose the name in quotation marks. If you omit an Oracle user name and password, then the default Oracle user OPS$sysid is used, if it is enabled. The **user** option must be used with the **password** option.

password=*password*
: specifies an Oracle password that is associated with the Oracle user name. If you omit the **password** option, then the password for the default Oracle user ID OPS$sysid is used, if it is enabled.

schema=*schema*
: (optional) enables you to register multiple repositories using the same Oracle user account. When you omit the **schema** option, the software uses the default schema, which is the requesting user ID, to create the repository. A schema name such as SASRMGR is recommended when you are creating the repository manager. The schema name must be entered in uppercase letters and be a valid SAS name that is unique for the **user**, **password**, and **path** options.

<RPOSMGR> Attributes Used to Connect to a DB2 Database

When you edit the file **omaconfig.xml** in order to create a repository manager in a DB2 database, you must specify the attributes shown below for the **<RPOSMGR>** element:

LIBREF
: specifies the libref that SAS will use to connect to the repository manager. The usual value of this attribute is **RPOSMGR**.

ENGINE
: specifies the SAS/ACCESS engine that you are using to connect to your database. In this case, the value will be **DB2**.

PATH
: specifies the file-system path to the repository manager directory. Since the repository manager will be created in a database, set the value of this attribute to the empty string (`""`).

OPTIONS

contains a list of the LIBNAME options that SAS/ACCESS should use when it connects to the DB2 database.

The LIBNAME options that make up the value of the **OPTIONS** attribute include the following:

reread_exposure=yes
: specifies that the SAS/ACCESS engine will behave like a random access engine for the scope of the connection.

preserve_names=yes
: preserves spaces, special characters, and mixed case in DBMS column and table names.

dbchar_constant_is_spoofed=yes
: required and is valid only when used for access to SAS Metadata Repositories by the DB2 engine.

connection=shared
: specifies that all tables that are opened for reading by this LIBNAME or libref share this connection.

readbuff=0
: specifies the number of rows in each DB2 fetch. A value of 0 is required.

insertbufff=1
: specifies the number of rows in a single DB2 insert operation. A value of 1 is required.

spool=no
: prevents spooling in the DB2 engine. Spooling data by the DB2 engine is unnecessary because the SAS Metadata Server uses the SAS In Memory Database for the repository tables. Although the SAS Metadata Server should not cause spooling to occur in the DB2 engine, explicitly setting the option to **no** will guarantee that no spooling occurs.

utilconn_transient=no
: specifies that a utility connection is maintained for the lifetime of the libref.

dbmax_text=4096
: (optional) on 64-bit platforms, specifies the length of any very long DBMS character data that is read into SAS or written from SAS using a SAS/ACCESS engine.

db=*database*
: specifies the DB2 data source or database to which you want to connect.

user=*user*
: enables you to connect to a DB2 database with a user ID that is different from the one you used to log on to your computer. The **user** and **password** connection options are optional for DB2. If you specify **user**, then you must also specify **password**. If you omit **user**, then your operating system user ID is used.

password=*password*
: specifies the DB2 password that is associated with your DBMS user ID. Specify a password only if you specified a DB2 user ID.

schema=*schema*
: (optional) enables you to register multiple repositories using the same DB2 user account. When you omit the **schema** option, the software uses the default schema, whose name matches the requesting user ID, to create the repository. A schema name such as SASRMGR is recommended when you are creating the repository

manager. You must enter the schema name in uppercase, and the name must be a valid SAS name that is unique for the **user**, **password**, and **db** options.

The omaconfig.xml File: Reference

The **omaconfig.xml** file specifies changes to standard SAS Metadata Server and repository manager features.

SAS Metadata Server features that can be changed include the following:

- the names and locations of the **adminUsers.txt** and **trustedUsers.txt** files
- the number of threads that are allowed to run concurrently on the SAS Metadata Server (contingent on the number of available processors)
- workunit journaling options

Repository manager features that can be changed include the location, the libref, and the engine used by the repository manager.

Sometimes the **omaconfig.xml** file includes an option that is added by a SAS release to activate a SAS Metadata Server or repository manager feature that is needed by a particular SAS application. When this is the case, do not remove the option or change its setting unless instructed to do so by SAS Technical Support.

Metadata server options are specified as attributes of an **<OMA>** XML element in the **omaconfig.xml** file. Repository manager options are specified as attributes of an **<RPOSMGR>** XML element in that file.

<OMA> Options

The **<OMA>** options include the following:

ADMINUSERS="*filename*"
specifies the name and location of the **adminUsers.txt** file. The planned installation creates the file in *SAS-config-dir***\Lev1\SASMain\MetadataServer**.

ALERTEMAIL="*email-address*"
specifies an e-mail address to which the metadata server should send a notification message when an error occurs that prevents the disk update process for the workunit journal from completing.

ALERTEMAILATTACHMENT="*filename*"
specifies an existing file to attach to the **ALERTEMAIL** message. Specify a relative pathname to identify the file. The path should be relative to the directory from which the metadata server is started. For example, if you want to attach the **omaconfig.xml** file, and the file is located in the same directory as the script that starts the server, you can simply enter the filename **omaconfig.xml**. If the **omaconfig.xml** file is located in a **config** directory that is parallel to the directory from which the metadata server is started, you can specify **..\config\omaconfig.xml**.

ALERTEMAILTEST
enables you to verify that the **ALERTEMAIL** and **ALERTEMAILATTACHMENT** options are set correctly. **ALERTEMAILTEST** is a *temporary* **omaconfig.xml** option that causes a test message to be sent at server startup in order to verify that the e-mail address and filename specified in the **ALERTEMAIL** and **ALERTEMAILATTACHMENT** options are valid. An error will cause the e-mail not to be sent and the metadata server to be stopped, so that the administrator can investigate which option has a problem. The metadata server is also stopped when the e-mail is sent correctly so that the

administrator can remove the **ALERTEMAILTEST** option from the **omaconfig.xml** file. The SAS Metadata Server will not run with the **ALERTEMAILTEST** option set. Be sure to specify a string value for the **ALERTEMAILTEST** options. Using the option without a value can also prevent the e-mail from being sent.

GROUPSFORMEMBERASSOCDELETE="*comma-separated-list-of-groups*"
invokes special functionality needed by the SAS Information Delivery Portal. The **GROUPSFORMEMBERASSOCDELETE** option specifies the names of groups that allow a member to be deleted by a user who has WriteMetadata permission for the member, even though the user does not have WriteMetadata permission for the group itself. Only the groups listed on this option allow the behavior; other groups continue to use the default security for the membership list. The following groups need special processing:

- DESKTOP_PORTALPAGES_GROUP
- Portlet Template Group
- OrderingGroup
- DESKTOP_PAGEHISTORY_GROUP
- Portal Collection

The names are case sensitive. The **GROUPSFORMEMBERASSOCDELETE** option is added to the **omaconfig.xml** file by the planned SAS installation. Do not change the names unless instructed to do so by SAS Technical Support. If this option is removed from the **omaconfig.xml** file, owners of information maps, reports, and stored processes, as well as others who have WriteMetadata permission to the metadata describing these items, will not be able to delete the items if someone has created a bookmark to them in the SAS Information Delivery Portal.

MAXACTIVETHREADS="*number-of-threads*"
specifies the number of threads that are allowed to run concurrently on the SAS Metadata Server. The number of processors determines the number of concurrent queries that can be made; therefore, the recommended setting is the number of processors on the host machine. For example, if you are running the server on a single-processor host, the recommended setting is **1**. If you are running the server on an eight-processor host, the recommended setting is **8**. A setting that is higher or lower than recommended one can affect performance.

Beginning in SAS 9.1.3 Service Pack 4, the planned SAS installation automatically sets the value of the **MAXACTIVETHREADS** option to the number of processors on the metadata server host. It also sets the values of the related THREADSMIN and THREADSMAX metadata server invocation options. For information about how the metadata server uses threads, and recommendations for setting the THREADSMIN and THREADSMAX options, see “Controlling the Number of Threads Used by the Server” on page 26.

TRUSTEDUSERS="*filename*"
specifies the name and location of the **trustedUsers.txt** file. The planned installation creates the file in *SAS-config-dir***Lev1\SASMain\MetadataServer**.

WORKUNITJOURNALPATH="*filename*"
activates workunit journaling and specifies a location to create the workunit journal file. Do not specify the **WORKUNITJOURNALPATH** option without also specifying a filename for the location. If the **WORKUNITJOURNALPATH** option is specified without a filename, the metadata server will fail to initialize.

We recommend that you specify a file that is in the directory from which the metadata server is started and that you name it something recognizable, such as **WorkunitJournal.dat**. If you need to use a different location and name, keep these things in mind:

- Put the workunit journal file on a native file system, such as the C: (system) drive. Do not place the file on a network-mounted file system or network appliance.
- Use a relative pathname to identify the file's location. The pathname should be relative to the directory from which the metadata server is started.
- The standard metadata server configuration creates repositories and the metadata server log on the same file system in which the SAS Metadata Server is started. If it is at all likely that this file system will fill up, consider moving the repositories to another file system. If the server file system fills up, the workunit journal file is not threatened because it is a fixed size. However, the metadata server log will need space to be able to record any disk-full messages in order to send them to the metadata server administrator.

WORKUNITJOURNALSIZE="*number-of-bytes*"
specifies a size, in bytes, for the workunit journal file. This space is reserved in the file system and is unavailable to other processes, whether the workunit journal file is full or not.

This option does not have a default value. The size that you assign to the workunit journal file depends on the total disk space available on the SAS Metadata Server host machine. The recommended size is 20MB (20000000), which should be enough for most sites. If clients regularly experience delays in getting their add, update, and delete requests processed, check the SAS Metadata Server log. If you find messages similar to the following, you might consider setting a higher **WORKUNITJOURNALSIZE** value:

```
WARNING: insufficient file space available in workunit journal file.
```

When an entry is added to the workunit journal file, it is marked as Pending. When entries are committed to the disk repository, they are marked as Done, and the space becomes available for new pending entries. If the rate at which entries are being added exceeds the rate at which they are completed, the workunit journal file can become full, and not enough space is available for new entries. When this occurs, the metadata server waits until the update process can catch up. When the contents of the workunit journal file are safely committed, the first new update is written directly to repositories on disk. Then, workunit journaling resumes.

Note: When you change the **WORKUNITJOURNALSIZE** value, or any other **omaconfig.xml** option, you must stop and re-start the metadata server for the change to take effect. △

WORKUNITJOURNALYIELDCOUNT="*number-of-observations*"
is designed to prevent large repository updates (for example, requests that add hundreds of objects) from monopolizing the server. It specifies the number of records that the update process can apply before yielding to allow other internal processing tasks, such as writing to the metadata server log, to occur and then resuming operation. For example, if **WORKUNITJOURNALYIELDCOUNT** were set to 20 and a workunit consisted of 100 records, the update process would apply 20 records, stop to allow internal processes to function, apply 20 more records, and so on, until all 100 records were applied to repositories.

This option does not affect the order in which updates are applied to repositories. Metadata updates are applied in the order in which they are received, and all repositories are locked for reading and writing at proper times. **WORKUNITJOURNALYIELDCOUNT** improves the overall performance of the metadata server by allowing those updates that have been performed on the in-memory metadata copy to be applied to the permanent copy on disk during periods of CPU availability. The in-memory copy remains up-to-date at all times.

The maximum setting of the **WORKUNITJOURNALYIELDCOUNT** option is 100 records; the minimum setting is 5 records. The default value is 14 records. Do not adjust the default value unless instructed to do so by SAS Technical Support.

<RPOSMGR> Options

The **<RPOSMGR>** options include:

LIBREF="*name*"
specifies an alternate libref for the repository manager. *Name* must be a valid SAS name. The name can be up to eight characters. The first character must be a letter (A, B, C, . . ., Z) or an underscore (_). Subsequent characters can be letters, numeric digits (0, 1, . . ., 9), or underscores. You can use uppercase or lowercase letters. SAS processes names as uppercase, regardless of how you type them. The default value is **RPOSMGR**.

ENGINE="*libname-engine*"
specifies the LIBNAME engine that is to be used to create the repository manager. The default value is an empty string; it defaults to Base SAS. Other supported values are **ORACLE** and **DB2**, representing the SAS/ACCESS to Oracle and SAS/ACCESS to DB2 LIBNAME engines. When changing the default value, note that engine values must be specified in uppercase letters. For information about using a repository manager that exists in an external DBMS, see "Creating the Metadata Repository in a Database Management System" on page 36.

PATH="*pathname*"
specifies an alternate location for the repository manager. The planned installation creates the repository manager in *SAS-config-dir***\Lev1\SASMain\MetadataServer\rposmgr**. *Pathname* can be an absolute reference or a relative reference.

Note: When defining the repository manager in an external DBMS, specify an empty string in this parameter (for example, **PATH=""**). △

OPTIONS="*libname-options*"
specifies connection options for the SAS/ACCESS to Oracle or SAS/ACCESS to DB2 LIBNAME engines. The required options are described in "Create the Repository Manager and a Metadata Repository in an Oracle or a DB2 Database" on page 37.

Reserved omaconfig.xml Options

The SAS Metadata Server supports a number of reserved **omaconfig.xml** options that you should not specify in the **omaconfig.xml** file unless you have been instructed to include them by SAS Technical Support. There are two categories of reserved options: SAS Metadata Server options and debugging options. The SAS Metadata Server options control features that are considered important for the optimized operation of the metadata server, but can be turned off if necessary. Because they are turned on by default, the only reason to include them in the **omaconfig.xml** file is to turn them off.

The debugging options can increase the processing overhead of the metadata server; therefore, they are not recommended for normal use.

The reserved server options are specified along with other server options in the **<OMA>** XML element in the **omaconfig.xml** file. The debugging options are specified in an **<OPTIONS>** XML element in the **omaconfig.xml** file.

Reserved <OMA> Options

BLOCK_ADDS="0 | 1"

optimizes server processing for adding metadata. A value of **1** (the default) optimizes processing by writing all objects that need to be added to a repository container data set in a single engine call. A value of **0** causes each object to be added by a separate engine call.

LOCK_DEPENDENT_REPOS="0 | 1"
determines the repository locking algorithm used by the server. A value of **0** (the default) specifies to lock only repositories that are explicitly referenced in update and delete requests. A value of **1** specifies to lock all repositories that have a hierarchical relationship with the repository on which update and delete requests are issued. Locking all repositories in a repository hierarchy can have a negative effect on server performance.

PHYSICAL_DELETE="0 | 1"
(deprecated) determines the disposition of deleted metadata records in a repository. A value of **0** specifies that deleted records are logically deleted from a repository. A value of **1** specifies that deleted records are physically deleted from a repository. The default is to physically delete metadata records.

WORKUNIT_INDEXES="0 | 1"
optimizes server processing for update requests by using indexes in workunits. A value of **0** causes internal searches to apply WHERE clauses to each object in the workunit list. A value of **1** (the default) causes internal searches of workunit updates to use indexes, which significantly speeds up the search.

Debugging Options

DEBUG
supports the debugging macros documented in the *SAS Internal Programmer's Guide* for SAS Scalable Architecture applications.

GENERIC="0 | 1"
supports the $GENERIC system option for SAS Scalable Architecture applications. A value of **1** turns the option on. A value of **0** (the default) turns the option off.

Options for the Metadata Server Invocation Command: Reference

The SAS Metadata Server supports the invocation parameters described in these sections:

- "Recommended Parameters" on page 46
- "Optional Parameters" on page 49

Recommended Parameters

The following parameters are recommended for the optimal operation of the SAS Metadata Server in all host environments. Not all of the parameters are required to start the server; however, they are included here because they are considered helpful to the operation of the server.

Some host environments have additional recommended parameters. These are described in "Additional Windows Parameters" on page 48, "Additional UNIX Parameters" on page 49, and "Additional z/OS Parameters" on page 49. Other parameters support optional functionality that is configured in conjunction with the **omaconfig.xml** file. These are described in "Optional Parameters" on page 49.

Recommended Parameters for All Host Environments

The following parameters are recommended for all host environments:

- the pathname of the non-interactive SAS executable at your site.
- the environment control parameters NOAUTOEXEC, NOTERMINAL, SASUSER, and RSASUSER:

  NOAUTOEXEC
  specifies not to process the SAS autoexec file, even if one exists. Processing of the file can result in unpredictable server behavior.

  NOTERMINAL
  specifies to run the server in batch mode.

  SASUSER="*library-specification*"
  specifies the SAS data library to contain the server's profile catalog. This library must be dedicated to the metadata server and is typically a directory named **sasusrms** or **sasuser** under the **MetadataServer** directory.

  RSASUSER
  limits access to the SASUSER library to read-only mode.
- the log output parameters LOG, LOGPARM, PAGESIZE, and LINESIZE:

  LOG="*file-specification*"
  enables metadata server logging and specifies a location to write the SAS log. The SAS Metadata Server is a non-interactive SAS session; therefore, this parameter is required to enable logging. If the LOG option is omitted, then SAS log entries display in the SAS Message Log window, which is closed during a non-interactive SAS session. The default location for the log file is *SAS-config-dir***\Lev1\SASMain\MetadataServer\logs**.

  LOGPARM="WRITE=*value* ROLLOVER=*value* OPEN=*value*"
  controls when SAS log files are opened and closed. Specify the parameter **"WRITE=IMMEDIATE"**; otherwise, the server buffers log entries in memory. If your server runs in a 24/7 environment, you might also want to specify **"ROLLOVER=AUTO OPEN=REPLACEOLD"** to partition the log into daily or weekly reports.

  PAGESIZE=*value*
  specifies the number of lines that make up a page of SAS output. A setting of **MAX** is recommended to reduce the occurrence of SAS log page headings and carriage controls.

  LINESIZE=*value*
  specifies the line size of SAS output. A setting of **MAX** is recommended to reduce log entry wrap-around.
- the encryption parameter NETENCRYPTALGORITHM:

  NETENCRYPTALGORITHM="*algorithm1*" | ("*algorithm1*", "*algorithm2*", ...)
  set in conjunction with the CLIENTENCRYPTIONLEVEL OBJECTSERVERPARMS option, specifies the algorithm(s) to be used for encrypted client-server data transfers. If you specify more than one algorithm, enclose all the algorithm names in parentheses and use commas to separate them. If there are embedded blanks in an algorithm name, enclose each algorithm within quotation marks. Valid values are **SASPROPRIETARY**, **RC2**, **RC4**, **DES**, and **TRIPLEDES**. NETENCRALG is an alias for NETENCRYPTALGORITHM.
- OBJECTSERVER, to invoke an Integrated Object Model (IOM) server.
- OBJECTSERVERPARMS, to pass IOM parameters to the server:

CLIENTENCRYPTIONLEVEL=NONE | CREDENTIALS | EVERYTHING
set in conjunction with the NETENCRYPTALGORITHM option, this option specifies the degree of encryption that the server uses when making outbound calls.

NONE	Nothing is encrypted. Clients transmit user IDs and passwords as plain text.
CREDENTIALS	Clients encrypt user IDs and passwords.
EVERYTHING	Clients encrypt all client-server communications.

CEL is an alias for CLIENTENCRYPTIONLEVEL.

PROTOCOL=*value*
identifies the network protocol used to communicate with the server. The only valid value is **bridge**.

PORT=*port-number*
specifies the TCP port on which the metadata server will listen for requests and that clients will use to connect to the server. The *port-number* value must be a unique number between 0 and 64K.

CLASSFACTORY=*factory-number*
specifies the type of IOM server to instantiate. The value **2887e7d7-4780-11d4-879f-00c04f38f0db** identifies a SAS Metadata Server.

TRUSTSASPEER=trustedPeers.xml
enables peer servers used by a SAS metadata client to connect to the metadata server using the credentials of the connecting client. SAS applications that require this support will state the requirement in their documentation. For more information, see "Setting up Trusted Peer Connections for SAS Sessions" in the *SAS Integration Technologies: Server Administrator's Guide* at **http://support.sas.com/rnd/itech/doc9/admin_oma/security/auth/security_imptrust.html**.

THREADSMIN=(*number-of-processors* * 2) + 1
specifies the minimum number of threads in the server's thread pool. When the number of threads falls below this number, IOM mechanisms automatically create new threads. TMIN is an alias for THREADSMIN.

Note: It is recommended that TMIN=TMAX. △

THREADSMIN and THREADSMAX are set in conjunction with the **MAXACTIVETHREADS** server configuration option.

THREADSMAX=(*number-of-processors* * 2) + 1
specifies the maximum number of threads in the server's thread pool. When the number of threads exceeds this number, IOM mechanisms automatically delete threads. TMAX is an alias for THREADSMAX.

Note: It is recommended that TMIN=TMAX. △

THREADSMIN and THREADSMAX are set in conjunction with the **MAXACTIVETHREADS** server configuration option.

Additional Windows Parameters

A Windows server also requires the following parameters:

the system administration parameter MEMSIZE=*value*
specifies the amount of memory to allocate to the metadata server process. We recommend a value that allows the operating system to set the memory limits on the server. This can be accomplished by setting a value of **0** (zero), **MIN**, or **MAX**.

the display parameter NOSPLASH
suppresses the display of SAS logo and copyright information when invoking the SAS session.

Additional UNIX Parameters

A UNIX server requires the following parameter in addition to those described in "Recommended Parameters for All Host Environments" on page 47:

the system administration parameter MEMSIZE=*value*
specifies the amount of memory to allocate to the metadata server process. We recommend a value that allows the operating system to set the memory limits on the server. This can be accomplished by setting a value of **0** (zero), **MIN**, or **MAX**.

Additional z/OS Parameters

A z/OS server requires the following parameters in addition to those described in "Recommended Parameters for All Host Environments" on page 47:

the file control parameter FILESYSTEM=HFS
specifies to run the server in a hierarchical file system.

the OBJECTSERVERPARMS parameter THREADSSTACKSIZE=1048576
specifies the stack size to give each thread.

Optional Parameters

The following parameters invoke optional functionality in a SAS Metadata Server in all host environments:

- The access control parameter AUTHPROVIDERDOMAIN:

  AUTHPROVIDERDOMAIN=(*provider*:*domain-name*)
  specifies an alternative authentication provider. When the AUTHPROVIDERDOMAIN parameter is omitted from the metadata server start command, the metadata server uses host authentication to authenticate users. AUTHPD is an alias for AUTHPROVIDERDOMAIN. For usage information, see "Using LDAP or Active Directory" in the chapter "Customizing the Authentication Configuration" in the *SAS Intelligence Platform: Security Administration Guide*.

  Note: In UNIX operating environments, you must insert an escape character before each parenthesis. For example:

  ```
  -authproviderdomain = \(ADIR:MyDomain\)
  ```

  △

- The workunit journaling parameters EMAILSYS, EMAILHOST, and EMAILID:

  EMAILSYS SMTP
  specifies that an e-mail message is sent by the metadata server using the SMTP (Simple Mail Transfer Protocol) e-mail interface.

  EMAILHOST *server-network-address*

specifies the network address of the enterprise's SMTP server, for example, *mailhost.company.com*.

EMAILID "*server-email-address*"
specifies the e-mail address in the **From** field for ALERTEMAIL messages. The e-mail address can be entered in the form "*server-name<user-account@domain>*" or simply as "*<user-account@domain>*".

Note: These EMAIL parameters should not be used unless workunit journaling is configured. For more information, see "Turning on Workunit Journaling" on page 27. △

- The OBJECTSERVERPARMS parameters:

  APPLEVEL=0 | 1 | 2 | 3
  controls IOM server logging. The default value (**1**) enables logging; therefore, this parameter is optional. APPLEVEL=0 disables IOM server logging. Setting APPLEVEL=0 is highly discouraged because the server will not have a log. Other APPLEVEL options invoke additional logging functionality. For more information about logging levels and the possible values of this option, see "Logging Events and Errors" on page 31.

  IOMLEVEL= 0 | 1
  specifies whether input and output XML strings are captured in the SAS log. The default value of **0** disables capturing of input and output XML strings. A value of **1** enables capturing of input and output XML strings. For more information, see "Capturing XML in the Log" on page 33.

  JNLSTRMAX=*number*
  used in conjunction with IOMLEVEL, specifies the maximum journal string length for XML input and output strings. A minimum value of 500 is recommended for the SAS Metadata Server.

  JNLLINEMAX=*number*
  used in conjunction with IOMLEVEL, specifies the maximum journal line length for XML input and output strings. A minimum value of 1,000 is recommended for the SAS Metadata Server.

CHAPTER

4

Maintaining the SAS Metadata Server

Maintaining the SAS Metadata Server

SAS Metadata Server maintenance tasks are discussed in the sections listed below:

□ "Starting and Stopping the SAS Metadata Server" on page 52
□ "Backing Up and Restoring the SAS Metadata Server" on page 53
□ "Checking the SAS Metadata Server's Access State" on page 69
□ "Quickly Recovering Memory in the SAS Metadata Server" on page 69
□ "Using the ARM_OMA Subsystem" on page 70
□ "SAS Metadata Server Troubleshooting" on page 80

Unless otherwise noted, you must be at least an *administrative user* on the SAS Metadata Server in order to perform server maintenance tasks. You are an *administrative user* if you connect to the metadata server using the sasadm user ID, or if your user ID is listed in the **`adminUsers.txt`** file in the **`MetadataServer`** directory.

Starting and Stopping the SAS Metadata Server

Generally, you start and stop the metadata server in the same way that you start and stop other SAS servers, as explained in "Starting and Stopping Your SAS Servers" on page 18. On Windows systems, the easiest way to start and stop the server is to use the **`Start`** menu. Select **Start ▶ Programs ▶ SAS ▶ *configuration-directory* ▶ Start SAS Metadata Server** to start the server and **Start ▶ Programs ▶ SAS ▶ *configuration-directory* ▶ Stop SAS Metadata Server** to stop it.

Note: If your metadata server runs as a service—which is recommended—your metadata server will be started any time you restart the host computer. △

On UNIX systems, you use the script *SAS-config-dir*/**`Lev1/SASMain/MetadataServer/MetadataServer.sh`** to start and stop the server. From the **`MetadataServer`** directory, type **`MetadataServer.sh start`** to start the server, or **`MetadataServer.sh stop`** to stop it. On z/OS systems, you start and stop the server by starting or stopping the associated started task.

In addition to using these methods that apply to all SAS servers, you can stop and start the metadata server by using SAS Management Console and PROC METAOPERATE. In SAS Management Console, in the Metadata Manager, right-click the **`Active Server`** and select **`Stop`** from the pop-up menu to stop the server.

Backing Up and Restoring the SAS Metadata Server

Introduction

SAS provides the %OMABAKUP macro to enable you to back up the SAS Metadata Server. A batch interface, the %OMABAKUP macro makes a system copy of SAS metadata repositories and the repository manager. You can use the system copy to restore the repositories and the repository manager to the state that they were in when the backups were made.

The SAS Metadata Server locks repository files when they are in use. This step prevents operating system and third-party backup tools from making backup copies. Using %OMABAKUP, you can copy SAS metadata repositories to an unlocked SAS backup location. The backup copies can then be included in your regular backups. Beginning in SAS 9.1.3 Service Pack 4, you can also use %OMABAKUP to restore repositories to a running a metadata server by setting a RESTORE option.

The default behavior of the macro is to back up the server's repository manager and all repositories that are registered in the repository manager. Beginning in SAS 9.1.3 Service Pack 4, the macro also supports selective backups by means of a REPOSITORYLIST option. The REPOSITORYLIST option can also be set with the RESTORE option to perform selective restores. Because of interdependencies that exist between repositories, selective backups and restores should be done with caution.

When %OMABAKUP is executed in backup mode, the following additional functionality is optionally available:

REORG
: reclaims unused disk space left by previously deleted metadata from SAS metadata repositories. For more information on this topic, see "Reclaiming Unused Disk Space by Using the %OMABAKUP Macro" on page 58.

RUNANALYSIS
: executes a process to analyze repository data sets for information to improve repository memory usage. This is new in SAS 9.1.3 Service Pack 4. For more information on this topic, see "Optimizing Repository Memory Usage by Using the %OMABAKUP Macro" on page 58.

The macro's functionality is described in detail in the sections that follow.

How the Backup Process Works

The %OMABAKUP macro uses PROC COPY to copy repositories. This is an outline of how the backup process works:

1. You must create the backup destination directory before running the %OMABAKUP macro. The macro does not create it for you, even if you use the suggested backup destination path.
2. Upon execution, %OMABAKUP performs the following tasks:
   a. It creates a subdirectory in the destination directory for each repository included in the backup.
   b. It copies each repository's data sets to the appropriate backup subdirectory.
   c. It creates a DS_DEFN data set for each repository in its registered location. This feature began in Service Pack 2. DS_DEFN is a control data set that specifies how character variables in repository data sets are to be treated in memory. This data set is used to load repositories back into memory when client activity is resumed on the metadata server.

d Beginning in Service Pack 4, %OMABAKUP creates two additional catalogs in the destination directory:

- OMA_backup.sas7bcat stores information about repository source and target destinations.
- OMA_varan.sas7bcat contains a job that SAS processes can optionally use to analyze repository data sets in the backup subdirectories.

These catalogs are updated each time a backup is performed.

e When executed with the RUNANALYSIS option, %OMABAKUP creates a DS_VCHAR data set in each backup subdirectory. The DS_VCHAR data sets are described in "Optimizing Repository Memory Usage by Using the %OMABAKUP Macro" on page 58.

Each subdirectory in the backup destination path is named after the value stored in the repository's **Name** attribute. For example, for two repositories registered on the metadata server using the following pathnames and names:

```
\MetadataServer\MetadataRepositories\Foundation (Name="Foundation")
\MetadataServer\MetadataRepositories\Custom1 (Name="MyRepository")
```

%OMABAKUP creates subdirectories with the following names in the destination directory. **SASBackup** is the name of the default backup destination:

```
\SASBackup\RPOSMGR
\SASBackup\Foundation
\SASBackup\MyRepository
```

RPOSMGR is the system-supplied name of the repository manager. The repository manager is copied whether or not a full or selective backup is performed.

Beginning in SAS 9.1.3 Service Pack 4, %OMABAKUP supports repository **Name** values of up to 60 characters. Earlier releases support names of up to 32 characters. The full pathname of a backup repository cannot exceed 200 characters.

If a repository name contains characters that are not allowed in directory names by the host operating environment, %OMABAKUP replaces any unsupported characters with underscores. On Windows systems, the following characters are not supported in repository names: \, /, :, *, ?, ", ', <, >, [,], and |. On UNIX systems, the following characters are not supported in repository names: \, /, ", ', [,], ?, and * .

How the Restore Process Works

The %OMABAKUP macro uses the SAS Metadata Server to process requests; therefore, the metadata server must be running in order to restore repositories using the RESTORE option. If the metadata server is not responsive, you will need to restore repositories from backup by using operating system commands. See "Manually Restore a Repository on an Unresponsive SAS Metadata Server" on page 66 for more information. If you are using a SAS software version that is earlier than SAS 9.1.3 Service Pack 4, see "Manually Restore a Repository on a Running SAS Metadata Server" on page 67 for information on restoring the SAS Metadata Server.

The general process for performing a full restore using %OMABAKUP is as follows:

1. Execute %OMABAKUP with the settings RESTORE="YES" and REPOSITORYLIST="RPOSMGR" to restore the repository manager. The repository manager must be restored in advance of the repositories. %OMABAKUP deletes the current repository manager and replaces it with the backup repository manager.
2. Execute %OMABAKUP again with the setting RESTORE="YES", this time either listing all repositories in the REPOSITORYLIST option, specifying

REPOSITORYLIST="ALL", or omitting the REPOSITORYLIST option altogether. When REPOSITORYLIST is omitted, %OMABAKUP reads the active (restored) repository manager to determine which repositories to restore and their target locations. It will restore all repositories that are registered in the repository manager. The macro deletes existing repositories before copying the backup repositories.

3 Follow the instructions in "Enabling and Disabling Repository Auditing" on page 90 to disable auditing on any repositories that have auditing enabled.
   - After you disable auditing on those repositories, you can re-enable auditing to a new audit path location.
   - If you want to continue using the existing audit path locations, perform these three steps *before* executing the restore:
     - **a** Disable auditing on repositories that have auditing enabled.
     - **b** Move all existing audit files to another location, so that auditing can be enabled on empty audit path locations.
     - **c** Re-enable auditing on the repositories.

To perform a selective restore, follow these steps:

1 Execute %OMABAKUP with the setting RESTORE="YES" and specify the repositories that you want to restore in the REPOSITORYLIST option. %OMABAKUP reads the active repository manager to determine the target locations for the repositories. The existing repositories are deleted before the backup repositories are copied.

2 Follow the instructions in "Enabling and Disabling Repository Auditing" on page 90 to disable auditing on any of the specified repositories that have auditing enabled, and re-enable auditing to a new audit path location. Or, follow the steps given above to empty the existing audit path locations.

In a selective restore, the repository manager is usually not restored. See "Planning a Restore" on page 59 for information to help you determine whether or not the repository manager should be restored. If you choose to restore it, note that the repository manager must be restored before the repositories are restored.

The DS_DEFN and DS_VCHAR data sets and OMA_varan.sas7bcat and OMA_backup.sas7bcat catalogs are not restored in a full or a selective restore. These data sets will be re-created the next time that you execute %OMABAKUP with appropriate options in backup mode.

Using the %OMABAKUP Macro

Preparing to Execute the %OMABAKUP Macro

The %OMABAKUP macro is provided in your SAS autocall libraries and runs in a dedicated SAS session. In order to execute the macro, create a SAS file that contains a SAS Metadata Server connection statement and supplies values for macro parameters. You can then execute this SAS file manually, or use operating system commands to automate its execution at periodic intervals. For instructions on how to create this file, see "Execute %OMABAKUP on Windows or UNIX" on page 63 or "Execute %OMABAKUP on z/OS" on page 63.

Usage Requirements

- %OMABAKUP must be executed on the computer that hosts the SAS Metadata Server.

- In a Windows environment, you must be logged in as a member of the Windows Administrators group or use the user ID that was used to start the SAS Metadata Server to execute the macro. On UNIX and z/OS, you need to use the user ID that was used to start the SAS Metadata Server.
- In all host environments, the user ID executing %OMABAKUP must also have *administrative user* status on the SAS Metadata Server, as well as full operating system access to the **MetadataServer** directory, the **rposmgr** directory, and all repository and backup directories. For more information about *administrative user* status, see "Rights Granted to Administrative, Unrestricted, and Trusted Users" on page 29.

Usage Recommendations

- %OMABAKUP pauses client activity on the specified repositories before making backup copies, and then resumes client activity when the backup is complete. In order to minimize the impact of the repository downtime on clients, %OMABAKUP should be executed when demand for the metadata server is low.
- In order to safeguard the integrity of your backups, consider alternating backups to multiple backup locations. For example, you might alternate backups to different network-accessible drives, depending on the day of the week. In this way, if a backup becomes corrupted, or a backup host becomes damaged, you have other locations from which to restore the SAS Metadata Server. For more information, see "Alternating Backups to Different Destinations" on page 64.
- To safeguard the integrity of your metadata, make regular full backups of the metadata server. A full backup copies the repository manager and all repositories registered on the metadata server. In addition to your regular full backup schedule, you should also perform a full backup after any of the following occur:
  - You add a repository.
  - You delete a repository.
  - You change the dependencies between two repositories.

  The SAS Metadata Server supports dependencies between repositories to enable cross-references to be defined between the repositories. These dependencies are managed by the repository manager. It is important to have a current backup copy of the active repository manager and the repositories represented in the repository manager, because restoring a repository to an out-of-date repository manager can cause serious problems if the dependencies do not match. Selective repository backups are supported to enable you to save copies of changes within a specific repository until a full backup can be performed. Whenever a specific repository is backed up, you should also back up any repositories that depend on that repository in order to preserve any cross-references that might exist between them. Otherwise, if a restore becomes necessary, a selective restore could damage cross-references between the repositories. You can identify the repositories that depend on a given repository in the Metadata Manager in SAS Management Console. For instructions, see "Identifying Repository Dependencies" on page 57. Specific recommendations for backing up and restoring repositories are provided in "Planning Backups" on page 56 and "Planning a Restore" on page 59.

Planning Backups

Benefits of Full Backups

As previously stated, the default behavior of the %OMABAKUP macro is to make a full backup of the SAS Metadata Server to the location specified in the

DESTINATIONPATH parameter. A full backup includes the SAS Metadata Server's repository manager and all SAS metadata repositories that are registered on the metadata server.

Note: A full backup does *not* include metadata server configuration files, such as the **`omaconfig.xml`** file, the **`adminUsers.txt`** and **`trustedUsers.txt`** files, or repository audit trails. Use operating-system commands or some other tool to back up these files. △

A full backup takes a complete snapshot of all registered repositories to enable you to restore all metadata and repository relationships to the exact state they were in when the backup copy was made. When a full backup is restored, any changes made to the metadata or repository definitions between the time the backup was made and the time it was restored will be lost, but the repositories will be returned to a known consistent state. Because they ensure consistency, regular full backups and full restores are recommended as the best approach for backing up and restoring the SAS Metadata Server.

Backing Up Specific Repositories

You can back up specific repositories by executing %OMABAKUP with the REPOSITORYLIST option. Reserve selective backups for special situations, because they introduce the potential for inconsistencies between repositories.

These inconsistencies can be mitigated by backing up as a unit any repositories that have dependencies defined between them. That is, whenever a specific repository is backed up, you should also back up all repositories that depend on that repository to preserve cross-references that might exist between them. Use the following guidelines to determine what repositories need to be backed up as a unit:

- All custom and project repositories depend on the foundation repository. Therefore, when you back up the foundation, you should also back up all of the custom and project repositories (full backup).
- In addition to having dependencies on the repositories that it services, a project repository potentially holds locks on objects in these repositories. To prevent these locks from being orphaned, back up all repositories that the project repository services along with the project repository. A project repository can service one or many repositories. If one project repository services many repositories, a full backup is recommended.
- Customers can also define site-specific dependencies between custom repositories. When backing up a custom repository, check to see if any other custom repositories depend on it. Follow the steps in "Identifying Repository Dependencies" on page 57 to view both the implicit and site-specific dependencies that might have been defined for a given repository.

Note: When you specify a repository name in the REPOSITORYLIST option, the %OMABAKUP macro does not automatically include repositories that have dependencies in the list. You must specify each repository that has a dependency in the list. △

Identifying Repository Dependencies

You can view a repository's dependencies in the Metadata Manager in SAS Management Console. To view a repository's dependencies:

1. Expand the **`Metadata Manager`** node to display the **`Active Server`**.
2. Expand the **`Active Server`** node to display a list of the repositories that are registered on the metadata server.
3. Highlight the repository in question in the navigation tree. The software displays the repository's dependencies in the right pane:

- The repositories that depend on this repository have **Depended On By** next to their name.
- The repositories on which the repository depends have **Depends On** next to their name.

When determining which repositories to back up with a custom repository, note repositories that have **Depended On By** next to their name.

When determining which repositories to backup with a project repository, note repositories that have **Depends On** next to their name.

Note: If you are considering a selective backup of a custom repository, and the custom repository has a project repository that depends on it, back up all repositories that have a **Depends On** relationship to the project repository, in addition to repositories that have a **Depended On By** relationship to the custom repository. △

Reclaiming Unused Disk Space by Using the %OMABAKUP Macro

When metadata is deleted from a SAS metadata repository, the record is removed from both memory and disk; however, the space allocated for the record remains in the data set after the record is removed. To reclaim unused disk space left by previously deleted metadata objects from SAS metadata repositories, you can additionally set the REORG option in the %OMABAKUP command. When REORG="YES", %OMABAKUP re-creates repository data sets as repositories are copied to the backup destination, and then copies the re-created data sets back to the production location. Because of the overhead associated with re-creating the data sets, you might not want execute this option for every backup. The default value is REORG="NO". Consider using this option on a monthly basis and after large amounts of metadata have been deleted.

Optimizing Repository Memory Usage by Using the %OMABAKUP Macro

The RUNANALYSIS Option for the %OMABAKUP Macro

The SAS Metadata Server is an *in-memory* server. This means that in addition to being stored on disk, all queried and updated metadata is held in memory for the duration of the server session in order to optimize performance. The records are read into memory as they are requested by clients and remain there until client activity on the metadata server is either paused and resumed, or the metadata server is stopped. SAS provides tools to optimize memory usage by these in-memory repository data sets. One such tool is the %OMABAKUP RUNANALYSIS option.

When %OMABAKUP is executed with the setting RUNANALYSIS="YES", the macro executes a subprogram that analyzes the handling of character variables in backup copies of repository data sets. The analyses are run on backup copies after client activity is resumed on the metadata server. The analyses check for such things as unnecessary indexes, long character variables that can be stored as variable-length strings, and duplicate strings. The program stores its results in an output DS_VCHAR data set in each backup subdirectory.

Because the analyses are performed after the backup process completes, the results of the analyses are not immediately applied to the memory copies of repositories. The results are applied the next time that %OMABAKUP is run to the backup destination. Therefore, consider executing another backup immediately after the analyses are done. When you execute this backup, %OMABAKUP merges the information in the DS_VCHAR data sets with the DS_DEFN data set in each registered repository

directory when the backup completes. It then uses the updated DS_DEFN data sets to load repositories back into memory.

Background on the RUNANALYSIS Option

RUNANALYSIS is one way to analyze repository data sets. Another way is to use the %OMARUNAN macro. The difference between RUNANALYSIS and %OMARUNAN is that RUNANALYSIS executes on the metadata server host immediately after a backup, which can consume host resources at a time when client requests have resumed. %OMARUNAN can be executed at any time from any computer that has SAS licensed and has a network connection to the computer storing the backups. The results of both analyzes are applied by %OMABAKUP the next time a backup is performed.

To minimize the impact of the analyses on the metadata server machine, you might want to direct the backups that you want to analyze to a drive that the metadata server can access over the network, and execute %OMARUNAN on that machine. If you do not have another networked computer that has a SAS license, you might still want to use %OMARUNAN so that you can run the analyses in a period of low activity for the metadata server. For more information about %OMARUNAN, see "Optimizing Repository Memory Usage" on page 96.

It is not necessary to perform character variable analyses on a regular basis. Variable usage is unlikely to change much unless you add or delete a significant percentage of the total metadata in a repository. The analyses are most useful after metadata is first loaded in SAS metadata repositories and after significant updates are applied.

Important Considerations for Using RUNANALYSIS

- When running analyses, be sure that other %OMABAKUP options support the analyses that you want to perform. For example, if you want to analyze all repositories that are registered on the metadata server, run %OMABAKUP with the setting REPOSITORYLIST="ALL" or omit the REPOSITORYLIST option. To analyze specific repositories, identify them in the REPOSITORYLIST option. See "Planning Backups" on page 56 for guidelines to follow when making selective backups. In both cases, run a full backup immediately following the analyses to ensure that the results are applied to all intended repositories and to back up any cross-references between repositories that might have been missed by the selective backup.
- If you run %OMABAKUP on an analyzed backup location in restore mode before the optimizations are applied, the optimizations will be lost. The RESTORE option deletes the DS_VCHAR data sets from the backup directories before restoring repositories to their registered locations.

Planning a Restore

Important Considerations When Planning a Restore

A full restore is necessary when the SAS Metadata Server host computer is destroyed or damaged, or when one or more metadata repositories have been corrupted. When repositories have been corrupted, you do not have to restore the repository manager; however, it is a good idea to do so so in order to ensure that the metadata server is restored to a consistent state.

Like a selective backup, a selective restore needs to be planned with care to avoid damaging dependencies and cross-references that exist between repositories. See "Planning Backups" on page 56 for information about dependencies that can affect which repositories should be backed up and restored as a unit.

Generally, if the foundation repository needs to be restored, it is a good idea to restore all repositories. When a custom repository needs to be restored, also restore any repositories that depend on it. When a project repository needs to be restored, repositories that it services should also be restored in order to protect potential locks in the repositories.

When deciding whether to restore the repository manager, consider the number and type of changes that have been made to repositories since the last full backup. For example:

- Did you add any repositories since the last full backup? If so, then you might want to keep the current repository manager. If you restore the repository manager, the new repository's registration will be overwritten. You can re-register the repository, but you must use the same engine and options.
- Is the purpose of the restore to recover a repository that was deleted by using the Delete action in SAS Management Console? The Delete action unregisters a repository and destroys the repository data sets. Therefore, you need to restore the repository manager in addition to the deleted repository, or the current repository manager will have no record of the repository.
- Are you restoring a repository that was accidentally deleted by using operating-system commands? Its registration remains in the repository manager. You only need to restore the repository files.

Using PROC COMPARE

If you are unsure of the changes that have been made since the last full backup, use PROC COMPARE to compare the MDASSOC and RPOSCTRL data sets in the current and backup repository manager directories. Here are sample statements:

```
libname origrpos '/users/omitest/rposmgr';
libname backup '/users/omitest/backup/REPOSMGR';

proc compare base=origrpos.mdassoc compare=backup.mdassoc; run;
proc compare base=origrpos.rposctrl compare=backup.rposctrl; run;
```

The MDASSOC data set contains repository dependencies. The RPOSCTRL data set contains repository registrations.

See "How the Restore Process Works" on page 54 for the specific steps that are needed to restore repositories and the repository manager with %OMABAKUP.

Audit Trails

Do not restore repository audit trails. After a restore, auditing should be resumed on emptied or new audit path locations.

%OMABAKUP Macro: Reference

The %OMABAKUP macro copies some or all of the repositories that are registered on a metadata server and the server's repository manager to a backup location.

Syntax

```
%omabakup(DestinationPath="pathname",
          ServerStartPath="pathname",
          RposmgrPath="pathname",
          <Reorg="Yes|No">,
```

```
<RepositoryList="repository-name<,...repository-name-n">>,
<Restore="Yes|No">,
<RunAnalysis="Yes|No">)
```

DESTINATIONPATH="*pathname*"
is a required parameter for a backup or a restore. For a backup, specify the pathname of the directory where backup copies are to be stored. Specify the absolute pathname of an existing directory; %OMABAKUP will not create the directory for you. See Table 4.1 on page 62 for a pathname that supports a planned SAS installation. For a restore, specify the pathname of the directory where backups are stored.

SERVERSTARTPATH="*pathname*"
is a required parameter for a backup and a restore. Specify the full path to the directory in which the metadata server process was started. See Table 4.1 on page 62 for the pathname that supports a planned SAS installation.

RPOSMGRPATH="*pathname*"
is a required parameter for a backup and a restore. Specify the relative location of the repository manager as defined in the **omaconfig.xml** file. See Table 4.1 on page 62 for the pathname that supports a planned SAS installation.

REORG="YES|NO"
is an optional parameter for a backup that enables you to reclaim unused disk space left from previously deleted metadata objects from SAS metadata repositories. **YES** (or **Y**) specifies to reclaim disk space. **NO** (or **N**) specifies to copy the repositories as they are. The default value is **NO**. This option is not valid when %OMABAKUP is executed in restore mode. For more information, see "Reclaiming Unused Disk Space by Using the %OMABAKUP Macro" on page 58.

REPOSITORYLIST="*repository-name<,...repository-name-n>*"
is an optional parameter for a backup or a restore that enables you to specify a list of repositories to back up or restore. Repositories are identified by the name stored in their Name attribute. If more than one repository name is specified, separate the names with commas. When entering names, note that the names are case-sensitive, and the repository manager must be identified as REPOSMGR. See "Planning Backups" on page 56 and "Planning a Restore" on page 59 for important information about performing selective backups and restores.

RESTORE="YES|NO"
is an optional parameter that enables you to execute %OMABAKUP in restore mode. **YES** (or **Y**) turns on restore mode. **NO** (or **N**) specifies that %OMABAKUP run in backup mode. The default value is **NO**. When RESTORE="YES", %OMABAKUP restores all repositories registered in the SAS Metadata Server's repository manager to their registered locations. If you want to do a selective restore, additionally set the REPOSITORYLIST option. For more information, see "How the Restore Process Works" on page 54 and "Planning a Restore" on page 59.

RUNANALYSIS="YES|NO"
is an optional parameter for a backup that specifies whether to analyze backup copies of repository data sets for information that can be used to optimize their memory footprint on the metadata server. The analysis is run immediately after client activity is resumed on the SAS Metadata Server. **YES** (or **Y**) specifies to analyze the backup repositories. **NO** (or **N**) specifies not to analyze the backup repositories. The default value is **NO**. This option is not valid when %OMBAKUP is executed in restore mode. For more information, see "Optimizing Repository Memory Usage by Using the %OMABAKUP Macro" on page 58.

Note: If you use pathnames for DESTINATIONPATH, SERVERSTARTPATH, and RPOSMGRPATH that are different from those shown in Table 4.1 on page 62, be aware

that the pathnames cannot contain embedded asterisks (*), backslashes (\), question marks (?), brackets ([]), or single or double quotation marks with the exception that the backslash is supported as a directory delimiter on hosts that support the backslash as a delimiter. △

Details

Before you can execute the %OMABAKUP macro, you must connect to a SAS Metadata Server. See "Server Connection Statement" on page 62 for details.

Table 4.1 on page 62 shows the recommended pathnames for the DESTINATIONPATH, SERVERSTARTPATH, and RPOSMGRPATH parameters in each host environment. These pathnames are based on the planned SAS installation. In the pathnames, *config-dir-name* represents the name of your configuration directory, which is determined by the value of the **`Name`** attribute of the **`<Group>`** element in your deployment plan (**`plan.xml`**). If the name of your configuration directory is **`BIArchitecture`**, on a Windows systems, you might enter **`C:\SAS\BIArchitecture\Lev1\SASMain`** for the SERVERSTARTPATH parameter and **`C:\SAS\BIArchitecture\Lev1\SASBackup`** for the DESTINATIONPATH parameter.

Table 4.1 Recommended %OMABAKUP Parameters

Host Environment	Recommended Pathnames
Windows	DestinationPath="C:\SAS*config-dir-name*\Lev1\SASBackup"
	ServerStartPath="C:\SAS*config-dir-name*\Lev1\SASMain"
	RposmgrPath="MetadataServer\rposmgr"
UNIX	DestinationPath="*installer's-home-dir*/SAS/*config-dir-name*/Lev1/SASBackup"
	ServerStartPath="*installer's-home-dir*/SAS/*config-dir-name*/Lev1/SASMain"
	RposmgrPath="MetadataServer/rposmgr"
z/OS (OS/390)	DestinationPath="/usr/lpp/SAS/*config-dir-name*/Lev1/SASBackup"
	ServerStartPath="/usr/lpp/SAS/*config-dir-name*/Lev1/SASMain"
	RposmgrPath="MetadataServer/rposmgr"

If you used other pathnames in the intelligence platform installation, or if you want to specify a different backup destination, be sure to verify that the pathnames in your backup execution file match before running the macro.

Server Connection Statement

The %OMABAKUP macro requires the following server connection statement to connect to the metadata server:

```
options metaserver='localhost'
        metaport=port-number
        metaprotocol=bridge
        metauser='administrative-userID'
        metapass='encoded-password'
        metarepository='Foundation';
```

The followin list explains how to set these options:

- For METASERVER, type **`localhost`** to connect to the metadata server on the same machine.

- For METAPORT, type the unique port number that identifies your metadata server. The default port number is 8561.
- For METAUSER and METAPASS, type the user ID and password of an *administrative user* that meets the requirements described in "Usage Requirements" on page 55. You should encode the password prior to this step by using PROC PWENCODE, and then specify the encoded password here.
- For METAREPOSITORY, specify the name of the default repository. The name that is recommended for all installations is Foundation.

Executing the %OMABAKUP Macro

Execute %OMABAKUP on Windows or UNIX

To create and execute an %OMABAKUP execution file on Windows and UNIX, perform these steps:.

1. Create the backup directory at the location recommended in Table 4.1 on page 62.
2. Start SAS.
3. In the Program Editor, create a SAS file that contains a server connection statement and %OMABAKUP parameters and name it **BackupRepositories.sas**. The parameters can be submitted as follows:

```
options metaserver='localhost'
        metaport=8561
        metaprotocol=bridge
        metauser='administrative-userID'
        metapass='encoded-password'
        metarepository='Foundation';

%omabakup(DestinationPath="absolute-pathname",
          ServerStartPath="absolute-pathname",
          RposmgrPath="relative-pathname")
```

4. Save the file to the directory where the SAS Metadata Server is started (STARTSERVERPATH).
5. Define operating system permissions that deny access to the **SASBackup** directory and to the **BackupRepositories.sas** file to anyone other than the person who will run the backups.
6. To execute a backup, submit one of the following commands:

   Windows command:

   ```
   "SAS-install-dir\sas.exe" BackupRepositories.sas
   ```

   UNIX command:

   ```
   "SAS-install-dir/sas" BackupRepositories.sas
   ```

   Alternatively, set up **BackupRepositories.sas** to be executed by a scheduling utility supplied by the operating system.

Execute %OMABAKUP on z/OS

To create and execute an %OMABAKUP execution file on z/OS, perform these steps.

1. Create the backup directory at the location recommended in Table 4.1 on page 62.

2. Create a CONFIG data set that contains a server connection statement as described in "Server Connection Statement" on page 62.
3. Create a batch job that (a) starts SAS, (b) references the CONFIG data set that contains the server connection statement, and (c) specifies values for the %OMABAKUP parameters.
4. Use operating system commands to submit the batch job or to automate execution of the macro program.

This example CONFIG data set member contains the server connection statement:

```
metaserver=hostname
metaport=8561
metaprotocol=bridge
metauser=administrative-userID
metapass=encoded-password
metarepository=Foundation
```

Here is an example of a batch job that runs the backup:

```
//BACKUP JOB (R4221),'myname',

//  MSGCLASS=A,
//  TIME=(5)
*-----------------------------------------------------------------
  RUN BACKUP
*-----------------------------------------------------------------
//SAS91  EXEC SAS91,

//           CONFIG=&prefix.WO.SRVCFG(BACKUP)

//TKMVSENV DD DISP=SHR,DSN=&prefix.WO.SRVENV(TKMVSENV)
//SYSIN    DD *

%OMABAKUP(DestinationPath="/usr/lpp/SAS/config-dir-name/Lev1/SASBackup",
          ServerStartPath="/usr/lpp/SAS/config-dir-name/Lev1/SASMain",
          RposmgrPath="./MetadataServer/rposmgr");

/*

//
```

Note that the %OMABAKUP parameters are specified as UNIX pathnames.

Alternating Backups to Different Destinations

The following is an example of a **BackupRepositories.sas** file for a Windows metadata server host that executes %OMABAKUP twice to write backup copies to backup destinations on different computers. The second invocation of the macro creates the backup in a different directory each day of the week.

```
/* These system options establish a connection to the metadata server */
options metaserver='D1234'
        metaport=8561
        metaprotocol=bridge
        metauser='administrative-userID'
        metapass='encoded-password'
        metarepository='Foundation';
```

```
/* This %OMABAKUP command creates a backup on the D1234 machine in
 C:\SAS\config-dir-name\Lev1\SASBackup */

%omabakup(DestinationPath="C:\SAS\config-dir-name\Lev1\SASBackup",
          ServerStartPath="C:\SAS\config-dir-name\Lev1\SASMain",
          RposmgrPath="MetadataServer\rposmgr")

/*Daily backups are stored at different locations on a D2345 machine,
depending on the day of the week*/
data _null_;
/* Function to get the day of the week */
dayofweek = weekday(date());
/* Using the case and select statement for the day of the week */
select (dayofweek);
   when (1)  call symput('day','Sun');
   when (2)  call symput('day','Mon');
   when (3)  call symput('day','Tue');
   when (4)  call symput('day','Wed');
   when (5)  call symput('day','Thu');
   when (6)  call symput('day','Fri');
   when (7)  call symput('day','Sat');
   otherwise;
end;
run;
%put _user_;
%put &day

/*This command creates backups on the D2345 machine */
%omabakup(DestinationPath="\\D2345\Dmt01 Repos Backups\&day",
          ServerStartPath="C:\SAS\config-dir-name\Lev1\SASMain",
          RposmgrPath="MetadataServer\rposmgr")
```

In the example, the first %OMABAKUP command backs up all the repositories to the C: drive on the local host. This backup file is overwritten every day. It represents the most recent backup image. The second %OMABAKUP command backs up the same repositories to the D2345 machine via the network pathname **\\D2345**, and puts the backup in the **Dmt01 Repos Backups\&day** folder, where **&day** is a folder named **Sun**, **Mon**, **Tue**, **Wed**, **Thu**, **Fri**, or **Sat**. This strategy preserves seven days of backup images of these repositories. Having seven days of backup images gives the site more restoration options in case the backups become corrupted. Substitute the name of your remote machine for **\\D2345** in this script to back up repositories to a different system.

Error Logging

On UNIX and Windows systems, errors are written to a file named **BackupRepositories.log** in the directory where **BackupRepositories.sas** was executed.

On z/OS, you must include job control language (JCL) that specifies where to write the log in the batch job that invokes %OMABAKUP—for example:

```
//SASLOG DD DSN=USERID.OMABAKUP.LOG, DISP=SHR
```

Warning Messages

The following warning messages are commonly generated by %OMABAKUP and are not cause for concern:

WARNING: Apparent invocation of macro Directory not resolved.
This warning is generated if you have a repository name that contains a '%'— for example, **my%directory**. %OMABAKUP ignores the warning and creates the backup subdirectory anyway.

WARNING: No matching members in directory.
%OMABAKUP clears the backup location prior to writing a new backup copy. This warning is generated when a backup is performed to a new or empty directory and there are no members to clear.

WARNING: Repository not found, using the first repository on the server.
%OMABAKUP copies repositories in the order they are found. This warning message is generated if the $METAREPOSITORY option is set in the host SAS session and it specifies a repository that is not first in line to be copied.

Manually Restore a Repository on an Unresponsive SAS Metadata Server

Use this procedure to restore the SAS Metadata Server if it does not respond. This procedure assumes that metadata server components such as the **omaconfig.xml** file, **adminUsers.txt** file, and **trustedUsers.txt** file exist and are intact. If this is not the case, re-install the metadata server before using this procedure.

1. Follow the steps to stop a metadata server in "Starting and Stopping the SAS Metadata Server" on page 52 to ensure the server is stopped.
2. If the SAS Metadata Server was running with auditing enabled in one or more repositories, locate the audit trails and move all audit files to other directories. The active audit directories need to be empty so that when the repositories are restored, the metadata server does not write new audit records with the old records, which are no longer valid.
3. Open the directory where the SAS Metadata Server is started. In a planned SAS installation on Windows, this is usually *SAS-config-dir***\Lev1\SASMain\MetadataServer**.
4. Use the appropriate operating system commands to delete the contents of the current **rposmgr** and repository directories.

   Windows commands:

   ```
   rmdir /S rposmgr
   rmdir /S "MetadataRepositories\dir-name"
   ```

   UNIX commands:

   ```
   rm -rf rposmgr
   rm -rf MetadataRepositories/dir-name
   ```

5. Change directories to the backup directory, and use operating system commands to copy the backup versions of the repository manager and each repository to the emptied directories.

   Windows commands:

```
cd backup-location
xcopy srcdir destdir /E /C /I /H /K /O /X /Y
```

UNIX commands:

```
cd backup-location
cp -rp srcdir destdir
```

When specifying the *srcdir* and *destdir* arguments in the copy command, remember that the backup versions of the repository manager and repositories are stored in the backup directory in subdirectories that are named after the value in each repository's **Name** attribute, and the backup files need to be copied to locations that use the repository's registered pathnames. For example, the backup copy of the repository manager is stored in a backup subdirectory named **REPOSMGR**, while the repository manager must be created in a directory named **rposmgr**. A repository that was registered with the following pathname and name exists in the backup directory in a subdirectory named **MyRepository**:

```
\MetadataServer\MetadataRepositories\Custom1 (Name="MyRepository")
```

Be sure to enter each repository's **Name** value in the *srcpath* and its directory name in the *destpath* argument.

Windows example:

```
xcopy REPOSMGR "C:\SAS\config-dir-name\Lev1\SASMain\MetadataServer\rposmgr"
  /E /C /I /H /K /O /X /Y
xcopy MyRepository "C:\SAS\config-dir-name\Lev1\SASMain\MetadataServer\
  MetadataRepositories\Custom1" /E /C /I /H /K /O /X /Y
```

UNIX example:

```
cp -rp REPOSMGR /SAS/config-dir-name/Lev1/SASMain/MetadataServer/rposmgr
cp -rp MyRepository /SAS/config-dir-name/Lev1/SASMain/MetadataServer/
  MetadataRepositories/Custom1
```

6 Use the instructions in "Starting and Stopping the SAS Metadata Server" on page 52 to start the metadata server.

7 Follow the instructions in "Enabling and Disabling Repository Auditing" on page 90 to verify that auditing is enabled with the appropriate settings.

Manually Restore a Repository on a Running SAS Metadata Server

Use this procedure to restore the repository manager and repositories if your SAS Metadata Server is responsive, but you are using a SAS software version that is earlier than SAS 9.1.3 Service Pack 4. If the metadata server does not respond, use the procedure in "Manually Restore a Repository on an Unresponsive SAS Metadata Server" on page 66.

1 Stop or pause the metadata server. Both actions halt client activity on the metadata server and unlock repository files so that a backup can be performed.
   - The Stop action halts client activity by dropping client connections and terminating the server process. The Stop action is the recommended way of halting client activity in preparation for a manual restore if you are using a SAS software version earlier than SAS 9.1.3 Service Pack 4.
   - The Pause action retains client connections and causes the metadata server to temporarily stop accepting requests.

   See "Starting and Stopping the SAS Metadata Server" on page 52 for instructions on how to stop the server.

You can pause the SAS Metadata Server by using SAS Managment Console or by issuing the METAOPERATE procedure as follows:

```
/* Example of PROC METAOPERATE PAUSE action */
PROC METAOPERATE
   SERVER="host-name"
   PORT=8561
   USERID="administrative-userID"
   PASSWORD="password"
   PROTOCOL=BRIDGE

   ACTION=PAUSE
   OPTIONS="<Repository id=""REPOSMGR"" Status=""READONLY""/>
            <Repository id=""ALL"" Status=""OFFLINE""/>"
   NOAUTOPAUSE;
RUN;
```

2 If the SAS Metadata Server was running with auditing enabled in one or more repositories, locate the audit trails and move all audit files to other directories. The active audit directories need to be empty so that when the repositories are restored, the metadata server does not write new audit records with the old records, which are no longer valid.

3 Delete the current **rposmgr** and repository directories and copy the backup copies of the repository manager and repositories by following the instructions in "Manually Restore a Repository on an Unresponsive SAS Metadata Server" on page 66.

4 Resume or restart the metadata server.

To resume the metadata server, issue the following PROC METAOPERATE call:

```
/* example of PROC METAOPERATE RESUME action */

PROC METAOPERATE
   SERVER="host-name"
   PORT=8561
   USERID="administrative-userID"
   PASSWORD="password"
   PROTOCOL=BRIDGE

   ACTION=RESUME
   OPTIONS="<Repository id=""ALL""/>
            <Repository id=""REPOSMGR""/>"
   NOAUTOPAUSE;
RUN;
```

To restart the metadata server, use the instructions in "Starting and Stopping the SAS Metadata Server" on page 52.

5 Follow the instructions in "Enabling and Disabling Repository Auditing" on page 90 to view the audit settings for each repository and verify that auditing is enabled with the appropriate settings.

Checking the SAS Metadata Server's Access State

You might want to check the access state of the SAS Metadata Server to determine whether it is running, paused, or stopped. To check the access state of the SAS Metadata Server, use PROC METAOPERATE to issue a STATUS action. The following is an example of PROC METAOPERATE statements that determine a metadata server's access state.

```
PROC METAOPERATE
  SERVER='hostname'
  PORT=port-number
  USERID='userID'
  PASSWORD='password'
  PROTOCOL=BRIDGE

  ACTION=STATUS;

RUN;
```

When the SAS Metadata Server is running normally, the server will write information similar to the following to the SAS Output window:

```
NOTE: Server dXXXX.us.company.com SAS Version is 9.1.2.
NOTE: Server dXXXX.us.company.com SAS Long Version is 9.1.2.01B0D03312003.
NOTE: Server dXXXX.us.company.com Operating System is XP_PRO.
NOTE: Server dXXXX.us.company.com Operating System Family is WINHOST.
NOTE: Server dXXXX.us.company.com Client is userid@DOMAIN.
NOTE: Server dXXXX.us.company.com Metadata Model is Version 3.02.
NOTE: Server dXXXX.us.company.com is RUNNING on 01Apr03:14:47:57.
```

When the SAS Metadata Server is paused, the server writes information similar to the following:

```
NOTE: Server dXXXX.us.company.com SAS Version is 9.1.2.
NOTE: Server dXXXX.us.company.com SAS Long Version is 9.1.2.01B0D03312003.
NOTE: Server dXXXX.us.company.com Operating System is XP_PRO.
NOTE: Server dXXXX.us.company.com Operating System Family is WINHOST.
NOTE: Server dXXXX.us.company.com Client is userid@DOMAIN.
NOTE: Server dXXXX.us.company.com Metadata Model is Version 3.02.
NOTE: Server dXXXX.us.company.com is PAUSED on 01Apr03:14:53:53.
```

A PAUSED state indicates that client activity in all repositories and the repository manager has been purposely paused by an *administrative user*. When the repository manager is paused to an offline state, users may or may not be able to write to repositories, depending upon whether or not the repository files were open previously. The repositories and repository manager will remain paused until an *administrative user* issues a Resume action to restore client activity.

A SAS Metadata Server that is stopped will not respond to a PROC METAOPERATE request.

Any user who can login to the SAS Metadata Server can issue a Status action.

Quickly Recovering Memory in the SAS Metadata Server

The SAS Metadata Server supports two approaches for quickly recovering memory on a running server:

- Issue a Pause action followed by a Resume action.
- Issue a Refresh action.

The actions can be performed on individual repositories or on the entire SAS Metadata Server.

- The Pause action instructs the SAS Metadata Server to stop accepting new client requests for the specified repositories, or for the entire metadata server, and closes repository containers.
- The Resume action resumes client activity in the specified repositories or on the entire SAS Metadata Server. Repository containers are re-opened as clients request metadata from them.
- The Refresh action pauses and resumes the specified repositories or the entire SAS Metadata Server in one step.

Pause and Resume actions can be issued in the Metadata Manager in SAS Management Console or by using PROC METAOPERATE. The Pause and Resume actions provided by SAS Management Console operate on the entire SAS Metadata Server. You cannot use SAS Management Console to pause specific repositories. To selectively pause repositories, you must use PROC METAOPERATE. The Refresh action is supported only in PROC METAOPERATE.

See the documentation for the METAOPERATE procedure in the *SAS Open Metadata Interface: Reference* for examples of procedure statements that issue Pause, Resume, and Refresh actions on specific repositories.

When pausing repositories to restore memory, it is not necessary to include options that change the repositories' access state.

Using the ARM_OMA Subsystem

Overview

Application Response Measurement (ARM) is a standard for managing distributed application performance. ARM API calls have been added to the SAS Metadata Server that return detailed information when the following events occur:

- a SAS Open Metadata Interface method is started
- certain metadata server functions are started
- an XML request is issued
- one of the following subtransactions occurs:
  - an association is added, retrieved, or removed
  - an association overwrites an existing association
  - a list of metadata objects is retrieved
- a SAS Open Metadata Interface method, function, or XML request is stopped.

The information in the ARM_OMA log is useful for performance analysis and basic method call debugging. For background information about the ARM system, see "Monitoring Performance Using Application Response Measurement (ARM)" in *SAS Language Reference: Concepts*.

Starting the ARM_OMA Subsystem

You can enable ARM logging for continous operation at server invocation, or you can enable it periodically on a running SAS Metadata Server by submitting ARM logging system options via PROC METAOPERATE.

Start the ARM_OMA Subsystem at Server Startup

To initialize the ARM_OMA subsystem at server invocation, include the following SAS system options in the SAS Metadata Server start command:

ARMSUBSYS enables and disables ARM logging. To enable ARM logging on the metadata server, specify the value ARM_OMA. To disable ARM logging, specify the value ARM_NONE.

ARMLOC specifies a location to which to write the ARM log file.

ARMAGENT specifies an ARM agent, which is an executable module that contains a vendor's implementation of the ARM API. If an agent is not specified, the default collector for the SAS Metadata Server is sasarmmg.dll.

For details about these system options, see the *SAS Language Reference: Dictionary*.

Here is an example of a metadata server start command that invokes the ARM_OMA subsystem on a Windows host:

```
"SAS-install-dir\sas.exe" -nosplash -noterminal -noautoexec
-log "MetadataServer.log" -pagesize max -memsize max
-linesize max -logparm "rollover=auto open=replaceold write=immediate"
-armsubsys "(arm_oma)" -armloc 'omaarmlog.log'
-objectserver -objectserverparms "protocol=bridge port=8561
classfactory=2887E7D7-4780-11D4-879F-00C04F38F0DB"
```

Start or Stop ARM Logging on a Running Server

The following is an example of PROC METAOPERATE statements that enable ARM logging on a running SAS Metadata Server:

```
PROC METAOPERATE
   SERVER="host-name"
   PORT=port-number
   USERID="administrative-userID"
   PASSWORD="password"

   ACTION=REFRESH
   OPTIONS="<ARM ARMSUBSYS=""(ARM_OMA)"" ARMLOC=""logs/armfile.log""/>"
   NOAUTOPAUSE;

RUN;
```

Note: The double-double quotation marks are required in the OPTIONS string. △

These statements specify to create the ARM log in a subdirectory of the **MetadataServer** directory that is called **logs**. The file is named **armfile.log**.

To disable ARM logging, execute a similar request, except specify **"(ARM_NONE)"** in the ARMSUBSYS= system option, and omit the ARMLOC system option.

Enable Other ARM Subsystems

When logging is enabled at SAS Metadata Server invocation, the ARM_DSIO and ARM_IOM subsystems can be enabled in addition to the ARM_OMA subsystem. To include information from all three subsystems in the SAS Metadata Server's ARM log, specify the ARMSUBSYS= system option as follows:

```
-armsubsys "(ARM_DSIO,ARM_IOM,ARM_OMA)"
```

The remainder of this section describes information collected by the ARM_OMA subsystem. For information about the ARM_IOM and ARM_DSIO subsystems, see the descriptions in the ARMSUBSYS= documentation.

Understanding the Records Written by the ARM_OMA Subsystem

The ARM_OMA subsystem writes records to the ARM log that collect processing information for SAS Metadata Server methods, functions, and XML requests.

Notes:

- The user ID field in ARM_OMA records is a fixed 8 characters in length and contains the last 8 characters of the requesting user's host user ID.
- The last field in records that write the *metadata-type*::*method* or *metadata-type*::*function* is limited to 32 characters; therefore, the text in this field might be truncated.
- Field 9 in P-records and in U-records that have a value is a fixed 8 characters in length. Valid values are a **0** (zero), a **1**, a **2**, or a return code. When the value is **0**, **1**, or **2**, the number is right-justified in field 9. For example:

  ```
  U,1326479606.58200,1,1,4,2.403456,3.989155,sastst ,      0,Columns::add
  ```

  When the return code is non-zero, it is expressed as hexadecimal:

  ```
  U,1326479606.58200,1,1,4,2.403456,3.989155,sastst  ,807FE835,TablePackage::get
  ```

These records are written to the ARM log:

I (initialization) record — is an initialization record. One record is written at session start, regardless of how many ARM subsystems have been armed. It starts with an I, followed by the SAS datetime value for session start, an application ID, a user start time, a system start time, an application name, and a user ID. Here is an example:

```
I,1346020250.357000,1,0.330475,0.410590,SAS,sastst
```

G (GetID) record — is a transaction ID record, with one record written to define each method, function, or XML request for which information will be logged. It starts with a G, followed by the SAS datetime value that indicates when the record was written, an application ID from the I record, a transaction class ID, the transaction name, and field definitions for the transaction description. The transaction name is written as *Type::Method*. Here is an example:

```
G,1347567099.146000,1,1,Type::Method,,User Identification,
ShortStr,Return Code,ShortStr,Type::Method,LongStr
```

S (start) record — is a start record, with one record written for each instance of an ARM transaction (OMA method call, function call, XML request). The record starts with an S, followed by the SAS datetime value that indicates when the record was written, an application ID from the I record, a transaction class ID from the G record, a transaction ID, a user start time, a system start time, a user ID, an unused field, and the metadata type name and method or function name. The metadata type, method, and function names are written as *metadata-type*::*method* or *metadata-type*::*function*. Here is an example:

```
S,1346209891.518000,1,1,18,9.283348,4.596609,sastst  ,,
request::AddMetadata
```

C (Start Correlated Transaction) record

is a start ARM subtransaction instance record, with one record written for each instance of an ARM transaction class that is started as an ARM subtransaction—for example, for a function that is launched within a method, or for a method that is launched within another method. The record starts with a C, followed by the SAS datetime value that indicates when the record was written, the application ID from the I record, the transaction class ID from the G record, a transaction ID, the parent transaction class ID from the G record, a parent transaction ID, a user start time, a system start time, a user ID, an unused field, and the metadata type name and method or function name. The metadata type, method, and function names are written as *metadata-type*::*method* or *metadata-type*::*function*. Here is an example:

```
C,1346020312.316000,1,1,3,0,0,8.412096,2.323340,sastst  ,,
RepositoryBase::omiPersistElement
```

U (update) record

is an update record. One U record is written each time a given event occurs within an ARM transaction instance. The record starts with a U, followed by the SAS datetime value that indicates when the record was written, the application ID from the I record, the transaction class ID from the G record, the transaction ID from the associated S record, a user start time, a system start time, and event-specific values. Events are the following:

- when an association is added in an UpdateMetadata request. The additional fields are a user ID, the return code after processing the association, and the association name and function name. The association name and function name are written as *association-name*::*function-name*, where *function-name* is one of **add**, **get**, **remove**, or **set**. Here is an example:

  ```
  U,1326479606.48000,1,1,4,2.403456,3.915630,sastst ,
  0,Columns::add
  ```

- when an association is added in an UpdateMetadata or AddMetadata request, to indicate whether the association was added to a new or existing object. The additional fields are a user ID, a value where **0** or **1** indicates a new element and **2** indicates an element exists, and the object type and function name **omiPersistElement**, written as *object-type***::omiPersistElement** Here is an example:

  ```
  U,1326479508.508000,1,1,2,1.612318,2.443513,sastst ,
  1,Column::omiPersistElement
  ```

- when an association is added in an UpdateMetadata or AddMetadata request, to record the object identifier of the associated object. Additional fields are a user ID, an unused field, the object type and method name, written as *metadata-type*::*method*, and either **Id=***objectID* or **ObjRef=***objectID*. Here is an example:

  ```
  U,1326479508.508000,1,1,2,1.612318,2.443513,sastst  ,,
  Column::AddMetadata,ObjRef=A9000001.A7EADO4N
  ```

- when a list of objects is retrieved by the GetMetadataObjects method. Additional fields are a user ID, the return code after

processing the request, and the object type and function name **ResumePeersAll**, written as *metadata-type*::**ResumePeersAll**. Here is an example.

```
U,1326479486.406000,1,1,1,1.321900,2.229080,sastst ,
0,Column::
ResumePeersAll
```

P (stop) record

is a stop ARM transaction instance record, with one record written upon exiting each transaction instance. It starts with a P, followed by the SAS datetime value that indicates when the record was written, the application ID from the I record, the transaction class ID from the G record, the transaction ID from the associated S record, a user start time, a system start time, a 0 (zero), a user ID, a return code, and the metadata type and method or function names. The metadata type, method, and function names are written as *metadata-type*::*method* or *metadata-type*::*function*. Here is an example:

```
P,1346209891.949000,1,1,18,9.463608,4.696753,0,sastst ,
0,request::AddMetadata
```

E (end) record

is an end record. One record is written at session end (regardless of how many ARM subsystems are active). It starts with an E, followed by the SAS datetime value that indicates when the record was written, the application ID from the I record, a user stop time, and a system stop time. Here is an example:

```
E,1326480210.737000,1,2.533643,4.257880
```

ARM_OMA Log Examples

This section contains examples of ARM records that might be written for typical SAS Open Metadata Interface method requests. Each example includes a SAS Open Metadata Interface XML request, an excerpt from the ARM_OMA log that shows the records written to describe the event, and a description of each record. SAS clients such as SAS Management Console and SAS ETL Studio issue SAS Open Metadata Interface requests internally to create repositories and to read and write metadata.

Example 1: ARM Records Created When Registering a Repository

The following example shows the ARM records created when you register a repository. SAS Open Metadata Interface clients issue an AddMetadata method call to register a repository. The following AddMetadata request creates three repositories and dependency associations between them such that the repository named scratch1 depends on the repository named scratch2, and repository scratch2 depends on repository scratch3. The request is formatted in XML and is passed to the metadata server in the **inMetadata** parameter of the DoRequest method.

```
<AddMetadata>
 <Metadata>
   <RepositoryBase Name="scratch2" Desc="Repository 2" Path="scratch2">
    <DependencyUses>
      <RepositoryBase Name="scratch1" Desc="Repository 1" Path="scratch1"/>
    </DependencyUses>
    <DependencyUsedBy>
      <RepositoryBase Name="scratch3" Desc="Repository 3" Path="scratch3"/>
```

```
    </DependencyUsedBy>
   </RepositoryBase>
 </Metadata>
 <!--Reposid specifies Repository manager -->
 <Reposid>A0000001.A0000001</Reposid>
 <Ns>REPOS</Ns>
 <!-- OMI_TRUSTED_CLIENT flag -->
 <Flags>268435456</Flags>
 <Options/>
</AddMetadata>
```

Here are the ARM_OMA log records that correspond to the request:

```
S,1346212863.241000,1,1,2,1.502160,2.153096,sastst ,,request::AddMetadata
C,1346212863.262000,1,1,3,1,2,1.512174,2.153096,sastst ,,RepositoryBase::
  AddMetadata
C,1346212863.572000,1,1,4,1,3,1.592289,2.313326,sastst ,,RepositoryBase::
  omiPersistElement
C,1346212863.572000,1,1,5,1,4,1.592289,2.313326,sastst ,,RepositoryBase::
  AddMetadata
P,1346212863.953000,1,1,5,1.632347,2.433499,0,sastst ,    0,RepositoryBase::
  AddMetadata
P,1346212863.953000,1,1,4,1.632347,2.433499,0,sastst ,    0,RepositoryBase::
  omiPersistElement
U,1346212863.983000,1,1,3,1.652376,2.443513,1,sastst ,     0,DependencyUses::
  add
C,1346212863.983000,1,1,6,1,3,1.652376,2.443513,sastst ,,RepositoryBase::
  omiPersistElement
C,1346212863.983000,1,1,7,1,6,1.652376,2.443513,sastst ,,RepositoryBase::
  AddMetadata
P,1346212864.303000,1,1,7,1.772548,2.533643,0,sastst ,     0,RepositoryBase::
  AddMetadata
P,1346212864.303000,1,1,6,1.772548,2.533643,0,sastst ,     0,RepositoryBase::
  omiPersistElement
U,1346212864.323000,1,1,3,1.782563,2.533643,1,sastst ,     0,
  DependencyUsedBy::add
P,1346212864.323000,1,1,3,1.782563,2.533643,0,sastst ,     0,RepositoryBase::
  AddMetadata
P,1346212864.433000,1,1,2,1.802592,2.563686,0,sastst ,     0,request::
  AddMetadata
```

Here are the records in the log:

- Record 1: Identifies an AddMetadata request.
- Record 2: Logs adding a RepositoryBase object (**scratch2** in the XML).
- Record 3: Logs a call to process an associated element (**scratch1** in the XML).
- Record 4: Logs AddMetadata of a second RepositoryBase object (**scratch1** in the XML).
- Record 5: AddMetadata for scratch1 got a successful return code.
- Record 6: Processing associated element got a successful return code.
- Record 7: Logs adding DependencyUses dependency for scratch1.
- Record 8: Logs call to process an associated element (**scratch3** in the XML).
- Record 9: Logs AddMetadata of a third RepositoryBase object (**scratch3** in the XML).
- Record 10: AddMetadata for scratch3 got a successful return code.

- Record 11: Processing associated element got a successful return code.
- Record 12: Logs adding DependencyUsedBy dependency for scratch3.
- Record 13: AddMetadata for scratch2 got a successful return code.
- Record 14: AddMetadata request got a successful return code.

In the output, note the following:

- All of the records store the user ID in an 8-character field.
- Records 3, 6, 8, and 11 exceed the 32-character limit for *metadata_type*::*function*.
- The U records store the return code in an 8-character field.

Example 2: ARM Records Created When Adding a Table and Columns

SAS Open Metadata Interface clients also use the AddMetadata method to create metadata objects describing application elements. The following AddMetadata request creates a PhysicalTable metadata object, an association to an existing SASLibrary metadata object, and four Column metadata objects.

```
<AddMetadata>
 <Metadata>
   <PhysicalTable Name="Sales Offices" Desc="Sales offices in NW region">
      <TablePackage>
        <SASLibrary ObjRef="$CLIENT.ADDSASLibrary$" Name="NW Sales"/>
      </TablePackage>
      <Columns>
        <Column
          Name="City"
          Desc="City of Sales Office"
          ColumnName="City"
          SASColumnName="City"
          ColumnType="12"
          SASColumnType="C"
          ColumnLength="32"
          SASColumnLength="32"
          SASFormat="$Char32."
          SASInformat="$32."/>
        <Column
          Name="Address"
          Desc="Street Address of Sales Office"
          ColumnName="Address"
          SASColumnName="Street_Address"
          ColumnType="12"
          SASColumnType="C"
          ColumnLength="32"
          SASColumnLength="32"
          SASFormat="$Char32."
          SASInformat="$32."/>
        <Column
          Name="Manager"
          Desc="Name of Operations Manager"
          ColumnName="Manager"
          SASColumnName="Manager"
          ColumnType="12"
          SASColumnType="C"
          ColumnLength="32"
```

```
          SASColumnLength="32"
          SASFormat="$Char32."
          SASInformat="$32."/>
        <Column
          Name="Employees"
          Desc="Number of employees"
          ColumnName="Employees"
          SASColumnName="Employees"
          ColumnType="6"
          SASColumnType="N"
          ColumnLength="3"
          SASColumnLength="3"
          SASFormat="3.2"
          SASInformat="3.2"/>
      </Columns>
   </PhysicalTable>
 </Metadata>
 <Reposid>$CLIENT.ADDRepositoryBase$ </Reposid>
 <NS>SAS </NS>
 <Flags>268435456</Flags>
 <Options/>
</AddMetadata>
```

Here are the ARM_OMA log records that correspond to the request:

```
S,1346701949.827000,1,1,10,8.632412,2.193153,sastst ,,request::AddMetadata
C,1346701949.877000,1,1,11,1,10,8.662456,2.193153,sastst ,,PhysicalTable::
  AddMetadata
C,1346701949.917000,1,1,12,1,11,8.692499,2.203168,sastst ,,SASLibrary::
  omiPersistElement
P,1346701949.937000,1,1,12,8.702513,2.203168,0,sastst ,    0,SASLibrary::
  omiPersistElement
U,1346701950.7000,1,1,11,8.772614,2.203168,1,sastst ,     0,TablePackage::set
C,1346701950.7000,1,1,13,1,11,8.772614,2.203168,sastst ,,Column::
  omiPersistElement
C,1346701950.37000,1,1,14,1,13,8.782628,2.203168,sastst ,,Column::AddMetadata
C,1346701950.97000,1,1,15,1,14,8.832700,2.203168,sastst ,,PhysicalTable::
  omiPersistElement
C,1346701950.117000,1,1,16,1,15,8.852729,2.203168,sastst ,,*::
  omiVerifyAssociation
P,1346701950.127000,1,1,16,8.862744,2.203168,0,sastst ,     2,*::
  omiVerifyAssociation
P,1346701950.127000,1,1,15,8.862744,2.203168,0,sastst ,     0,PhysicalTable::
  omiPersistElement
U,1346701950.177000,1,1,14,8.912816,2.203168,1,sastst ,     0,Table::set
P,1346701950.177000,1,1,14,8.912816,2.203168,0,sastst ,     0,Column::
  AddMetadata
P,1346701950.177000,1,1,13,8.912816,2.203168,0,sastst ,     0,Column::
  omiPersistElement
C,1346701950.177000,1,1,17,1,11,8.912816,2.203168,sastst ,,Column::
  omiPersistElement
C,1346701950.187000,1,1,18,1,17,8.922830,2.203168,sastst ,,Column::AddMetadata
C,1346701950.207000,1,1,19,1,18,8.942859,2.203168,sastst ,,PhysicalTable::
  omiPersistElement
C,1346701950.227000,1,1,20,1,19,8.962888,2.203168,sastst ,,*::
```

```
  omiVerifyAssociation
P,1346701950.238000,1,1,20,8.972902,2.203168,0,sastst ,     2,*::
  omiVerifyAssociation
P,1346701950.238000,1,1,19,8.972902,2.203168,0,sastst ,     0,PhysicalTable::
  omiPersistElement
U,1346701950.288000,1,1,18,9.22974,2.203168,1,sastst ,     0,Table::set
P,1346701950.288000,1,1,18,9.22974,2.203168,0,sastst ,     0,Column::
  AddMetadata
P,1346701950.288000,1,1,17,9.22974,2.203168,0,sastst ,     0,Column::
  omiPersistElement
C,1346701950.288000,1,1,21,1,11,9.22974,2.203168,sastst ,,Column::
  omiPersistElement
C,1346701950.298000,1,1,22,1,21,9.32988,2.203168,sastst ,,Column::AddMetadata
C,1346701950.318000,1,1,23,1,22,9.53017,2.203168,sastst ,,PhysicalTable::
  omiPersistElement
C,1346701950.338000,1,1,24,1,23,9.73046,2.203168,sastst ,,*::
  omiVerifyAssociation
P,1346701950.348000,1,1,24,9.83060,2.203168,0,sastst ,     2,*::
  omiVerifyAssociation
P,1346701950.348000,1,1,23,9.83060,2.203168,0,sastst ,     0,PhysicalTable::
  omiPersistElement
U,1346701950.408000,1,1,22,9.143147,2.203168,1,sastst ,     0,Table::set
P,1346701950.408000,1,1,22,9.143147,2.203168,0,sastst ,     0,Column::
  AddMetadata
P,1346701950.408000,1,1,21,9.143147,2.203168,0,sastst ,     0,Column::
  omiPersistElement
C,1346701950.408000,1,1,25,1,11,9.143147,2.203168,sastst ,,Column::
  omiPersistElement
C,1346701950.418000,1,1,26,1,25,9.153161,2.203168,sastst ,,Column::AddMetadata
C,1346701950.438000,1,1,27,1,26,9.173190,2.203168,sastst ,,PhysicalTable::
  omiPersistElement
C,1346701950.458000,1,1,28,1,27,9.183204,2.213182,sastst ,,*::
  omiVerifyAssociation
P,1346701950.468000,1,1,28,9.193219,2.213182,0,sastst ,     2,*::
  omiVerifyAssociation
P,1346701950.468000,1,1,27,9.193219,2.213182,0,sastst ,     0,PhysicalTable::
  omiPersistElement
U,1346701950.528000,1,1,26,9.243291,2.223196,1,sastst ,     0,Table::set
P,1346701950.528000,1,1,26,9.243291,2.223196,0,sastst ,     0,Column::
  AddMetadata
P,1346701950.528000,1,1,25,9.243291,2.223196,0,sastst ,     0,Column::
  omiPersistElement
P,1346701950.528000,1,1,11,9.243291,2.223196,0,sastst ,     0,PhysicalTable::
  AddMetadata
P,1346701950.999000,1,1,10,9.493651,2.283283,0,sastst ,     0,request::
  AddMetadata
```

Here are the records in the log:

- Record 1: Identifies an AddMetadata request.
- Record 2: Logs adding a PhysicalTable object (**`Sales Offices`** in the XML).
- Record 3: Logs call to persist associated SASLibrary element (**`NW SALES`** in the XML).
- Record 4: Persist of associated SASLibrary element got a successful return code **`0`**.
- Record 5: Setting TablePackage association got a successful return code **`0`**.

- Record 6: Logs call to persist associated Column element (**City** in the XML).
- Record 7: Logs AddMetadata of a new Column object (**City** in the XML).
- Record 8: Logs call to persist associated PhysicalTable for required association.
- Record 9: Logs call to omiVerifyAssociation to check for required association.
- Record 10: Return code (**2**) from omiVerifyAssociation indicates no existing association.
- Record 11: Persist of associated PhysicalTable element got a successful return code **0**.
- Record 12: Setting required Table association got a successful return code **0**.
- Record 13: AddMetadata for Column **City** got a successful return code **0**.
- Record 14: Persist of associated Column **City** got a successful return code **0**.
- Record 15-23: Same as records 6-14 for Column **Address**.
- Record 24-32: Same as records 6-14 for Column **Manager**.
- Record 33-41: Same as records 6-14 for Column **Employees**.
- Record 42: AddMetadata for PhysicalTable **Sales Office** got a successful return code **0**.
- Record 43: AddMetadata request got a successful return code **0**.

In a multi-threaded environment, the ARM records describing one OMA transaction might be interleaved with records describing other OMA transactions in the ARM_OMA log. When this occurs, use the application identifier, ARM transaction class identifier, and ARM transaction identifiers to keep track of transactions. As an example, here are the first 14 ARM records from the previous example:

```
S,1346701949.827000,1,1,10,8.632412,2.193153,sastst ,,request::AddMetadata
C,1346701949.877000,1,1,11,1,10,8.662456,2.193153,sastst ,,PhysicalTable::
  AddMetadata
C,1346701949.917000,1,1,12,1,11,8.692499,2.203168,sastst ,,SASLibrary::
  omiPersistElement
P,1346701949.937000,1,1,12,8.702513,2.203168,0,sastst ,     0,SASLibrary::
  omiPersistElement
U,1346701950.7000,1,1,11,8.772614,2.203168,1,sastst ,     0,TablePackage::set
C,1346701950.7000,1,1,13,1,11,8.772614,2.203168,sastst ,,Column::
  omiPersistElement
C,1346701950.37000,1,1,14,1,13,8.782628,2.203168,sastst ,,Column::AddMetadata
C,1346701950.97000,1,1,15,1,14,8.832700,2.203168,sastst ,,PhysicalTable::
  omiPersistElement
C,1346701950.117000,1,1,16,1,15,8.852729,2.203168,sastst ,,*::
  omiVerifyAssociation
P,1346701950.127000,1,1,16,8.862744,2.203168,0,sastst ,     2,*::
  omiVerifyAssociation
P,1346701950.127000,1,1,15,8.862744,2.203168,0,sastst ,     0,PhysicalTable::
  omiPersistElement
U,1346701950.177000,1,1,14,8.912816,2.203168,1,sastst ,     0,Table::set
P,1346701950.177000,1,1,14,8.912816,2.203168,0,sastst ,     0,Column::
  AddMetadata
P,1346701950.177000,1,1,13,8.912816,2.203168,0,sastst ,     0,Column::
  omiPersistElement
```

The following is a description of how the ARM records are nested:

Record 1 — Logs the start of an ARM_OMA transaction. The third parameter (**1**) is the application identifier, the fourth (**1**) is the ARM transaction class identifier, and the fifth (**10**) is the ARM transaction identifier.

Record 2	Logs the start of an ARM_OMA subtransaction **11** (the fifth parameter). The sixth parameter (**1**) is the parent transaction class identifier, and the seventh (**10**) is the parent transaction identifier.
Record 3	Logs the start of an ARM_OMA subtransaction **12** (the fifth parameter). The sixth parameter (**1**) is the parent transaction class identifier, and the seventh (**11**) is the parent transaction identifier.
Record 4	Logs completion of an ARM_OMA subtransaction **12** (the fifth parameter).
Record 5	Logs detail information.
Record 6	Logs the start of an ARM_OMA subtransaction **13** (the fifth parameter). The sixth parameter (**1**) is the parent transaction class identifier, and the seventh (**11**) is the parent transaction identifier.
Record 7	Logs the start of an ARM_OMA subtransaction **14** (the fifth parameter). The sixth parameter (**1**) is the parent transaction class identifier, and the seventh (**13**) is the parent transaction identifier.
Record 8	Logs the start of an ARM_OMA subtransaction **15** (the fifth parameter). The sixth parameter (**1**) is the parent transaction class identifier, and the seventh (**14**) is the parent transaction identifier.
Record 9	Logs the start of an ARM_OMA subtransaction **16** (the fifth parameter). The sixth parameter (**1**) is the parent transaction class identifier, and the seventh (**15**) is the parent transaction identifier.
Record 10	Logs completion of an ARM_OMA subtransaction **16** (the fifth parameter).
Record 11	Logs completion of an ARM_OMA subtransaction **15** (the fifth parameter).
Record 12	Logs detail information.
Record 13	Logs completion of an ARM_OMA subtransaction **14** (the fifth parameter).
Record 14	Logs completion of an ARM_OMA subtransaction **13** (the fifth parameter).

Post-Processing the ARM_OMA Log

SAS provides macros for post-processing an ARM log. For more information, see "SAS ARM Macros" in the *SAS Language Reference: Dictionary*.

SAS Metadata Server Troubleshooting

Overview of Troubleshooting the SAS Metadata Server

There are two common problems you might encounter when using a SAS Metadata Server:

- The SAS Metadata Server will not start.
- Clients cannot connect to the SAS Metadata Server.

The sections that follow make troubleshooting recommendations for these problems.

For help with diagnosing user dialog error messages for a running server, consult the SAS Metadata Server log. Error messages from the I/O subsystem are prefaced with the statement "The I/O subsystem has returned error message *message-text*" and are followed by another message that describes the corresponding error event on the SAS Metadata Server.

Repository containers are SAS data sets. If you suspect that your repository containers are damaged, see "Repairing Damaged SAS Files" in the *SAS Language Reference: Concepts*.

How to Respond When the SAS Metadata Server Will Not Start

Check for a SAS Log File

When the metadata server fails to initialize, first check to see if a SAS log file exists.

- If the SAS log does not exist, one of the following statements is true:
  - The LOG parameter was omitted from the metadata server start command, or the destination filename does not match the name of the destination directory.
  - APPLEVEL=0 was specified in the metadata server start command. For a description of APPLEVEL values, see "Enable the Logging of Authentication Events" on page 32.
  - SAS could not initialize. See "Check for SAS Initialization Failure" on page 81.
- If the SAS log file exists, open it and check the text for a problem description. The SAS log file contains error messages that you can use to diagnose and correct initialization problems. Successful initialization is indicated by a message similar to the following:

  ```
  2003066:16.26.41.48: 00000005: Defined server listen connection (1) on
  port 7991.
  ```

  A failed initialization includes this message in the log:

  ```
  2003069:17.43.13.88: 00000005:ERROR: The SAS Metadata Supervisor failed to
  initialize properly because of the errors noted above. Contact Technical
  Support for assistance if these messages do not indicate a solution
  to the problem.
  ```

  Follow the instructions in the error messages. If the error messages indicate a problem in the **omaconfig.xml** file, open the file in an Internet browser. If you see the contents of the file with no other messages, then no syntax errors were found.
- If the file exists but is empty, or you do not understand the error messages, contact SAS Technical Support for help in diagnosing the problem.

Check for SAS Initialization Failure

Answer the following questions to identify problems with initialization.

Were the start-up parameters entered correctly?

Check the parameters in your start-up command against the information in "Options for the Metadata Server Invocation Command: Reference" on page 46 to be sure that the start-up parameters were entered correctly.

As a second verification, re-execute the start-up command, omitting the NOTERMINAL system option. When the log file cannot be created, SAS diagnostic

messages are written to the SAS Console Log. The NOTERMINAL option suppresses the SAS Console Log so that it does not interfere with server processing. The SAS Console Log will report any problems with server start-up parameters, if there are any.

If an error is found, edit the metadata server startup command to correct the problem and to add back the NOTERMINAL option, and then restart the server. See the companion documentation for your host environment for more information about the SAS Console Log.

Does the server invoker have Write permission to the server directory?

1. Using your favorite text editor, create a file named **verify.sas** in the server directory.
2. In the file, type this single line:

```
proc options;run;
```

If you are able to create the file, you have the necessary directory permissions.

Are you able to initialize SAS?

At an operating system prompt, issue this command:

```
sas -sysin verify
```

If a file named **verify.log** is created and that file's contents show no errors, then SAS initialized successfully. If the file is not created, error messages will indicate why it could not be created. Follow the instructions in the error messages to correct the problem.

Is the port number you are using to bring up the server in use by another TCP/IP process?

Some UNIX and Windows machines have a NETSTAT command that can be used to identify what ports are in use. You also can check your SERVICES file to see what port numbers have already been assigned to other processes. Just because a port number is in the file does not mean it is in use, but it could mean the specified process can use it at some time. Edit your server start-up command to use a port number not mentioned in the SERVICES file.

If you followed these steps and the server still is not working, then SAS initialization is not the problem.

How to Respond When a Client Cannot Connect to SAS Metadata Server

If the server is running and an attempt by a client to connect fails, then there could be several possible problems. Answer the following questions to identify the problem.

Does the server accessor have a valid user ID and password on the metadata server?

The SAS Metadata Server supports several authentication options. The specified user ID must meet the configured authentication provider's requirements and (if host authentication is used) the requirements of the host environment. Note that the value **localhost** cannot be used to indicate the domain in a domain-qualified user ID.

Do the host name and port number that you are using to connect to the server match the server host and the port number specified in the server start-up command?

A metadata server is identified by its host name and port number. Check the connection parameters to make sure that they are correct.

Does the server invoker user ID have the right system access permissions?

All customers should be running secure servers, which require the server invoker to have documented additional user rights such as **Act as part of the operating system**. Determine if the correct user rights have been set. Then, restart the computer to ensure that the permissions have taken effect.

Does the server accessor user ID have the right system access permissions?

Some host environments require server accessors to have the **`Log on as a batch job`** user right set. Ensure the appropriate user rights are set and make any necessary corrections. Note that if you change the user rights, then you will have to restart the computer for them to take effect.

If you are connecting to a server under UNIX, was SAS set up to run as root?

The SAS Metadata Server and many other SAS processes require that some files in the **`!SASROOT/utilities/bin`** directory be setuid and owned by root. To ensure that the appropriate permissions are set, invoke the setup utility while logged in as root. Then, access the Configure User Authorization utility from the **`Run Setup Utilities`** menu to set the permissions on the necessary files.

If you are connecting to a server under Windows, are you specifying a domain-qualified user ID to connect to the server?

For user accounts that are authenticated by a domain controller, the server must see a domain qualifier on the user ID. A network domain can be specified as *domain\userID* or *userID@domain*. For a local user ID, you use the machine name as the domain qualifier—for example, *machine-name\userID*. Depending on the interfaces being used, you might have to enter more than one backslash in order for the server to receive a single backslash. A double-backslash typically is not required to connect from SAS Management Console.

CHAPTER

5

Maintaining SAS Metadata Repositories

Overview of Maintaining SAS Metadata Repositories

Administrative and programmer interfaces are available for performing most repository maintenance tasks. For example, you can use the Metadata Manager in SAS Management Console or PROC METAOPERATE to perform most repository maintenance tasks.

Unless otherwise noted, you must be an *administrative user* or *unrestricted user* on the SAS Metadata Server in order to perform server maintenance tasks. You are an

unrestricted user if you connected to the metadata server using the sasadm user ID. You are an *administrative user* if your user ID is listed in the **adminUsers.txt** file in the **MetadataServer** directory. For more information about these special users, see "Defining Administrative, Unrestricted, and Trusted Users" on page 29.

Repository maintenance tasks include those discussed in the following sections:

Creating Repositories

SAS Management Console is the preferred method for creating repositories because this tool also creates default authorization metadata in the repository. This default authorization metadata includes a Default Access Control Template (Default ACT) that you can use to restrict user access to the repository. SAS Management Console also creates resource templates to assist you in defining global metadata, such as metadata that describes users, groups, and servers. Repositories created by a tool other than SAS Management Console must be opened and initialized in SAS Management Console to obtain this default metadata.

SAS Management Console provides an Add Repository wizard to step you through the repository creation process. To start the wizard, open the Metadata Manager, right-click the **Active Server**, and select **Add Repository** from the pop-up menu. For information about how to run the wizard, see the wizard's Help.

By default, repositories are created using the SAS Base engine. If you have the appropriate SAS/ACCESS software installed, you can also create repositories in Oracle and IBM DB2 databases. SAS/ACCESS and database software requirements are described in "Creating the Metadata Repository in a Database Management System" on page 36.

For best results when creating a repository, follow these guidelines:

- Use operating system commands to create your repository directories in advance; SAS Management Console will not create them for you. A repository directory can be located anywhere on the network as long as the SAS Metadata Server can access it. The only requirement is that each repository have a path that does not include the path to the repository manager. That is, a repository directory should not be nested within the **rposmgr** directory.

  If you create repository directories within the **MetadataServer/MetadataRepositories** directory created for you by the planned installation, the repositories will inherit any file and directory permissions that you defined for the **MetadataServer** directory. If you create the repositories elsewhere, you will need to set file and directory access permissions for each repository.

- When registering a repository, use a relative pathname for the repository location. The use of a relative pathname makes a repository easier to copy and move. A SAS Metadata Server that was set up by the planned SAS installation must have

its repositories identified relative to the **SASMain** directory—for example, **MetadataServer/MetadataRepositories/***repository-name*.

- Repository names must be unique.
- A repository directory should not contain any files that are not part of the repository.
- Do not overwrite an old repository with a new one.

Note: The metadata server must add, update, and delete repositories when no other activity is taking place on the server, so it automatically delays other client requests until the repository changes are complete. This delay might have a small effect on server performance. △

Deleting Repositories

There might be times when it is desirable to clear the contents of a repository. SAS Management Console and PROC METAOPERATE support the following actions for removing metadata from a repository:

DELETE — destroys a repository. DELETE deletes all metadata and metadata containers and removes the repository's registration from the repository manager.

REINITIALIZE — deletes the contents of a repository, but enables you to continue using the repository. REINITIALIZE deletes the metadata in a repository, but does not remove the repository's registration from the repository manager. In SAS Management Console, this option is called FORMAT.

Neither action affects a repository's audit files. In SAS 9.1, a repository's audit files are deleted independently of the repository.

Checking a Repository's Access State

A SAS metadata repository can exist in one of three states: full access, read-only, and offline. A repository's intended access mode is set when the repository is created and is recorded in the repository's **Access** attribute. For example, if a repository was configured to be read-only at creation, then it will have an **Access** value of **OMS_READONLY**. The default access mode is full access.

A repository's access mode can be temporarily changed to a more restrictive state by issuing a Pause action using PROC METAOPERATE. For example, a repository that was created with an access mode of full access can be paused to a read-only or offline state. A repository that was created with an access mode of read-only can be paused to an offline state; it cannot be paused to full access.

There are only a few reasons to change the access state of a repository. One reason is that you are about to move a repository. For information on this subject, see "Moving a Repository to a New Location on the Same SAS Metadata Server" on page 93. Another reason is that you are about to perform a selective restore of a repository. See "Backing Up and Restoring the SAS Metadata Server" on page 53. A third is that you need to quickly recover memory on the SAS Metadata Server. See "Quickly Recovering Memory in the SAS Metadata Server" on page 69 for information.

Repositories are most often paused to another state by the internal process of an application that needs temporary exclusive use of a repository, such as the

%OMABAKUP macro. In this case, the application automatically restores the repository to its intended access mode when the process is complete.

If one or more repositories become unavailable on the SAS Metadata Server, it is a good idea to check their access states to see if a process has the repositories paused. To check the access state of your repositories, use PROC METADATA to issue a SAS Open Metadata Interface GetRepositories method call that sets the OMI_ALL flag. The following code provides an example of PROC METADATA statements that issue such a GetRepositories method call (the OMI_ALL flag is set by specifying a **1** in the **<Flags>** element):

```
PROC METADATA
    SERVER="hostname"
    PORT=port-number
    USERID="your-user-ID"
    PASSWORD="your-password"

    IN="<GetRepositories>
        <Repositories/>
        <Flags>1</Flags>
        <Options/>
        </GetRepositories>";
    RUN;
```

The method will return the **id**, **Name**, **Desc**, **Access**, and **PauseState** attributes of all repositories that are registered on the SAS Metadata Server.

- The **id** attribute specifies the repository's unique object identifier.
- The **Name** attribute specifies the name assigned to the repository.
- The **Desc** attribute contains a description of the repository.
- The **Access** attribute indicates the access mode defined when the repository was created. Valid values are **OMS_FULL**, indicating full access, and **OMS_READONLY**, indicating read-only access. A repository cannot be registered in an offline mode.
- The **PauseState** attribute indicates whether a repository is paused and its temporary access state. Valid values are an empty string, **READONLY**, and **OFFLINE**. An empty string means that the repository is not paused.

Information similar to the following will be displayed in the SAS Output window:

```
<GetRepositories>
<Repositories>
 <Repository id="A0000001.A5HZY944" Name="Test Repository 1"
   Desc="First repository created for GetRepos
   test" DefaultNS="SAS" Access="OMS_READONLY" PauseState=""/>
 <Repository id="A0000001.A5G3R7J5" Name="Test Repository 2"
   Desc="Second repository created for GetRepos
   test" DefaultNS="SAS" Access="OMS_FULL" PauseState="READONLY"/>
 <Repository id="A0000001.A5IDMA0U" Name="Test Repository 3"
   Desc="Third repository created for GetRepos
   test" DefaultNS="SAS" Access="OMS_FULL" PauseState=""/>
 <Repository id="A0000001.A0000001" Name="REPOSMGR" Desc="The Repository Manager"
    DefaultNS="REPOS" Access="OMS_FULL" PauseState=""/>
</Repositories>
<Flags>1</Flags>
<Options/>
</GetRepositories>
```

In the example, the output means:

- Test Repository 1 is not paused. It was created with an access mode of read-only. Therefore, it is partially available by design.
- Test Repository 2 is paused to a read-only state; therefore, it is only partially available. Its normal access mode is full access.
- Test Repository 3 is available.
- REPOSMGR (which represents the repository manager) is available.

PROC METAOPERATE enables you to change the access state of the repository manager in addition to that of repositories. An access state of read-only or offline for the repository manager is typically reflected in the following behavior:

Read-only	Clients can read and write metadata in repositories, but they cannot create or delete repositories or change their registrations.
Offline	The behavior can vary depending on whether a repository is available in memory. Assume that repositories are unavailable for reading or writing to all users.

CAUTION:
Do not change the access state of the repository manager to offline unless all repositories are taken offline as well. △

For more information about PROC METADATA, see the documentation for the METADATA procedure in the *SAS Open Metadata Interface: Reference*. The GetRepositories method is also documented in the *SAS Open Metadata Interface: Reference*.

Note: Someone who is not an *administrative user* can list repositories using the GetRepositories method if they have ReadMetadata permission to all repositories on the SAS Metadata Server. △

Invoking Repository Auditing

Overview of Repository Auditing

The SAS Metadata Server supports saving information about changes to a repository in a set of audit files. Auditing is performed on individual repositories and is invoked in SAS Management Console after a repository is created. The audit files, referred to as an *audit trail*, provide a record of changes that you can view and use to recover individual metadata records from the repository if they are accidentally deleted or logically damaged.

You can choose which metadata records to audit:

- deleted metadata records only
- new metadata records only
- updated metadata records only
- all metadata records (added, deleted, and updated).

CAUTION:
Complete recovery of a repository from the audit trail should not be attempted unless all metadata record transactions are audited. △

In the current release, the availability of the audit trail is limited as follows:

- You can view and recover audit records from Windows hosts.
- You can view audit records on UNIX hosts, but you cannot recover them.

- You cannot view or recover audit records on z/OS (OS/390) hosts.
- You must recover audit records individually using SAS processes. Automated recovery of large amounts of metadata is not available.
- Audit records are created by the SAS Base engine, whether the repositories themselves were created in SAS data sets, Oracle, or DB2.

The steps for viewing and restoring an audit record on Windows systems are documented in "Restoring a Metadata Record on Windows Hosts" in "Invoking a Repository Audit Trail" in the *SAS Open Metadata Interface: User's Guide*. Additional steps are required to view audit records from UNIX hosts. Contact Technical Support for this information.

In the current release, it is best to use your system backups to restore repositories. A system backup is a complete copy of all repositories on the metadata server and the dependencies defined among them. You can make a system backup by using the %OMABAKUP autocall macro. Instructions for using this macro are provided in "Backing Up and Restoring the SAS Metadata Server" on page 53. It is recommended that you execute this macro on a daily basis. Note the instructions for handling audit files before and after a restore.

Enabling and Disabling Repository Auditing

You enable and disable repository auditing in the Metadata Manager in SAS Management Console. Both tasks are performed on an existing repository. You enable repository auditing to begin storing records of metadata add, update, and delete transactions. You might disable repository auditing before restoring a repository from a backup, depending on the restore approach taken. After a restore, you should retire the old audit trail and start a new audit trail.

To enable repository auditing, perform these steps:

1. In the Metadata Manager, expand the **Active Server** node. The software will list all repositories that are registered on the SAS Metadata Server in the navigation tree.
2. Highlight a repository with your mouse, and select **Actions ▶ Audit** from the menu bar. The software will open an Audit Properties dialog box.
3. In the Audit Properties dialog box, perform these steps:
   a. Check **Enable Auditing**.
   b. Select a type from the **Audit Type** drop-down list. See the Help for a description of each type.
   c. Specify a location to write the audit trail in the **Audit Path** field.
   d. Click **OK** to save your selections and close the Audit Properties dialog box.

To disable repository auditing, perform these steps:

1. In the Metadata Manager, expand the **Active Server** node.
2. Highlight a repository in the navigation tree, and select **Actions ▶ Audit** from the menu bar.
3. In the Audit Properties dialog box, and perform these steps:
   a. Clear the **Audit Path** field.
   b. Clear the **Enable Auditing** check box.
   c. Click **OK** to save your selections and close the Audit Properties dialog box.

If you are disabling repository auditing in preparation for a server restore, repeat these steps for all repositories that are registered on the SAS Metadata Server.

Copying, Promoting, Importing, and Exporting Your Metadata

Tools for Copying, Promoting, Importing, and Exporting Metadata

The SAS Intelligence Platform includes several tools to assist you in copying, promoting, importing, and exporting your metadata. Some tools apply to an entire metadata repository, while others apply to specific types of metadata. Some of the tools are useful in promotion scenarios—for example, when you are creating a new development environment based on an existing production environment, or when you are moving objects from a test environment to a production environment.

The tools that are available include the following:

- Metadata Manager Promotion Wizard and Replication Wizard in SAS Management Console (to copy an entire metadata repository)
- import, export, copy, and paste features in SAS Management Console and in SAS Data Integration Studio (to copy selected objects within a repository or from one repository to another)
- Metadata Manager Import and Export Wizards in SAS Management Console (to copy metadata for relational data and DBMSs to and from external sources)
- SAS Data Integration Studio Import and Export Wizards (to copy metadata for relational data and DBMSs to and from external sources with change analysis)
- %OMAPORT macro (to port a metadata repository from one operating system environment to another)

The wizards are described briefly in the following paragraphs. For information about the %OMAPORT macro, see "Porting a Repository to a Different Host Environment" on page 94

About the Promotion Wizard and the Replication Wizard in SAS Management Console

The Promotion Wizard and the Replication Wizard are features of SAS Management Console that enable you to copy an entire metadata repository from one host machine to another. The two wizards are identical, except that the Promotion Wizard enables you to make modifications to the metadata as you copy it. For example, you can change port numbers, host names, schema names, and path names. You can save the promotion or replication process as a job and rerun the process at any time without having to redefine the process parameters.

The Promotion and Replication wizards completely replace the metadata that exists in the target environment. The wizards copy only metadata; you must use other methods to copy the associated content.

The Promotion and Replication wizards enable you to replicate or promote a repository between hosts in the same operating environment. For example, you can replicate a repository from a Windows host to another Windows host, a Solaris host to another Solaris host, or an HP-UX host to another HP-UX host. However, you cannot replicate a repository from a Windows host to a Solaris host or from a Solaris host to a z/OS host.

The Promotion and Replication wizards are located in the Metadata Manager **`Job Definitions`** folder. For detailed information, see Chapter 6, "Using the Promotion and Replication Wizards to Copy an Entire Metadata Repository," on page 101.

About the Import, Export, Copy, and Paste Features in SAS Management Console and SAS Data Integration Studio

SAS Data Integration Studio 3.4 and SAS Management Console provide the following set of common features for exporting, importing, copying, and pasting metadata in the SAS Open Metadata Architecture format:

- an Export Wizard and an Import Wizard that enable you to selectively promote individual objects (including data explorations, documents, external files, folders, generated transformations, information maps, jobs, libraries, mining results, notes, reports, tables, and stored processes) from one metadata repository to another. For most object types, the wizards can export and import both the content and the metadata for the selected objects. For example, when you export metadata for reports, the Export Wizard also exports the associated XML (.srx) files. The Import Wizard enables you to establish connections with the appropriate resources (such as servers) in the destination repository.

  For detailed information, see Chapter 7, "Promoting Individual BI Objects," on page 133 and the Help for the Export Wizard and Import Wizard.

- the ability to copy individual objects (including data explorations, documents, external files, folders, generated transformations, information maps, jobs, libraries, mining results, notes, reports, tables, and stored processes) from one location in the metadata repository and paste them into another location in the same repository. For most object types, the Paste Special feature enables you to also copy the content that is associated with the objects and to establish connections with resources.

  For detailed information, see the Help for the Paste Special Wizard.

About the Metadata Manager Import and Export Wizards in SAS Management Console

The Metadata Manager Import and Export Wizards enable you to copy metadata for relational data and DBMSs to and from external sources. You can import and export the following types of metadata objects: DatabaseSchema, PhysicalTable, Column, Column:type, UniqueKey, ForeignKey (and associated KeyAssociation), and Index.

This feature is useful when you have created a data model using other data warehousing software, and you want to import it for use with SAS; or when you have created a data model in SAS, and want to export it for use with other data warehousing software.

To access this feature, navigate to the **`Metadata Manager`** node in SAS Management Console. Expand the **`Active Server`** node under **`Metadata Manager`**, and then right-click a repository name. On the menu that appears, you can select **`Import Metadata`** or **`Export Metadata`**.

By default, Metadata Manager imports and exports metadata in the Common Warehouse Metamodel (CWM) format. Additional import and export formats are supported if you install the Model Bridge software from Meta Integration Technology, Inc. You can download Model Bridge from Meta Integration Technology's Web site, **`http:\\www.metaintegration.net`**. In order to access the software, you need to install the JAR files in the **`plugins`** directory and reboot the computer on which the software is installed.

The following table maps the SAS metadata types to the corresponding metadata types in the Common Warehouse Model.

Table 5.1 Metadata Type Conversions

Common Warehouse Model Type	SAS Metadata Type
CWMRDB:Schema	DatabaseSchema
CWMRDB:BaseTable	PhysicalTable
CWMRDB:Table	
CWMRDB:View	
CWMRDB:Column	Column
CWMRDB:SQLDistinctType	Column:type
CWMRDB:SQLSimpleType	
CWMRDB:PrimaryKey	UniqueKey
CWMRDB:ForeignKey	ForeignKey (and associated KeyAssociation)
CWMRDB:SQLIndex	Index

Notes:

1. The wizards import metadata to, and export metadata from, a single SAS Metadata Repository; they do not support repository dependencies.
2. Error messages generated by the wizards are written to the directory where SAS Management Console is installed in a file named **`errorlog.txt`**.

See the *SAS Management Console: User's Guide* for instructions on how to use the wizards.

About the SAS Data Integration Studio Import and Export Wizards (for External Data Sources)

If you have SAS Data Integration Studio 3.3 or higher, then you can use that product's enhanced Import and Export wizards to copy metadata for relational data and DBMSs to and from external sources.

These wizards provide a change analysis capability that enables you to compare the imported metadata to existing metadata in the repository and selectively apply changes. For detailed information, see the product Help for SAS Data Integration Studio.

Moving a Repository to a New Location on the Same SAS Metadata Server

The server supports two approaches for moving a repository to a new location while keeping its registration on the same SAS Metadata Server, depending on which is more important, repository availability or disk space.

The first approach enables clients to continue to read the metadata until the repository is ready to be registered in the new location. However, the host machine must have enough disk space to temporarily hold two copies of the repository. You should use this approach when you are moving foundation repositories and other repositories on which child repositories depend. Perform these steps:

1. Pause client activity in the repository that you want to move to a read-only state. (In the current release, the pausing of individual repositories is supported only in PROC METAOPERATE.)

2 Use host commands to copy the contents of the repository to a new location.

3 When the files are in the new location, use the tools in the SAS Management Console Metadata Manager to delete the repository in the old location and to register it in the new location.

   - To delete a repository, highlight it in the server's navigation tree and select **Actions ▶ Delete**. The Delete action deletes the contents of a repository and removes the repository's registration from the repository manager.
   - To register a repository, highlight the active-server icon, and select **Actions ▶ Add Repository**. When you are prompted for the repository path, specify the repository's new location. The SAS Metadata Server will read the existing files in the specified location to create the new repository's registration.

The second approach does not require that you create a copy of the repository. However, this approach will make the repository temporarily unavailable to clients. Perform these steps:

1 In the Metadata Manager, expand the **`Active Server`** node.

2 Right-click the repository that you want to move, and select **`Unregister`** from the pop-up menu.

3 Use host commands to move the contents of the repository to a new location.

4 When the files are in the new location, re-register the repository using the Add Repository Wizard.

Note: Information about repository dependencies is stored in the repository manager. Moving a repository on the same SAS Metadata Server has the following effects:

- The old dependency is deleted when the original repository's registration is deleted.
- You must re-create dependencies when you re-register the repository in its new location.

△

Porting a Repository to a Different Host Environment

Using the %OMAPORT Macro

SAS provides the %OMAPORT autocall macro for porting a SAS Metadata Repository created in one host environment to another host environment. %OMAPORT is run within SAS on the target host. The target host must have a SAS Metadata Server defined on it, and you must copy the source repository to a working directory on the target host by using one of the following methods before executing the macro:

- NFS or some other network convention
- binary FTP
- PROC CPORT/CIMPORT
- PROC DOWNLOAD/UPLOAD

After you have copied the source repository to the target host, use %OMAPORT to convert the repository data sets to the appropriate internal format. The macro uses information from the target host's repository manager to accomplish the conversion and writes the converted data sets to a destination directory.

Notes:

- The macro can convert the data sets of one repository at a time. Therefore, if you are porting a series of dependent repositories, you will have to copy and convert them separately.
- The macro does not register a ported repository on the target server. After the data set conversion is complete, you will need to connect to the target SAS Metadata Server with SAS Management Console and register the ported repository. This is also the time to redefine any dependencies between ported repositories.
- %OMAPORT does not affect metadata values in the source repository. You must change any host-dependent values that are stored in the metadata, such as pathnames and host names, manually.

You can port a repository between any two hosts supported by the SAS Metadata Server. Ask your SAS Support Representative for a list of the platforms supported for the SAS Intelligence Platform servers.

%OMAPORT Autocall Macro: Reference

The %OMAPORT macro converts a repository's data sets to the internal format of a different host environment.

Syntax

```
libname oldrpos 'source-repository-path';
libname newrpos 'target-repository-path';
libname newrmgr 'target-repository-manager-path';

%omaport(oldrpos,newrpos,newrmgr)
```

oldrpos
: specifies the pathname of the source repository. If the source and target hosts are connected in a network such as an NFS network, and you specify an absolute pathname for the *source-repository-path*, then %OMAPORT references the source repository on the source host. Otherwise, you must use one of the copy methods listed in "Details" on page 96 to move the source repository to a working directory on the target host and specify this working directory as the *source-repository-path*.

newrpos
: specifies the path to the target directory on the target host in which to create the converted repository data sets. Create this directory before running the macro; the macro will not create it for you.

  For example, if you are porting a foundation repository to the **`MetadataRepositories`** directory of a planned SAS installation location,
  you should specify a *target-repository-path* similar to this: *SAS-config-dir***`Lev1\SASMain\MetadataServer\MetadataRepositories\`***repository-name*.

newrmgr
: specifies the pathname of the target SAS Metadata Server's repository manager. The planned SAS installation typically creates the repository manager in a subdirectory of the **`MetadataServer`** directory that is named **`rposmgr`**. So the pathname might look like this: *SAS-config-dir***`Lev1\SASMain\MetadataServer\rposmgr`**. In order for %OMAPORT to work, the repository manager pathname must exist, and the **`rposmgr`** directory must contain valid repository manager data sets for the target host. The SAS Metadata Server

populates the **rposmgr** directory with repository manager data sets upon invocation.

Details

Although the %OMAPORT macro does not connect to it, a SAS Metadata Server must be defined on the target host in order for the macro to work.

To ensure that you are copying the latest version of a repository, it is recommended that you either unregister or pause the source repository to a read-only or offline state before copying it. This prevents updates to the source repository during the time that it takes to copy the repository. Later, you can choose to delete or continue using the source repository, depending on whether your intention is to promote or to replace the source repository.

You can pause, resume, and delete an individual repository from within SAS by using PROC METAOPERATE. For more information on PROC METAOPERATE, see the section "METAOPERATE Procedure" in the *SAS Open Metadata Interface: Reference*.

When you are porting a repository, also keep in mind the following information:

- The SAS Metadata Server stores password values entered for Login metadata as variable-length attributes in encrypted form. A variable-length attribute is one that accepts a value of any length, but stores values that exceed a certain length as overflow. %OMAPORT cannot convert repositories that contain Login objects with Password attributes that are stored as overflow. The maximum password length allowed is 42 characters.
- In SAS 9, the SAS Metadata Server created repository data sets with the encoding set to "ANY". Beginning in SAS 9.1, the SAS Metadata Server creates repository data sets with the encoding set to "ASCIIANY" or "EBCDICANY". The macro cannot distinguish whether the source host data sets are ASCII or EBCDIC unless the encoding is set correctly. Repository data sets with outdated encoding will cause the macro to terminate abnormally with an error message.

Optimizing Repository Memory Usage

%OMARUNAN Autocall Macro

The SAS Metadata Server is an *in-memory* server. In addition to being stored on disk, all queried and updated metadata is held in memory for the duration of the SAS Metadata Server session in order to optimize performance. Records are read into memory as they are requested by clients and remain in memory until the metadata server is either paused and resumed, or stopped.

SAS provides tools to optimize memory usage by these in-memory repository data sets. One such tool is the %OMARUNAN macro. %OMARUNAN is in your autocall library. The macro analyzes the handling of character variables in backup copies of repository data sets, checking for such things as unnecessary indexes, long character variables that can be stored as variable-length strings, and duplicate strings. It stores the results of this analysis in output DS_VCHAR data sets in each repository backup directory. The optimizations in the DS_VCHAR data sets are applied to production repositories by the %OMABAKUP macro the next time a backup is run to the same destination directory. %OMABAKUP merges the information in the DS_VCHAR data sets with a DS_DEFN data set that it creates in the current **rposmgr** and repository directories, and uses the updated information in DS_DEFN to load repositories back into memory after a backup. The optimizations are not applied to repository data sets on disk.

Comparing the %OMARUNAN Macro and the %OMABAKUP RUNANALYSIS Option

The %OMARUNAN macro has the same functionality as the %OMABAKUP RUNANALYSIS option. Both tools analyze the handling of character variables in backup copies of SAS Metadata Repositories and return information to optimize the memory usage of SAS Metadata Repositories. The tools replace the %OMAOPTMZ autocall macro that was provided for this purpose in releases prior to SAS 9.1.3 Service Pack 4.

The difference between these new tools is that RUNANALYSIS runs the analyses when backups are performed. %OMARUNAN runs independently of other programs and the SAS Metadata Server, so it can be executed at any time on any computer that has SAS software licensed and a network connection to the computer on which the backups are stored.

For a further comparison of the tools, see "Optimizing Repository Memory Usage by Using the %OMABAKUP Macro" on page 58.

Using the %OMARUNAN Macro

Because the %OMARUNAN macro analyzes backup copies of SAS Metadata Repositories, a backup copy of SAS Metadata Repositories must have been created using %OMABAKUP before you run %OMARUNAN.

%OMABAKUP also creates the OMA_Varan.sas7bcat catalog that is used by %OMARUNAN. This has two implications for running %OMARUNAN:

- The catalog is created to the specifications of the host operating environment; therefore, you need to execute %OMARUNAN in the same operating environment in order to be able to read the catalog. That is, if the backups are created on a UNIX host, then you need to execute %OMARUNAN from another UNIX host.
- The information in OMA_Varan.sas7bcat reflects the last backup that was made. That is, if a full backup was made, then OMA_Varan.sas7bcat catalog contains information to analyze all repositories that are registered on the SAS Metadata Server. If a selective backup was made, then OMA_Varan.sas7bcat contains only the information necessary to analyze the specified repositories. Keep this in mind when you choose backups that you want to analyze.

Because %OMARUNAN does not interact directly with the SAS Metadata Server, no special user IDs or server privileges are required to execute the macro. However, the user executing the macro needs full operating system access to the SAS Metadata Server backup destination path and all subdirectories.

%OMARUNAN Autocall Macro: Reference

The %OMARUNAN macro analyzes SAS metadata repositories for information to optimize their memory usage.

Syntax

```
%omarunan(DestinationPath="pathname")
```

DESTINATIONPATH
specifies the path to a SAS Metadata Server backup directory. %OMARUNAN reads a VarAnalyzeAll source entry in an OMA_Varan.sas7bcat catalog in this directory to locate and analyze the repository backups in the specified location.

Details

The %OMARUNAN macro must be executed on a computer that has SAS software installed.

To minimize the impact of the character variable analyses on the SAS Metadata Server host, direct the backup copies that you want to analyze to a network-accessible drive, and execute %OMARUNAN on that computer. Note that %OMARUNAN must be executed in the same operating environment that was used to create the backups.

It is not necessary to perform character variable analyses on a regular basis. Character variable usage is unlikely to change much unless you add or delete a significant percentage of metadata from the total metadata in a repository. Execute %OMARUNAN after metadata is first loaded in SAS Metadata Repositories and after significant updates are applied.

Note: If you run %OMABAKUP on an analyzed backup location in restore mode before the optimizations are applied, the %OMARUNAN optimizations will be lost. The restore process deletes the DS_VCHAR data sets that contain the optimizations from the backup directories before restoring repositories to their registered locations. △

Avoiding Changes to the Repository That Adversely Affect Performance

The SAS Metadata Server optimizes its performance by relying on the original internal structure of repository data sets. Therefore, you should not use SAS options that affect the internal structure of these data sets, because such changes will adversely affect performance. For example, do not set the data set options COMPRESS and REUSE to **Yes** in order to compress observations and to reuse compressed space in compressed data sets. In addition, using a procedure like PROC SORT can cause problems. The problems result from the procedure's changing the order in which observations are stored in the data sets.

Determining Whether the COMPRESS or REUSE Attribute Has Been Set

It is possible that the installer at your site set the COMPRESS and REUSE options on your repository data sets when he or she set up your system. When the installer creates your foundation repository by using SAS Management Console, he or she has the chance to specify data set options such as COMPRESS=Yes and REUSE=Yes, and may have done so in an attempt to conserve disk space. In addition, these options can be set inadvertently if they are specified as global options in a system command line, a configuration file, or an autoexec file when you start the metedata server or run the %OMABAKUP macro.

To determine whether COMPRESS or REUSE has been set on a repository, run PROC CONTENTS against the repository directory. For example, you might run this program:

```
libname reposdir 'C:\SAS\EntBIServer\Lev1\SASMain\MetadataServer\
MetadataRepositories\Foundation';
proc contents data=reposdir._all_;
run;
```

The output of this procedure will contain a list of variables and attributes for each data set processed. If the list for a data set indicates that the COMPRESS or REUSE option has been set, you will know that the internal structure of that data set has been altered.

Repairing Problematic Data Sets

To repair repository data sets that have been compressed, perform these steps:

1. Create a temporary repository directory to hold the uncompressed data sets.
2. Use PROC COPY or a DATA step to create an uncompressed version of the data sets in the temporary location. For example, you can use code similar to the following:

```
/* set the correct system options for the SAS session and assign librefs
to the original and temporary repository directories */
options compress=no reuse=no;
libname reposdir 'C:\SAS\EntBIServer\Lev1\SASMain\MetadataServer\
MetadataRepositories\Foundation';
libname tempdir 'C:\SAS\EntBIServer\Lev1\SASMain\MetadataServer\
MetadataRepositories\FoundationTemp' compress=no;

/* list the contents of the repository directory */
proc contents data=reposdir._all_;
run;

/* use proc copy to re-create the contents of the original repository
directory in the temporary directory */
proc copy in=reposdir out=tempdir noclone;
run;

/* use proc contents on the new directory to verify that compress and
reuse are set correctly */
proc contents data=tempdir._all_;
run;
```

3. Use an operating system command to swap the directory names.

When you use PROC COPY, you must have enough space to store two copies of all of the repository data sets. If only a few data sets are affected, you might prefer to use the DATA step. The following code fragment illustrates the use of a DATA step:

```
/* set the correct system options for the SAS session and assign a libref
to the original and temporary repository directories */
options compress=no reuse=no;
libname reposdir 'C:\SAS\EntBIServer\Lev1\SASMain\MetadataServer\
MetadataRepositories\Foundation';
libname tempdir 'C:\SAS\EntBIServer\Lev1\SASMain\MetadataServer\
MetadataRepositories\FoundationTemp' compress=no;

/* execute a data step similar to the following one for each data set that
needs to be repaired */
data tempdir.data-set-name;
set reposdir.data-set-name;
run;
```

After re-creating the data sets, use operating system commands to copy the modified data sets to the source directory.

CHAPTER

6

Using the Promotion and Replication Wizards to Copy an Entire Metadata Repository

When to Use the Promotion and Replication Wizards

The Replication Wizard and the Promotion Wizard enable you to copy an entire metadata repository from one host machine to another.

The two wizards are identical, except that the Promotion Wizard enables you to make modifications to the metadata as you copy it. For example, you can change port numbers, host names, schema names, and path names.

The Promotion Wizard and the Replication Wizard are useful in situations such as the following:

- You have implemented the SAS Intelligence Platform in a production environment, and you want to use that environment as the basis for creating or refreshing a test environment. In the process, you need to make modifications such as changing your server definitions so that they refer to hosts in the test environment.
- You have developed a completely new implementation of the SAS Intelligence Platform in a test environment, and you want to move the new implementation to a production environment. In the process, you need to make modifications such as changing your server definitions so that they refer to hosts in the production environment.
- You have created a metadata repository for a particular organizational unit, and you want to create another instance of the repository for another organizational unit.

CAUTION:
The Promotion Wizard and the Replication Wizard will completely overwrite all of the metadata in the target environment, including server definitions, library definitions, user definitions, access controls, personal folders, user-defined portal content, and portal preferences. △

Because of this behavior, the promotion wizard is useful primarily when you are creating or refreshing a development environment or test environment based on an existing production environment, or when you are creating a completely new production environment based on a development or test environment.

To promote metadata for only selected BI objects (including selected information maps, reports, stored processes, and data explorations), use BI Manager's Export and Import Wizards. For details, see Chapter 7, "Promoting Individual BI Objects," on page 133.

Features and Capabilities of the Promotion and Replication Wizards

Both the Promotion Wizard and the Replication Wizard provide a series of dialog boxes that guide you through the process of creating a job definition. The dialog boxes prompt for the following information:

- the name of the source metadata server and the repository that you want to promote.
- the name of the target metadata server.
- credentials to log on to the source metadata server and the target metadata server.
- for promotion jobs, the values for any metadata attribute substitutions that should be made, including values for port numbers, host machine names, schema names, and paths. For each of these metadata attributes, the wizard displays the values

that currently exist in the source metadata repository. For each value, you can enter a corresponding substitution. The wizard checks the validity of each substitution value that you enter.

- paths for work directories on both the source metadata server and the target metadata server.
- the path to a repository backup location on the target metadata server. The promotion or replication job will back up the existing metadata repository and place it in this location before replacing it with the promoted or replicated repository.
- a name for the job definition.
- the action to take with the job definition. You can choose to save the job and run it later, or you can go ahead and run the job and also save the job definition.

Based on the information that you provide, the wizard generates the SAS code that is necessary to promote or replicate your metadata repository. The code is saved using the job definition name that you specify. The substitution values that you enter are saved with the job.

You can create as many different promotion and replication jobs as necessary to handle different scenarios. The jobs can be rerun at any time.

Special Considerations

Before choosing to use the Promotion Wizard or the Replication Wizard, you should be aware of the following special considerations:

- The Promotion and Replication wizards will completely replace the metadata repository in the target environment. This means that all of the metadata in the target environment will be overwritten, including the following:
  - server definitions
  - library definitions
  - user definitions
  - access controls
  - personal folders that users have created to store reports
  - preferences that users have selected in the SAS Information Delivery Portal
  - portal content (including pages, page templates, portlets, links, syndication channels, and applications) that users have created or modified
- The Promotion and Replication wizards copy all of the metadata in the repository. To promote metadata for only selected BI objects (including selected information maps, reports, stored processes, and data explorations), use BI Manager's Export and Import Wizards. For details, see Chapter 7, "Promoting Individual BI Objects," on page 133.
- Although the Promotion Wizard automatically replaces some metadata values based on information that you provide, you will need to manually enter some metadata changes after the promotion job has completed. Manual changes will be required for server definitions, user logins, content mapping for reports, Foundation Services deployment definitions, and cube definitions. For details, see "Making Modifications after Running a Promotion Job" on page 124.
- The Promotion and Replication wizards promote only your metadata. They do not promote associated content, such as the following:
  - WebDAV folders and their contents (for example, report definition files and portal content)

  - other portal content, including custom portlets, custom themes, and applications
  - source files for stored processes
  - physical tables and databases

  For information about moving this content, see "Moving Associated BI Content Following a Promotion Job" on page 130.
- Before you can promote or replicate a dependent repository to a target machine, you must first promote or replicate its set of parent repositories to that same host. For details, see "Promoting Dependent Repositories" on page 105.
- When you use the Promotion and Replication wizards, the source environment and the target environment must have the same operating system. For example, you can promote or replicate a metadata repository from a Windows host to another Windows host, or from a UNIX host to another UNIX host.

  If you want to promote or replicate metadata from one operating system to another, you must copy the repository to the target environment and then run a SAS macro that converts the repository data sets to the appropriate internal format. For details, see "Porting a Repository to a Different Host Environment" on page 94.
- SAS/CONNECT software must be installed in both the source environment and the target environment.
- Before you use the Promotion or Replication wizard, you must prepare the source environment, the target environment, and a separate administration environment with the appropriate software and server definitions. In both the source environment and the administration environment, you will need to set up special administrative users. For details, see "Preparing Your Site for a Promotion Job" on page 105.
- Before you use the Promotion or Replication wizard, you should back up the existing target repository if there is any chance that you might need to recover the metadata.

Overview of the Promotion or Replication Process

The remainder of this chapter explains the following procedures:

- how to set up your source and target environments (for example, production and test environments) for a promotion or replication operation
- how to set up an administration environment from which you can create and run a promotion or replication job
- how to use SAS Management Console to create and run a promotion or replication job
- how to customize a promotion or replication job
- how to troubleshoot a failed promotion or replication job
- how to move content that is associated with the metadata following a promotion or replication job

Note: Throughout the rest of this chapter—except where we explicitly state otherwise—we use the term *promotion* to mean promotion or replication. △

The topics listed above and others are discussed in the following sections:

If you want to promote a dependent repository, you should first read "Promoting Dependent Repositories" on page 105.

Promoting Dependent Repositories

Promoting a dependent repository is a special case. Before you can promote a dependent repository to a target machine, you must promote its set of parent repositories to that same host. The promotion process uses the permissions that are defined in the server's foundation repository and inherited by the dependent repository.

So the general procedure for promoting a dependent repository is as follows:

1. Promote all parent repositories of the dependent repository from the source server to the target server.
2. Promote the dependent repository from the source server to the target server.
3. On the target server, define the dependency relationships between the dependent repository and its parents so that they are the same as the dependencies between the corresponding repositories on the source server.

It is only necessary to create the dependencies between the parent and child repositories on the target server the first time that you promote the repositories. The dependency relationships will remain in place for subsequent promotions of the repositories. However, if the dependency structure changes on the source server, you must establish an equivalent dependency structure on the target server.

Preparing Your Site for a Promotion Job

Before you can run a promotion job, you must correctly set up three *environments*, each consisting of a metadata server, a metadata repository, and some related software components:

- A *source environment*, which includes the metadata server in an existing installation of the SAS Intelligence Platform. This environment contains the metadata repository that you want to promote to the target environment.

  In the source environment, you must have SAS/CONNECT software installed because the promotion process uses this software to move the repository data sets to the target environment. In addition, if you want to be able to run the promotion job from the Promotion Wizard, the source environment must include an IOM object spawner and a workspace server (which means that SAS Integration Technologies software must be installed). The workspace server and SAS/CONNECT software must be installed on the same host.

  Note: If you plan to run the promotion job in batch mode, you do not need the workspace server. However, the remainder of the chapter assumes that you want to be able to run the promotion job from the Promotion Wizard. △

- A *target environment*, which is the environment to which you want to promote the source environment's metadata repository.

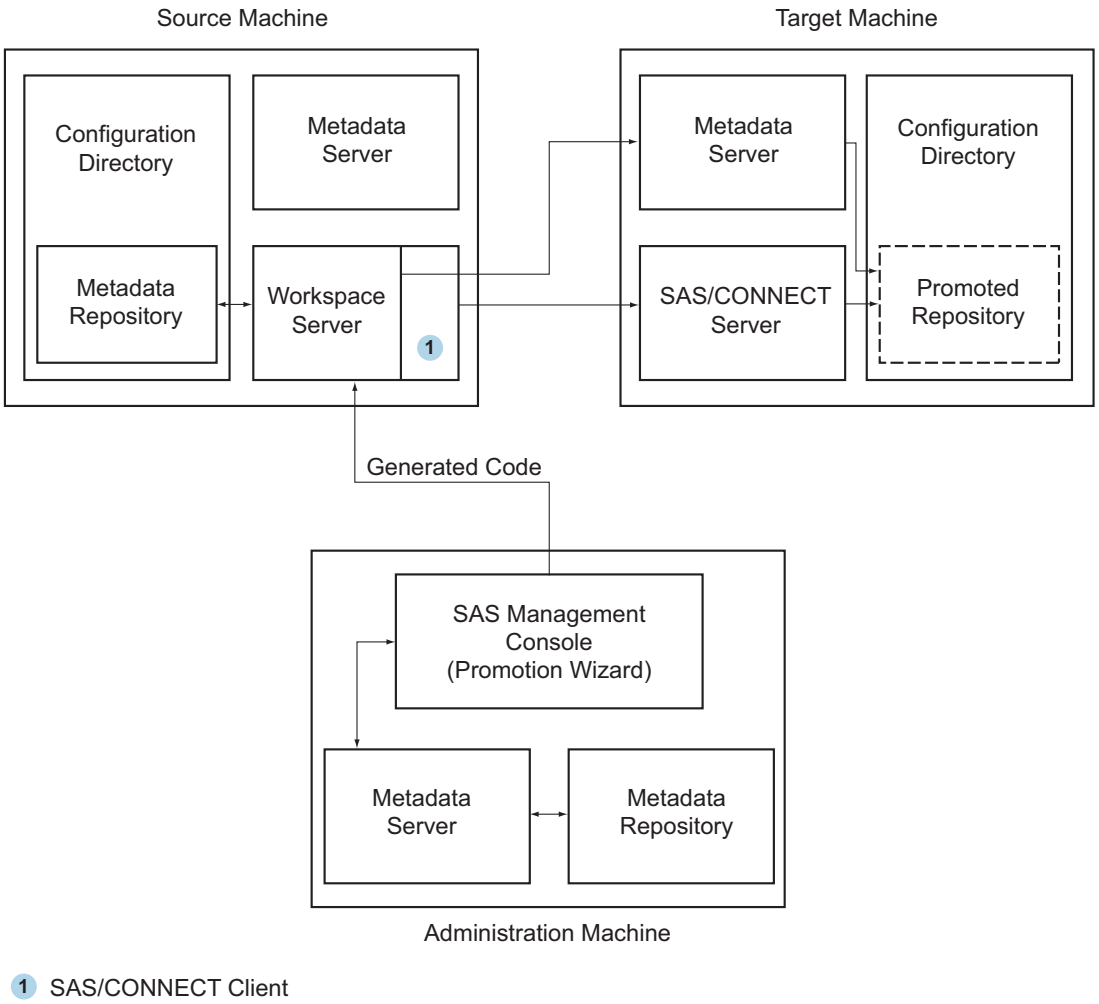
For the most part, you should install and configure in this environment the same SAS software that you originally installed in the source environment. An important exception to this rule is that you should not create a metadata repository in the target environment during configuration. Basically, this means not following any of the instructions in the **`instructions.html`** files produced by the SAS Configuration Wizard. Beyond this, the target environment must include a SAS/CONNECT server for the reasons mentioned above.

- An *administration environment*, which includes a metadata repository that contains information about the SAS servers and spawners in the source and target environments. It is also the environment from which you run the Promotion Wizard to create and execute a promotion job. In this environment, you must install Base SAS software and SAS Management Console, and you must configure a metadata server.

Note: Throughout the rest of this chapter, it is assumed that the source environment is Level 1, the target environment is Level 2, and the administration environment is Level 0. When the SAS Configuration Wizard creates a directory called **`Lev1`** in the path to your configuration directory, the **`1`** in the name means "production". A test environment would use **`Lev2`** in the path, and an administration environment would use **`Lev0`** in the path. △

The diagram below illustrates the primary components of the promotion infrastructure.

Figure 6.1 Required Infrastructure for a Promotion Job



As you can see, the Promotion Wizard generates code that performs the promotion. The wizard then submits this code to a workspace server running in the source environment. The code uses SAS/CONNECT software to copy the source environment's metadata repository to the target environment.

Note: For a much more detailed explanation of the promotion process, see "What Happens When You Run a Promotion Job?" on page 129. You may need this extra detail to customize the promotion process or to troubleshoot a problem with promotion. △

You may be wondering how the environments shown above map to different computers. Generally, the source environment is installed on one or more computers, and the target environment is installed on a different set of one or more computers. The administration environment is typically installed on a machine by itself or on the metadata-server host in the source environment. For the sake of simplicity, in this chapter, we will consider the case where each environment is installed on a single machine that is separate from the other environments. (We will also explain how the process would change if you were to have the source and administration environments on the same machine, since this is a common scenario.)

For detailed information about how to set up the source, target, and administration environments prior to creating and running a promotion job, see the following subsections.

Setting Up the Source Environment for a Promotion Job

On the source machine, you need to perform the following setup:

- Make sure that an IOM object spawner and a workspace server have been configured on the system and that SAS/CONNECT software has been installed. The SAS code that performs the promotion runs on the workspace server, and the SAS/CONNECT software enables the workspace server to move the repository data sets to the target machine.

  You can determine whether SAS/CONNECT software is installed on the source machine by looking in the SAS installation area and seeing whether the folder **`SAS 9.1`** contains a folder named **`connect`**. To determine whether an object spawner and workspace server have been configured, look in the folder *SAS-config-dir***`Lev1\SASMain`**, and see whether it contains directories named **`ObjectSpawner`** and **`WorkspaceServer`**.
- Create an administrative user, and make sure that this user has full access to the necessary parts of the configuration directory on the source machine. The user must have administrative access in order to pause and resume the metadata server.
- Create a metadata access file that is used to establish communication between the administration metadata server and the source metadata server.

The following subsections explain how to perform these tasks.

Install and Configure the Necessary SAS Software

If SAS/CONNECT software is installed on the source machine and a workspace server has been configured, you can skip this section. Otherwise, perform these tasks:

1. Install SAS/CONNECT software (if it is not already installed) and SAS Integration Technologies software (if it is not already installed). Both of these components are part of the SAS 9.1 Foundation, so you can install them by performing a Software Index installation of the SAS Foundation. (For information on how to perform a Software Index installation, see the *SAS Intelligence Platform: Installation Guide*.) During the installation of the SAS Foundation, you will have an opportunity to specify which components the installer should install.
2. If you installed Integration Technologies in step 1, use the SAS Configuration Wizard to configure an object spawner and a workspace server. (For information about how to run the SAS Configuration Wizard outside the context of a planned installation, see the *SAS Intelligence Platform: Installation Guide*.)

Set Up an Administrative User

Before you can run a promotion job, you must set up an administrative user on the source machine. This user will have to be authenticated by the source machine's authentication provider and will need full access to the **`MetadataRepositories`** and **`ReplicationWorkArea`** directories.

To set up such a user on a Windows system, follow these steps:

1. Create an operating-system user account for a user with the ID srcadmin.
2. Make this user an administrative user by adding the user's ID to the file **`adminUsers.txt`**, which is located in the directory *SAS-config-dir***`Lev1\SASMain\MetadataServer`**. This file should already contain an entry for the SAS Administrator (sasadm). Add a similar entry for srcadmin,

but do *not* place an asterisk before the user ID. You will have to restart the metadata server for this new setting to take effect.

3 Give the user srcadmin full control of the following directories, either explicitly or by adding the user to the Administrators group:
   - the directory *SAS-config-dir*`\Lev1\SASMain\MetadataServer\ReplicationWorkArea` and its subdirectories
   - the directory that holds your repository data sets, typically *SAS-config-dir*`\Lev1\SASMain\MetadataServer\Foundation`

If your source machine is a UNIX box, you should make the SAS user (sas) an administrative user by adding this user to the file **`adminUsers.txt`**. Do not place an asterisk before the user ID. This user should already be the owner of the **`MetadataRepositories`** and **`ReplicationWorkArea`** directories.

Create a Metadata Access File

To create the metadata access file, follow the directions below.

1 Before you actually create the file, you must decide on some values that you will use when you set up the administration machine:
   - The number of the port on which the metadata server on this machine will listen. By default, this will be port 8560.
   - The user ID that you will use for the SAS Replication Administrator account, by convention, sasrpadm. This user will be an administrative user on the administration machine.
   - The password that you will assign to the SAS Replication Administrator.

2 Create a file named **`mdrepljs.sas`** in the directory *SAS-config-dir*`\Lev1\SASMain`. The content of this file should look like this:

```
options metaserver='host-name'
        metaport=port-number
        metaprotocol=BRIDGE
        metauser='user-ID'
        metapass='encoded-password';
```

You need to substitute actual values for the variable values shown in italics. The table below explains how to replace the variables in this text.

Table 6.1 Creating the File mdrepljs.sas

Variable	Replace With
host-name	The host name of the administration machine.
port-number	The number of the port on which the metadata server on the administration machine will listen (8560).

Variable	Replace With
user-ID	The user ID of the SAS Replication Administrator. By convention, this user is named sasrpadm. If the administration machine is a Windows system, you must qualify this ID with a domain or machine name, for example, **`D1234\sasrpadm`**.
encoded-password	The encoded password for the SAS Replication Administrator. To determine the encoded form of the password, start an interactive SAS session, and submit the following code in the Editor: **`proc pwencode in='`*`unencoded-pwd`*`';`** **`run;`** The encoded password will appear in the Log window.

Setting Up the Target Environment for a Promotion Job

On the target machine, you need to perform the following setup:

- Install the same components of the SAS Intelligence Platform that you installed on the source machine when you first created your system. (The target server is typically a mirror image of the source machine.) But do not create a metadata repository. The target metadata repository is created when you run your promotion job.
- Make sure that SAS/CONNECT software has been installed and that a SAS/CONNECT server has been configured. A SAS/CONNECT spawner must be running on the target machine in order for the workspace server on the source machine to move the repository data sets to the target machine.
- Manually update the script that starts the SAS/CONNECT spawner so that you can run it without registering the SAS/CONNECT server in the metadata. (You do not have a metadata repository on the target machine yet.) Then, start the spawner.
- Set up an unrestricted user, who will be the process owner for the SAS/CONNECT server.

The following subsections explain how to perform these tasks.

Install the Platform Components Called for by Your Plan

When you originally installed the SAS Intelligence Platform, you installed and configured a set of products on the source machine. Under normal circumstances, you want to install and configure the same set of products on your target machine. To do this, you follow basically the same procedure that you used to install the platform on your source machine:

1. Perform the pre-installation tasks required for your platform and the products that you are installing on your target machine.
2. Use the SAS Software Navigator to install the set of products listed in a deployment plan. This may be the same deployment plan that you used when you installed software on the source machine, or a similar one that contains different host names.
3. Use the SAS Configuration Wizard to configure the products that you installed, but *do not* use SAS Management Console to create a metadata repository, and do not follow any of the instructions in the file **`instructions.html`**. You create the metadata repository on the target machine when you run your promotion job.

   When you run the SAS Configuration Wizard, be sure to do the following things:

a When you are prompted for the location of the configuration directory, specify the same value that you used when you configured the source machine.

b When you are prompted for the port on which the metadata server will listen, enter the value **8562**. On your source machine, the name of the Level directory (**Lev1**) and many of the port numbers end with the number 1. This number indicates that these items are part of your production environment. Similarly, the number 2 indicates that an element is part of your first derived configuration, which is typically a test environment.

c When you reach the Advanced Properties Editor screen, open the properties file, and make the changes shown below:

- Change **Lev1** to **Lev2**.
- Change the port number **8591** to **8592**, if necessary. This is the workspace server port.
- Change the port number **7551** to **7552**, if necessary. This is the SAS/CONNECT server port.

Install and Configure the SAS/CONNECT Software

To install SAS/CONNECT software and create a script that you can use to start a SAS/CONNECT spawner, follow these directions:

1. Install SAS/CONNECT software by performing a Software Index installation of the SAS Foundation. (For information on how to perform a Software Index installation, see the *SAS Intelligence Platform: Installation Guide*.) During the installation of the SAS Foundation, you will have an opportunity to specify that you want to install only the SAS/CONNECT component of the SAS Foundation.
2. Use the SAS Configuration Wizard to configure the SAS/CONNECT software. For information about how to run the SAS Configuration Wizard outside the context of a planned installation, see the *SAS Intelligence Platform: Installation Guide*. *Do not* follow the directions in the **instructions.html** file generated by the configuration wizard.

Start the SAS/CONNECT Spawner

When you ran the SAS Configuration Wizard in the previous step, it created a script that you can use to start a SAS/CONNECT spawner. Edit this script as described below; then run the script as directed in step 5 to start the spawner.

On Windows systems, you edit and run the script *SAS-config-dir***\Lev2\SASMain\ConnectServer\ConnectServer.bat** as follows:

1. In the file **ConnectServer.bat**, locate the line that sets the value of the **USEMETADATA** option, and change it so that it reads:

```
set USEMETADATA=0
```

2. Locate the line that sets the value of the **IsService** option, and change it so that it reads:

```
set IsService=0
```

3. Locate the line that begins **echo Manual start of Spawner**. Make sure that the **start** command on the next line contains the **-security** option. For example, the finished **start** command might look like this:

```
start \b "Connect Spawner" "%sasdir%\spawner" -security -service %CONNPORT%
  -SASCMD %CONNCMD%
```

4. Save your changes to **ConnectServer.bat**.

5 Start the spawner by running the script **StartConnectServer.bat**, which is located in the same directory as the **ConnectServer.bat** script. To start a spawner using a command that contains the **-security** option, you must have the following Windows user rights:
   - Act as part of the operating system
   - Adjust memory quotas for a process (or Increase quotas)
   - Bypass traverse checking
   - Log on locally
   - Replace a process level token

On UNIX systems, you edit and then run the script *SAS-config-dir***\Lev2\SASMain\ConnectServer\ConnectServer.sh**.

1 In the file **ConnectServer.sh**, locate the line that sets the value of the **USEMETADATA** option, and change it so that it reads:

```
set USEMETADATA=0
```

2 Save your changes to **ConnectServer.sh**.

3 Start the spawner by running the script **ConnectServer.sh**. Use the following command:

```
ConnectServer.sh start
```

Give the Target Administrator the Required Directory Permissions

In the promotion procedure discussed in this chapter, a special account on the target machine is used to start a SAS/CONNECT server process and to communicate with the metadata server on that machine. This user must be an unrestricted user and must have full access to certain directories.

On Windows systems, we recommend that you use the SAS Administrator account (sasadm) for this purpose. This user is already an unrestricted user, but must be granted full access to the following directories in order for the promotion to succeed:

- *SAS-config-dir***\Lev2\SASMain\MetadataServer\ReplicationWorkArea** and its subdirectories
- *SAS-config-dir***\Lev2\SASMain\MetadataServer\MetadataRepositories\Foundation**

On UNIX systems, we recommend that you use the SAS User account (sas). This user already has access to the requisite directories; however, the user is not an unrestricted user of the metadata. To make sas an unrestricted user, enter the following line in the file **adminUsers.txt**:

```
*sas
```

Setting Up the Administration Environment for a Promotion Job

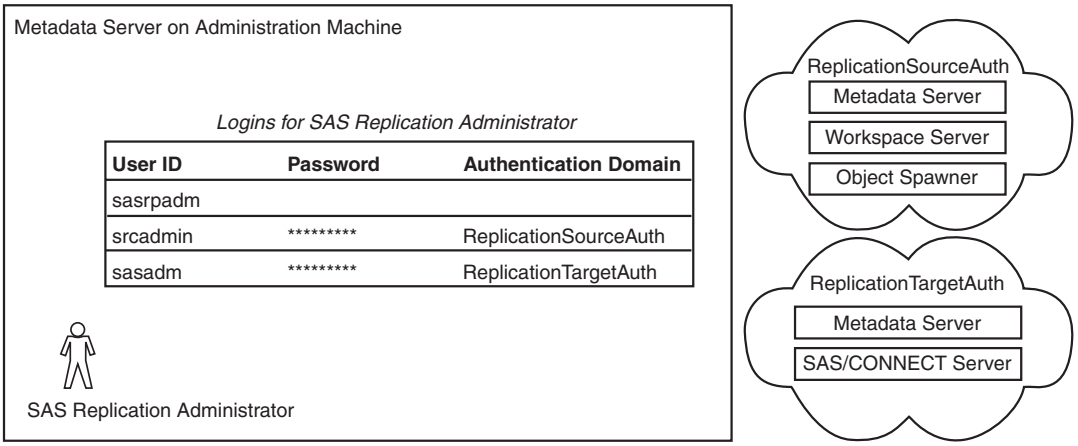
Setting up the administration machine involves performing the following tasks:

- installing Base SAS software and SAS Management Console
- configuring a metadata server
- creating an operating-system user account for a SAS Replication Administrator
- creating metadata objects for the SAS Replication Administrator and the SAS servers that are running on the source and target machines

In order to run the Promotion Wizard, you log into SAS Management Console as the SAS Replication Administrator. This same user later needs to be able to start a

workspace server on the source machine and a SAS/CONNECT server on the target machine. To make sure that the SAS Replication Administrator is authenticated when he or she attempts to start these servers, you need to place the SAS servers on the source machine in one authentication domain, place the SAS servers on the target machine in a different authentication domain, and create three logins for the replication administrator. This setup is shown below.

Figure 6.2 Logins for the SAS Replication Administrator



Note: The figure above shows the logins that the SAS Replication Administrator would have if you were promoting a repository from one Windows system to another. If you are promoting a repository on a UNIX machine, the second and third logins should contain the credentials of the SAS User (sas). △

You can find details about how to set up these authentication domains and logins—and all of the other tasks mentioned above—in the sections below.

Install and Configure the Necessary Software

To install Base SAS software and SAS Management Console on the administration machine and to configure a metadata server there, follow these directions:

1. Perform the set of pre-installation tasks required for your platform and the set of products that you plan to install.

   Note: If your source and administration environments will reside on the same machine, you will already have performed this step. △

2. Use the SAS Software Navigator to perform a Software Index installation of the SAS Foundation and SAS Management Console. During the installation of the SAS Foundation, you will have an opportunity to specify which components of the foundation you want to install. You need to install only Base SAS software.

   Note: If your source and administration environments will reside on the same machine, you will already have performed this step. △

3. Use the SAS Configuration Wizard to configure a metadata server on this host, following the directions in the *SAS Intelligence Platform: Installation Guide*. The only instructions specific to the current situation are listed below.

   a. When you are prompted for the metadata server's port number, enter **8560**. (The 0 in this port number indicates that this instance of the metadata server is part of the administration environment.)

b Use the Advanced Properties Editor to change the value of the **LEVEL** property from **Lev1** to **Lev0**.

c When the configuration wizard generates the instruction file **instructions.html**, follow only the instructions for creating a Foundation repository. Do not follow the instructions that have you enter metadata in the repository.

Create an Operating System Account for the SAS Replication Administrator

You need to set up a user account for a new user and make this user an administrative user.

1 Create an operating-system user account for a user with the ID sasrpadm.

2 Make this user an administrative user by adding the user's ID to the file **adminUsers.txt**, which is located in the directory *SAS-config-dir***\Lev0\SASMain\MetadataServer**. This file should already contain an entry for the SAS Administrator (sasadm). Add a similar entry for sasrpadm, but do *not* place an asterisk before the user ID. You will have to restart the metadata server for this new setting to take effect.

Define the Source Metadata Server in the Metadata

The remaining instructions for setting up the administration machine involve creating metadata definitions for the SAS servers that are running on the source machine and the target machine and creating a definition for the SAS Replication Administrator. To perform each of these tasks, you should be logged in to SAS Management Console as the SAS Replication Administrator. You use SAS Management Console to create the server and user definitions.

To create a definition for the metadata server running on the source machine:

1 Right-click the **Server Manager**, and select **New Server**. The New Server Wizard appears.

2 Run the New Server Wizard, supplying the values shown below:

Note: For any properties not listed below, you should accept the default value. △

- Server type: **SAS Application Server**
- Name: **Lev 1 -** *application-server-name* (for example, **Lev 1 - SASMain**)
- Type of server to add to application server: **Metadata Server**
- Authentication Domain: **ReplicationSourceAuth**

  Note: Click **New** to create this authentication domain, if necessary. △

- Host Name: Name of source machine
- Port Number: **8561**

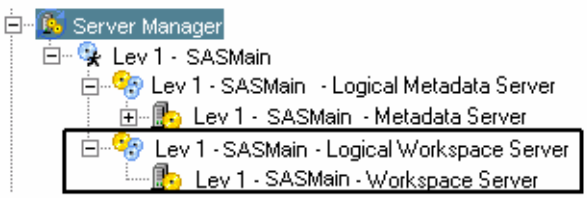
When you click **Finish**, the nodes shown below are added to the Server Manager tree.

Define the Source Workspace Server in the Metadata

To create a definition for the workspace server running on the source machine:

1 Right-click the node for the application server you defined in the previous section (**`Lev1 -`** *application-server-name*), and select **`Add Application Server Component`**. The New Application Server Component Wizard appears.
2 Run the wizard, supplying the values shown below:
   - Type of server to add to application server: **`Workspace Server`**
   - Authentication Domain: **`ReplicationSourceAuth`**
   - Host Name: Name of source machine
   - Port Number: **`8591`**

When you click **`Finish`**, the nodes shown below are added to the Server Manager tree.



Define the Source Object Spawner in the Metadata

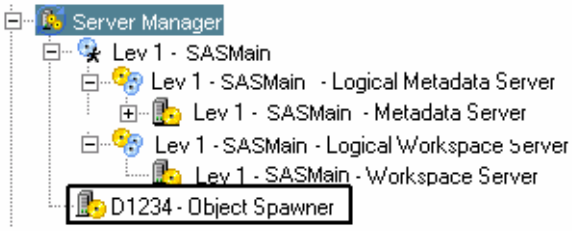
To create a definition for the object spawner running on the source machine:

1 Right-click the **`Server Manager`**, and select **`New Server`**. The New Server Wizard appears.
2 Run the wizard, supplying the values shown below:

   Note: For any properties not listed below, you should accept the default value. △

   - Server type: **`Object Spawner`**
   - Name: *hostname* **`- Object Spawner`** (for example, **`D1234 - Object Spawner`**)
   - Associated machine: Name of source machine
   - Selected Servers: **`Lev 1 -`** *application-server-name* **`- Workspace Server`**
   - Authentication Domain: **`ReplicationSourceAuth`**
   - Host Name: Name of source machine
   - Port Number: **`8581`**

When you click **`Finish`**, the node shown below is added to the Server Manager tree.



Define the Target Metadata Server in the Metadata

To create a definition for the metadata server running on the target machine:

1 Right-click the **`Server Manager`**, and select **`New Server`**. The New Server Wizard appears.
2 Run the New Server Wizard, supplying the values shown below:

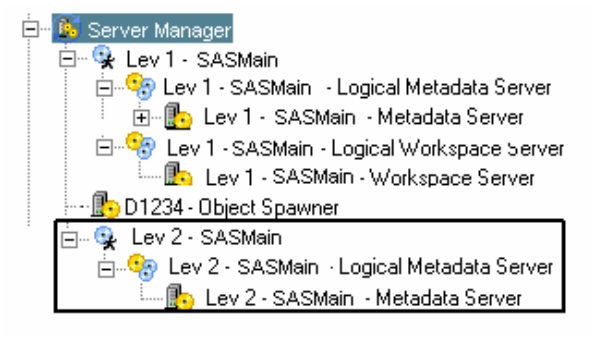
Note: For any properties not listed below, you should accept the default value. △

- Server type: **`SAS Application Server`**
- Name: **`Lev 2 -`** *application-server-name* (for example, **`Lev 2 - SASMain`**)
- Type of server to add to application server: **`Metadata Server`**
- Authentication Domain: **`ReplicationTargetAuth`**

  Note: Click **`New`** to create this authentication domain, if necessary. △

- Host Name: Name of target machine
- Port Number: **`8562`**

When you click **`Finish`**, the nodes shown below are added to the Server Manager tree.



Define the Target SAS/CONNECT Server in the Metadata

To create a definition for the SAS/CONNECT server running on the target machine:

1. Right-click the node for the application server you defined in the previous section (**`Lev2 -`** *application-server-name*), and select **`Add Application Server Component`**. The New Application Server Component Wizard appears.
2. Run the wizard, supplying the values shown below:
   - Type of server to add to application server: **`SAS/CONNECT Server`**
   - Authentication Domain: **`ReplicationTargetAuth`**
   - Host Name: Name of target machine
   - Port Number: **`7552`**

When you click **`Finish`**, the nodes shown below are added to the Server Manager tree.

Define the SAS Replication Administrator in the Metadata

When you run a promotion job, you run it as the SAS Replication Administrator. Because the replication administrator must be able to start a workspace server on the

source machine and a SAS/CONNECT server on the target machine, this user must have a total of three logins:

- one for the administration machine
- one for the source machine
- one for the target machine

To define this user and the user's three logins, follow the directions below:

1. Right-click the **User Manager** and select **New ▶ User**. The New User Properties dialog box appears.
2. On the **General** tab, enter **SAS Replication Administrator** in the **Name** text box.
3. Select the **Logins** tab.
4. Define a login for the user sasrpadm.
   a. Click **New** to bring up the New Login Properties dialog box.
   b. Enter the user ID for the sasrpadm account in the **User ID** field. On Windows systems, this ID must be qualified with a domain name or a machine name.
   c. Use the **Authentication Domain** list box to clear the associated text field.
   d. Click **OK**.
5. Define a login using the credentials for the srcadmin account (Windows) or the sas account (UNIX) on the source machine. Include a password, and set the **Authentication Domain** to **ReplicationSourceAuth**.
6. Define a login using the credentials for the sasadm account (Windows) or the sas account (UNIX) on the target machine. Include a password, and set the **Authentication Domain** to **ReplicationTargetAuth**. (This login needs to contain information about an unrestricted user.)
7. Click **OK** in the New User Properties dialog box.

At the end of this step, you have created the necessary infrastructure and are ready to run the promotion wizard to create the promotion job.

Creating and Running a Promotion Job

Once you have set up your source, target, and administration machines, you are ready to use SAS Management Console to create a promotion job and to run that job. After you run the promotion job, you complete the promotion manually by editing some of the metadata in the target repository. These topics are discussed in the following sections:

- "Building a Promotion Job" on page 117
- "Running a Promotion Job" on page 124
- "Making Modifications after Running a Promotion Job" on page 124

Building a Promotion Job

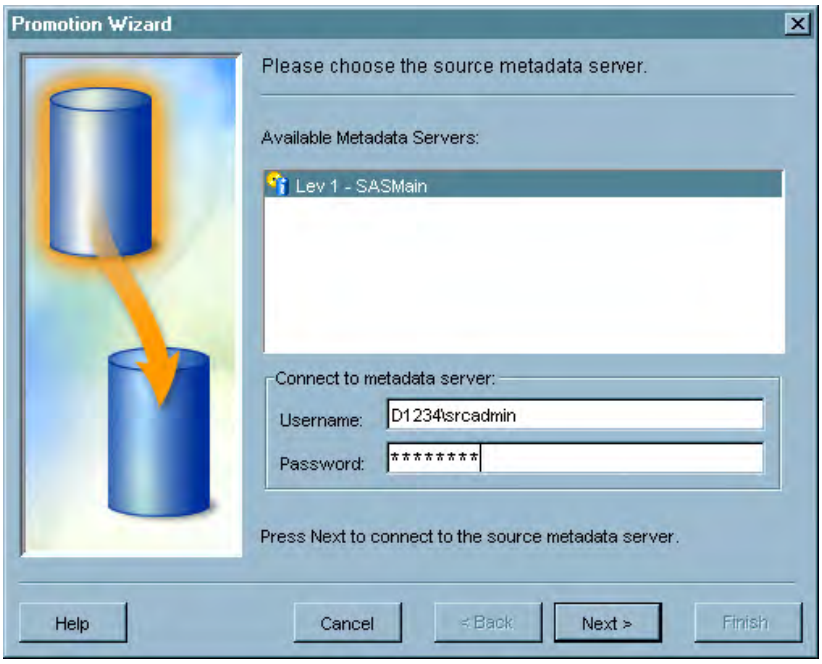
You create a promotion job—that is, you generate the SAS code required to promote a metadata repository—by running the SAS Management Console Promotion Wizard. To run this wizard, follow the directions below.

1. On the administration machine, start SAS Management Console, and connect to the metadata server on that machine using a metadata profile that contains the credentials of the SAS Replication Administrator (sasrpadm).

2 Expand the **Metadata Manager** and **Job Definition** nodes in the SAS Management Console tree. Underneath the **Job Definition** node, you will see icons labeled **Promotion** and **Replication**.

3 Right-click the **Promotion** icon, and select **New Definition** from the pop-up menu that appears. The Promotion Wizard starts.

4 The wizard's first page asks you to choose the source metadata server.



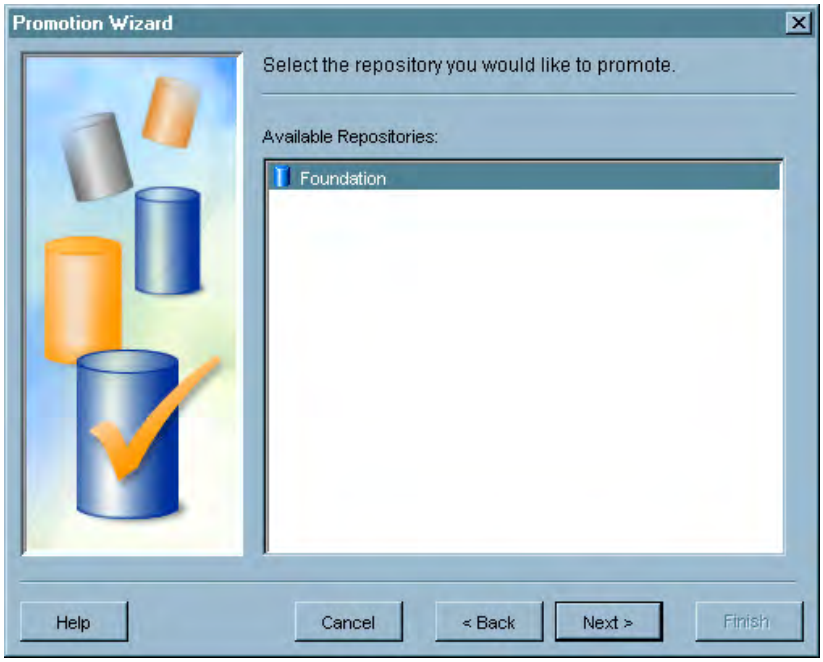
Provide the following information:

a In the **Available Metadata Servers** list, select the SAS application server that contains the metadata server on the source machine.

b In the **Connect to metadata server** area, supply the credentials of an administrative user on the source machine. Normally, you enter the user ID and password for the srcadmin or sas account. ("Set Up an Administrative User" on page 108 explains how to give this user administrative rights.)

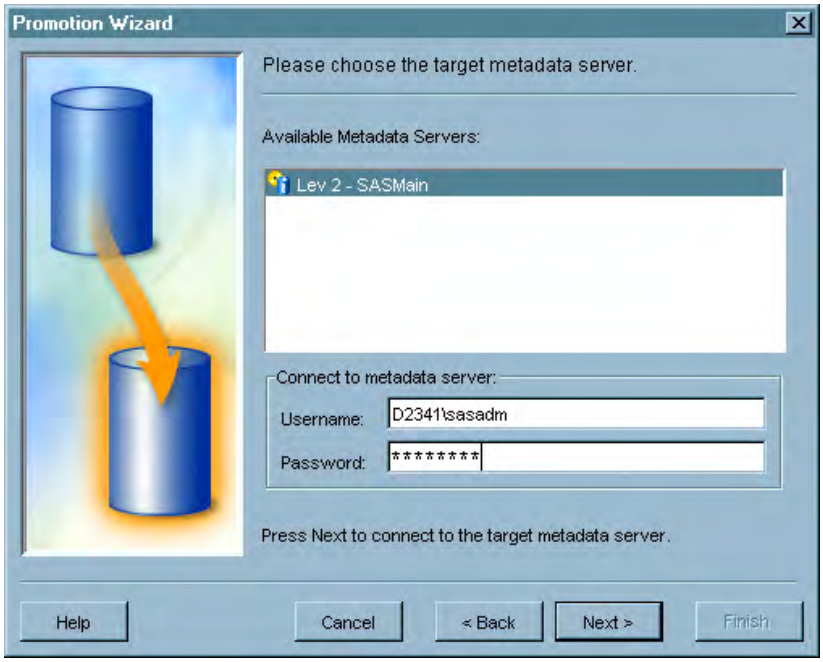
Then click **Next**.

5 The wizard's next page asks you to select the repository that you want to promote.



From the list of metadata repositories on the source machine, select the repository that you want to promote, and then click **Next**.

6 The wizard's next page asks you to choose the target metadata server.

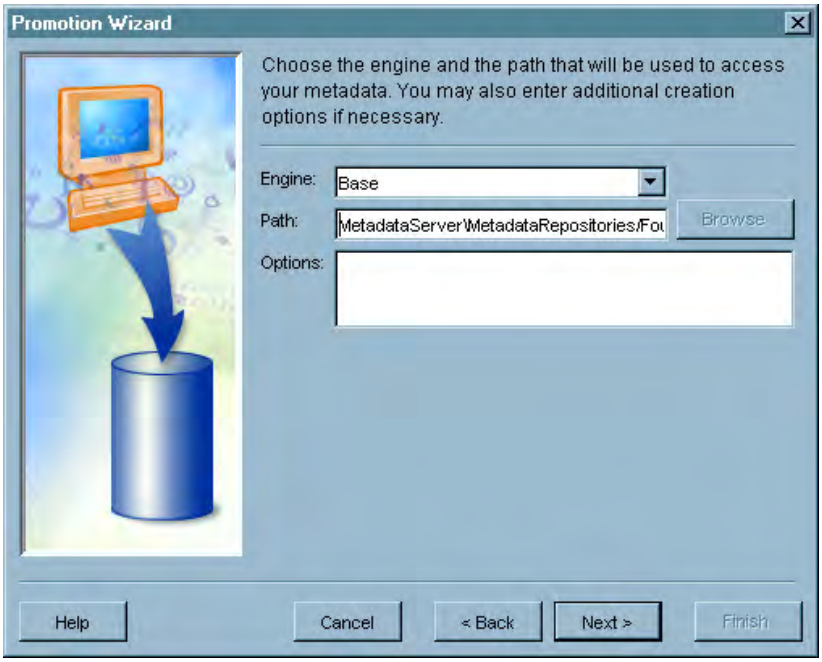


Provide the following information:

- **a** In the **Available Metadata Servers** list, select the SAS application server that contains the metadata server on the target machine.
- **b** In the **Connect to metadata server** area, supply the credentials of the SAS Administrator (sasadm) or SAS User (sas) on the target machine.

Then click **Next**.

7 If the repository that you selected has not been defined on the target machine—which will be the case the first time that you promote a repository—the wizard's next page asks you to specify the type and location of the new repository.



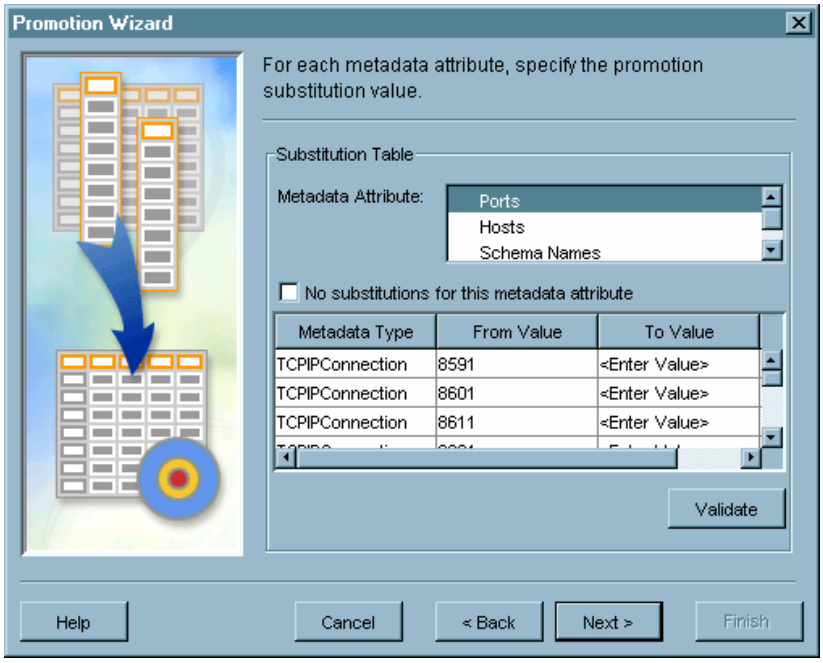
Provide the following information:

a From the **Engine** list, select the engine that will be used to access the metadata repository. Because the repository normally consists of a group of SAS data sets, typically you should select **Base**.

b In the **Path** field, enter the path from the **SASMain** directory in your configuration directory to the directory for the metadata repository. If you are promoting your foundation repository, this path will typically be **MetadataServer/MetadataRepositories/Foundation**.

Click **Next**.

8 If you are performing a promotion, the wizard's next page enables you to specify what substitutions the resulting promotion job should make in the metadata. (The Replication Wizard does not have this page.)



The wizard makes substitutions in the following four areas:

- port numbers
- host names
- schema names
- file system paths

For each type of information, follow this procedure:

a From the **`Metadata Attribute`** list, select an attribute such as **`Ports`**. If you select **`Ports`**, the table in the middle of the page is populated with a list of the port numbers in the source metadata repository. Typically, you want to specify a replacement value for each value in the **`From Value`** column. However, you can also select **`No substitutions for this metadata attribute`** to indicate that you do not want to make any changes for a particular attribute.

Note: Before you proceed to the next page in the wizard, you must supply substitution information for all four metadata attributes. △

b Unless you have chosen the no-substitutions check box for an attribute, for each value in the **`From Value`** column of the substitutions table, specify a value in the **`To Value`** column. For example, you might indicate that you want each occurrence of the port number **`8591`** in the source metadata repository to be changed to **`8592`** in the target repository. The table below provides some brief recommendations on how to choose a **`To Value`**.

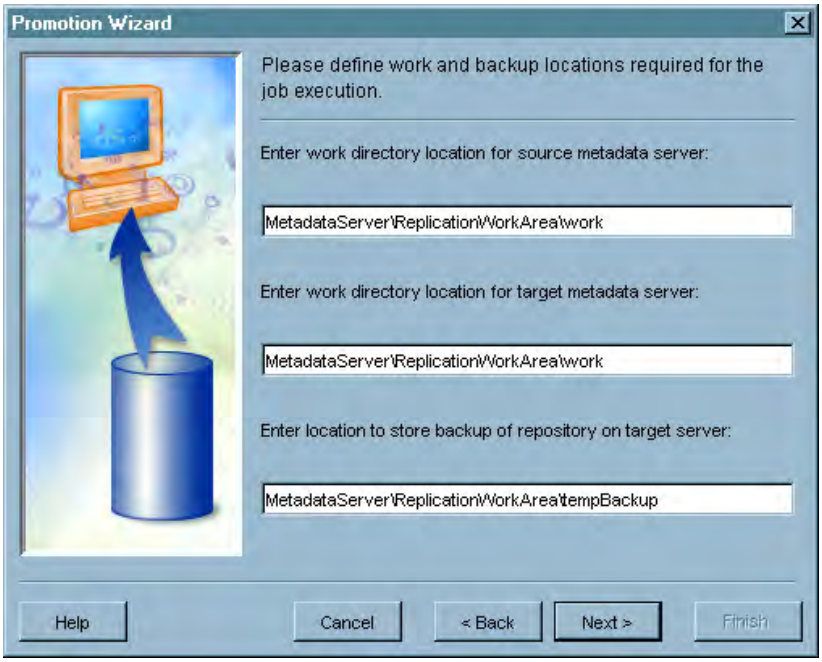
Table 6.2 Substitutions to Make in the Target Metadata

For this metadata attribute	Make this change to the source values
Ports	Add 1 to each port number that ends in 1. For example, if a port number is **4321**, change the value to **4322**.
Hosts	Change host names in the source metadata to names that will be valid in the target system. Use fully qualified host names. For example, if a source host name is **D1234**, the target host name might be **D1235.na.abc.com**.
Schema Names	Change any schema names to ones that will be appropriate on the target server.
Paths	Change any paths that do not exist on the target system to paths that are valid there. Leave all relative paths unchanged.

c Click **Validate**. If you have entered an acceptable **To Value** in each row of the table (or if you chose not to make any substitutions), a green check appears to the right of the currently selected metadata attribute. If you see a red **X** instead, one or more of the values you entered is not a valid replacement value. Fix each problematic row—a red **X** appears in the rightmost column of each such row—and then click **Validate** again. Repeat this set of steps as many times as necessary.

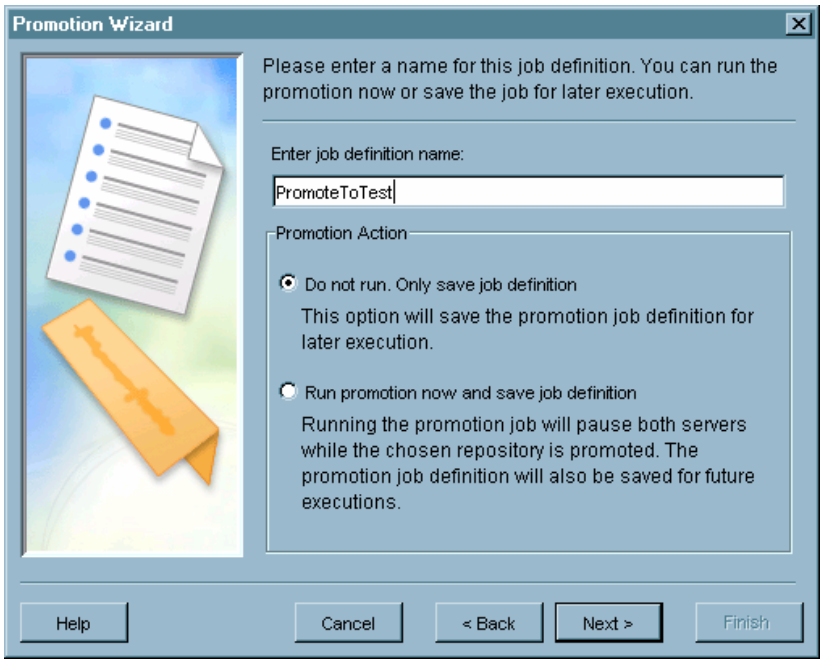
When all four metadata attributes have green check marks next to them, click **Next**.

9 The wizard's next page asks you to specify work directories on the source and target metadata server hosts and to specify a backup directory on the target host.



Enter values similar to those shown in the display above; then click **Next**.

10 The wizard's next page prompts you for the name of your promotion job and asks whether you want to run the promotion job now.



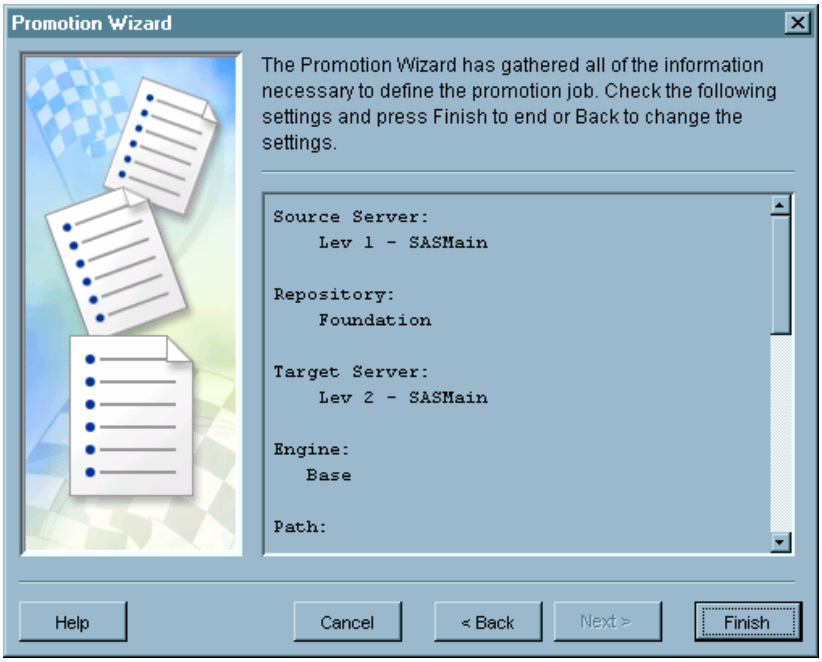
Supply the following values:

a In the **`Enter job definition name`** text box, enter a name for the replication job. We recommend that you use a name like **`Source1 Dev to Test - Foundation`**. In this name, "Source1" is a description of the metadata server associated with the repository being promoted; this part of the name is optional if, as usual, the environment contains only one metadata server. The "Dev to Test" part of the name describes the source and target environments. And the "Foundation" part of the name is the name of the repository being promoted.

b Select one of the radio buttons in the **`Promotion Action`** area. If you select the top button, the job will not be run when the wizard exits; a definition of the job will be saved in the metadata, and you can run the job later. If you select the bottom button, the job will be run when the wizard exits *and* a definition of the job will saved so that you can run it later.

Click **`Next`**.

11 The wizard's last page displays all of the information that you have specified in the wizard.



To make changes, click **Back** until you reach the appropriate page. If all of the information is correct, then click **Finish** to create the replication job definition and to run the job, if you elected to do so.

Running a Promotion Job

To run a promotion or replication job that has already been created, follow these steps in SAS Management Console:

1. In the SAS Management Console tree, expand the **Metadata Manager** and **Job Definitions** nodes. Under the **Job Definitions** node, you see **Promotion** and **Replication** icons.
2. Select the **Promotion** or **Replication** icon. If you select the **Promotion** icon, you see a list of promotion jobs in the right pane, and if you select the **Replication** icon, you see a list of replication jobs.
3. Right-click a promotion or replication job in the right pane; then select **Run Job** from the pop-up menu.

Making Modifications after Running a Promotion Job

After you run a promotion job, you must make some manual changes to the following:

- the script used to start the SAS/CONNECT spawner
- the promoted metadata

The following sections explain how to make these modifications.

Note: You do not need to modify the metadata after running a replication job. △

Modifications to the SAS/CONNECT Server

On the target machine, you must stop the SAS/CONNECT spawner, edit the **ConnectServer.***ext* script, and then restart the spawner.

On Windows systems, perform the following steps:

1 Modify the **ConnectServer.bat** file, which is typically located in *SAS-config-dir***Lev2\SASMain\ConnectServer**.

   a Locate the **USEMETADATA** option, and change the line containing it to read as follows:

```
USEMETADATA=1
```

   b If you are going to install the SAS/CONNECT spawner as a service, locate the **IsService** option and change the line containing it to read as follows:

```
IsService=1
```

   c Save your changes.

2 To install the SAS/CONNECT spawner as a service, run the script **InstallConnectServer.bat**. This script is located in the same directory as the **ConnectServer.bat** script.

3 Start the SAS/CONNECT spawner.

   □ If you installed the SAS/CONNECT spawner as a service, start the service using either the Start menu or the Control Panel.

   □ If you did not install the spawner as a service, run the script **StartConnectServer.bat**.

On UNIX systems, perform the following steps:

1 In the file **ConnectServer.sh**, locate the line that sets the value of the **USEMETADATA** option, and change it so that it reads as follows:

```
set USEMETADATA=1
```

2 Save your changes to **ConnectServer.sh**.

3 Start the spawner by running the script **ConnectServer.sh**. Use the following command:

```
ConnectServer.sh start
```

Modifications to Server Definitions

Perform the following tasks on the metadata-server host in the target system:

1 In SAS Management Console, connect to the target metadata server using the credentials of the SAS Administrator (sasadm).

2 In the SAS Management Console tree, expand the **Server Manager** and, under that node, any SAS application servers.

3 If a SAS application server contains a logical workspace server, expand the logical workspace server, and perform the following steps for the physical workspace server(s):

   a Right-click the icon for the workspace server, and select **Properties** from the pop-up menu. A Workspace Server Properties dialog box appears.

   b In the Workspace Server Properties dialog box, select the **Options** tab.

   c Update the command in the **Command** text field, as appropriate. On a Windows system, enter the following command:

```
sas -config "SAS-config-dir\Lev2\SASMain\sasv9.cfg"
```

   On a UNIX system, enter the following command:

```
SAS-config-dir/Lev2/SASMain/sas.sh
```

   d Click **OK**.

4 If a SAS application server contains a logical stored process server, expand the logical stored process server, and perform the following steps for the physical stored process server(s):

   a Right-click the icon for the stored process server, and select **Properties** from the pop-up menu. A Stored Process Server Properties dialog box appears.
   b In the Stored Process Server Properties dialog box, select the **Options** tab.
   c Update the command in the **Command** text field, as appropriate. On a Windows system, enter the following command:

```
sas -config "SAS-config-dir\Lev2\SASMain\StoredProcessServer\
  sasv9_StorProcSrv.cfg"
```

   On a UNIX system, enter the following command:

```
SAS-config-dir/Lev2/SASMain/StoredProcessServer/sas_SPS.sh
```

   d Click **OK**.

5 If a SAS application server contains a logical OLAP server, expand the logical OLAP server, and perform the following steps for the physical SAS OLAP Server:

   a Right-click the icon for the OLAP server, and select **Properties** from the pop-up menu. A properties dialog box appears.
   b On the **Options** tab, click **Advanced Options** to open the Advanced Options dialog box.
   c On the **Performance** tab, edit the value of the **Path for temporary working files** field, if necessary. If the path in this field is a relative path, leave the value unchanged. If it contains a full path, change the value so that it points to a work directory on the target machine.
   d If you made changes on the **Performance** tab, click **OK** to save your changes; otherwise, click **Cancel**. You are returned to the properties dialog box.
   e Click **OK**.

6 If a SAS application server contains a logical SAS/CONNECT server, expand the logical SAS/CONNECT server, and perform the following steps for the physical SAS/CONNECT server:

   a Right-click the icon for the SAS/CONNECT server, and select **Properties** from the pop-up menu. A Connect Server Properties dialog box appears.
   b In the Connect Server Properties dialog box, select the **Options** tab.
   c Update the command in the **SASCMD Option** text field, as appropriate. On a Windows system, enter the following command:

```
SAS-config-dir\Lev2\SASMain\sasconnect.bat
```

   On a UNIX system, enter the following command:

```
SAS-config-dir/Lev2/SASMain/sasconnect.sh
```

   d Click **OK**.

7 If a SAS application server contains a SAS/SHARE server, perform the following steps for the this server:

   a Select the icon for the SAS/SHARE server. Information about a SAS/SHARE connection appears in the righthand pane.
   b Right-click the connection name, and select **Properties** from the pop-up menu. A properties dialog box appears.
   c In the properties dialog box, select the **Options** tab.
   d On the **Options** tab, change the values for the **Server Host**, **Remote Session ID**, and the **Server ID** to values that are appropriate for the machine that is hosting the target metadata server.

e Click **Connection Information Options** and, if necessary, change the **Host Name** value to the machine that is hosting the target metadata server. Click **OK** to return to the properties dialog box.

f Click **OK** in the properties dialog box.

8 If a SAS application server contains a logical SAS Data Step Batch Server, expand the logical batch server, and perform the following steps for the physical batch server:

a Right-click the icon for the batch server, and select **Properties** from the pop-up menu. A SAS Data Step Batch Server Properties dialog box appears.

b In the SAS Data Step Batch Server Properties dialog box, select the **Options** tab.

c In the **Command Line** text field, enter the following command:

```
SAS-config-dir\Lev2\SASMain\BatchServer\sasbatch
```

d In the **Logs Directory** text field, enter the following command:

```
SAS-config-dir\Lev2\SASMain\BatchServer\logs
```

e Click **OK**.

Modifications to Users

If necessary, modify the logins for local users who were promoted to the target system.

1 In SAS Management Console, select the **User Manager**. A list of users and groups appears in the righthand pane.

2 Right-click a user or group, and select **Properties** from the pop-up menu. A properties dialog box appears.

3 In the properties dialog box, select the **Logins** tab.

4 Modify any logins that contain a user ID that was valid on the source system, but is not valid on the target system. For example, you must change the user IDs for local Windows users so that they contain the correct hostname.

Modifications to Content Mapping for Reports

If your metadata repository includes report objects, then you must modify the content mapping properties of the root folder that contains the reports. Content mapping establishes the location of the XML (.srx) files that define the reports. The location can be either a path on a WebDAV server or a path on the file system.

To modify the content mapping properties:

1 In SAS Management Console, identify the root folder under BI Manager that contains reports. In default installations, this root-level folder is called **BIP Tree**.

2 Right-click this folder, and then click **Properties**.

3 On the **Content Mapping** tab, enter information about the target system's report content. If you specify a WebDAV server location, you must enter the user ID and password for the SAS Web Administrator user in the appropriate fields.

Note: The Promotion Wizard does not copy your report content to the target system. For information about copying report content and other BI content from the source system to the target system, see "Moving Associated BI Content Following a Promotion Job" on page 130. △

Modifications to Foundation Services

If you are using SAS Web Report Studio and SAS Information Delivery Portal at your site, you must use the Foundation Services Manager plug-in to SAS Management Console to delete—and then reimport—the following service deployments:

- ID Portal Local Services
- Query and Reporting
- Remote Services

To update your service deployment metadata, follow these steps:

1. In SAS Management Console, expand the **Foundation Services Manager** node. A list of service deployments appears.
2. For each of the service deployments listed above, right-click the service deployment name, and select **Delete** from the pop-up menu.
3. Right-click the **Foundation Services Manager**, and select **Import Service Deployment** from the pop-up menu. An Import Service Deployment dialog box appears.
4. Click **Add** in this dialog box. A file browser (an Open dialog box) appears.
5. Use the file browser to display the contents of the directory *SAS-config-dir***\Lev2\web\Deployments\Portal**.
6. Select the following files and click **Open**.
   - **sas_services_idp_remote_omr.xml**
   - **sas_services_idp_local_omr_xml**

   The names of these files will be displayed in the Import Services Deployment dialog box.
7. In the same way, add the file *SAS-config-dir***\Lev2\web\Deployments\WebReportStudio** *hostname***_sas_pfs_queryandreporting.xml** to the list.
8. Click **OK** in the Import Service Deployment dialog box.

Modifications to Cubes

If you promoted cube metadata from the source metadata server to the target metadata server, you must modify the work path, index path, and data path for each promoted cube. The following instructions explain how to use SAS Data Integration Studio to edit a cube's structure:

1. Use SAS Data Integration Studio to connect to the target metadata server. You can use the login information for any user who has WriteMetadata access to the cube metadata that must be modified.
2. In the SAS Data Integration Studio inventory tree, select the cube.
3. To launch the Cube Designer wizard, right-click the cube name and select **Edit Cube Structure**.
4. On the General window, change the **Work Path** to the correct path for the target machine.
5. Click **Next** until the Generated Aggregations window appears. On that window, click **Advanced** to open the Performance Options dialog box. In this dialog box, make these changes on the **Global** tab:
   - **a** Change the value of the **Location of index component files** to the correct path for the target machine.
   - **b** Change the value of the **Location of partitions in which to place aggregation table data** to the correct path for the target machine.
   - **c** Click **OK** to close the dialog box and return to the Cube Designer wizard.
6. Click **Next** until the Finish window appears.
7. Depending on whether you want to save the metadata and rebuild the cube or just save the metadata, select the appropriate radio button.

8 Click **Finish**.

What Happens When You Run a Promotion Job?

This section presents a more in-depth view of what happens when you run a promotion job than the one the we have presented so far. It should be of interest if you are debugging a promotion problem or are interested in customizing such a job.

When you run a promotion job, the Promotion Wizard submits generated code to the workspace server on the source machine. This code performs the following actions:

1 It establishes a connection to the metadata server on the source machine, which in turn establishes a connection to the metadata server on the target machine.

If the job fails at this point, check the following items:

a Open the file *SAS-config-dir***Lev1\SASMain sasv9.cfg**, and make sure that the values of the **metaserver**, **metaport**, and **metaprotocol** options are correct.

b Open the file *SAS-config-dir***Lev1\SASMain\MetadataServer\MetadataServer.***ext**, and make sure that the **OMAADMIN** and **OMAAPW** variables hold the user name and password of the source administrator. On Windows, this is typically the SAS Administrator (sasadm), and on UNIX, it is typically the SAS installer account (sas).

c Make sure that the file **mdrepljs.***ext* exists in the directory *SAS-config-dir***Lev1\SASMain**.

d Open the file **mdrepljs**.*ext*, and make sure that it contains the information needed to establish a connection to the metadata server on the administration machine.

2 It retrieves the three logins for the SAS Replication Administrator from the metadata repository on the administration machine.

If the job fails at this point, the most likely cause is that the users identified by these logins have not been defined as administrative or unrestricted users on the appropriate machine. (See Figure 6.2 on page 113.)

- The user referred to in the first login must be an administrative user on the administration machine.
- The user referred to in the second login must be an administrative user on the source machine.
- The user referred to in the third login must be an unrestricted user on the target machine.

3 It connects to the source metadata server to ensure that the repository to be promoted still exists.

If the repository does not exist and you are rerunning the job from outside the Promotion Wizard, the job will fail.

4 It connects to the target metadata server to determine whether a repository with the same object ID as the source repository exists.

If such a repository does not exist and you are rerunning your job outside of the Promotion Wizard, the job will fail.

5 It validates the substitution tables defined for the repository promotion. For all four substitution tables (Hosts, Ports, Schema, and Paths), the job retrieves the substitution values from the metadata on the administration server. It then gets the values that will be replaced from the source metadata server and matches/merges the substitutions with the current metadata values. Finally, the job

creates an XML file that it will use later when it applies changes to the metadata on the target machine.

At the end of this step, a macro called **mduval.sas** is run. If you want to perform any custom validation, you should add your code here.

6 It performs any optional preprocessing.

By default, no preprocessing occurs. You can run custom preprocessing code on the source metadata server host by adding code to the macro **mdpresrc.sas**.

7 It runs PROC DATASETS to delete any data sets in the work directory on the source machine.

8 It pauses the source metadata server.

9 It uses PROC COPY to copy the repository from its location at *SAS-config-dir***Lev1\SASMain\MetadataServer\MetadataRepositories***repository-name* to *SAS-config-dir***Lev1\SASMain\MetadataServer\ReplicationWorkArea\work**.

A failure at this point indicates that the source administrator does not have the necessary file-system permissions to copy the repository.

10 It resumes the source metadata server.

11 It calls the macro **mdpstsrc.sas**. By default, this macro does nothing. However, you can add code to it to perform any desired postprocessing on the source metadata server host.

12 It performs any optional (user-written) preprocessing on the target server. You can specify such preprocessing by adding code to the macro **mdpretrg.sas**.

13 It pauses the target metadata server.

14 It copies the metadata being promoted from the source machine to the target machine.

During this step, the job does the following:

a Uses PROC UPLOAD to copy the repository from the work directory on the source machine to a work directory on the target machine

b Cleans out the backup directory on the target machine and then moves the existing repository (if any) to the backup directory

c Cleans out the target repository location and then copies the repository being promoted from the work directory on the target machine to its final location

Any errors that occur at this stage in the process are typically the result of the target administrator not having the necessary file-system permissions.

15 It resumes the target metadata server.

16 It performs substitutions on the target metadata based on the contents of the XML file mentioned earlier.

17 It performs any optional postprocessing on the target machine by executing the macro **mdpsttrg.sas**. By default, this macro is just a stub; however, you can add code to it to perform custom postprocessing.

Moving Associated BI Content Following a Promotion Job

About Moving BI Content

Since the Promotion and Replication Wizards promote only your metadata, you must use other methods to move the BI content that is associated with your metadata. This content includes the following:

- WebDAV folders and their contents (for example, report definition files and portal content). If you are using Xythos WebFile Server (Xythos WFS), then you can use utilities that are delivered with Xythos WFS to promote this content. For instructions, see "Moving Xythos WebFile Server Content" on page 131.
- other portal content, including custom portlets, custom themes, and applications. For instructions, see "Moving Portal Content" on page 132.
- source files for stored processes. Using an appropriate utility or operating system command, move these files to the paths that you specified for stored process source code when you ran the Promotion Wizard.
- physical tables and databases. Using an appropriate utility or operating system command, move these files to the locations that you specified as your library paths when you ran the Promotion Wizard.

Moving Xythos WebFile Server Content

If you are using Xythos WFS, then you can use the WFSDump and WFSRestore utilities to move your WebDAV content. These utilities copy both the WebDAV content and the associated WebDAV properties, including quotas and permissions.

Follow these steps:

1. On the source environment's Xythos WFS machine, navigate to the path where the utilities are installed:
   - For Xythos WFS 4.2.34, the path is *drive***:\xythos\custom\bin** (Windows) or *install-location***/xythos/custom/bin** (UNIX).
   - For Xythos WFS 4.0.48, the path is *drive***:\xythos\wfs-4.0.48\utils** (Windows) or *install-location***/xythos/wfs-4.0.48/utils** (UNIX).
2. Use a text editor to open the file **WFSDump.bat** (Windows) or **WFSDump.sh** (UNIX), and edit the file as follows:
   - For Xythos WFS 4.2.34, change the value of XYTHOS_HOME to the appropriate value for your environment.
   - For Xythos WFS 4.0.48, change the values of XYTHOS_HOME and JDBC_DRIVER to the appropriate values for your environment.

   Then save the file.
3. Create an empty directory in your file system for the utility to store the output files.
4. At an operating system prompt, enter the following command:

   ```
   WFSDump path
   ```

   where *path* is the path for the directory that you created in the previous step.

   In this directory, WFSDump writes a file for each unique content resource that it finds on Xythos WFS. It also creates a file called **WFSDump.xml**, which contains metadata for the content resources.
5. If the directory is not accessible from the target machine, copy the directory and its contents to a location that is accessible from the target machine.
6. On the target machine, use a text editor or XML editor to open **WFSDump.xml**. To this file, make any changes that are necessary to reflect the target environment's Xythos WFS installation. At a minimum, you will need to edit the machine names that are included in domain-qualified user IDs. Do not, however, edit machine names that represent the WebDAV server, because WFSRestore will modify those. For example, in the expression

   ```
   principal='machine-name\sasdemo@machine-name'
   ```

you would edit the first instance of *machine-name*, but not the second instance.

7 On the target machine, navigate to the path where the utilities are installed, as described in step 1. Use a text editor to open the file **WFSRestore.bat** (Windows) or **WFSRestore.sh** (UNIX), and edit the file as follows:

- For Xythos WFS 4.2.34, change the value of XYTHOS_HOME to the appropriate value for your environment.
- For Xythos WFS 4.0.48, change the values of XYTHOS_HOME and JDBC_DRIVER to the appropriate values for your environment.

8 Enter the following command:

```
WFSRestore path
```

where *path* is the path for the directory that contains **WFSDump.xml** and the content resource files.

WFSRestore writes the folders and files to the target Xythos WebFile Server. For each resource, the utility assigns the properties that are specified in **WFSDump.xml**.

Moving Portal Content

To finish moving portal content following a promotion job, you might need to perform the following tasks on the middle tier of the target environment:

- deploy custom themes
- deploy PAR files for custom portlets
- deploy Web applications that are associated with custom remote portlets
- deploy other applications that are called by the portal
- move WebDAV content (see "Moving Xythos WebFile Server Content" on page 131)

For instructions on deploying portal themes, portlets, and applications, see the *SAS Web Infrastructure Kit: Administrator's Guide* at **http://support.sas.com/rnd/itech/library/toc_portaladmin.html**.

CHAPTER

7

Promoting Individual BI Objects

Overview of Promoting Individual BI Objects

Wizards for Promoting Individual BI Objects

SAS Data Integration Studio 3.4 and SAS Management Console provide a common set of wizards for importing and exporting metadata in SAS Open Metadata Architecture format. You can use the Export Wizard and the Import Wizard in these applications to selectively promote the following types of metadata objects:

- data explorations
- documents
- external files
- folders
- generated transformation

- information maps
- jobs
- libraries
- mining results
- notes
- reports
- stored processes
- tables

With the Export Wizard and Import Wizard, you can promote individual objects or groups of objects from one repository to another, either on the same host machine or on different host machines. The wizards are useful in the following situations:

- You want to move newly developed (or newly modified) metadata objects and their associated content from a test environment to a production environment.
- You want to distribute metadata objects among repositories that belong to different organizational entities.

The source metadata server and the destination metadata server can run on host machines that have different operating systems. For example, you can import objects into a metadata repository that is running on a UNIX system even though those objects were exported from a Windows system.

The Export Wizard and Import Wizard work with dependent repositories as well as foundation repositories. For example:

- You can export objects from a foundation repository, and then import them to a custom repository that is dependent on the foundation repository.
- You can export objects from a dependent repository, and then import them to the foundation repository.

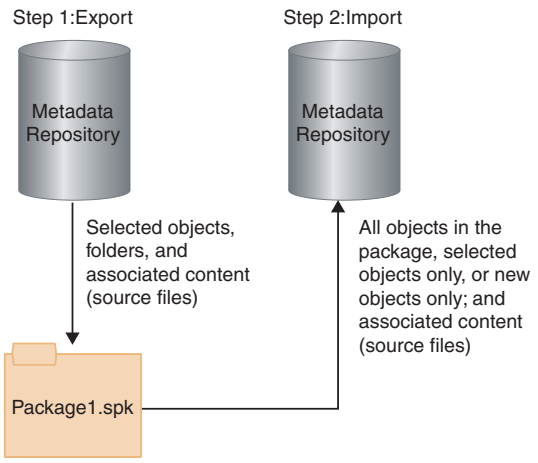
Note: SAS Web Report Studio cannot access metadata that is stored in a dependent repository. △

Overview of the Promotion Process

Promotion is a two-step process:

- First, you export the selected objects or folders from the source metadata repository.

  The Export Wizard places the exported objects in a SAS Package (SPK) file. If you want the SPK file to be available remotely, then you can instruct the wizard to place the package in a shared location within your file system. You can also e-mail the package after it has been created.
- Second, you import the SPK file to another metadata repository, which can be located either on the same host machine or a different host machine. You can use either the same instance of SAS Management Console or SAS Data Integration Studio, or a different instance to perform the import.

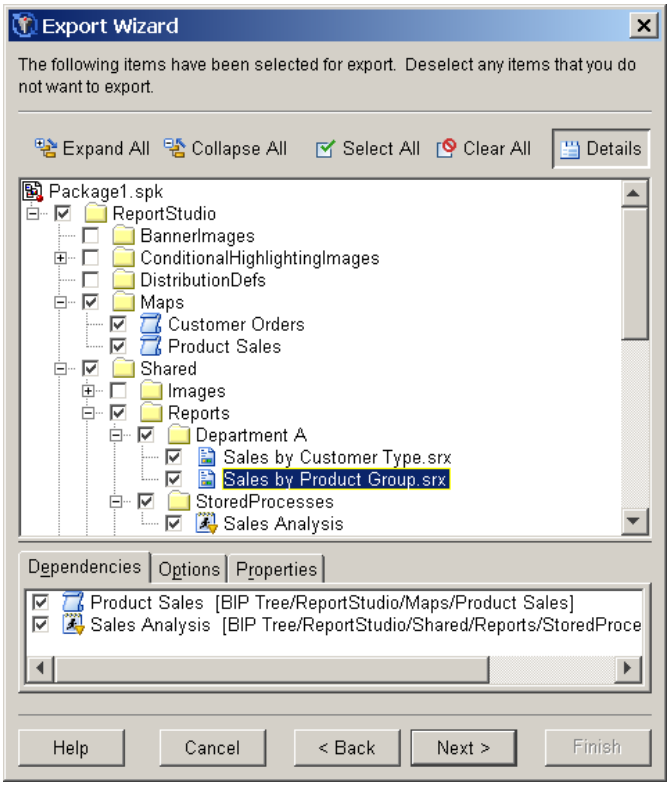


The package format is the same regardless of whether the package is created by SAS Management Console or SAS Data Integration Studio. A package that is exported using SAS Management Console can be imported using SAS Data Integration Studio, and vice versa.

Selective Promotion

The wizards provide the following options for selecting objects to export or import:

- You can select multiple nested folders.
- In SAS Data Integration Studio, you can select an entire repository.
- You can include all of the objects that are contained in a folder (or repository), or just selected objects, as shown in the following example:



- As shown in the preceding example, you can highlight one or more objects in the Export Wizard and view their dependencies. The wizard identifies the following types of dependent objects and enables you to select which ones you want to include in the export:

Table 7.1 Object Dependencies Displayed by the Export Wizard

Object	Dependencies
Information map	Libraries, tables, and stored processes that are used by the information map
Library	Tables that are contained in the library
Job	Nested jobs, mining results, tables, and external files that are used by the job
Report	Information maps and stored processes that are used by the report

- When importing a package, you can import all objects in the package or just new objects (that is, objects that do not already exist in the destination metadata repository).

Promotion of Content Along With Metadata

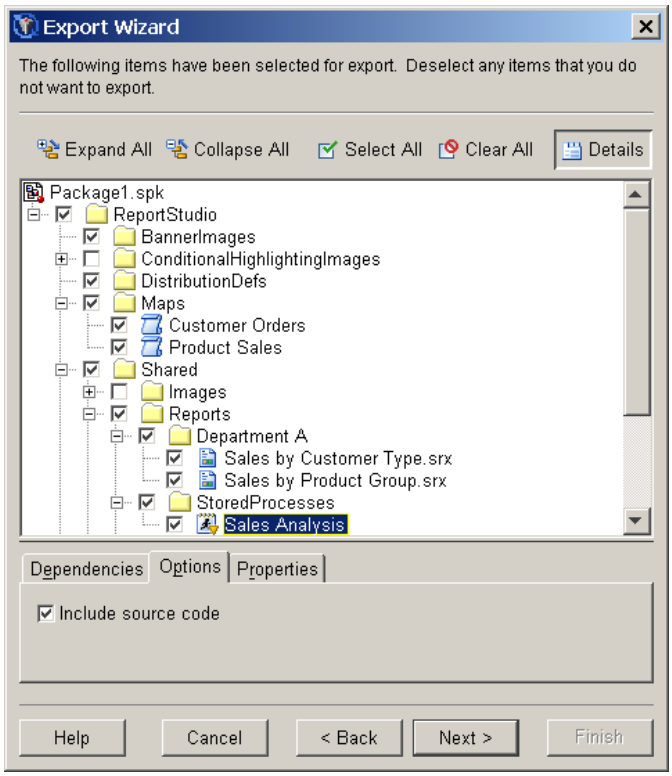
For most object types, the wizards export and import both the content and the metadata. Some content is promoted automatically with the metadata, and other content is promoted optionally, as described in the following table.

Table 7.2 Promotion of Content with Metadata Objects

Object	Associated Content	Promotion Details
External file	Physical file	Promoted optionally
	External format file (EFF)	Promoted optionally
Job	Custom code, preprocesses, and postprocesses	Promoted automatically
Library	XML files and XML maps that are contained in the library	Promoted optionally
Report	XML (.srx) file that defines the report	Promoted automatically to a target WebDAV server or file system path that you designate

Object	Associated Content	Promotion Details
Stored process	Source code (.sas) file	Promoted optionally to the source code repository that you designate. A metadata definition for the source code repository must be present on the target system.
Table	Physical table	Promoted optionally. Use this option with small data sets only. Since data sets are packaged in the .spk file, large data sets could require lengthy processing times.

Content that is promoted optionally is displayed in the wizards' **Options** tab. In the following example, a stored process object is highlighted, and the **Options** tab displays a check box for including the source code with the stored process. The check box is selected by default.



When you export or import an external file, a library, or a table, the option to include the content is not selected by default.

Restoration of Metadata Associations

The Import Wizard displays prompts that assist you in associating imported objects with the appropriate objects and entities in the target environment. Depending on the

types of objects that you are importing, you might be prompted to establish associations with the following objects and entities on the target system:

- a SAS Application Server
- a WebDAV server
- base path on a WebDAV server
- libraries
- tables
- external files
- physical locations for external files or for libraries
- OLAP schemas
- mining results
- a source code repository and an archive path (for stored processes)

You can choose to restore the same metadata associations that were established in the source environment, or you can establish different associations.

Optional Promotion of Access Controls

You can include direct assignments of access control entries (ACEs) and access control templates (ACTs) in the promotion process. The following limitations apply to the promotion of access controls:

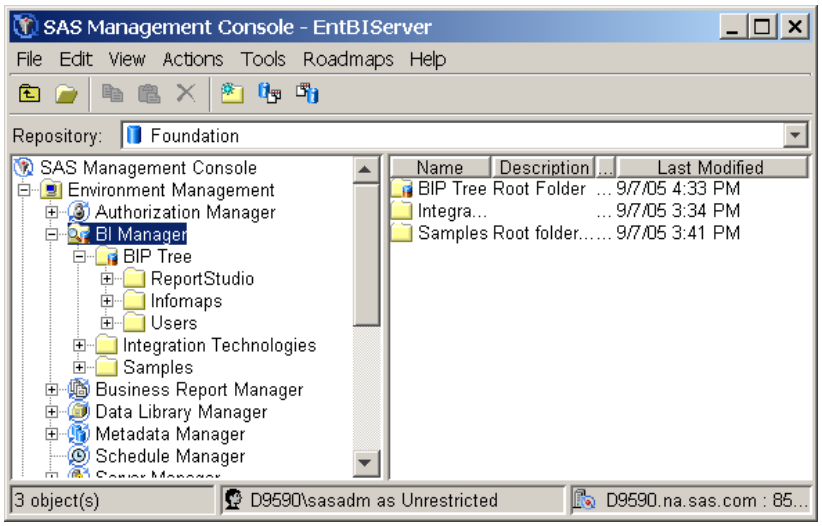
- The wizards cannot promote ACTs, but they can promote direct associations between ACTs and objects. In order for an object's ACT association to be promoted, an ACT of the same name must exist in the destination metadata repository.
- Access control entries and ACT associations that are inherited from a folder are not promoted. For inheritance to take effect in the destination metadata repository, you must also promote the folder that contains the direct access control entries or ACT association.
- Access control entries are not promoted if they refer to users or groups that do not exist in the destination metadata repository (or in the parent Foundation repository, if the destination is a dependent repository).

If you do not specify that you want to include access controls in the promotion process, then access controls are applied as follows:

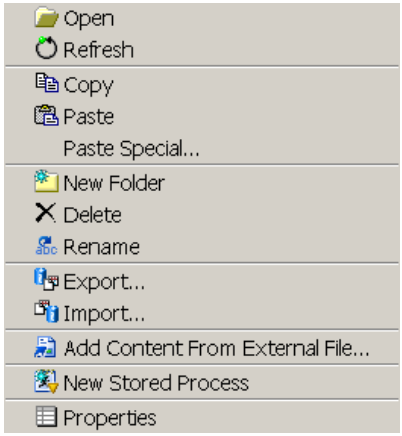
- If you import an object that already exists in the destination metadata repository, then the permissions that have been applied to the existing object are preserved when you overwrite the object.
- If you import an object for the first time, then the object inherits permissions from the folder into which the object is imported.

How to Access the Export and Import Wizards

To access the Export Wizard and Import Wizard in SAS Management Console, navigate to the **`BI Manager`** node, as shown here:



Right-click any folder or object under the **BI Manager** node (or highlight a folder or object, and then click **Actions** in the menu bar). The applicable BI Manager actions are displayed in a menu, as shown here:



To access the Export Wizard and Import Wizard in SAS Data Integration Studio, right-click any repository, folder, or object in either the Custom tree or the Inventory tree. Or you can highlight a repository, folder, or object in the Custom tree or the Inventory tree, and then click **File** in the menu bar. The menu that appears will include the **Export** and **Import** options, if they are applicable to the object that you have selected.

Special Considerations

Before choosing to use the Export Wizard and Import Wizard to promote metadata, you should be aware of the following important considerations:

- The wizards can be used to promote only the following metadata objects: data explorations, documents, external files, folders, generated transformations, information maps, jobs, libraries, mining results, notes, reports, tables, and stored processes.

- In order for objects to function properly in the target environment, you must import the resources that objects depend on, unless those resources already exist in the target environment. For example, if you want reports to function properly, you must import the information maps that the reports depend on. If a report has stored processes or images associated with it, then you must import those objects in order for the report to function properly.
- The wizards do not export or import the servers, database schemas, source code repositories, or archive paths that are associated with the BI objects. During the import process, the wizard prompts you to associate the imported objects with the appropriate servers, database schemas, source code repositories, and archive paths in the target environment.
- If you choose to export physical tables along with metadata, you should be aware that large data sets could require lengthy processing times (since data sets are packaged in the .spk file along with the metadata).
- When you export physical tables along with DBMS table objects, do not log on as an unrestricted user (for example, **sasadm**). Unrestricted users cannot access the servers that are necessary to export physical DBMS tables.

Preparing to Use the Export and Import Wizards

Set Up Servers and Start Spawners

Before using the Export Wizard and Import Wizard in BI Manager or SAS Data Integration Studio, make sure that the appropriate servers are set up and running, as follows:

- The object spawner must be running in the source environment.
- If you are promoting stored processes, then a stored process server or workspace server (depending on which type of server is used to run the stored process) must exist in the target environment.
- If you are promoting tables, then a workspace server must exist and an object spawner must be running in the target environment.
- If your stored processes create packages that are stored permanently in a WebDAV location, then a WebDAV server must exist in the target environment.
- If you are promoting reports and want to store report content on a WebDAV server, then a WebDAV server must exist and must be running in the target environment.
- The user who launches the Export Wizard must have ReadMetadata access to the objects that are being exported. The user who launches the Import Wizard must have WriteMetadata access to the objects that are being imported. The user must also have the appropriate access to associated content that is being exported and imported.

In SAS Management Console, Collect Tables and Libraries under BI Manager

If you want to use SAS Management Console to export libraries and tables, the tables must appear in the BI Manager tree. To make them appear under the BI Manager tree, follow these steps:

1. Right-click the **BI Manager** node.
2. On the menu that appears, click **Collect Ungrouped Library Metadata**.

BI Manager places pointers to your libraries and tables in a folder called **`Shared Data`**. From this folder, you can export and import library and table objects.

You only need to complete this procedure once. From this point forward, new library and table objects that you create in Data Library Manager will automatically appear in the BI Manager tree after you click **View ▶ Refresh**. The library and table objects will also continue to appear under the **`Data Library Manager`** node. If you delete any of the objects from the **`Shared Data`** folder, the objects will also be deleted from the Data Library Manager.

Set Up Folders in the Destination Metadata Repository

If you will be using folders to control access to metadata, then you might want to set up folders in the destination metadata repository before you import your metadata. If you use a folder structure that corresponds to the folder structure in the source metadata repository, then you should import objects at the same level in the folder hierarchy that you exported them from.

For Reports, Set Up Content Mapping in the Destination Repository

Make sure that content mapping has been specified in the destination metadata repository if you are planning to promote reports. Content mapping establishes the location of the XML (.srx) files that define the reports. The location can be either a path on a WebDAV server or a path on the file system.

To specify content mapping:

1. In SAS Management Console, identify the root folder under BI Manager that contains reports. In default installations, this root-level folder is called **`BIP Tree`**.
2. Right-click this folder, and then click **`Properties`**.
3. On the **`Content Mapping`** tab, enter information about the content location. If you specify a WebDAV server location, you must enter the user ID and password for the SAS Web Administrator user (saswbadm) in the appropriate fields.

Set Up Security for the Folders or Objects

To implement security for promoted objects, you can choose from a variety of options, including the following:

Option 1: Apply ACTs and ACEs to folders and objects in the source environment, and promote the access controls along with the folders and objects.

This option assumes that you have done the following in the source environment:

1. You have set up user groups that mirror the groups that will exist in the target environment, and you have defined a small number of users in each group for testing purposes.
2. You have created ACTs that define the permission levels for each user group, applied the ACTs to objects and folders, and tested the ACTs to ensure that the appropriate level of security is applied.
3. Instead of (or in addition to) ACTs, you have applied ACEs to folders and objects to define the permission levels for each user group, and you have tested the ACEs to ensure that the appropriate level of security is applied.

If you use this option, then your promotion process needs to include the following steps:

1. Create copies of the ACTs in the target environment. Be sure to assign the ACTs the same names in both environments.
2. When you export and import the objects, select the **Include access controls** option. The ACT associations and ACEs will be promoted and will be re-established in the target environment.

   Note: In order for the ACT associations and the ACEs to be promoted, you must promote the objects or folders to which the ACTs were directly applied. Inherited ACT associations and ACEs are not promoted. △

Option 2: Apply ACTs or ACEs to folders and objects directly in the target environment.

For this option, you need to define user groups, ACTs, and ACEs only in the target environment instead of in both environments. You do not select **Include access controls** in the Export Wizard and Import Wizard.

This option involves fewer steps, but it does not provide the opportunity to test your security settings before promotion.

For Stored Processes, Set Up a Source Code Repository and an Archive Path (If Needed) in the Target Environment

If you are promoting stored processes, then you must ensure that a location for the source code files (called a source code repository) exists in the target environment. A metadata definition for the source code repository must also exist in the destination metadata repository.

If the stored processes will create packages that are stored permanently in the file system, then you must ensure that an archive path for the packages exists in the target environment. A metadata definition for the archive path must also exist in the destination metadata repository.

Follow these steps to set up the source code repository and the archive path:

1. On the file system of the target application server, identify (or create) the physical directory where the source code files for stored processes will be stored. Ensure that the person who will run the Import Wizard has Write permissions for the source code directory.
2. If the stored processes will create packages that are stored permanently in the file system, then identify (or create) the physical directory for the archive path on the target application server.
3. *If the destination metadata repository already contains one or more stored process definitions,* then follow these steps to determine whether metadata definitions already exist for the source code repository and the archive path:

   a. In SAS Management Console, right-click any stored process object, and choose **Properties**.
   b. Click the **Execution** tab.
   c. Click the **Source code repository** list to see the list of source code repositories that have been defined.

      If a definition already exists for the source code repository that you plan to use, then you do not need to define one. If the definition does not exist, then you must add it as described in step 5.
   d. If your stored processes require an archive path, then complete these steps:

      i. On the **Execution** tab, select **Permanent package** from the **Output** list, and click **Details**.
      ii. On the Permanent Package Details dialog box, click the **Publish to server file system** radio button, and then click the **Archive path** list.

If a definition already exists for the archive path that you plan to use, then you do not need to define one. If the definition does not exist, then you must add it as described in step 6.

4 *If the destination metadata repository does not have any stored process definitions,* then you must create a stored process strictly for the purpose of creating the source code repository and archive path definitions. If you click **Cancel** at the end of this procedure, the source code repository definition and the archive path definition will be saved, but the stored process definition will not be saved. Follow these steps:

   a In SAS Management Console, right-click any folder under BI Manager. In the menu that appears, click **New Stored Process**.
   b In the **Name** field of the New Stored Process Wizard, type any name (for example, **Test**). Then click **Next**.
   c Select the SAS Server that is associated with the intended location of the source code files.
   d Use the procedure in step 5 to create the source code repository definition. Use the procedure in step 6 (if needed) to create the archive path definition.

5 To add a new source code repository definition, complete these steps either on the **Execution** tab of the Properties dialog box or on the second page of the New Stored Process Wizard:

   a Click **Manage**.
   b In the Manage Source Code Repositories dialog box, click **Add**.
   c In the Add Source Code Repository dialog box, type the fully qualified path for the directory. The path should point to the folder that you created in step 1. Click **OK**.
   d In the Manage Source Code Repositories dialog box, click **OK**.

6 To add a new archive path definition, complete these steps either on the **Execution** tab of the Properties dialog box or on the second page of the New Stored Process Wizard:

   a If you have not already done so, select **Permanent package** from the **Output** list, and click **Details**.
   b On the Permanent Package Details dialog box, click the **Publish to server file system** radio button, and then click **Manage**.
   c In the Manage Archive Paths dialog box, click **Add**.
   d In the Add Archive Path dialog box, enter the fully qualified path for the folder that is to be used to store packages. The path should point to the folder that you created in step 2. You can also enter a description. Then click **OK**.
   e Click **OK** in the Manage Archive Paths dialog box and in the Permanent Package Details dialog box.

7 Click **Cancel** in the Properties dialog box or the New Stored Process Wizard.

Using the Export and Import Wizards: Example Usage Scenario

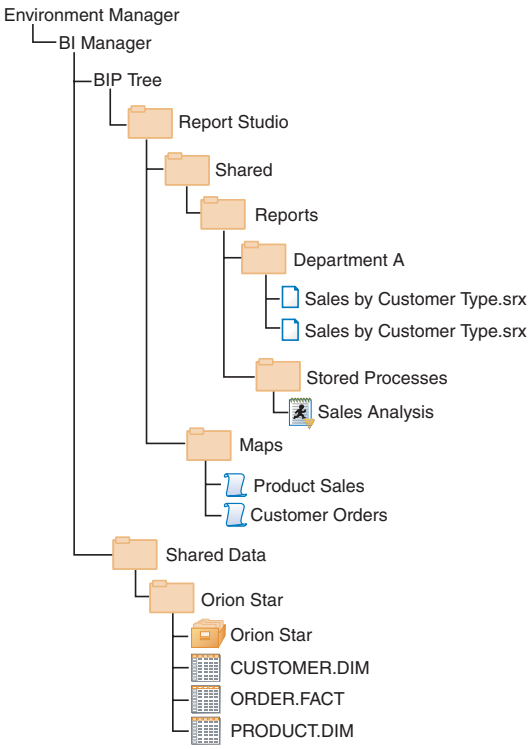
Description of Scenario

In this example, a set of objects has been developed using a SAS Metadata Repository in Environment 1. The objects will be promoted to a different SAS Metadata Repository in Environment 2. The objects to be promoted are as follows:

- a report called Sales by Customer Type

- a report called Sales by Product Type
- a stored process called Sales Analysis
- an information map called Customer Orders
- an information map, called Product Sales
- a library called Orion Data
- three SAS tables: CUSTOMER_DIM, PRODUCT_DIM, and ORDERS_FACT

The metadata for these objects is stored in the **`BIP Tree`** in SAS Management Console, as shown in this figure:



Note: This example scenario uses SAS Management Console to perform the promotion. If you use SAS Data Integration Studio, the procedure is the same except where noted. △

Before promoting these objects, it is important to understand their associations with one another, as well as their associations with other objects and entities that do not appear in the tree. The following table describes these associations. It also describes the preparatory actions that must be taken in the target environment to ensure that the associations are restored following the promotion.

Table 7.3 Example Scenario: Objects To Be Promoted and Their Associations

Metadata Object or Folder Name	Associated Objects or Content in the Source Environment	Action Needed to Prepare the Target Environment
Department A folder	Reports_ A ACT	Create the Reports_A ACT. Set up the user groups that the ACT refers to.
Sales by Customer Type report	Sales by Customer Type.srx file (on WebDAV server)	Ensure that a WebDAV server exists.
	Customer Orders information map	None
Sales by Product Group report	Sales by Product Group.srx file	Ensure that a WebDAV server exists.
	Product Sales information map	None
	Sales Analysis stored process	None
Stored Processes folder	Processes_ A ACT	Create the Processes_A ACT. Set up the user groups that the ACT refers to.
Sales Analysis stored process	Analysis.sas source file	Set up a source code directory in the file system, and apply appropriate permissions.
	source file repository definition	Ensure that a workspace server exists. Create a source file repository definition that corresponds to the source code directory and the workspace server.
	stored process server	Ensure that a stored process server exists.
Orion Star library	CUSTOMER_DIM, PRODUCT_DIM, and ORDER_FACT tables	Identify a physical location for the library.
CUSTOMER_DIM, PRODUCT_DIM, and ORDER_FACT tables	physical tables	None

These associations include the following:

- The **`Department A`** folder and the **`Stored Process`** folder are each associated with an ACT. If the export and import operations include access controls, then BI Manager will expect to find the ACTs already present in the destination metadata repository.
- Each of the reports is associated with an XML file (with the suffix .srx) on the WebDAV server. For these to be promoted, BI Manager will expect a WebDAV server to exist in the target environment.
- Each of the reports is associated with a different information map. The information maps must be promoted along with the reports in order for the reports to function properly.

- Each of the information maps is associated with specific tables. The table metadata must be promoted along with the information maps in order for the information maps to function properly. The physical tables must also be promoted, or must already exist in the target environment.
- One of the reports is associated with a stored process. The stored process must be promoted in order for the report to function properly.
- The stored process is associated with a source code file that is stored on the file system. For this to be promoted, BI Manager will expect a source code repository to exist on the target application server.
- The stored process is associated with a SAS application server and a source code repository definition. BI Manager will prompt you to establish an association between the stored process and an application server and a source code repository definition that exist in the target environment.

Prerequisite Actions

In the example steps that follow, it is assumed that the following preparatory actions have already been completed:

- The following physical items must be created (or already present) in the target environment:
  - a WebDAV server
  - a file path for the source code file for the stored process
  - physical tables that correspond to the tables in the source environment (or file paths for these tables, if they are being exported along with the metadata)
- In SAS Management Console, the library and table metadata must appear in folders under BI Manager (see "In SAS Management Console, Collect Tables and Libraries under BI Manager" on page 140). When you use SAS Data Integration Studio, this requirement does not apply.
- The following metadata must be created (or already present) in the destination metadata repository:
  - ACTs that correspond to, and have the same names as, the ACTs in the source metadata repository
  - user group definitions for the groups that the ACTs refer to
  - a server definition for the WebDAV server
  - an application server definition that includes a stored process server and a workspace server
  - a source code repository definition

Step 1: Export the Folders and Objects

Follow these steps to export the folders and objects from the source environment. These steps assume that you have already completed all of the preparatory actions that are described in the preceding section.

1. Open SAS Management Console, and connect to the source metadata repository (that is, the repository from which you want to promote the folders and objects).
2. Select the **`BI Manager`** node.

   Note: In SAS Data Integration Studio, you would select the repository that contains the objects that you want to export. △

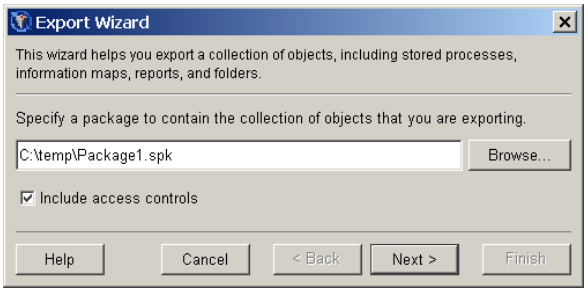
3 Right-click to display the menu, and click **Export**.

4 On the first Export Wizard page, complete the following steps:

   a Enter a name for the SPK file that will contain the exported objects. The default name is **Package1.spk**.

   b Click **Browse**, and navigate to the location on the file system where you want the SPK file to be saved. For convenience, you might want to choose a location that will be accessible from the target environment.

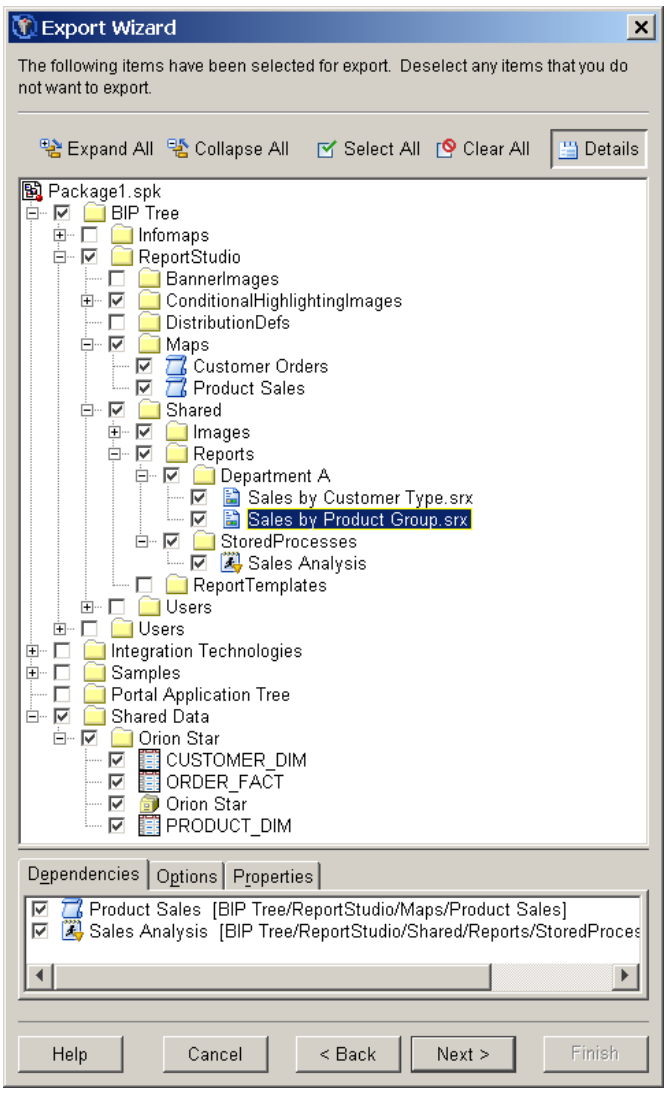
   c Click **Include access controls**. This option will retain the associations that have been established between folders and ACTS.



   d Click **Next**.

5 On the next Export Wizard page, do the following:

   a Select and unselect the checkboxes to specify which folders and objects are to be exported, as shown here. Be sure to select the folders whose access controls you want to promote.



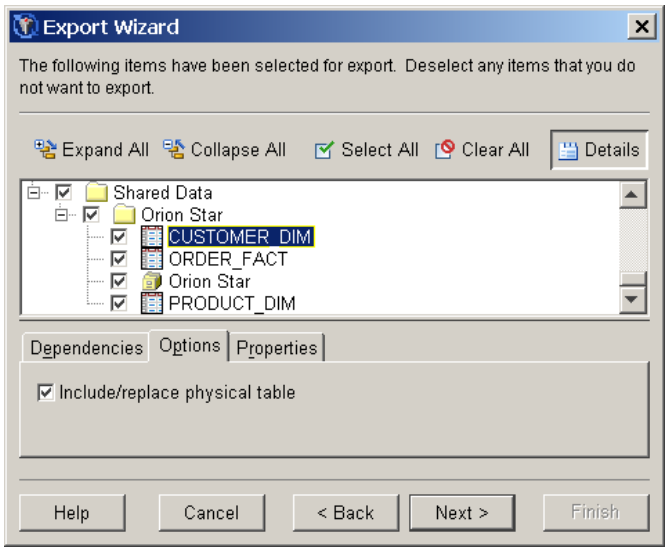
b Highlight each information map, report, and library object that you have selected. As you highlight each object, view the **Dependencies** tab to ensure that you have included the objects that the highlighted object depends on.

In the preceding example, the report **Sales by Product Group** is highlighted. The **Dependencies** tab lists the **Product Sales** information map and the **Sales Analysis** stored process, which are used by this report. Both objects have been selected to export.

Note: If you press the **Ctrl** key and highlight multiple objects, the **Dependencies** tab displays dependencies for all of the highlighted objects. △

c Highlight each of the table objects, and use the **Options** tab to specify whether the associated physical table is to be exported with the metadata.

In the following example, the **CUSTOMER_DIM** table is highlighted. On the **Options** tab, the **Include/replace physical table** check box has been selected to indicate that the physical table is to be included in the export.



Note:

- If you choose the option to export physical tables, you should be aware that large data sets could require lengthy processing times (since data sets are packaged in the .spk file along with the metadata).
- If you press the **Ctrl** key and highlight multiple objects of the same type, the **Options** tab enables you to select options for all of the highlighted objects.

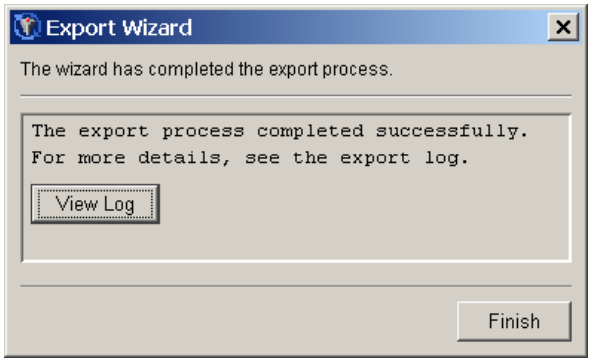
△

d After you have selected all of the appropriate metadata objects, dependencies, and options, click **Next**.

6 The next Export Wizard page displays a list of the objects and folders that you selected. If the list is correct, click **Export**.

7 As the export process begins, BI Manager might display a prompt asking you to log on to the application server. Enter your SAS administrator's user ID and password at the prompt.

8 When the export process finishes, a log with a date and time stamp is saved in your user directory and a message like the following is displayed:



If errors or warnings are present, you can click **View Log** to view explanatory messages. For assistance in resolving errors, see "Troubleshooting the Export and Import Wizards" on page 152.

Note: For complete step-by-step instructions for using the Export Wizard, see the product Help. △

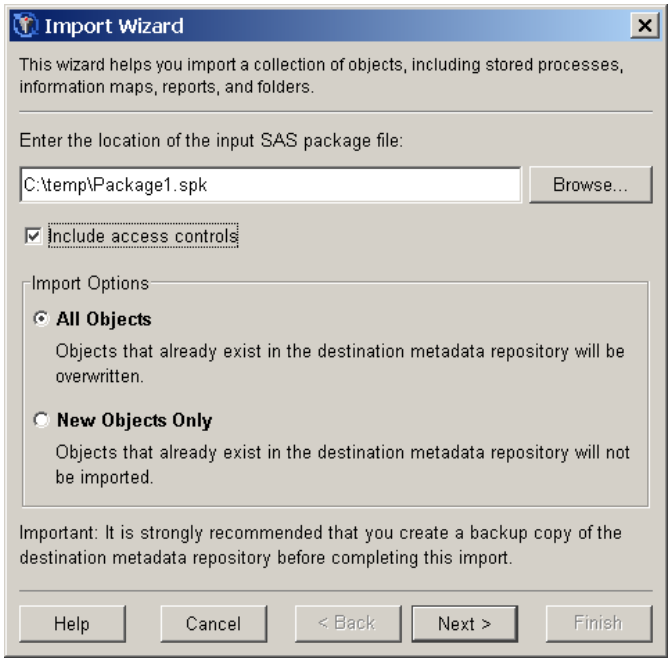
Step 2: Import the SPK File

Follow these steps to import the folders and objects from the SPK file into the target environment. These steps assume that you have completed Step 1 and that no errors occurred. These steps also assume that you have completed all of the preparatory actions.

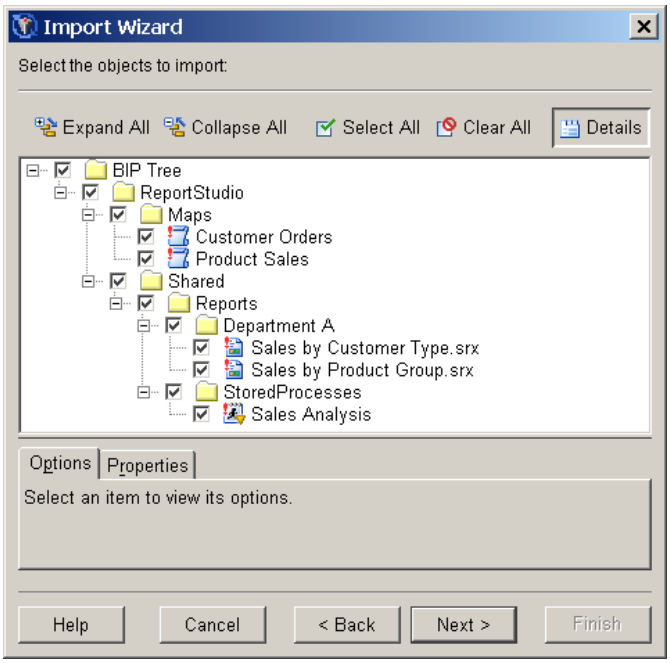
1. Locate the SPK file that was created by the Export Wizard. If necessary, copy it, move it, or e-mail it to a location that is accessible from the target environment.
2. Open SAS Management Console, and connect to the destination metadata repository (that is, the repository to which you want to promote the folders and objects).
3. Select the **BI Manager** node.

   Note: In SAS Data Integration Studio, you would select the repository to which you want to import the objects. △
4. Right-click to display the BI Manager menu, and click **Import**.
5. On the first Import Wizard page, complete the following steps:

   a. Click **Browse**, and navigate to the SPK file that contains the folders and objects.
   b. Click **Include access controls**. This option will retain the associations that have been established between folders and ACTS.
   c. Click the **All Objects** radio button.



   d. Click **Next**.
6. The next Import Wizard page displays a tree which shows the objects that are included in the package.

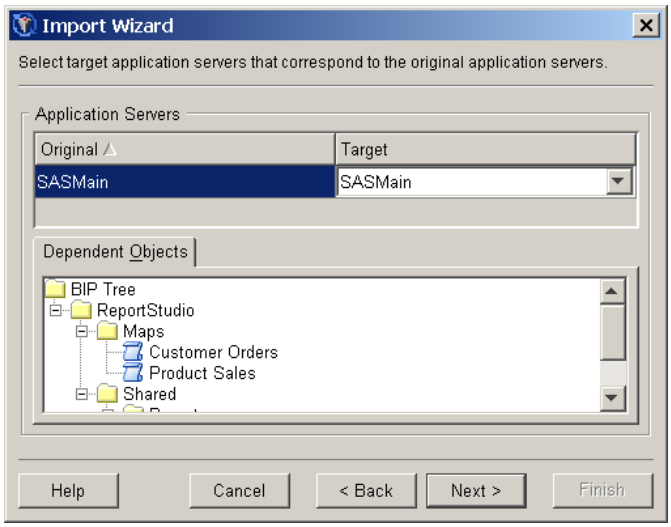


On this page, you can highlight the table objects, and then use the **Options** tab to indicate whether the physical tables are to be imported along with the metadata. You can also edit the table names. (These options are available only if you exported the physical tables.)

Then click **Next**.

7 The next Import Wizard page provides advance information about the metadata values that you will need to select on subsequent windows. For this scenario, these include values for application servers, library locations, and source code repositories. Click **Next**.

8 In the **Target** column of the next page, select the application server that the imported objects will be associated with in the target environment.



Click **Next**.

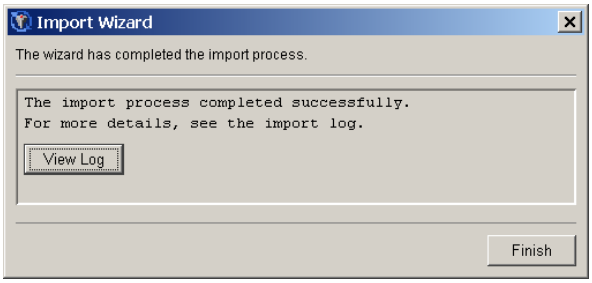
9 In the **Target** column of the next page, enter (or browse to) the path where the library is to be located in the target environment. Then click **Next**.

10 In the **Target** column of the next page, select the path where the source code file for the stored process should be stored in the target environment. This path is defined in the source code repository definition on the destination metadata repository. Then click **Next**.

11 The next page displays a list of the objects that will be imported. The metadata associations for these objects, both in the source environment and the target environment, are also listed. If the information is correct, click **Import**.

12 As the import process begins, the wizard might display a prompt asking you to log on to the application server in the target environment. Enter your SAS administrator's user ID and password at the prompt.

13 When the import process finishes, a log with a date and time stamp is saved in your user directory, and a message like the following is displayed:



If errors or warnings are present, you can click **View Log** to view explanatory messages. For assistance in resolving errors, see "Troubleshooting the Export and Import Wizards" on page 152.

Note: The preceding steps are for the example scenario. Additional wizard pages might appear, depending on the types of objects you are promoting. For complete step-by-step instructions for using the Import Wizard, see the product Help. △

Troubleshooting the Export and Import Wizards

The following error messages might appear when you run the Export Wizard or Export Wizard in SAS Management Console or in SAS Data Integration Studio:

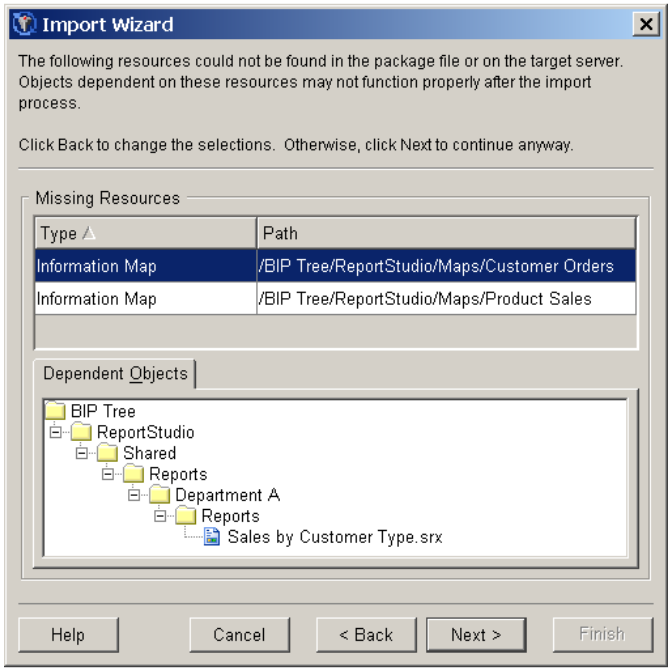
Destination folder has invalid or missing content mapping; cannot import reports.

If this message appears when you try to run the Import Wizard, click **OK** to close the message box and click **Cancel** to stop the import process.

Specify content mapping on the destination metadata repository (see "For Reports, Set Up Content Mapping in the Destination Repository" on page 141), and start the Import Wizard again.

The following resources could not be found in the package file or on the target server. Objects dependent on these resources may not function properly after the import process.

When you run the Import Wizard, this message could appear in a dialog box similar to the following:



This message means that the objects that you are importing depend on objects that are not present in the destination metadata repository.

You will need to import the resources that objects depend on, unless those resources already exist in the target environment. For example, in order for a report to function properly, you must import the information map that the report depends on, as well as any stored processes that are associated with the report.

Could not import "*filename.sas*" in "*path*" on application server "*server name*". Reason: Insufficient authorization to access *path*.

This message appears in the error log at the end of the import process. It means that the user who ran the wizard does not have Write permission for the specified directory in the file system of the target machine. Grant the user Write permission to the directory, and run the wizard again.

Stored Process "*stored process name*" (fqid: "*id*") does not have a Directory metadata object associated with its source File metadata object. This stored process was imported but needs to be fixed. No content was imported.

This message appears in the error log at the end of the import process. It means that the source code file for the stored process was not imported because a source code repository had not been defined. Define a stored process repository on the target system (see "For Stored Processes, Set Up a Source Code Repository and an Archive Path (If Needed) in the Target Environment" on page 142), and run the Import Wizard again.

Libname *library name* is not assigned.

This message could appear in the error log if you try to export physical tables along with DBMS table objects while you are logged on as an unrestricted user. Unrestricted users (for example, **sasadm**) cannot access the servers that are necessary to export physical DBMS tables. Log on as a different user, and start the export process again.

PART 3

Data Administration

CHAPTER

8

Overview of Common Data Sources

Overview

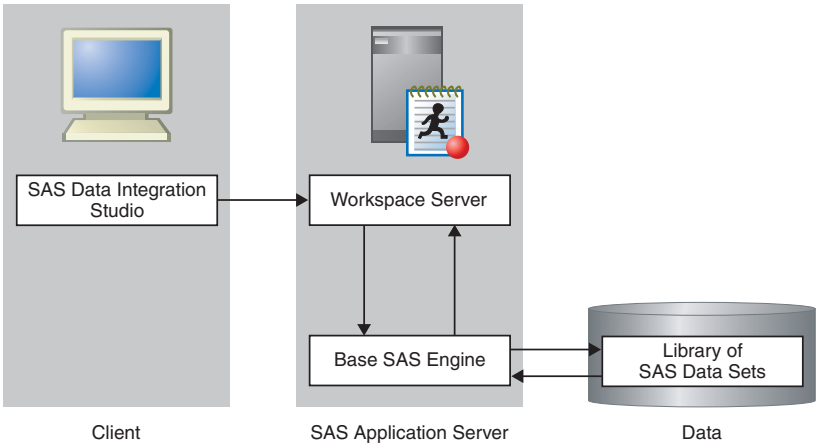
This chapter provides a brief introduction to the most common data sources that you encounter as you perform administrative tasks. The features of each data source are described. In addition, a simple diagram is provided for each data source that shows how the data flows as connections are established between source storage, SAS engines and servers, and SAS applications.

Chapter 9, "Connecting to Common Data Sources," on page 167 provides detailed examples of the connection process for each the data sources covered in this chapter. Chapter 10, "Assigning Libraries," on page 197 explains how to assign libraries so that the SAS servers in the environment know where the libraries are located and how to access them. Chapter 10, "Assigning Libraries," on page 197 explains how to use PROC METALIB to manage table metadata.

SAS Data Sets

SAS data sets (tables) are the default SAS storage format. You can use them to store data of any granularity. A SAS table is a SAS file stored in a SAS data library that SAS creates and processes. A SAS table contains data values that are organized as a table of observations (rows) and variables (columns) that can be processed by SAS software. A SAS table also contains descriptor information such as the data types and lengths of the columns, as well as which engine was used to create the data. For more information about using default SAS storage, see *SAS Language Reference: Concepts* and *SAS Language Reference: Dictionary*.

Display 8.1 Establishing Connectivity to SAS Data Sets



See "Establishing Connectivity to a Library of SAS Data Sets" on page 168 for a detailed example of a SAS data set connection.

Shared Access to SAS Data Sets

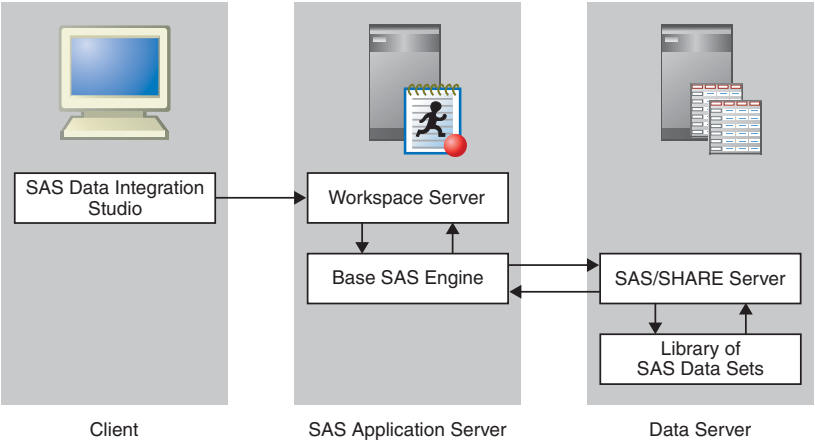
SAS/SHARE software provides concurrent update access to SAS files for multiple users. SAS/SHARE is often required for transaction-oriented applications where multiple users need to update the same SAS data sets at the same time. Data entry applications where multiple users are entering data to the same dataset are a good example of this type of usage. SAS/SHARE software provides both member- and record-level locking. Therefore, two or more users can update different observations within the same data set, and other users can print reports from the same data set.

SAS/SHARE supports multi-user read/write access to both SAS data files and SAS catalogs. Multi-user access to SAS catalogs simplifies the maintenance of applications by allowing users and developers to share the same program libraries. Users can execute applications at the same time that developers update the source programs.

SAS/SHARE software also acts as a data server that delivers data to users for their processing needs. This capability provides data administrators both a centralized point of control for their data and a secure environment to control who accesses the data. SAS/SHARE is also designed to be a reliable data server that functions as long as the system that the server is running on is operational.

Finally, SAS/SHARE allows you to exploit SAS software's ability to define views of your data. This allows administrators to restrict certain users to subsets of data for security or efficiency purposes. Access to rows and columns in SAS tables can be defined using this technique.

Display 8.2 Establishing Shared Access to SAS Data Sets



See "Establishing Shared Access to SAS Data Sets" on page 170 for a detailed example of a share SAS data set connection.

External Files

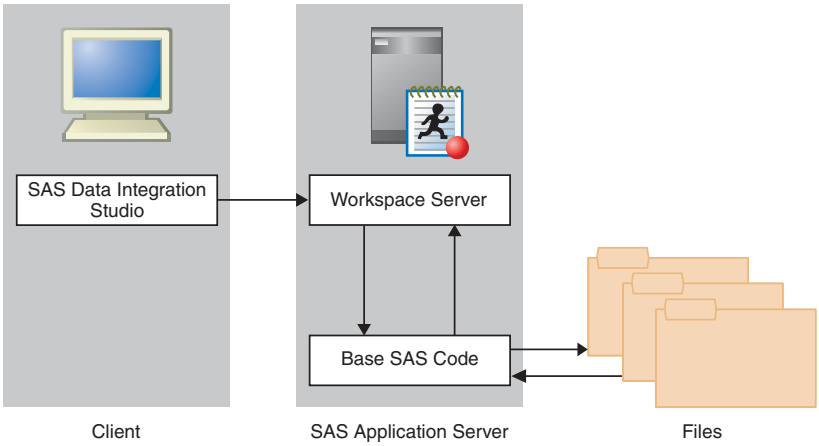
An external file is a file that is maintained by the machine operating environment or by a software product other than SAS. A flat file with comma-separated values is one example. SAS Data Integration Studio provides three source designer wizards that enable you to create metadata objects for external files:

- the delimited external file wizard for external files in which data values are separated with a delimiter character. This wizard enables you to specify multiple delimiters, nonstandard delimiters, missing values, and multi-line records.
- the fixed-width external file wizard for external files in which data values appear in columns that are a specified number of characters wide. This wizard enables you to specify non-contiguous data.
- the user-written external file wizard for complex external files that require user-written SAS code to access their data.

The external file source designer wizards enable you to do the following:

- display a raw view of the data in the external file
- display a formatted view of the data in the external file, as specified in the SAS metadata for that file
- display the SAS DATA step and SAS INFILE statement that the wizard generates for the selected file
- display the SAS log for the code that is generated by the wizard
- specify options for the SAS INFILE statement that is generated by the wizard, such as National Language Support (NLS) encoding
- override the generated SAS INFILE statement with a user-written statement
- supply a user-written SAS DATA step to access an external file

Display 8.3 Establishing Connectivity to External Files



See "Establishing Connectivity to a Flat File" on page 172 for a detailed example of an external file connection.

XML Data

The XML LIBNAME engine works in a way similar to other SAS engines. A LIBNAME statement is executed so that a libref is assigned and an engine is specified. That libref is then used throughout the SAS session.

Instead of the libref being associated with the physical location of a SAS data library, the libref for the XML engine is associated with a physical location of an XML document. When you use the libref that is associated with an XML document, SAS either translates the data in a SAS data set into XML markup or translates the XML markup into SAS format.

The XML LIBNAME engine can read input streams from a Web service input and write an output stream to a Web service output. The XML LIBNAME engine supports reading XML files in complex structures using XMLMaps. An XMLMap is a user-defined file that contains XML tags that tell the XML LIBNAME engine how to interpret an XML document. XMLMaps are defined using the SAS XML Mapper product. For additional information, see the *SAS XML LIBNAME Engine User's Guide*.

XML files are written by the XML Writer transformation provided by SAS Data Integration Studio. The XML LIBNAME engine supports Output Delivery System (ODS) tag sets; XMLmaps are not supported for writing. The XML Writer transformation in SAS Data Integration Studio ships with a sample ODS tag set, if needed. An output XML document can either be:

- used by a product that processes XML documents
- moved to another host for the XML LIBNAME engine to then process by translating the XML markup back to a SAS data set

Since the XML LIBNAME engine is designed to handle tabular data, all the data sent to or from a Web service must be in table form.

When you are writing an XML file, we recommend that you define the library specifically for your write operation.

Relational Database Sources

SAS/ACCESS

Data also can be stored in third-party hierarchical and relational databases such as IMS, DB2, Oracle, SQL Server, and NCR Teradata. SAS/ACCESS interfaces provide fast, efficient reading and writing of data to these facilities.

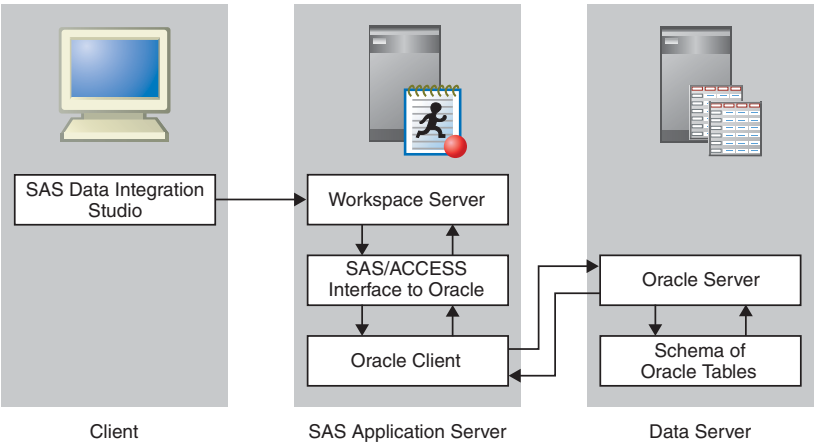
Several of the SAS/ACCESS engines support threaded reads. This enables you to read entire blocks of data on multiple threads instead of reading data just one record at a time. This feature can reduce I/O bottlenecks and enables thread-enabled procedures to read data quickly. These engines and DB2 on z/OS also have the ability to access database management system (DBMS) data in parallel by using multiple threads to the parallel DBMS server.

The following SAS/ACCESS engines support this functionality:

- Oracle
- Sybase
- DB2 (UNIX and PC)
- SQL Server
- Teradata

For more information about using the SAS/ACCESS interfaces, see *SAS/ACCESS for Relational Databases: Reference*.

Display 8.4 Establishing Connectivity to Oracle Databases



See “Establishing Connectivity to an Oracle Database” on page 175 for a detailed example of an Oracle connection.

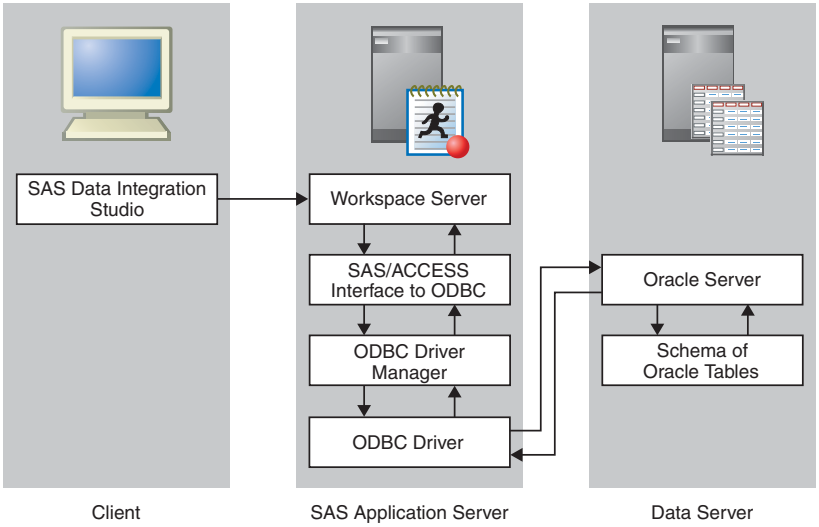
ODBC Sources

Open database connectivity (ODBC) standards provide a common interface to a variety of databases, including AS/400, dBASE, Microsoft Access, Oracle, Paradox, and Microsoft SQL Server databases. Specifically, ODBC standards define application programming interfaces (APIs) that enable an application to access a database if the ODBC driver adheres to the specification.

The basic components and features of ODBC include the following:

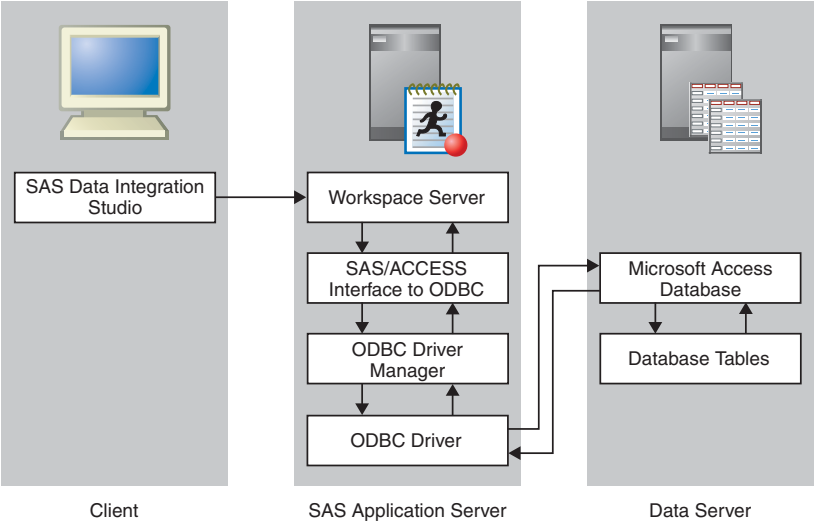
- ODBC functionality is provided by three components: the client interface, the ODBC driver manager, and the ODBC driver. SAS provides the SAS/ACCESS interface to ODBC, which is the client interface. For PC platforms, Microsoft developed the ODBC Administrator, which is used from the Windows Control Panel to perform software administration and maintenance activities. The ODBC driver manager also manages the interaction between the client interface and the ODBC driver. Other software vendors provide the ODBC manager with their ODBC drivers, which process requests for external data.
- The ODBC administrator defines a data source as the data that is used in an application and the operating system and network that are used to access the data. You create a data source by using the ODBC Administrator in the Windows Control Panel and then selecting an ODBC driver. You then provide the information (for example, data source name, user ID, password, description, server name) that is required by the driver to make a connection to the desired data. The driver displays dialog boxes in which you enter this information. During operation, a client application usually requests a connection to a named data source, not just to a specific ODBC driver.
- An ODBC Administrator tool is not available in a UNIX environment such as HP-UX, AIX, or Solaris. During an install, the driver creates a generic **`.odbc.ini`** file that can be edited to define your own data sources.

Display 8.5 Establishing Connectivity to Oracle Databases By Using ODBC



See “Establishing Connectivity to an Oracle Database by Using ODBC” on page 179 for a detailed example of an ODBC-based Oracle connection.

Display 8.6 Establishing Connectivity to Access Databases By Using ODBC



See "Establishing Connectivity to a Microsoft Access Database by Using ODBC" on page 184 for a detailed example of an ODBC-based Access connection.

Scalable Performance Data Servers

Both the SAS Scalable Performance Data Engine (SPD Engine) and the SAS Scalable Performance Data Server (SPD Server) are designed for high-performance data delivery. They enable rapid access to SAS data for intensive processing by the application. The SAS SPD Engine and SAS SPD Server deliver data to applications rapidly by organizing the data into a streamlined file format that takes advantage of multiple CPUs and I/O channels to perform parallel input/output functions.

The SAS SPD Engine is included with Base SAS software. It is a single-user data storage solution that shares the high-performance parallel processing and parallel I/O capabilities of SAS SPD Server, but it lacks the additional complexity of a full-blown server. The SAS SPD Server is available as a separate product or as part of the SAS Intelligence Storage bundle. It is a multi-user parallel-processing data server with a comprehensive security infrastructure, backup and restore utilities, and sophisticated administrative and tuning options. The SAS SPD Server libraries can now be defined using SAS Management Console.

The SAS SPD Engine and SAS SPD Server use multiple threads to read blocks of data very rapidly and in parallel. The software tasks are performed in conjunction with an operating system that enables threads to execute on any of the machine's available CPUs.

Although threaded I/O is an important part of both product offerings' functionality, their real power comes from the way that the software structures SAS data. They can read and write partitioned files and, in addition, use a specialized file format. This data structure permits threads, running in parallel, to perform I/O tasks efficiently.

Although not intended to replace the default Base SAS engine for most tables that do not span volumes, SAS SPD Engine and SAS SPD Server are high-speed alternatives for processing very large tables. They read and write tables that contain billions of observations.

The SAS SPD Engine and SAS SPD Server performance are boosted in these ways:

- support for terabytes of data

- scalability on symmetric multiprocessing (SMP) machines
- parallel WHERE selections
- parallel loads
- parallel index creation
- partitioned tables
- parallel I/O data delivery to applications
- implicit sorting on BY statements

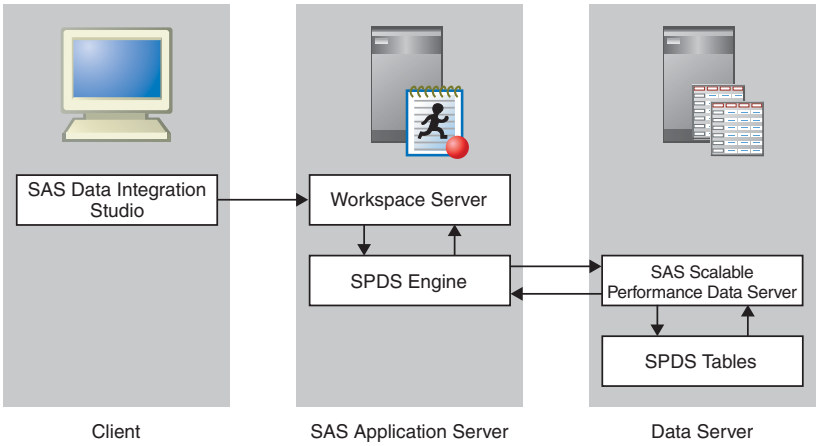
The SAS SPD Engine runs on UNIX, Windows, z/OS (on HFS and zFS file systems only), and OpenVMS for Integrity Servers (on ODS-5 file systems only) platforms. The SAS SPD Server runs on Tru64 UNIX, Windows Server, HP-UX, and Sun Solaris platforms.

Symmetric Multiprocessing:

The SAS SPD Engine exploits a hardware and software architecture known as symmetric multiprocessing (SMP). An SMP machine has multiple CPUs and an operating system that supports threads. An SMP machine is usually configured with multiple disk I/O controllers and multiple disk drives per controller. When the SAS SPD Engine reads a data file, it launches one or more threads for each CPU; these threads then read data in parallel. By using these threads, a SAS SPD Engine that is running on an SMP machine provides the quick data access capability that is used by SAS in an application.

For more information about using the SAS SPD Engine, see *SAS Scalable Performance Data Engine: Reference* and **`support.sas.com/rnd/scalability/spde`**.

Display 8.7 Establishing Connectivity to an SPDS Server



See “Establishing Connectivity to an Scalable Performance Data Server” on page 187 for a detailed example of an SPDS server connection.

ERP Systems

Enterprise Resource Planning (ERP) systems are composed of thousands of tables, columns, variables and fields. Although they contain a wealth of data, they lack several key features:

- the ability to provide integration with other data sources
- the ability to do backward-looking drill-down analysis into what caused the effect (Business Intelligence)
- the ability to do forward-looking cause and effect analysis (Business Analytics)

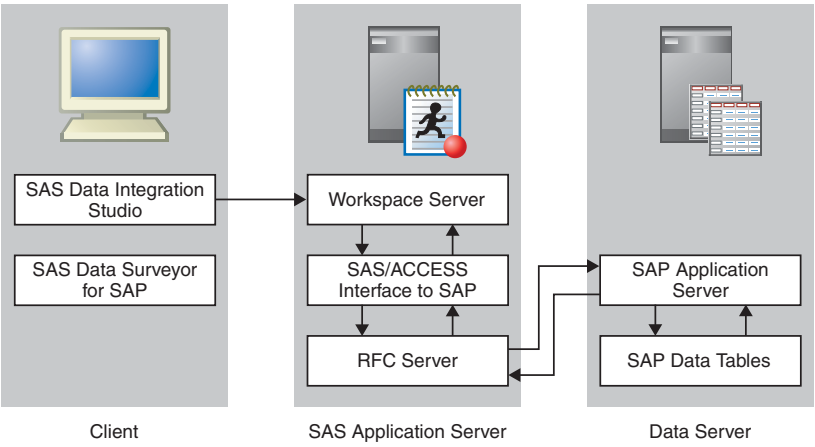
To make it possible to get to the data the ERP systems contain, SAS provides data surveyors for each of the following ERPs:

- SAP
- Peoplesoft
- Oracle
- Siebel

Data surveyors contain Java plug-ins to SAS Data Integration Studio and SAS Management Console, plus the required SAS/ACCESS engine necessary to get the information out of the DBMS system. Understanding the metadata of these business applications is at the heart of the data surveyor. Each data surveyor has knowledge about the specific application it is designed for. This knowledge contains information about the ERP metadata that allows you to do the following:

- understand complex data structures
- navigate the large amounts of tables (SAP is over 20,000)

Display 8.8 Establishing Connectivity to an SAP Server



See “Establishing Connectivity to an SAP Server” on page 191 for a detailed example of an SAP server connection.

CHAPTER

9

Connecting to Common Data Sources

Overview

This chapter consists of detailed examples for establishing a connection to each of the common data sources introduced in Chapter 8, "Overview of Common Data Sources," on page 157. Some of the connection processes covered in this chapter have common elements that might be applied to similar data sources. For example, the description of the process of using SAS/ACCESS to connect to an Oracle database might be useful when you connect to other relational databases such as DB2, Sybase, and Informix. Also, the descriptions of ODBC connections to Oracle and Microsoft Access databases and the account of the connection to an SAP source can be helpful when you connect to similar data sources.

The chapter also explains a connection verification process that imports tables from the data sources and enables you to view their data in a SAS application. More detailed information about managing table metadata can be found in the Chapter 11, "Managing Table Metadata," on page 211.

Establishing Connectivity to a Library of SAS Data Sets

Stage 1: Define the SAS Base Engine Library

After you have installed the required SAS software, you need to set up a connection from a SAS server to a SAS data set. This connection requires that you define a SAS Base Engine Library metadata object in the metadata repository. In addition, you must import any user-defined formats that have been created for the data set in order to view or operate on the data. Assume that the SAS software has already been loaded by using the standard installation wizard and that the data set is stored in a location that can be accessed.

Define the SAS Base Engine Library metadata object by using SAS Management Console. This metadata enables your SAS applications to access the data sets that you need to work with. For this example, the dataset contains information about customers of the Orion Gold enterprise. Perform the following steps:

1. In SAS Management Console, double-click **`Data Library Manager`**. Right-click **`SAS Libraries`**. Then, select the **`New Library`** option to access the first page of the New Library Wizard.
2. Select **`SAS Base Engine Library`** from the **`SAS Libraries`** list. Click **`Next`**.
3. Enter an appropriate library name in the **`Name`** field (for example, **`Orion Gold Customers`**). Note that you can supply an optional description if you wish. Click **`Next`**.
4. Enter the following library properties:

Table 9.1 Library Properties

Field	Sample Value
Libref	**ORGOLD**
Engine	**BASE**
Path Specification	**C:\SAS\EntBIServer\Lev1\SASMain\Data** (Enter the fully-qualified path to the library. This path is specified differently in different operating systems. Make sure that the appropriate path is displayed in the **Selected items** field.)

You can also click **Advanced Options** to perform tasks such as pre-assignment and setting host-specific and LIBNAME options. Click **Next** to access the next page of the wizard.

5 (Optional) Select one or more SAS servers. (You might not need to select a server. The need to select a server depends on the applications that are included in your environment.) The library is assigned to the server or servers that you select from this list. Click **Next**.

6 Examine the final page of the wizard to ensure that the proper values have been entered. Click **Finish** to save the settings.

At this point, you can import tables, as explained in "Verifying Access to Tables" on page 194.

Working with User-Defined Formats

If you have existing SAS data sets, you might also have a catalog of user-defined formats and informats. You have two options for making these formats available to applications such as SAS Data Integration Studio and SAS Information Map Studio:

- Give the format catalog a standard name and place it in an expected location. This is the preferred method.
- Create a user-defined formats configuration file, and use the FMTSEARCH system option to point to the format catalog.

Use a Standard Name and Location for the Format Catalog

The preferred way to make a format catalog available is to complete these steps:

1 Name the format catalog formats.sas7bcat.

2 Place the catalog in the directory *SAS-config-dir***\Lev1\SASMain\SASEnvironment\SASFormats**.

Note: In the z/OS environment, you must perform an additional step. In the SASRX REXX exec, add the following LIBRARY ALLOCATE command, which points to the **SASFormats** directory.

```
allocate dd(library) path('SAS-config-dir/Lev1/SASMain/SASEnvironment/SASFormats')
```

△

Create a User-Defined Formats Configuration File

Alternatively, you can create a user-defined formats configuration file in which you point to the location of the formats catalog.

On Windows and UNIX systems, complete these steps:

1 To the SAS configuration file *SAS-config-dir***\Lev1\SASMain\sasv9.cfg**, add the CONFIG system option, and use it to point to the user-defined formats configuration file.

```
-config "SAS-config-dir\Lev1\SASMain\userfmt.cfg"
```

2 Then, use the FMTSEARCH system option in the same configuration file to point to the format catalog:

```
-set fmtlib1 "SAS-config-dir\Lev1\Data\orformat"
-fmtsearch (fmtlib1.orionfmt)
```

In this example, *SAS-config-dir***\Lev1\Data\orformat** is the location of the format catalog, and orionfmt (filename orionfmt.sas7bcat) is the name of the format catalog.

Note: On UNIX systems, you must enter the variable name in uppercase. For example, you would enter **FMTLIB1** instead of **fmtlib1**. △

On z/OS systems, perform the following steps:

1 Add the AUTOEXEC system option to the SAS launch command as shown in the following example.

```
SAS-config-dir/Lev1/SASMain/startsas.sh
   o("autoexec="./WorkspaceServer/userfmt.sas"")
```

In this example, **startsas.sh** is your SAS launch command script, and **userfmt.sas** is the name of the SAS autoexec file. When you enter the command, you must enter it all on one line.

2 In the autoexec file, use the LIBNAME statement to assign the format library and the OPTIONS statement to set the FMTSEARCH system option. For example, you might specify the following statements:

```
libname fmtlib1 'SAS-config-dir/Lev1/Data/orformat' repname=Foundation;
options fmtsearch=(fmtlib1.orionfmt);
```

Establishing Shared Access to SAS Data Sets

Overview of Establishing Shared Access

Base SAS libraries allow the following access:

- Any number of users can read data.
- A single user can write or update data.

This access can be extended through the use of the SAS/SHARE server. A SAS/SHARE server permits multiple users to update the same items in a SAS library.

You can share access to a library of existing SAS data sets by using a SAS/SHARE server to manage access to the data. Assume that the SAS/SHARE software has already been loaded by using the standard installation wizard, and that you have created a SAS/SHARE server metadata object (for example, SHAREServer). Also assume that the other user has already created a SAS Base Engine Library, as described in “SAS Data Sets” on page 157. Setting up shared access to a SAS data set can be seen as a three-stage process, as follows:

1. Assign the SAS Base Engine Library to the SAS/SHARE server.
2. Create a SAS/SHARE Remote Engine Library metadata object.
3. Enable library pre-assignment. In this example, you're assigning the Orion Gold Customers library to the SHAREServer server.

Stage 1: Assign the SAS Base Engine Library to the SAS/SHARE Server

You need to assign the SAS Base Engine Library for the library that you need to share to the SAS/SHARE Server. In this example, a library that contains information about customers of the Orion Gold enterprise is being shared. Perform the following steps:

1. In SAS Management Console, double-click **Data Library Manager**. Double-click **SAS Libraries**. Then, right-click the icon for the library that you need to share (for example, **Orion Gold Customers**). Finally, select the **Properties** option to access the Properties dialog box.
2. Click the **Assign** tab to see a list of available servers. Click the name of the SAS/SHARE server. This step enables the SAS/SHARE server to access the data in the library that you need to share.
3. Click **OK** to save the server assignment.

Stage 2: Create a SAS/SHARE Remote Engine Library

Perform the following steps:

1. In SAS Management Console, double-click **Data Library Manager**. Right-click **SAS Libraries**. Then, select the **New Library** option to access the New Library Wizard.
2. Select **SAS/SHARE Remote Engine Library** from the **SAS Libraries** list. Click **Next**.
3. Enter an appropriate library name in the **Name** field (for example, **SharedAccessToOrionGold**). Note that you can supply an optional description if you wish. Click **Next**.
4. Enter the following library properties:

   Table 9.2 Library Properties

Field	Sample Value
SAS/SHARE Server	**SHAREServer**
Default Login	**(None)** (This default login is used to resolve conflicts between multiple logins to an authentication domain. In such cases, the default login is used.)
SAS/SHARE Server Library	**Orion Gold Customers** (Use the drop-down menu to select the library that you need to share.)

   Click **Next**.
5. Select one or more SAS servers. The library is assigned to the servers included in this list. Click **Next**.
6. Examine the final page of the wizard to ensure that the proper values have been entered. Click **Finish** to save the settings.

Stage 3: Enable Library Pre-Assignment

In order to gain access to the tables, you must perform two additional steps to enable pre-assignment for the library that you need to share, as follows:

1. In SAS Management Console, double-click **Data Library Manager**. Double-click **SAS Libraries**. Then, right-click the icon for the library that you shared (for example, **Orion Gold Customers**). Finally, select the **Properties** option to access the Properties dialog box.
2. Click the **Options** tab. Then, click **Advanced Options** to access the Advanced Options dialog box.
3. Select the **Library is pre-assigned** option to enable pre-assignment for the shared library. Click **OK** to return to the **Options** tab.

   Note: When **Library is pre-assigned** is selected, SAS applications no longer try to assign a library. Instead, the pre-assigned library is referenced by the applications. △
4. Click **OK** to close the Properties dialog box and save the pre-assignment.
5. Update the SAS/SHARE configuration file to include the METAAUTORESOURCES value for the SAS/SHARE server. Perform the following steps:
   a. Navigate in your local file system to find the SAS/SHARE configuration file (for example, **C:\SAS\EntBIServer\Lev1\SASMain\ShareServer\sasv9_ShareServer.cfg**).
   b. Open the SAS/SHARE configuration file in a text editor (such as Notepad).
   c. Add the following text to the end of the file:

      ```
      -metaautoresources "omsobj:ServerComponent?@Name='SASMain - SHAREServer'"
      ```

   d. Save the configuration file to save the new setting.

At this point, you can import tables, as explained in "Verifying Access to Tables" on page 194.

Establishing Connectivity to a Flat File

You can connect to a flat file using the External File Source Designer in SAS Data Integration Studio. Setting up a connection from SAS to a flat file can be seen as a three-stage process, as follows:

1. Connect to the flat file.
2. Define the columns in the external file object.
3. Save the external file object.

Assume that the SAS software has already been loaded by using the standard installation wizard, and that the flat file is stored in a location that can be accessed. This example focuses on a comma-delimited flat file. A similar process is used for other types of flat files, but some steps are different.

Stage 1: Connect to the Flat File

Perform the following steps to establish a connection to the flat file:

1 Open the SAS Data Integration Studio application. Then, select **Tools ► Source Designer** to access the Source Designer wizard.
2 Select **Delimited External File** from the **External Files** list. Click **Next**.
3 Enter the fully-qualified path to the flat file in the **File name** field (for example, *SAS-config-dir***sources\customer_data.dat**). Click **Next**.

Stage 2: Define the Columns in the External File Object

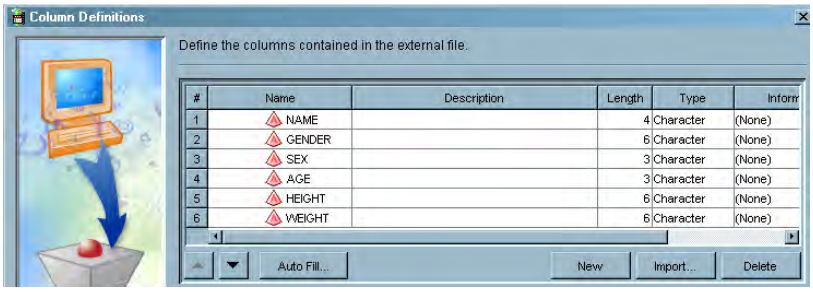
Perform the following steps to define the columns in an external file object:

1 On the Delimiters and Parameters page of the wizard, deselect the **Blank** option in the **Delimiters** group box. Then, select the **Comma** option. Click **Next** to access the Column Definitions page of the wizard.
2 Perform the following steps to define the columns in the external file object:
   a Click **Refresh** to view the data from the flat file in the **File** tab in the view pane at the bottom of the page.
   b Click **Auto Fill** to access the Auto Fill Columns dialog box. Change the value entered in the **Start record** field in the **Guessing records** group box to **2**. This setting is based on the assumption that the first data record of the flat file contains header information, and that the record is unique because it holds the column names for the file. Therefore, excluding the first data record from the guessing process yields more accurate preliminary data because it is excluded when the guessing algorithm is run.
   c Click **OK** to return to the Column Definitions page.
3 Click **Import** to access the Import Column Definitions dialog box. The following four methods are provided for importing column definitions:
   - Get the column definitions from other existing tables or external files.
   - Get the column definitions from a format file.
   - Get column definitions from a COBOL format file.
   - Get the column names from column headings in the file.

   In most cases, you will either get the column definitions from an external format file or get the column names from the column headings in the external file. Here is an example of a format file:

```
# Header follows
Name,SASColumnType,SASColumnName,SASColumnLength,
SASInformat,SASFormat,Desc,ReadFlag
# Column definition records follow
Namë,C,name,8,,$char8.,cl@ss name column,y
Sëx,C,sex,1,,,cl@ss sex column,n
Agë,N,age,3,,,cl@ss age column,y
# Comma within quotation marks below is not a delimiter
Description,C,Description,32,#char32.,,'Description, Comments, etc.',y
```

   A sample of the output is shown in the following figure:



For this example, select the **Get the column names from column headings in the file** radio button. Keep the default settings for the fields underneath it.

Note: If you select **Get the column names from column headings in the file**, the value in the **Starting** record field in the **Data** tab of the view pane in the Column Definitions dialog box is automatically changed. The new value is one greater than the value in the **The column headings are in file record** field in the Import Column Definitions dialog box. △

4 Click **OK** to return to the Column Definitions page.

5 The preliminary data for the external file object is displayed in the columns table at the top of the page. The **Informat** and **Format** columns for the rows in the table are based on the values that are included in the sample data that is processed by the guessing function. The results are accurate for this particular set of records, but you should still examine them to make sure that they are representative of the data in the rest of the flat file. Edit the values by clicking directly on the cells in the column table and making the necessary changes.

6 Click the **Data** tab at the bottom of the Column Definitions page. Then, click **Refresh**. The data should be properly formatted. If not, edit the cells in the column table and check the results by refreshing the **Data** tab. You can repeat this process until you are satisfied. You also can review the SAS log for more details.

Note: To view the code that will be generated for the external file, click the **Source** tab. To view the SAS log for the generated code, click the **Log** tab. The code that is displayed in the Source tab is the code that will be generated for the current external file. △

Stage 3: Save the External File Object

Perform the following steps to name and save the external file object:

1 Click **Next** to access the Select Group page. Assume that you do not want to specify a custom group for the table in this example.

2 Click **Next** to access the General page. Enter an appropriate name in the **Name** field (for example, **Customer Data**). Note that you can supply an optional description if you wish.

3 Click **Finish** to save the metadata and exit the wizard.

At this point, you can import tables, as explained in “Verifying Access to Tables” on page 194.

Establishing Connectivity to XML Data

Connect to the XML Data

The following steps describe how to specify a SAS XML library in SAS Management Console. Assume that the XML library will point to an XML file that contains climate information (**climate.xml**). The XML file is in GENERIC format, as defined for the SAS XML LIBNAME engine. For additional information, see the *SAS XML LIBNAME Engine User's Guide*.

1. In SAS Management Console, double-click **Data Library Manager**. Right-click **SAS Libraries**. Then, select the **New Library** option to access the New Library Wizard.
2. Select **SAS XML library** from the **SAS Libraries** list. Click **Next**.
3. Enter an appropriate library name in the **Name** field (for example, **XML Lib**). Click **Next**.
4. Enter information about the library, such as the following:

Table 9.3 Library Properties

Field	Sample Value
Name	**XML Lib**
Libref	**xmllib**
Engine	**XML**
XML File	**C:\sources\xml\climate.xml**
XML Type	**GENERIC**
Library Access	**READONLY**

5. Click **Finish** to save the wizard settings.

Establishing Connectivity to an Oracle Database

Setting up a connection from SAS to a database management system can be seen as a three-stage process, as follows:

1. Define the database server metadata object.
2. Define the database schema metadata object.
3. Define the database library metadata object.

This example illustrates the process for establishing a SAS connection to an Oracle database. It assumes that the software for the database has already been loaded by using the standard installation wizard for the database client. The following prerequisites have been satisfied:

- installation of SAS/ACCESS Interface to Oracle. For requirements information, go to the Install Center at **http://support.sas.com/documentation/**

installcenter/index.html and select the operating system. Then, select the SAS version and click the Planning Installation Edition Kit link. Finally, select the appropriate System Requirements for SAS Foundation document in the Installation category.

- installation of a supported Oracle Database Client.
- validation that the Oracle client can communicate with the Oracle server.
- configuration of SAS/ACCESS environmental variables. Create a UNIX script file that sets up all the environment variables (for example, LD_LIBRARY_PATH_64, LD_LIBRARY_PATH, ODBCHOME, ODBCINI, ORACLE_HOME, SYBASE, INSTHOME, LIBPATH, and SHLIB_PATH). Refer to the Post-Installation Guide or the Configuration Guide in your Installation Kit to see whether this configuration is needed. If this configuration is necessary, follow the instructions in the appropriate document. For information about setting environmental variables when you use SAS/ACCESS to connect to data on UNIX systems, see "Set SAS/ACCESS Environment Variables" on page 176.

Set SAS/ACCESS Environment Variables

If you are attempting to connect to data sources located on UNIX by using SAS/ACCESS, you must set the environmental variables for the workspace server. Use the following steps to set the appropriate environment variables in the SAS Workspace Server:

1. Use the SASENV invocation option to call the DBMS environment script file. From SAS Management Console, select the **Options** tab from the Workspace Server Properties dialog box. Add **-sasenv pathtoscript.txt --** after **sas.sh** in the **Workspace Server Launch Command** field. For example, enter the following:

   /BIArchitecture/Lev1/SASMain/sas.sh -sasenv dbmsscript.txt --

   Note: The trailing **--** is required. The file name **pathtoscript.txt** is any script file name containing the DBMS environment variable settings provided by your DBA. △
2. Restart the object spawner using the **objectspawner.sh** script file. For example, enter the following:

   /BIArchitecture/Lev1/SASMain/ObjectSpawner/objectspawner.sh
3. In SAS Management Console, right-click the **Workspace Server** connection and select the **Test Connection** option to verify that the workspace server starts correctly with the new DBMS environment script file.

Stage 1: Define the Database Server

Perform the following steps to define the database server metadata object:

1. Open the SAS Management Console application.
2. Right-click **Server Manager** and select the **New Server** option to access the New Server Wizard.
3. Select **Oracle Server** from the **Database Servers** list. Then, click **Next**.
4. Enter an appropriate server name in the **Name** field (for example, **Oracle Server**). Note that you can supply an optional description if you wish. Click **Next**.
5. Enter the following server properties:

Table 9.4 Server Properties

Field	Sample Value
Major Version Number	**10**
Minor Version Number	**2**
Software Version	**10.2.0**
Vendor	**Oracle Corporation**

Click **Next**.

6 Enter the following connection properties:

Table 9.5 Connection Properties

Field	Sample Value
Path	**NEWSERVER10G** (This value is contained in the **tnsnames.ora** file generated during the Oracle installation. The file is stored in an Oracle installation directory such as **C:\oracle\product\10.2.0\client_1\NETWORK\ADMIN\tnsnames.ora**. The alias for the connection information is contained in this file. See the following display.)
Authentication Domain	**OracleAuth** (You need to create a new authentication domain each time you connect to a new database server. The SAS Metadata Server, Workspace Server, and others require one set of credentials, while the database server requires another. Click **New** to access the New Authorization Domain dialog box. Then enter the appropriate value in the **Name** field and click **OK** to save the setting. Make sure that the authentication domain that you create here is added to the appropriate users and user groups. Add a login to these users and groups that includes a user ID, a password, and the authentication domain.)

Note: If you will have multiple users for the connection, consider creating a user group for them to avoid the inefficient process of creating separate user IDs and passwords for each. Use the Access Control Template in the Authorization Manager in SAS Management Console. For more information, see "Example: Use a Custom ACT" in the chapter "Securing a Deployment" in the *SAS Intelligence Platform: Security Administration Guide*. △

The following display shows a sample **tnsnames.ora** file:

```
# tnsnames.ora Network Configuration File:
C:\oracle\product\10.2.0\client_1\network\admin\tnsnames.ora
# Generated by Oracle configuration tools.

NEWSERVER10G =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP)(HOST = server.na.sas.com)(PORT = 1521))
    )
    (CONNECT_DATA =
      (SERVICE_NAME = server10G)
    )
  )
```

Note that the correct **Path** value is circled. Click **Next**.

7 Examine the final page of the wizard to ensure that the proper values have been entered. Click **Finish** to save the wizard settings.

Stage 2: Define the Database Schema

After you have defined the database server metadata, you can define the database schema metadata object. The server object must be defined first because the server object must be entered into the wizard when you define the schema object. (If needed, you can define several schemas per server.) Perform the following steps:

1 In SAS Management Console, double-click **Data Library Manager**. Then, right-click **Database Schema** and select the **New Database Schema** option to access the New Database Schema Wizard.

2 Select **Oracle Schema** from the **Database Schemas** list. Click **Next** to access the next page of the wizard.

3 Enter an appropriate schema name in the **Name** field (for example, **Oracle Schema**). Note that you can supply an optional description if you wish. Click **Next**.

4 Enter the following schema properties:

Table 9.6 Schema Properties

Field	Sample Value
Database Schema Name	**Schema Name** (This value needs to be the name of an existing schema).
Oracle Server	**Oracle Server** (Use the value that you entered in the **Name** field in the New Server Wizard when you defined the database server metadata object. In this case it must be **Oracle Server**, which is the name of the metadata object representing the Oracle server.)

Click **Next**.

Note: The case of the schema names depends upon the database type. For example, DB2 schema names are uppercased by default. △

5 Examine the final page of the wizard to ensure that the proper values have been entered. Click **Finish** to save the schema settings.

Stage 3: Define the Database Library

After you have defined the database server metadata object and the database schema metadata object, you can define the database library metadata object. It is important to

define the library object after the server and schema objects because these objects must be entered into the wizard when you define the library object. Perform the following steps:

1. In SAS Management Console, double-click **Data Library Manager**. Right-click **SAS Libraries**. Then, select the **New Library** option to access the New Library Wizard.
2. Select **Oracle Library** from the **Database Libraries** list. Click **Next**.
3. Enter an appropriate library name in the **Name** field (for example, **Oracle Library**). Note that you can supply an optional description if you wish. Click **Next**.
4. Enter the following library properties:

   Table 9.7 Library Properties

Field	Sample Value
Libref	**ORAREF**
Engine	**ORACLE**

   You can also click **Advanced Options** to perform tasks such as pre-assignment and optimization. Click **Next** to access the next page of the wizard.
5. Enter the following settings:

   Table 9.8 Advanced Option Settings

Field	Sample Value
Database Server	**OracleServer** (Use the database server that you created in the **New Server Wizard**.)
Default Login	**Login** (Select your login from the drop-down list. This default login is used to resolve conflicts between multiple logins to an authentication domain. In such cases, the default login is used.)
Database Schema	**OracleSchema** (Select the schema name that you entered in the **New Database Schema Wizard**.)

   Click **Next**.
6. (Optional) Select one or more SAS servers. (You might not need to select a server. The need to select a server depends on the applications that are included in your environment.) The library is assigned to the servers included in this list. Click **Next**.
7. Examine the final page of the wizard to ensure that the proper values have been entered. Click **Finish** to save the library settings. At this point, you can import tables, as explained in "Verifying Access to Tables" on page 194.

Establishing Connectivity to an Oracle Database by Using ODBC

Setting up a connection from SAS to an Oracle database management system by using ODBC can be seen as a four-stage metadata definition process, as follows:

1. Define an ODBC data source.
2. Define the database server metadata object.
3. Define the database schema metadata object.
4. Define the database library metadata object.

This example illustrates the process for establishing a SAS connection to an Oracle database. It assumes that the software for the database has already been loaded with the standard installation wizard for the database client. Before you begin, satisfy the following prerequisites:

- installation of SAS/ACCESS Interface to Oracle. For requirements information, go to the Install Center at **`http://support.sas.com/documentation/installcenter/index.html`** and select the operating system. Then, select the SAS version and click the Planning Installation Edition Kit link. Finally, select the appropriate System Requirements for SAS Foundation document in the Installation category.
- installation of a supported Oracle Database Client.
- validation that the Oracle client can communicate with the Oracle server.
- configuration of SAS/ACCESS environmental variables. Create a UNIX script file that sets up all the environment variables (for example, LD_LIBRARY_PATH_64, LD_LIBRARY_PATH, ODBCHOME, ODBCINI, ORACLE_HOME, SYBASE, INSTHOME, LIBPATH, and SHLIB_PATH). Refer to the Post-Installation Guide or the Configuration Guide in your Installation Kit to see whether this configuration is needed. If this configuration is necessary, follow the instructions in the appropriate document. For information about setting environmental variables when you use SAS/ACCESS to connect to data on UNIX systems, see "Set SAS/ACCESS Environment Variables" on page 176.

Stage 1: Define the Data Source

First, you must define the ODBC data source. On Window systems, you should follow these steps:

1. Open the Windows Control Panel. Then, double-click **`Administrative Tools`**. Finally, double-click **`Data Sources (ODBC)`** to access the ODBC Data Source Administrator dialog box.
2. Click **`Add`** to access the Create New Data Source dialog box. Click the Oracle driver listed in the window (for example, **`Oracle in OraClient10g_home1`**). Click **`Finish`** to access the Oracle ODBC Driver Configuration dialog box.

   Note: System data sources and user data sources store information about how to connect to the indicated data provider. A system data source is visible to all users with access to the system, including NT services. A user data source is only visible to a particular user, and it can only be used on the current machine. For this example, we are creating a system data source. △
3. Enter the following configuration settings:

Table 9.9 Configuration Settings

Field	Sample Value
Data Source Name	**Oracle_newserver**
TNS Service Name	**NEWSERVER10G** (Select the name entered in the tnsnames.ora file created during installation of the Oracle database from the drop-down menu. See the following display.)
User	**User Name**

The following display shows the **tnsnames.ora** file:

```
# tnsnames.ora Network Configuration File:
C:\oracle\product\10.2.0\client_1\network\admin\tnsnames.ora
# Generated by Oracle configuration tools.

NEWSERVER10G =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP)(HOST = server.na.sas.com)(PORT = 1521))
    )
    (CONNECT_DATA =
      (SERVICE_NAME = server10G)
    )
  )
```

Note that the correct **TNS Service Name** value is circled. You can click **Test Connection** to verify the settings.

4 Click **OK** to save the configuration settings and return to the ODBC Data Source Administrator dialog box. Then, click **OK** to save the data source.

Stage 2: Define the Database Server

Perform the following steps to define the database server metadata object:

1 Open the SAS Management Console application.

2 Right-click **Server Manager** and select the **New Server** option to access the New Server Wizard.

3 Select **ODBC Server** from the **Database Servers** list. Click **Next**.

4 Enter an appropriate server name in the **Name** field (for example, **ODBC Server**). Note that you can supply an optional description if you wish. One server is required for each DSN. Click **Next**.

5 Enter the following server properties:

Table 9.10 Server Properties

Field	Sample Value
Major Version Number	**3**
Minor Version Number	**7**
Data Source Type	**ODBC - Oracle**
Software Version	**10**

Field	Sample Value
Vendor	**Oracle**
Associated Machine	**newserver.na.sas.com** (Select this value from the drop-down list. If the value that you need is not available, click **New** to access the New Machine dialog box. Then enter the appropriate value in the **Host Name** field.)

Click **Next**.

6 Enter the following connection properties:

Table 9.11 Connection Properties

Field	Sample Value
Datasrc	**Oracle_newserver** (Use the value entered in the **Data Source Name** field in the ODBC Data Source Administrator dialog box.)
Authentication Domain	**ODBCAuth** (You need to create a new authentication domain each time you connect to a new database server. The SAS Metadata Server, Workspace Server, and others require one set of credentials, while the database server requires another. Click **New** to access the New Authorization Domain dialog box. Then enter the appropriate value in the **Name** field and click **OK** to save the setting. Make sure that the authentication domain that you create here is added to the appropriate users and user groups. Add a login to these users and groups that includes a user ID, a password, and the authentication domain.

Note: If you will have multiple users for the connection, consider creating a user group for them to avoid the inefficient process of creating separate user IDs and passwords for each. Use the Access Control Template in the Authorization Manager in SAS Management Console. △

Click **Next**.

7 Examine the final page of the wizard to ensure that the proper values have been entered. Click **Finish** to save the wizard settings.

Stage 3: Define the Database Schema

After you have defined the database server metadata, you can define the database schema metadata object. It is important to define the server object first because the server object must be entered into the wizard when you define the schema object. Perform the following steps:

1 In SAS Management Console, double-click **Data Library Manager**. Right-click **Database Schemas** and select the **New Database Schema** option to access the New Database Schema Wizard.

2 Select **ODBC Schema** from the **Database Schemas** list. Click **Next**.

3 Enter an appropriate server name in the **Name** field (for example, **ODBC Schema**). Note that you can supply an optional description if you wish. Click **Next**.

4 Enter the following schema properties:

Table 9.12 Schema Properties

Field	Sample Value
Database Schema Name	**ODBCSchema** (The database schema name is case sensitive.)
ODBC Server	**ODBC Server** (Select the server from the drop-down list.)

Click **Next**.

5 Examine the final page of the wizard to ensure that the proper values have been entered. Click **Finish** to save the schema settings.

Stage 4: Define the Database Library

After you have defined the database server metadata object and the database schema metadata object, you can define the database library metadata object. It is important to define the library object after the server and schema objects because these objects must be entered into the wizard when you define the library object. Perform the following steps:

1 In SAS Management Console, double-click **Data Library Manager**. Right-click **SAS Libraries** and select the **New Library** option to access the New Library Wizard.

2 Select **ODBC Library** from the **Database Libraries** list. Click **Next**.

3 Enter an appropriate library name in the **Name** field (for example, **ODBC Library**). Note that you can supply an optional description if you wish. Click **Next**.

4 Enter the following library properties:

Table 9.13 Library Properties

Field	Sample Value
Libref	**ODBCREF**
Engine	**ODBC**

You can also click **Advanced Options** to perform tasks such as pre-assignment and optimization. Click **Next** to access the next page of the wizard.

5 Enter the following settings:

Table 9.14 Advanced Option Settings

Field	Sample Value
Database Server	**ODBCServer** (Use the database server that you selected in the **New Server Wizard**.)
Default Login	**Login(ODBCAuth)** (Select your login from the drop-down list. This default login is used to resolve conflicts between multiple logins to an authentication domain. In such cases, the default login is used.)
Database Schema	**ODBCSchema** (Select the schema name that you entered in the **New Database Schema Wizard**.)

Click **Next**.

6 Select one or more SAS servers. The library is assigned to the servers included in this list. Click **Next**.

7 Examine the final page of the wizard to ensure that the proper values have been entered. Click **Finish** to save the library settings. At this point, you can import tables, as explained in "Verifying Access to Tables" on page 194.

Establishing Connectivity to a Microsoft Access Database by Using ODBC

Setting up a connection from SAS to a Microsoft Access database by using ODBC can be seen as a four-stage process, as follows:

1 Define an ODBC data source.
2 Define the database server metadata object.
3 Define the database schema metadata object.
4 Define the database library metadata object.

This example illustrates the process for establishing a SAS connection to an Access database. It assumes that the software for the database has already been loaded with the standard installation wizard for the database client. The following prerequisites have been satisfied:

- installation of SAS/ACCESS Interface to ODBC
- installation of Microsoft Access

For information about setting environmental variables when you use SAS/ACCESS to connect to Microsoft Access data on UNIX systems, see "Set SAS/ACCESS Environment Variables" on page 176.

Stage 1: Define the Data Source

First, you must define the ODBC data source. On Windows systems, follow these steps:

1 Open the Windows Control Panel. Then, double-click **Administrative Tools**. Finally, double-click **Data Sources (ODBC)** to access the ODBC Data Source Administrator dialog box.

2 Click **Add** to access the Create New Data Source dialog box. Click the Microsoft Access driver listed in the window (for example, **Microsoft Access Driver [*.mdb]**). Click **Finish** to access the Oracle ODBC Driver Configuration dialog box.

Note: System data sources and user data sources store information about how to connect to the indicated data provider. A system data source is visible to all users with access to the system, including NT services. A user data source is only visible to a particular user, and it can only be used on the current machine. △

3 Enter the following configuration settings:

Table 9.15 Configuration Settings

Field	Sample Value
Data Source Name	**MS Access**
Database	Click **Select** to browse for your Access database file, such as **Northwinds.mdb** in the Microsoft Office Samples directory.

4 Click **OK** to save the configuration settings and return to the ODBC Data Source Administrator dialog box. Then, click **OK** to save the data source.

Stage 2: Define the Database Server

Perform the following steps to define the database server metadata object:

1 Open the SAS Management Console application.

2 Right-click **Server Manager** and select the **New Server** option to access the New Server Wizard.

3 Select **ODBC Server** from the **Database Servers** list. Click **Next**.

4 Enter an appropriate server name in the **Name** field (for example, **MS Access Server**). One server is required for each DSN. Note that you can supply an optional description if you wish. Click **Next**.

5 Enter the following server properties:

Table 9.16 Server Properties

Field	Sample Value
Major Version Number	**3**
Minor Version Number	**7**
Data Source Type	**ODBC - Microsoft Access**
Software Version	**3.7.0**
Vendor	**Microsoft**
Associated Machine	**newserver.na.sas.com** (Select this value from the drop-down list. If the value that you need is not available, click **New** to access the New Machine dialog box. Then enter the appropriate value in the **Host Name** field.)

Click **Next**.

6 Enter the following connection properties:

Table 9.17 Connection Properties

Field	Sample Value
Datasrc	**MS Access** (Use the value entered in the **Data Source Name** field in the ODBC Data Source Administrator dialog box.)
Authentication Domain	**(None)**

Click **Next**.

7 Examine the final page of the wizard to ensure that the proper values have been entered. Click **Finish** to save the wizard settings.

Stage 3: Define the Database Schema

Most databases include schemas that must be identified as part of the connection process. Microsoft Access databases, though, require that you define a separate library for each DSN, and they do not use schemas. Nevertheless, SAS expects to connect to a schema. For this reason, you must create a blank schema object as a placeholder. Perform the following steps:

1 In SAS Management Console, double-click **Data Library Manager**. Then, right-click **Database Schemas** and select the **New Database Schema** option to access the New Database Schema Wizard.

2 Select **ODBC Schema** from the **Database Schemas** list. Click **Next**.

3 Enter an appropriate server name in the **Name** field (for example, **MS Access Schema**). Note that you can supply an optional description if you wish. Click **Next**.

4 Enter the following schema properties:

Table 9.18 Schema Properties

Field	Sample Value
Database Schema Name	Leave this field blank. (This field must be left blank because SAS expects a schema value, but Microsoft Access does not use one.)
ODBC Server	**MS Access Server** (Select the server from the drop-down list.)

Click **Next**.

5 Examine the final page of the wizard to ensure that the proper values have been entered. Click **Finish** to save the schema settings.

Stage 4: Define the Database Library

After you have defined the database server metadata object and the database schema metadata object, you can define the database library metadata object. It is important to define the library object after the server and schema objects because these objects must be entered into the wizard when you define the library object. Perform the following steps:

1. In SAS Management Console, double-click **Data Library Manager**. Then, right-click **SAS Libraries** and select the **New Library** option to access the New Library Wizard.
2. Select **ODBC Library** from the **Database Libraries** list. Click **Next**.
3. Enter an appropriate library name in the **Name** field (for example, **MS Access Library**). Note that you can supply an optional description if you wish. Click **Next**.
4. Enter the following library properties:

   Table 9.19 Library Properties

Field	Sample Value
Libref	**ACCESREF**
Engine	**ODBC**

   You can also click **Advanced Options** to perform tasks such as pre-assignment and optimization. Click **Next** to access the next page of the wizard.
5. Enter the following setting:

   Table 9.20 Advanced Option Setting

Field	Sample Value
Database Server	**MS Access Server** (Use the database server that you created in the New Server Wizard.)

   Click **Next**.
6. Select one or more SAS servers. The library is assigned to the servers included in this list. Click **Next**.
7. Examine the final page of the wizard to ensure that the proper values have been entered. Click **Finish** to save the library settings. At this point, you can import tables, as explained in "Verifying Access to Tables" on page 194.

Establishing Connectivity to an Scalable Performance Data Server

Setting up a connection from SAS to a Scalable Performance Data Server (SPD Server) can be seen as a four-stage process, as follows:

1. Define the server metadata object.
2. Define the server domain metadata object.
3. Define the library metadata object.
4. Configure the **libnames.parm** file.

This example illustrates the process for establishing a SAS connection to SPD Server. It assumes that the software for the database has already been loaded by using the standard installation wizard for the database client. The SPD Server client and server software must be installed before the connection can be established.

Stage 1: Define the Server

Perform the following steps to define the database server metadata object:

1 Open the SAS Management Console application.

2 Right-click **Server Manager** and select the **New Server** option to access the New Server Wizard.

3 Select **SAS Scalable Performance Data Server** from the **SAS Servers** list. Then, click **Next**.

4 Enter an appropriate server name in the **Name** field (for example, **SPDServer**). Note that you can supply an optional description if you wish. Click **Next**.

5 Enter the following server properties:

Table 9.21 Server Properties

Field	Sample Value
Major Version Number	**4**
Minor Version Number	**0**
Vendor	**SAS Institute**
SAS Compatibility	**SAS 9**

Click **Next**.

6 Enter the following connection properties:

Table 9.22 Connection Properties

Field	Sample Value
Host	**D1234**
Port Number or Name	**5200** (Enter the port number for the SPDS name server.)
Communication Protocol	**TCP**
Authentication Domain	**SPDSAuth** (You need to create a new authentication domain each time you connect to a new server. The SAS Metadata Server, Workspace Server, and others require one set of credentials, while the SPDS requires another. Click **New** to access the New Authorization Domain dialog box. Then enter the appropriate value in the **Name** field and click **OK** to save the setting. Make sure that the authentication domain that you create here is added to the appropriate users and user groups. Add a login to these users and groups that includes a user ID, a password, and the authentication domain.)

Note: If you will have multiple users for the connection, consider creating a user group for them to avoid the inefficient process of creating separate user IDs and passwords for each. Use the Access Control Template in the Authorization Manager in SAS Management Console. △

7 Examine the final page of the wizard to ensure that the proper values have been entered. Click **Finish** to save the wizard settings.

Stage 2: Define the Server Domain

After you have defined the server metadata, you can define the server domain metadata object. It is important to define the server object first because the server object must be entered into the wizard when you define the domain object. Perform the following steps:

1. In SAS Management Console, double-click **Data Library Manager**. Then, right-click **Database Schema** and select the New Database Schema option to access the New Database Schema Wizard.
2. Select **SAS Scalable Performance Data Server Domain** from the **Database Schemas** list. Then, click **Next**.
3. Enter an appropriate server domain name in the **Name** field (for example, **SPDServerDomain**). Note that you can supply an optional description if you wish. Click **Next**.
4. Enter the following domain properties:

   Table 9.23 Server Domain Properties

Field	Sample Value
LIBNAME Domain Name	**spdssrv** (This value will be entered into the **libnames.parm** file).
SPD Server	**SPDSServer** (Use the value that you entered in the **Host** field of the Connection Properties dialog box when you defined the server metadata object.)

   Click **Next**.
5. Examine the final page of the wizard to ensure that the proper values have been entered. Click **Finish** to save the schema settings.

Stage 3: Define the Library

After you have defined the server metadata object and the server domain metadata object, you can define the library metadata object. It is important to define the library object after the server and domain objects because these objects must be entered into the wizard when you define the library object. Perform the following steps:

1. In SAS Management Console, double-click **Data Library Manager**. Right-click **SAS Libraries**. Then, select the **New Library** option to access the New Library Wizard.
2. Select **SAS Scalable Performance Data Server v4 Library** from the **SAS Libraries** list. Click **Next**.
3. Enter an appropriate library name in the **Name** field (for example, **SPDServerLibrary**). Note that you can supply an optional description if you wish. Click **Next**.
4. Enter the following library properties:

Table 9.24 Library Properties

Field	Sample Value
Libref	**spdsrv**
Engine	**SASSPDS**

You can also click **Advanced Options** to perform tasks such as pre-assignment and optimization. Click **Next** to access the next page of the wizard.

5 Enter the following settings:

Table 9.25 Advanced Option Settings

Field	Sample Value
SPD Server	**SPDSServer** (Use the database server that you selected in the New Server Wizard.)
Default Login	**(None)** (Keep this default value. SPD Server does not use a schema, but SAS expects a value in this field.)
LIBNAME Domain	**spdsrv** (Select the domain name that you entered in the New Database Schema Wizard.)

Click **Next**.

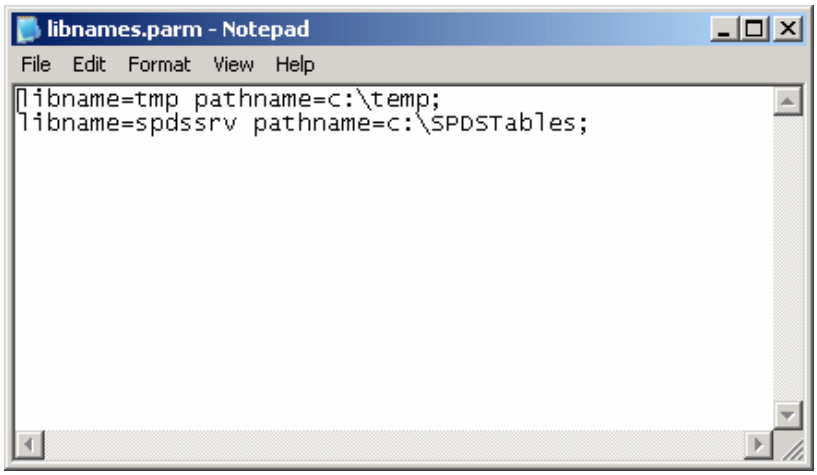
6 Select one or more SAS servers. (You might not need to select a server. The need to select a server depends on the applications that are included in your environment.) The library is assigned to the servers included in this list. Click **Next** to access the next page of the wizard.

7 Examine the final page of the wizard to ensure that the proper values have been entered. Click **Finish** to save the library settings. At this point, you can import tables, as explained in "Verifying Access to Tables" on page 194.

Configure the libnames.parm File

When you install the SPD Server software on Windows, a **libnames.parm** file is created in the **C:\Program Files\SAS Institute Inc\SPD v4.3\Site** directory. You must specify at least a LIBNAME and a pathname for the directory where the SPD Server tables will be saved (for example, **C:\SPDSTables**). For the LIBNAME, use the LIBNAME domain that you created earlier for the library (in this case, *spdsrv*).

A sample **libnames.parm** file is shown in the following display:



Establishing Connectivity to an SAP Server

Setting up a connection from SAS to an SAP server can be seen as a three-stage process, as follows:

1. Define the server metadata object.
2. Define the server domain metadata object.
3. Define the library metadata object.

This example illustrates the process for establishing a SAS connection to SAP. It assumes that the following software has already been loaded by using the standard installation wizard:

- SAP Front End. You must install this front end before you can install the SAS/ACCESS Interface to SAP R3.
- SAS/ACCESS Interface to SAP R3. This installation installs the SAS RFC service. Note that you must manually start this service on both Windows and UNIX each time that you start the SAS server.

Stage 1: Define the Server

Perform the following steps to define the SAS server metadata object:

1. Open the SAS Management Console application.
2. Right-click **Server Manager** and select the **New Server** option to access the New Server Wizard.
3. Select **SAP Server** from the **Enterprise Applications Servers** list. Then, click **Next**.
4. Enter an appropriate server name in the **Name** field (for example, **SAPServer**). Note that you can supply an optional description if you wish. Click **Next**.
5. Enter the following server properties:

Table 9.26 Server Properties

Field	Sample Value
Major Version Number	**4**
Minor Version Number	**6**
Software Version	**4.6**
Vendor	**SAP AG**

Click **Next**.

6 Enter the following connection properties:

Table 9.27 Connection Properties

Field	Sample Value
Authentication Domain	**SAPAuth** (You need to create a new authentication domain each time you connect to a new server. The SAS Metadata Server, Workspace Server, and others require one set of credentials, while the SAP server requires another. Click **New** to access the New Authorization Domain dialog box. Then enter the appropriate value in the **Name** field and click **OK** to save the setting. Make sure that the authentication domain that you create here is added to the appropriate users and user groups. Add a login to these users and groups that includes a user ID, a password, and the authentication domain.
RFC Server Host	**localhost** (If the SAP server is installed on the same machine as the workspace server, this value is populated automatically. If the SAP server is installed anywhere else, enter the appropriate value.)
RFC Server Port	**6999** (If the SAP server is installed on the same machine as the workspace server, this value is populated automatically. If the SAP server is installed anywhere else, enter the appropriate value.)
Client	**800** (This value is obtained from your SAP administrator.)
Language	**EN** (This value is obtained from your SAP administrator.)

Note: If you will have multiple users for the connection, consider creating a user group for them to avoid the inefficient process of creating separate user IDs and passwords for each. Use the Access Control Template in the Authorization Manager in SAS Management Console. △

7 Select **Application Server** and click **Options** to access the Application Server Host dialog box.

8 Enter the fully-qualified name of the server host that was supplied by the SAP administrator (for example, **sapsrv.na.sas.com**) in the **Application Server Host** field. Enter the system number that was supplied by the SAP administrator

(for example, **12**) in the **System Number** field. Then, click **OK** to return to the New Server Wizard.

9 Click **Next**.

10 Examine the final page of the wizard to ensure that the proper values have been entered. Click **Finish** to save the wizard settings.

Stage 2: Define the Schema

After you have defined the server metadata, you can define the schema metadata object. It is important to define the server object first because the server object must be entered into the wizard when you define the schema object. Perform the following steps:

1 In SAS Management Console, double-click **Data Library Manager**. Then, right-click **Database Schema** and select the **New Database Schema** option to access the New Database Schema Wizard.

2 Select **SAP Schema** from the **Enterprise Application Schemas** list. Then, click **Next**.

3 Enter an appropriate server domain name in the **Name** field (for example, **SAP Schema**). Note that you can supply an optional description if you wish. Click **Next**.

4 Enter the following schema properties:

Table 9.28 Server Domain Properties

Field	Sample Value
Database Schema Name	Leave this field blank.
SAP Server	**SAPServer** (Use the value that you entered in the **Name** field of the New Server Wizard when you defined the server metadata object.)

Click **Next**.

5 Examine the final page of the wizard to ensure that the proper values have been entered. Click **Finish** to save the schema settings.

Stage 3: Define the Library

After you have defined the server metadata object and the server domain metadata object, you can define the library metadata object. It is important to define the library object after the server and schema objects because these objects must be entered into the wizard when you define the library object. Perform the following steps:

1 In SAS Management Console, double-click **Data Library Manager**. Right-click **SAS Libraries**. Then, select the **New Library** option to access the New Library Wizard.

2 Select **SAP Library** from the **Enterprise Applications Libraries** list. Click **Next**.

3 Enter an appropriate library name in the **Name** field (for example, **SAP Library**). Note that you can supply an optional description if you wish. Click **Next**.

4 Enter the following library properties:

Table 9.29 Library Properties

Field	Sample Value
Libref	**SAPLib**
Engine	**SASIOSR3** (Accept the value that is populated automatically.)

Click **Next**.

5 Select the SAP server that you entered in the **Name** field of the New Server Wizard (for example, **SAP Server**) by using the **Database Server** drop-down list. Then, click **Next**.

6 (Optional) Select one or more SAS servers. (You might not need to select a server. The need to select a server depends on the applications that are included in your environment.) The library is assigned to the servers included in this list. Click **Next**.

7 Examine the final page of the wizard to ensure that the proper values have been entered. Click **Finish** to save the library settings. At this point, you can import tables, as explained in "Verifying Access to Tables" on page 194.

Verifying Access to Tables

You need to make sure that the end users of your SAS applications can gain access to tables in your data libraries. The exact steps and authorization requirements vary across applications and data types, but you must always log on to the application, create the needed metadata, and verify the existence of the tables. This example will focus on the process used to verify SAS tables in SAS Management Console.

Verifying your access to tables in SAS Management Console can be seen as a two-stage process, as follows:

1 Import the tables.

2 View the data in SAS Management Console.

Stage 1: Import the Tables

To import the tables, perform the following steps.

1 Open SAS Management Console, if necessary. Be sure to select the metadata profile of a user who is not an unrestricted user.

2 Double-click **Data Library Manager**. Then, double-click **SAS Libraries** to see the list of libraries. Right-click the library that contains the tables that you need to import. Then, select the **Import Tables** option to access the Connect to SAS page of the Import Tables wizard.

3 Select a server.

4 Select the library that contains the tables from the **SAS Library** drop-down list.

5 Verify that the values shown in the fields in the **Library details** group box are correct. Click **Next**.

6 Click the tables that you need to select. (Hold down the CTRL key and click to select more than one table.) Click **Next**.

7 Examine the final page of the wizard to ensure that the proper values have been entered. Click **Finish**.

Note: You can also import tables using the Source Designer in SAS Data Integration Studio. △

Stage 2: View the Data in a SAS Application

Open an application that can view SAS data in order to view the data in the imported tables and review the data. For example, you can perform the following steps in SAS Data Integration Studio:

1 Navigate to the **Inventory** tree and double-click **Tables**.
2 Right-click a table that you need to verify, and select the **Properties** option.
3 Click the **Columns** tab to see column data for the table.
4 Click the **Advanced** tab. Notice that the time and date that the metadata for the table was created is displayed in the **MetadataCreated** field. Click **Cancel** to dismiss the Properties dialog box.
5 Right-click the table again and select the **View Data** option. Examine the data contained in the table in the View Data window.
6 Close the View Data window.

CHAPTER

10

Assigning Libraries

Overview of Assigning Libraries

What Does It Mean to Assign a Library?

In Chapter 9, "Connecting to Common Data Sources," on page 167, you learned how to define SAS libraries in metadata. These libraries represented such things as the set of SAS data sets in a directory or the set of tables in a database schema. This chapter explains how to assign libraries so that the SAS servers in the environment know where the libraries are located and how to access them.

Assigning a library means letting a SAS session know that a libref—a shortcut name—is associated with the information that the SAS session needs to access a data library. For example, if you were writing a SAS program that needed to access a library of SAS data sets, your program might include the following statement:

```
LIBNAME ORGOLD BASE 'C:\SAS\EntBIServer\Lev1\SASMain\Data\orgold' repname=Foundation;
```

In this case, the libref ORGOLD tells the SAS session that it should access data sets in the directory **`C:\SAS\EntBIServer\Lev1\SASMain\Data\orgold`** using the BASE data-access engine.

SAS Intelligence Platform clients such as SAS Data Integration Studio, SAS OLAP Cube Studio, and SAS Information Map Studio generate SAS code that makes use of librefs. Before this code can execute, the corresponding library must have been assigned, and the server that will execute the code must know about that assignment.

Pre-assigning Libraries

There are two ways in which a server can find out about a library reference. One way is for you, as the administrator, to configure the environment so that the server finds out about the libref at server startup. This approach is referred to as *pre-assigning* the library, because the libref is established before any code that uses that libref is submitted. The other way is to let the client application define the libref for a server when it generates code for submission to that server.

Deciding whether to pre-assign a library or not has important consequences. These are described in "Data-Access Engines and the MLE" on page 199. SAS clients and stored processes can access a library using one of two engines:

- the engine specified in the library's metadata. This would be the Base SAS engine for libraries of SAS data sets, the ORACLE engine for Oracle libraries, and so forth.
- the metadata LIBNAME engine (MLE).

Which engine you use affects security and dictates what operations are possible.

Note: Avoid the "Pre-Assigned Library" template. When pre-assigning a library, be sure to choose the resource template specific to the type of data source library you are creating and select the **`This library is pre-assigned`** checkbox. Do not use the specialized "Pre-Assigned Library" template, which is intended for certain system libraries only; it will not work for other libraries. △

If you pre-assign libraries, *you* control which engine is used to access the data. If you do not pre-assign a library, the client that needs to access that library decides which engine to use, and different clients use different strategies. For example, SAS Data Integration Studio and SAS OLAP Cube Studio always use the engine stored in the library's metadata, while SAS Enterprise Guide can use either the MLE or its native engine. For more information, see "Managing Libraries" in the chapter "Managing Metadata Objects" in *SAS Management Console User's Guide*.

Having the server process assign libraries upon start-up based on information in the metadata results in library assignments that are identical and guaranteed across all SAS client applications and servers. Some environments where this approach to assigning libraries is desirable include the following:

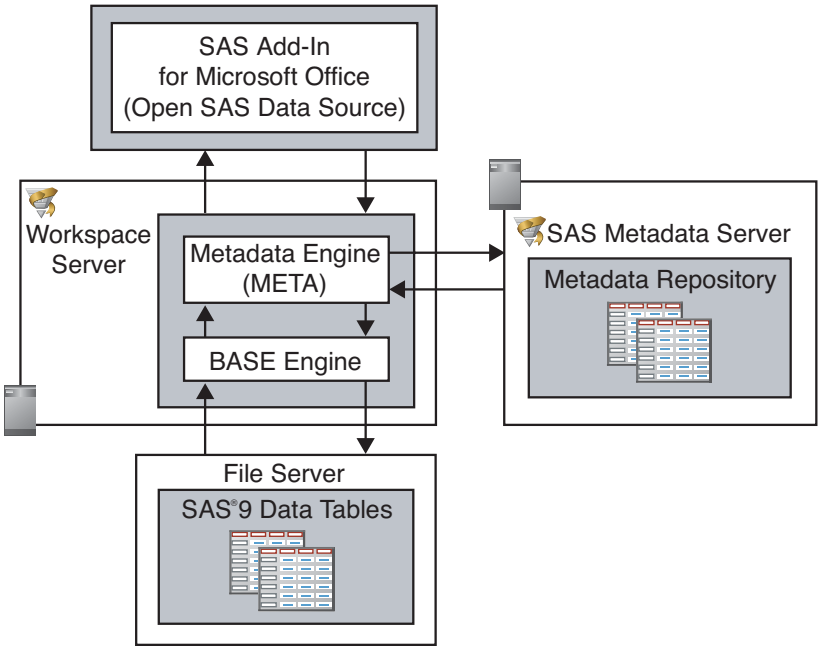
- Environments where users are executing stored processes, and you don't want programmers having to manage library assignments in their code or in autoexec files.
- Environments where the Data Step Batch Server is used to execute jobs created by SAS Data Integration Studio, and library assignments for these jobs should be identical to assignments used when the process was created.
- Environments where SAS Enterprise Guide or SAS Add-In for Microsoft Office users will be running tasks that need to create tables in the library defined in the metadata. Recall that when you define a client-assigned library (a library that is not pre-assigned), SAS Enterprise Guide and SAS Add-In for Microsoft Office will assign the library to use the META engine by default. Recall further that a library assigned with the MLE should not be used as the location for output or target tables.

When libraries are assigned by the client application, each application can assign the library in a way that is most suitable for its intended user base, and library connections are established only if needed. When libraries are assigned by the server, each library is available to all back-end server processes and is allocated exactly the same way for all client applications. A mixture of some server-assigned and some client

application-assigned libraries will most likely be required to meet the needs of all the users in your environment.

Data-Access Engines and the MLE

As mentioned previously, when you access the data in a data library, you can use the data-access engine stored in the metadata definition of the library, or you can use the MLE. As shown in the following display, the MLE invokes the SAS/ACCESS engine stored in the metadata.



One of the key enhancements made in SAS®9 has been the introduction of the SAS Open Metadata Architecture authorization facility. This authorization facility gives you, the administrator, the ability to control which users can access which metadata objects, such as SASLibrary, PhysicalTable, and LogicalServer. You manage their access to metadata by setting ReadMetadata and WriteMetadata permissions on the object or on the repository.

As depicted in the preceding diagram, when SAS users expand a library that they have ReadMetadata access to and that has been assigned to use the MLE, the engine first sends a request to the SAS Metadata Server asking for the users' metadata permissions on the tables in the library. A list of tables for which the users have at least ReadMetadata access will be returned and presented to them for selection. If they then attempt to perform some action against one of those tables, such as opening it, the MLE sends a query directly to the authorization facility asking for the their data-level permissions on that table. If the users or the group to which they belong have at least Read access to the table, the MLE will call upon the engine specified in the metadata to handle the request, and the table will be opened into the client application for reading.

Client applications contact the metadata server and request access to a metadata object as the user. The metadata server then queries the SAS authorization facility to determine if users have ReadMetadata, CheckInMetadata, or WriteMetadata permission to the object. These metadata-based permissions are the only permissions checked by the metadata server. Users attempting to access a table in a SAS

metadata-based library pre-assigned by the server will be successful if they have ReadMetadata access to the library and, in the case of SAS Data Integration Studio, SAS OLAP Cube Studio, and SAS Information Map Studio, also to the table.

In contrast, data-level authorizations of Read, Write, Create, and Delete are never checked by the metadata server, because a metadata object is not involved. As a result, if you need to use the SAS authorization facility to control which users can access a physical table or library, then you need to assign the server-side library using the MLE. When used with its default options, the MLE will query the metadata server for metadata-based permissions. The SAS authorization facility must be queried for data-level permissions. When MLE libraries are defined in an autoexec file through a LIBNAME statement, they are always pre-assigned.

The general form of a LIBNAME statement for the META engine is as follows:

```
LIBNAME libref META LIBID=id|LIBURI=URI-format|LIBRARY=name
   <connection-options><engine-options>;
```

Therefore, a META LIBNAME statement for the Orion Gold Customers library defined in the metadata would look something like the following:

```
ORGOLD META library="Orion Gold Customers" METAEEPOSITORY="Foundation";
```

This is the minimum information you would need to supply in the LIBNAME statement itself. However, this statement will only work if the META* options that contain the information necessary to connect to the metadata server have already been specified. These options, METASERVER, METAPORT, METAREPOSITORY, and METAPROTOCOL, are defined in the **sasv9.cfg** file already if you used the Configuration Wizard to set up your environment.

Having data requests flow through the MLE before they reach the engine that actually fulfills the request provides an important capability: the MLE enforces the data-level authorizations that are available in the SAS Authorization Manager. These include the Read, Write, Create, and Delete permissions. The other data-access engines ignore these permissions.

At the same time, using the MLE takes away some capabilities. Most important, the MLE should not be used in its default mode to create or delete tables. So it is most suited to read-only applications.

Using Libraries That Are Not Pre-assigned

By default, newly created libraries are not pre-assigned. When a library is not pre-assigned, the library is assigned by using the data-access engine that best suits the client application and its intended user base. Thus, the default assignments for applications such as SAS Data Integration Studio, SAS Add-In for Microsoft Office, SAS Enterprise Guide, SAS OLAP Cube Studio, SAS Enterprise Miner, and SAS Information Map Studio are employed. For example, if you do not pre-assign the library, SAS Data Integration Studio will assign it for the user in such a way that data-level authorizations of Read, Write, Create, and Delete are not imposed on the clients. Data requests are sent directly to the engine specified in the library's metadata (such as BASE) without checking data-level authorizations. This approach was chosen as a best practice, because it is assumed that in most cases SAS Data Integration Studio developers will be building processes that create or update tables in the library and that the underlying engine is the only engine that should be used for data-populating tasks.

How Do the Different Platform Clients Assign Libraries?

When libraries are not pre-assigned, each SAS platform client assigns libraries. Allowing each application to assign libraries as it deems appropriate for its user base results in the optimal security model for environments where users have different data access requirements to a library and where you want to capitalize on using metadata decisions enforced by the SAS authorization facility on top of the operating system or RDBMS authorization layer. An example of such an environment would be one with clients running at least SAS Enterprise Guide and SAS Data Integration Studio. In this environment, SAS Data Integration Studio processes update tables that are in turn used in ad hoc analysis within SAS Enterprise Guide. The SAS Data Integration Studio processes need to specify tables in the library as target tables (output), whereas the SAS Enterprise Guide user's activities largely involve querying and analyzing chunks of data (input).

Because SAS Data Integration Studio processes typically update or create target tables, when SAS Data Integration Studio assigns the library it does not use the META engine. Instead, it assigns the library using the engine specified in the metadata. Because SAS Data Integration Studio only works with tables that are defined in the metadata repository, you can use the SAS authorization facility to control a client's access to tables by setting ReadMetadata, WriteMetadata, and CheckInMetadata permissions on the library and table metadata objects.

SAS Information Map Studio always assigns the library by using a LIBNAME statement and the engine specified in the metadata, unless the library is explicitly defined by a SAS administrator (or SAS Data Integration Studio administrator) to use the META engine.

Note: The metadata authorization layer supplements operating system- and RDBMS-level security. It does not replace it. Operating system and RDBMS authorization layers can and should always be employed as the first means of securing access to tables. △

On the other hand, the SAS Add-In for Microsoft Office and SAS Enterprise Guide (shown in the following table) assign the library using the META engine by default, so that data-level authorizations of Read, Write, Create, and Delete, which are specified in the metadata, are enforced. If defining libraries so that they are not pre-assigned seems like a potential option for your environment, then you will want to explore this topic a little further and learn how to ensure that these libraries will be available to server processes that do not receive direct requests from client applications. For example, you will need to learn how to manually assign the library in server processes such as the stored process server and Data Step Batch Server (if present), as discussed in the next section.

Table 10.1 Platform Client Default Library Assignments

Application	Pre-Assigned	Library Engine Used	Minimum Metadata Authorizations Required
SAS Add-In for Microsoft Office	No	META	Library: ReadMetadata Table: ReadMetadata and Read
SAS Enterprise Guide	No	META	Library: ReadMetadata Table: ReadMetadata and Read

Application	Pre-Assigned	Library Engine Used	Minimum Metadata Authorizations Required
SAS Data Integration Studio	No	Underlying data engine	Library: ReadMetadata Table: ReadMetadata
SAS OLAP Cube Studio	No	Underlying data engine	Library: ReadMetadata Table: ReadMetadata
SAS Information Map Studio	No	Underlying data engine	Library: ReadMetadata Table: ReadMetadata

Processing a Stored Processes When the Library is Not Pre-assigned

In the SAS®9 Intelligence Platform, a stored process is a SAS program that is stored on a server and can be executed as requested by clients who have ReadMetadata access to the stored process program's metadata. SAS Stored Processes can be executed by either a SAS Workspace Server or a SAS Stored Process Server. If a library is not pre-assigned, it is the responsibility of the stored process program's author or the SAS administrator to ensure that the library is assigned to a specific location and physical path. This can be done either directly in each stored process program or from an external file that is linked to the stored process with an %INCLUDE statement.

These methods have the following advantages and disadvantages:

- Method: Define a META engine library in the stored process program.
  - Advantage: Data-level authorizations specified for the library and table metadata objects are enforced by the SAS authorization facility.
  - Disadvantage: Library and table metadata for any table called in the program must be defined in the metadata repository, thus preventing a stored process from accessing tables that might reside in the library but are not defined in the metadata.
  - Disadvantage: Changes to the library metadata object's name or repository location would require that each stored process that references the library be updated.
  - Disadvantage: Metadata inconsistencies or corruptions can result if the stored process modifies a table's structure through the library. Examples of this modification include adding or removing columns.
- Method: Define the library in the stored process program and use only the underlying data engine.
  - Advantage: A table does not have to be defined in the metadata repository in order for the stored process to access it.
  - Advantage: Tables in the library can be re-created or updated and new tables created without directly impacting the metadata. Note, however, that changes to the structure of a table that has been defined previously in the metadata repository can still cause synchronization issues between the table and the metadata.
  - Disadvantage: The metadata repository is no longer a single point of management, because library definitions are stored in multiple places.
  - Disadvantage: Changes to the library path or directory would require that each stored process that references the library be updated.

- Disadvantage: The SAS authorization facility has no role in managing access to tables called by the stored process. Thus, the SAS General Server User can access data in any table in the library for which he has been granted Read access at the OS or RDBMS layer.

- Method: Store the library assignment(s) in an external file and then include it in the stored process program.
  - Advantage: Library assignments are defined in one file or directory location that all stored process programs can reference.
  - Advantage: Multiple files that contain library assignments can be created and referenced as needed in the stored process so that things such as connections to databases are established only when absolutely required.
  - Advantage: Other advantages depend on how the library is defined in the file. See the previous two methods.
  - Disadvantage: The file(s) referenced in the stored process must be created and maintained by someone who has Write (and Modify) access to the file's location on the system.
  - Disadvantage: Stored processes created through point and click applications such as SAS Enterprise Guide will have to be modified manually to replace the library assignment with manually generated %INCLUDE syntax.
  - Disadvantage: Changes to the file's location or name would require that all stored processes, including that file, be updated.

Pre-Assigning Libraries Using Engines Other Than the MLE

Pre-assigning a library ensures that the library will always be available to and assigned by SAS server processes on a server-by-server basis when the server starts, rather than assigned by the client application or later in SAS code. Two types of pre-assignment are possible. First, you can pre-assign a library so that it will be accessed by the engine defined in the metadata. Second, you can pre-assign a library so that it will be accessed by the MLE. In either case, pre-assignment allows you to designate an assignment method for use by all of the applications that use the library. However, it should not be used when you create output from the applications if the assignment is made by using the META engine.

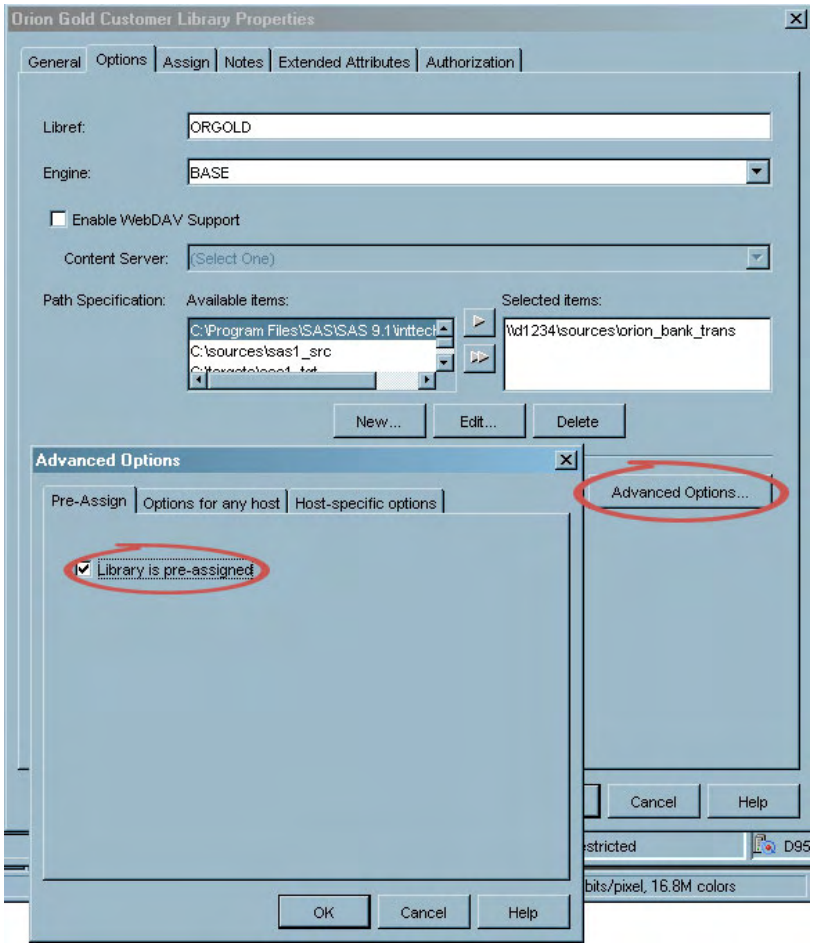
Pre-assigning a library to a non-MLE server requires the following steps:

1. Flag the library as already assigned by the server by selecting the **`Library is pre-assigned`** advanced option in SAS Management Console.
2. Set up the server process to retrieve library metadata by adding the METAAUTOINIT object server parameter option or the METAAUTORESOURCES SAS system option to the server's start-up definition.

Note: Library definitions (and therefore, table definitions) that are pre-assigned using the METAAUTORESOURCES option must exist in the same repository in order to support pre-assigned librefs. The code that pre-assigns the libraries on start-up of the server does not look down the repository dependency chain to find library definitions. This is a known limitation with pre-assigned libraries and the METAAUTORESOURCES option. Libraries must be defined in the same repository as the server context definition. △

Assume that we are pre-assigning the Orion Gold Customers library. The library can be set up to be assigned by the server process by either selecting the **`Library is pre-assigned`** advanced option when the library is being defined or by modifying the

library's properties after the fact. When you select the **`Library is pre-assigned`** option, any server processes that have been configured to do so will connect into the metadata server on start-up and assign the library. The libraries can be assigned to SAS application servers (which may include a workspace server, a stored process server, and/or an OLAP server) or to a server defined outside an application server context, such as a SAS/SHARE server. To access the advanced options for the library, select the following in SAS Management Console: **Data Library Manager ▶ SAS Libraries ▶ Orion Gold Customers ▶ Properties**. Then, click the **`Options`** tab and click **`Advanced Options`**.

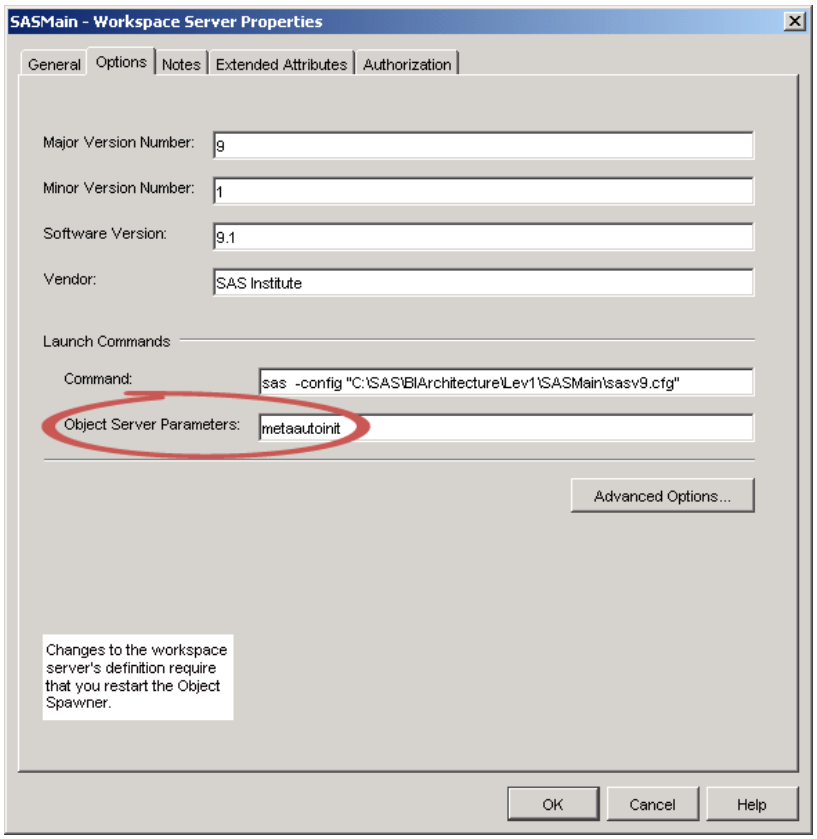


With the Orion Gold Customers library flagged to be assigned at start-up, the next step is to define which server processes should assign libraries. The option or parameter that you use to tell a server process to connect into the SAS Metadata Server during start-up depends on the type of server that is handling the request.

The server process must contain the METAAUTOINIT object server parameter in the server start-up definition for the following IOM (Integrated Object Model) servers: the workspace server, the stored process server, and the OLAP server. If the server process is any other SAS session, you use the METAAUTORESOURCES system option. (These servers include Data Step Batch Servers, the SAS Display Manager System, the SAS/CONNECT Server, and the SAS/SHARE Server.)

Pre-assignment Using Information Stored in the Metadata

The METAAUTOINIT option can be specified in the IOM server's definition through SAS Management Console or in the configuration file defined by the configuration option in the start-up command. The following example shows how you supply the METAAUTOINIT option in the **`Object Server Parameters`** field in the server's properties to servers that are started by the object spawner. Modifying the server's metadata definition requires that the object spawner be restarted in order for it to obtain the changes. To access the server's properties, select the following in SAS Management Console: **Server Manager ▸ SASMain ▸ Workspace Server ▸ Properties**. (Obviously, the path would be different for different servers).



Note: Server processes that are currently running will not pick up the new configuration options until they are restarted. Therefore, load-balanced processes on stored process servers that use the default configuration of staying open once started will need to be shut down. The same is true for pooled workspace servers. The object spawner will restart them as needed.

The METAAUTOINIT OBJECTSERVERPARMS option could be referenced in a configuration file called by the workspace server instead of being supplied in the server's definition. However, because the default configuration file called by the workspace server (**`sasv9.cfg`**) is also used by non-IOM servers executing on the host, it is recommended that you create a custom configuration file and place it in the workspace server directory, which is *SAS-config-dir***`\Lev1\SASMain\WorkspaceServer`**. Use this file to store the METAAUTOINIT option. Options specified in a configuration

file take precedence over conflicting options in the server's definition. The following custom configuration files are created during installation for the other IOM servers:

- Metadata server:
  SAS-config-dir`\Lev1\SASMain\MetadataServer\sasv9_MetadataServer.cfg`
- OLAP server:
  SAS-config-dir`\Lev1\SASMain\OLAPServer\sasv9_OLAPServer.cfg`
- Stored Process server:
  SAS-config-dir`\Lev1\SASMain\StoredProcessServer\sasv9_StorProcSrv.cfg`

△

Display 10.1 Options Specified in a Configuration File

```
*/
-config "C:\SAS\BIArchitecture\Lev1\SASMain\sasv9.cfg"

-nosplash
-noterminal

-log "C:\SAS\BIArchitecture\Lev1\SASMain\StoredProcessServer\logs\StoredProcessServer_%v.log"
-logparm "rollover=session open=replaceold write=immediate"

-objectserverparms "metaautoinit"

-pagesize max
-linesize max

-sasuser "C:\SAS\BIArchitecture\Lev1\SASMain\StoredProcessServer\sasuser"
```

SAS servers that are not IOM servers connect into the metadata server to retrieve definitions through the use of META* system options, which are typically specified in the configuration file. These options provide a SAS session with metadata information such as the name of the metadata server host, its port number, and optionally the metadata identity, to use for authentication decisions in the metadata repository. In addition, the METAUTORESOURCES option tells the SAS session which specific metadata object it should retrieve in order to determine libraries to assign at start-up. The following example shows how the METAAUTORESOURCES option can be used to reference a specific SAS Application Server in the configuration file for the Data Step Batch Server or SAS Display Manager session, causing it to retrieve pre-assigned libraries that are associated with the SASMain server context. The configuration file is located in *SAS-config-dir*`\Lev1\SASMain\sasv9.cfg`.

Display 10.2 Metaautoresources Option

```
sasv9.cfg - Notepad
File Edit Format View Help
-config "C:\Program Files\SAS\SAS 9.1\sasv9.cfg"
-sasinitialfolder "C:\SAS\BIArchitecture\Lev1\SASMain"
-set library ("SASEnvironment/sasFormats")
-sasautos ("SASEnvironment/sasMacro" SASAUTOS)
-dqlocale (ENUSA)
-dqsetuploc "dqsetup.txt"
-rsasuser
-metaserver "d9576.na.sas.com"
-metaport 8561
-metarepository "Foundation"
-metaprotocol BRIDGE
-metaautoresources "SASMain"
-autoexec "C:\SAS\BIArchitecture\Lev1\SASMain\appserver_autoexec.sas"
```

The METAAUTORESOURCES system option is set differently for a SAS/SHARE server. Recall that a SAS/SHARE server is not defined within a server context, such as SASMain, for example. Because it is not defined within a server context, the SAS/SHARE server will not use the METAAUTORESOURCES option specified in the previous example. To tell the SAS/SHARE server to connect into the metadata and retrieve pre-assigned metadata-based libraries, you must add the specific SAS/SHARE server component that it should retrieve to its configuration file in the following path: *SAS-config-dir***\Lev1\SASMain\ShareServer**. To do this, add a line similar to the following to the **sasv9.cfg** file:

```
-metaautoresources "omsobj:ServerComponent?@Name=SASMain - SAS/SHARE Server"
```

Pre-assignment Using Information in an Autoexec File

Recall that a SAS autoexec file is a text file that contains SAS statements that are executed upon start-up of the server process. If an autoexec file is being used in your environment, it is important to note that libraries assigned by an autoexec file take precedence over same-named libraries assigned by the server via METAAUTORESOURCES or METAAUTOINIT. (Use the autoexec file created during installation, which is *SAS-config-dir***\Lev1\SASMain\appserver_autoexec.sas**.) For example, if ORGOLD is defined in the metadata to be server-assigned, and ORGOLD is defined in a LIBNAME statement in an autoexec file defined to the server, the ORGOLD library will be assigned using the information in the autoexec file. Simply put, the library assignment in the autoexec file always wins.

Display 10.3 Library Assignment in an Autoexec File

```
/* Notice this LIBNAME maps ORGOLD to a different location than      */
/* the location in the metadata. This can cause unexpected           */
/* results and definite failures in metadata-dependent applications. */
LIBNAME ORGOLD BASE "D:\OrionStar\NotGold" access=readonly;
LIBNAME TABLES 'c:\SAS\Training\Lev1\SASApp\Data' access=readonly;
LIBNAME MISC 'c:\SAS\Training\Lev1\SASApp\Data';
LIBNAME GLDMACRO
"C:\SAS\Training\Lev1\SASApp\SASEnvironment\SASMacro\GoldProgrammers"
access=readonly;
option fmtsearch= (MISC.MOREFMTS) nofmterr;
options MSTORED SASMSTORE=GLDMACRO;
/* initialize some macros */
%let dbdsoptions=;
%let dbtempschema=;
%let dbmstemp=;
```

Pre-assigning Libraries to Use the MLE

The MLE is a data access engine that enforces the data-level permissions of Read, Write, Create, and Delete that are set on table objects in the repository. It also enforces the Create and Delete permissions that are set on library objects. The META engine acts as a gatekeeper that determines which users can access which metadata-based libraries and tables.

To define a library that uses the MLE, complete the following steps:

1. Define metadata for the library in the SAS Metadata Repository.
2. Mark the library as pre-assigned.
3. Construct a LIBNAME statement that uses the same libref specified in the metadata and META as the engine.

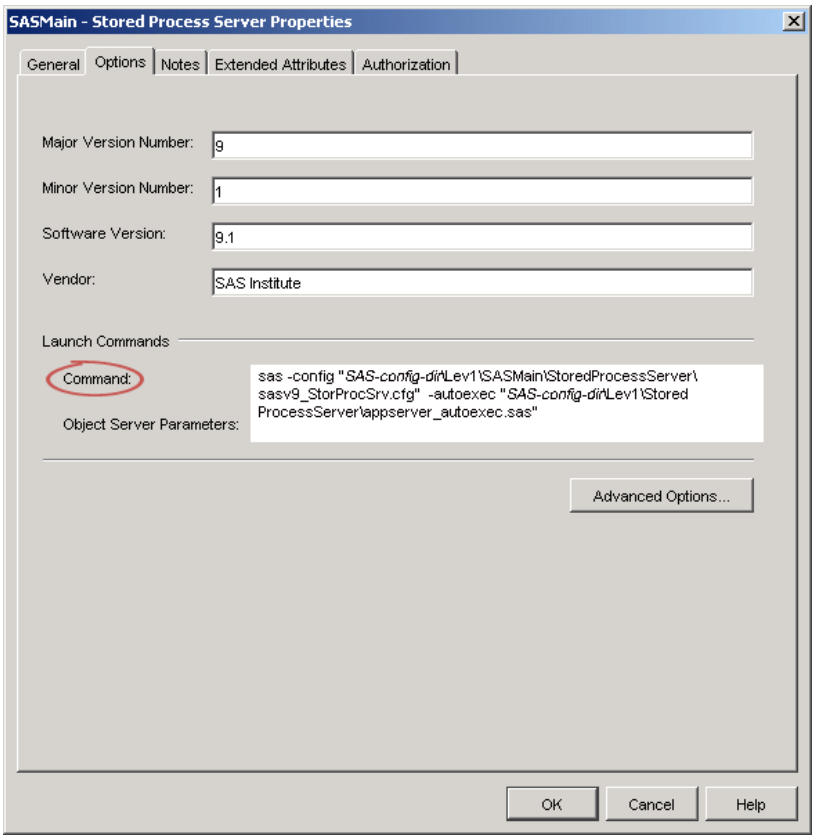
4 Add the metadata LIBNAME statement to an autoexec file. During the configuration process, the Configuration Wizard created a file named **`appserver_autoexec.sas`** and placed it in the same directory as the SAS Intelligence Platform's configuration file, **`sasv9.cfg`**. In platform implementations that do not include a SAS Solution, such as Marketing Automation, this file is typically empty. The purpose of this text file is to serve as the location where SAS solutions along with administrators like you can place SAS statements that need to be executed immediately after the SAS server process initializes and before any user input is accepted. This is considered the SAS Intelligence Platform's autoexec file. Add the LIBNAME statement created in Step 3 to this file as shown.

5 Add the autoexec file to the start-up definition of any server that should assign the library using the META engine. There are a variety of ways to tell a SAS server process that it should use the autoexec file created in the previous step. Here are the two approaches you've already seen in this chapter:

- Call the autoexec file from the server process's configuration file (ideally a shared configuration file). Find the **`sasv9.cfg`** file and enter text similar to the following:

```
-autoexec 'SAS-config-dir\Lev1\SASMain\appserver_autoexec.sas'
```

- Add the autoexec system option to each server's metadata object definition. Note that this option is only available to IOM servers.



6 Restart the object spawner and any server processes whose autoexec files have been modified. The SAS/SHARE server must be restarted in order to pick up changes to the configuration file. On Windows it can be restarted using the **`NET STOP`** and **`NET START`** commands at a command prompt on the SAS/SHARE server's

host, or through the Windows Service Manager. For more information about restarting the servers, see "Starting and Stopping Your SAS Servers" on page 18.

Note: We recommended that you treat libraries mapped with the META engine as read-only and do not allow any users Create access to the library or its tables. △

Verifying Pre-Assignments by Reviewing the Logs

After you specify that a library is to be assigned by the server (by specifying the METAAUTOINIT or METAAUTORESOURCES system option), the SAS server process will start up as follows:

1. Connect to the metadata server.
2. Retrieve library metadata.
3. Assign the library using the engine specified in the library definition.

For example, if the Orion Gold Customers library is assigned by the workspace server, then the library assignment would be equivalent to a SAS programmer submitting a LIBNAME statement such as the following:

```
LIBNAME ORGOLD BASE "D:\OrionStar\Gold" repname=Foundation;
```

In the case of an IOM server (using METAAUTOINIT), you can verify the pre-assignment of this library by the server process by enabling logging and observing the note generated from the first GetMetadata method call in the server's log, as in the following sample log:

Display 10.4 Verification of Pre-assignment in a Server Log

```
NOTE: Libref ORGOLD successfully assigned from logical server.
…
:: IOM RETURN 0={compRef:23b2b00}->OMIProxy::GetMetadata():
outMetadata=
< SASLibrary Id="A5R8WXTZ.B6000001" Libref="ORGOLD" Engine="BASE"
     IsDBMSLibname="0" IsPreAssigned="1"> …
```

For non-IOM servers using the METAAUTORESOURCES option, a note like the following would be written to its log file:

```
NOTE: Libref ORGOLD successfully assigned from logical server.
```

See "Configure the Logging of XML" on page 35 for information about setting logging levels. To verify pre-assignment, you should set the logging level to **1**.

CHAPTER

11

Managing Table Metadata

Overview of Managing Table Metadata

As explained in “Verifying Access to Tables” on page 194, one way to create metadata for the tables in a library is to use the Import Tables feature of SAS Management Console. You can also create this metadata programmatically by using PROC METALIB. In addition, PROC METALIB provides you with options for maintaining your table metadata that are not available in SAS Management Console. For example, by default PROC METALIB creates metadata definitions for any physical tables that are not registered in the metadata—for instance, tables that have been added since the table definitions were first created—and updates table definitions that do not reflect the current structure of the tables that they represent.

By using optional statements, you can also use PROC METALIB to perform the following tasks:

- Delete table definitions for tables that have been removed from the library.
- Produce a report that lists the changes made by the procedure—or the changes that will be made when the procedure is executed.
- Operate on a subset of the tables in a library.

Note: For detailed information about PROC METALIB and its syntax, see “METALIB Procedure” in the *SAS Open Metadata Interface: Reference*. △

Note: PROC METALIB cannot work with a library whose metadata is defined by using the "Pre-assigned Library" resource template. When pre-assigning a library, be sure to choose the resource template specific to the type of data source library you are creating and select the **`This library is pre-assigned`** checkbox. Do not use the specialized “Pre-Assigned Library” template. △

The remainder of the chapter presents examples of how PROC METALIB is commonly used. The examples assume that you have set the following metadata server connection options in your SAS session.

```
options METAUSER = "metadata-server-userid"
        METAPASS = "metadata-server-password"
        METAPORT = metadata-server-port
        METASERVER = "metadata-server-machine";
```

If you have not set these options, you can use PROC METALIB parameters to specify this information.

Creating Table Metadata for a New Library

When you first define a SAS library, it has no related table metadata. You can add this metadata by using the Import Tables wizard in SAS Management Console (see "Overview of Managing Table Metadata" on page 211), or by using PROC METALIB, as shown in the example below.

Before you can successfully run this PROC METALIB code, you must have Create, ReadMetadata, and WriteMetadata access to SASLibrary metadata object. To check your permissions, use the Authorization Manager in the SAS Management Console. (See "Managing Permissions" in the chapter "Managing Authorizations" in the *SAS Management Console: User's Guide*.)

Example: Creating Table Metadata

The following example shows how to use PROC METALIB to create initial table definitions for the tables in a library. The REPORT statement causes the procedure to write information to SAS output about the table definitions that it creates.

```
proc metalib;
  omr  (library=&mlibname repname= &mrepname );
  report;
run;
```

The report that this code writes would resemble the following sample.

```
                              The METALIB Procedure

                         Summary Report for Library sas91 lib2
                              Repository Meta Proc repos
                                       17MAR2005

                               Metadata Summary Statistics

                          Total tables analyzed           2
                          Tables Updated                  0
                          Tables Added                    2
                          Tables matching data source     0
                          Tables not found                0


 ------------------------------------------------------------------------------------
```

```
                                        Tables Added
-------------------------------------------------------------------------------


Metadata Name                      Metadata ID          SAS Name

COUNTRY                            A5HJ58JU.AX001LPV    COUNTRY
POSTAL                             A5HJ58JU.AX001LPW    POSTAL
```

Assessing Potential Changes in Advance

Before you use PROC METALIB to update existing table metadata, it is a good idea to execute the procedure with the NOEXEC and REPORT statements. The NOEXEC statement tells the procedure not to actually add, update, or delete any metadata. The REPORT statement tells the procedure to create a report that explains what actions it would have taken had the NOEXEC statement not been present. If you want to make all of the changes that are shown in the report, you can then remove the NOEXEC statement and rerun the procedure to update the metadata.

Example: Using the NOEXEC and REPORT Statements

The following example shows how to use the NOEXEC and REPORT statements to assess potential metadata changes.

```
ods html "myfile";
proc metalib;
omr (library=&mlibname repname= &mrepname );
update_rule=(delete);
noexec;
report;
run;
```

Note: The UPDATE_RULE statement tells the procedure to delete table definitions for any tables that have been deleted from the library. For more information about this statement, see "Changing the Update Rule" on page 215. △

Here is the resulting SAS log.

```
55   proc metalib;
56   omr (library=&mlibname  repname= &mrepname );
57   update_rule=(delete);
58   noexec;
59   report;
60   run;

NOTE: A total of 22 tables were analyzed for library "SAS91 lib".
NOTE: NOEXEC statement in effect.  No Metadata changes applied.
NOTE: Metadata for 4 tables would have been updated.
NOTE: Metadata for 2 tables would have been deleted.
NOTE: Metadata for 2 tables would have been added.
NOTE: Metadata for 13 tables matched the data sources.
NOTE: 0 other tables were not processed due to error or UPDATE_RULE.
NOTE: PROCEDURE METALIB used (Total process time):
      real time           31.26 seconds
```

```
cpu time            8.12 seconds
```

SAS output is the default. This example specifies ODS output. Specifying ODS produces reports in both ODS and SAS output formats unless you specify the following to suppress SAS output:

```
ods listing close;
```

Here is the resulting ODS output.

The SAS System

The METALIB Procedure

Summary Report of Potential Changes for Library sas91 lib
Repository Meta Proc repos
23FEB2006

Metadata Summary Statistics	
Total tables analyzed	22
Tables to be Updated	4
Tables to be Deleted	2
Tables to be Added	2
Tables matching data source	13
Other tables not processed	0

Tables to be Updated							
Table			Updates				
Metadata Name	Metadata ID	SAS Name	Metadata Name	Metadata ID	SAS Name	Metadata Type	Change
MYSTATE3	A5HJ58JU.AX0018LR	MYSTATE3	state	A5HJ58JU.AY001BMF	STATE	Column	Deleted
			continent			Column	Added
			prim_key3	A5HJ58JU.B3000U3K	prim_key3	Index	Column continent added
							Column deleted
					population	Index	Added
			POPULATION	A5HJ58JU.B3000U3J	POPULATION	Index	Deleted
			MYSTATE3.unq_key3	A5HJ58JU.B4000N5L	unq_key3	UniqueKey	Column population added
			MYSTATE3.Primary	A5HJ58JU.B400111E	prim_key3	UniqueKey	Column state added
							Column continent added
SASWINTER	A5HJ58JU.AX000EI9	SASWINTER	country	A5HJ58JU.AY000GQT	country	Column	IsNullable
USPOST3	A5HJ58JU.AX0018LS	USPOST3	name	A5HJ58JU.AY001BMI	NAME	Column	Deleted
			cont			Column	Added
			for_key3	A5HJ58JU.B3000U3L	for_key3	Index	Column name added
							Column deleted
WINTER	A5HJ58JU.AX0013ZA	WINTER	country	A5HJ58JU.AY00170T	country	Column	Desc
					total	Index	Added

Tables to be Deleted		
Metadata Name	**Metadata ID**	**SAS Name**
NONULL1	A5HJ58JU.AX001723	NONULL1
UPDTAB	A5HJ58JU.AX001AX3	UPDTAB

Tables to be Added		
Metadata Name	**Metadata ID**	**SAS Name**
		MYSTATES
		USPOSTAL

Updating Your Table Metadata to Match Data in Your Physical Tables

Adding and Updating Table Metadata

By default, PROC METALIB creates table definitions for any tables in the library that do not have table definitions and updates any table definition that does not reflect the current structure of the table that it represents. It does not, however, delete table metadata.

Use REPORT when you want an output listing that summarizes metadata changes, either before changes are made (by using NOEXEC) or to see afterward what changes were actually made. SAS output is the default.

Example: Default PROC METALIB Behavior

The following example uses the default PROC METALIB behavior. Summary notes are written to the SAS log regardless of whether you request a report. Unlike the example shown in “Assessing Potential Changes in Advance” on page 213, the summary does not mention any deleted tables.

```
proc metalib;
  omr  (library=&mlibname repname= &mrepname );
run;
```

Here is the resulting SAS log.

```
85   proc metalib;
86   omr (library=&mlibname repname= &mrepname );
87   run;

NOTE: A total of 1 tables were analyzed for library "v9SASlib".
NOTE: Metadata for 0 tables was updated.
NOTE: Metadata for 1 tables was added.
NOTE: Metadata for 0 tables matched the data sources.
NOTE: 0 other tables were not processed due to error or UPDATE_RULE.
NOTE: PROCEDURE METALIB used (Total process time):
      real time           19.06 seconds
      cpu time            5.39 seconds
```

Changing the Update Rule

By using the optional UPDATE_RULE statement, you can change PROC METALIB’s default behavior. The principal rules that you can specify are shown below:

NOADD specifies not to add table metadata to the metadata repository for physical tables that have no metadata.

NOUPDATE specifies not to update existing table metadata to resolve discrepancies with the corresponding physical tables.

DELETE specifies to delete table metadata if a corresponding physical table is not found in the specified library.

Examples: Adding, Updating, and Deleting Metadata

The following example shows how to use PROC METALIB to add metadata for new tables, update table definitions where necessary, and also delete table definitions that are no longer valid.

```
proc metalib;
omr (library=&mlibname repname= &mrepname );
update_rule=(delete);
report;
run;
```

The following example shows how to use UPDATE_RULE with DELETE, NOADD, and NO UPDATE to delete table definitions that are no longer valid, as well as suppress the default add and update actions.

```
proc metalib;
  omr  (library=&mlibname repname= &mrepname );
  update_rule (delete noadd noupdate);
  report;
run;
```

The resulting SAS output would resemble the following sample.

```
                              The METALIB Procedure

                         Summary Report for Library sas91 lib2
                              Repository Meta Proc repos
                                       17MAR2005

                               Metadata Summary Statistics

                          Total tables analyzed           2
                          Tables Updated                  0
                          Tables Added                    2
                          Tables matching data source     0
                          Tables not found                0


 ------------------------------------------------------------------------------
                                     Tables Added
 ------------------------------------------------------------------------------


 Metadata Name                     Metadata ID          SAS Name

 COUNTRY                           A5HJ58JU.AX001LPV    COUNTRY
 POSTAL                            A5HJ58JU.AX001LPW    POSTAL
```

Specifying Which Tables Are Affected

You can use the optional SELECT or EXCLUDE statements to perform an operation against a subset of the tables in a library. SELECT and EXCLUDE are mutually exclusive, so you should use only one or the other.

When you set the SELECT statement, you can select tables or table definitions for processing:

- For tables, specify their SAS name. If no table definition is found, it is created in the repository that contains the SASLibrary object. If a matching table definition is found, it is compared to the physical table. If differences are found, the table definition is updated.
- For table definitions, specify either the unique metadata identifier or the value in the SASTableName attribute. If you specify the metadata identifier, only the specified table definition is updated, not the first table definition in the association list.

You can use EXCLUDE to specify a single table or a list of tables to exclude from processing.

Examples: Specifying Tables

The following example shows how to use SELECT to process only a subset of tables.

```
ods html "myfile";
proc metalib;
omr (library=&mlibname repname= &mrepname );
select(spec_char_col ukeys ndx_multicol);
report;
run;
```

Here is the resulting ODS output.

The SAS System

The METALIB Procedure

Summary Report for Library SAS91 lib2
Repository Meta Proc repos
13FEB2006

Metadata Summary Statistics	
Total tables analyzed	3
Tables Updated	1
Tables Added	1
Tables matching data source	1
Tables not found	0
Other tables not processed	0

Tables Updated							
Table			Updates				
Metadata Name	Metadata ID	SAS Name	Metadata Name	Metadata ID	SAS Name	Metadata Type	Change
NDX_MULTICOL	A5HJ58JU.AX001H35	NDX_MULTICOL	multi_col	A5HJ58JU.B30011T5	multi_col	Index	Column silver added

Tables Added		
Metadata Name	Metadata ID	SAS Name
SPEC_CHAR_COL	A5HJ58JU.AX0026JV	SPEC_CHAR_COL

The following example shows how to use EXCLUDE to exclude a specific subset of tables.

```
proc metalib;
omr (library=&mlibname repname= &mrepname);
exclude(country postal mystate2);
noexec;
report;
run;
```

CHAPTER

12

Optimizing Data Storage

Overview of Optimizing Data Storage

For the purposes of querying, cube loading, and creating data marts and data warehouses, all four data storage structures (explained in Chapter 8, "Overview of Common Data Sources," on page 157) can be optimized to improve performance. Some optimization can be achieved, for example, by specifying transformation options in SAS Data Integration Studio. Some optimization requires hardware configuration, as in the case of SPD Engine tables. Cubes can be optimized for querying and loading during the cube loading process. For SAS tables, database tables, and SPD Engine tables, libraries can be defined in the metadata with options that enhance performance.

For more information, see these sections:

- "Compressing Data" on page 220
- "Indexing Data" on page 221
- "Sorting Data" on page 223
- "Buffering Data" on page 225
- "Using Threaded Reads" on page 226

Compressing Data

Compression is a process that reduces the number of bytes that are required to represent each table row. In a compressed file, each row is a variable-length record, while in an uncompressed file, each row is a fixed-length record. Compressed tables contain an internal index that maps each row number to a disk address so that the application can access data by row number. This internal index is transparent to the user. Compressed tables have the same access capabilities as uncompressed tables. Here are some advantages of compressing a file:

- reduced storage requirements for the file
- fewer I/O operations necessary to read from or write to the data during processing.

Here are some disadvantages of compressing a file:

- more CPU resources are required to read a compressed file because of the overhead of uncompressing each observation
- there are situations when the resulting file size might increase rather than decrease.

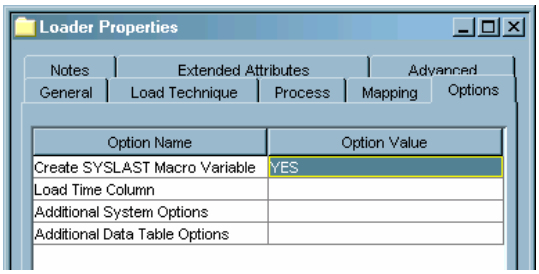
These are the types of compression that you can specify:

- CHAR to use the RLE (Run Length Encoding) compression algorithm, which works best for character data.
- BINARY to use the RDC (Ross Data Compression) algorithm, which is highly effective for compressing medium to large (several hundred bytes or larger) blocks of binary data. (The SPD Engine does not support binary compression.)

You can compress these types of tables:

- all tables that are created during a SAS session. Besides specifying SAS system options on the command line or inside a SAS program with the OPTIONS statement, you can use SAS Data Integration Studio to set system options. For example, you can use the **`Additional System Options`** field to set the COMPRESS= system option on a loader transformation. (A loader transformation generates or retrieves code that puts data into a specified target.)

Display 12.1 The Options Tab in a Loader Properties Dialog Box in SAS Data Integration Studio



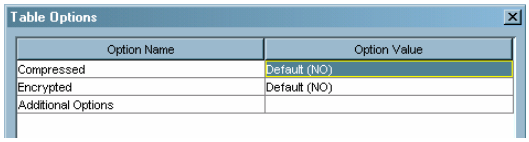
- all tables for a particular SAS data library. For example, when you define a Base SAS engine library in the metadata, you can specify the COMPRESS= option in

the **Other options to be appended** field on the **Options for any host** tab (see "Setting LIBNAME Options That Affect Performance of SAS Tables" on page 233). For third-party relational database tables, you can use the **Options to be appended** field on the **Other Options** tab (see "Setting LIBNAME Options That Affect Performance of SAS/ACCESS Databases" on page 234).

Note: You cannot specify compression for a SPD Engine data library. △

- an individual table. In SAS Data Integration Studio, SAS tables have a **Compressed** option that is available from the table properties dialog box. To use CHAR compression, you select **YES**. To use BINARY compression, you select **Binary**.

Display 12.2 The Table Options Dialog Box in SAS Data Integration Studio



For SPD Engine tables and third-party relational database tables, you can use the **Table Options** field in the table properties dialog box to specify the COMPRESS= option.

Note: The SPD Engine compresses the data component (.dpf) file by blocks as the engine is creating the file. (The data component file stores partitions for an SPD Engine table.) To specify the number of observations that you want to store in a compressed block, you use the IOBLOCKSIZE= table option in addition to the COMPRESS= table option. For example, in the **Table Options** field in the table properties dialog box, you might enter **COMPRESS=YES IOBLOCKSIZE=10000**. The default blocksize is 4096 (4k). △

When you create a compressed table, SAS records in the log the percentage of reduction that is obtained by compressing the file. SAS obtains the compression percentage by comparing the size of the compressed file with the size of an uncompressed file of the same page size and record count. After a file is compressed, the setting is a permanent attribute of the file, which means that to change the setting, you must re-create the file. To uncompress a file, you can, for example, in SAS Data Integration Studio, select **Default (NO)** for the **Compressed** option in the table properties dialog box for a SAS table.

For more information on compression, see *SAS Language Reference: Dictionary*.

Indexing Data

An index is an optional file that you can create to provide direct access to specific rows. The index stores values in ascending value order for a specific column or columns and includes information about the location of those values within rows in the table. In other words, an index enables you to locate a row by value. For example, if you use SAS to find a specific social security number (465-33-8613), SAS performs the search differently depending on whether there is an index on the row that contains the social security numbers.

- Without an index, SAS accesses rows sequentially in the order in which they are stored in the table. SAS reads each row, looking for SSN=465-33-8613 until the value is found or all observations are read.

- With an index on column SSN, SAS accesses the row directly. SAS satisfies the condition by using the index and going straight to the row that contains the value. SAS does not have to read each row.

When you create an index, you designate which columns to index. You can create two types of indexes:

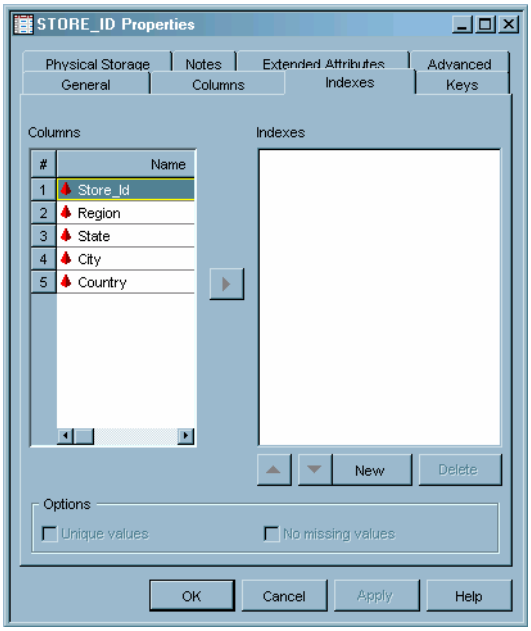
- a simple index, which consists of the values of one column.
- a composite index, which consists of the values of more than one column, with the values concatenated to form a single value.

For each indexed column, you also can perform these tasks:

- declare unique values. A unique index guarantees that values for one column or the combination of a composite group of columns remain unique for every row in the table. If an update tries to add a duplicate value to that column, then the update is rejected.
- keep missing values from using space in the index by specifying that missing values are not maintained by the index.

In addition to writing SAS code to create indexes, you can create indexes by using SAS Data Integration Studio. In SAS Data Integration Studio, you use the properties window for the table to index individual columns. When you create the index, you also can specify **`Unique values`** and **`No missing values`**.

Display 12.3 The Indexes Tab in the Properties Dialog Box for a Table Named STORE_ID



In general, SAS can use an index to improve performance in these situations:

- For cube loading, a composite index on the columns that make up the cube's hierarchies might provide best results.
- For WHERE processing, an index can provide faster and more efficient access to a subset of data. Note that to process a WHERE expression, SAS decides whether to use an index or to read the table sequentially.

  Note: For WHERE processing, the Base SAS engine uses a maximum of one index. The SPD Engine can use multiple indexes. △

Even though an index can reduce the time that is required to locate a set of rows, especially for a large table, there are costs that are associated with creating, storing, and maintaining the index. When deciding whether to create an index, you must consider increased resource usage, along with the performance improvement.

Once an index exists, SAS treats it as part of the table. That is, if you add or delete columns or modify values, the index is automatically updated.

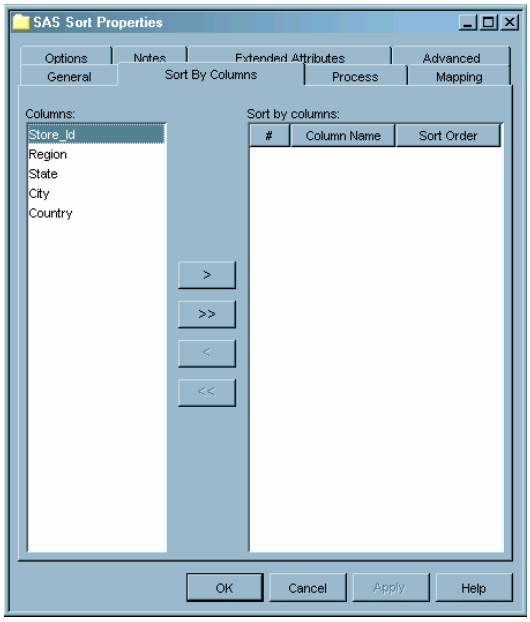
For more information about creating indexes, see *SAS Language Reference: Concepts*.

Sorting Data

You can sort table rows by the values of one or more character or numeric columns. For Base SAS tables and third-party relational database tables, the process either replaces the original table or creates a new table. You can perform sorting in two ways:

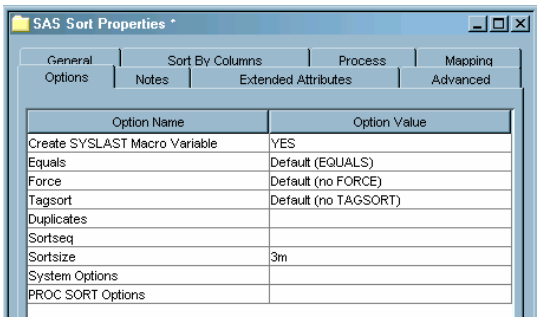
- using the SAS SORT procedure.
- setting properties for a SAS sort template in SAS Data Integration Studio.

Display 12.4 The Sort By Columns Tab in the SAS Sort Properties Dialog Box



To manage the memory that is used for the sorting process, you can specify the maximum amount of memory that is available to the sort. Generally, the sort size should be less than the physical memory available to the process. If the sorting requires more memory than you specify, then SAS creates a temporary utility file on disk. To specify a sort size in SAS Data Integration Studio, access the **`Options`** tab in the properties window for the sort template and enter a value in the **`Sortsize`** field.

Display 12.5 The Options Tab in the SAS Sort Properties Dialog Box



The SPD Engine has implicit sorting capabilities, which saves time and resources for SAS applications that process large tables. When the SPD Engine encounters a BY clause, if the data is not already sorted or indexed on the BY column, then the SPD Engine automatically sorts the data without affecting the permanent table or producing a new table. You can change the implicit sorting options when you define a SPD Engine library in the metadata. See "Setting LIBNAME Options That Affect Performance of SPD Engine Tables" on page 237.

For more information about the SORT procedure, see the *Base SAS Procedures Guide*.

Multi-Threaded Sorting

The SAS system option THREADS activates multi-threaded sorting, which achieves a degree of parallelism in the sorting operations. This parallelism is intended to reduce the real-time to completion for a given operation; however, the parallelism comes at the possible cost of additional CPU resources. For more information, see the section on "Support for Parallel Processing" in *SAS Language Reference: Concepts*.

The performance of the multi-threaded sort will be affected by the value of the SAS system option CPUCOUNT=. CPUCOUNT= indicates how many system CPUs are available for use by the multi-threaded sort. The multi-threaded sort supports concurrent input from the partitions of a partitioned table.

Note: For information about the support of partitioned tables in your operating environment, see the SAS documentation for your operating environment. △

For more information about THREADS and CPUCOUNT=, see the chapter on SAS system options in *SAS Language Reference: Dictionary*.

Sorting a Database Table

When you use a third-party database table, the column ordering that is produced by the SORT procedure depends on whether the DBMS or SAS performs the sorting. If you use the BEST value of the SAS system option SORTPGM=, then either the DBMS or SAS will perform the sort. If the DBMS performs the sort, then the configuration and characteristics of the DBMS sorting program will affect the resulting data order. Most database management systems do not guarantee sort stability, and the sort might be performed by the database table regardless of the state of the SORTEQUALS/NOSORTEQUALS system option and EQUALS/NOEQUALS procedure option.

If you set the SAS system option SORTPGM= to SAS, then unordered data is delivered from the DBMS to SAS and SAS performs the sorting. However, consistency in the delivery order of columns from a database table is not guaranteed. Therefore, even though SAS can perform a stable sort on the DBMS data, SAS cannot guarantee

that the ordering of columns within output BY groups will be the same, run after run. To achieve consistency in the ordering of columns within BY groups, first populate a SAS table with the database table, then use the EQUALS or SORTEQUALS option to perform a stable sort.

Buffering Data

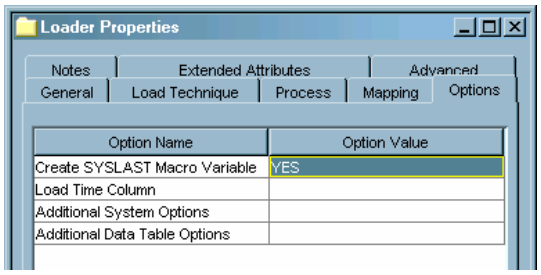
For Base SAS tables and some relational database tables, you can adjust page buffer settings to optimize CPU and I/O use. Different options are used for each type of table.

Base SAS Tables

For Base SAS tables, you might be able to make performance improvements by performing these tasks:

- tuning the size of table pages on disk by using the BUFSIZE= system option. SAS uses the BUFSIZE= option to set the permanent page size for the SAS table. The page size is the amount of data that can be transferred for an I/O operation to one buffer. If you know that the total amount of data is going to be small, you can set a small page size, so that the total table size remains small and you minimize the amount of wasted space on a page. Large tables that are accessed sequentially benefit from larger page sizes because sequential access reduces the number of system calls that are required to read the table.
- adjusting the number of open page buffers when the SAS table is processed. Increasing the value of the BUFNO= option can improve performance by enabling applications to read more data with fewer passes; however, your memory usage increases. You must determine the optimal value for your needs.

Besides specifying SAS system options on the command line or inside a SAS program with the OPTIONS statement, you can set the BUFSIZE= and BUFNO= system options in SAS Data Integration Studio. For example, you can set these **`Additional System Options`** in the properties window for a loader transformation.



For more information about the BUFSIZE= and BUFNO= options, see the *SAS Language Reference: Dictionary* and the documentation for your operating environment.

DB2 (UNIX and PC), ODBC, OLE DB, Oracle, SQL Server, and Sybase Tables

For DB2 (UNIX and PC), ODBC, OLE DB, Oracle, SQL Server, and Sybase, you can adjust page buffers by setting the INSERTBUFF= and READBUFF= options on the

library (see "Setting LIBNAME Options That Affect Performance of SAS/ACCESS Databases" on page 234) or on the individual table.

- The INSERTBUFF= option specifies the number of rows to insert. SAS allows the maximum that is supported by the DBMS. The optimal value for this option varies with factors such as network type and available memory. You might need to experiment with different values in order to determine the best value for your site.
- The READBUFF= option specifies the number of rows to hold in memory. SAS allows the maximum number that is supported by the DBMS. Buffering data reads can decrease network activities and increase performance. However, because SAS stores the rows in memory, higher values for READBUFF= use more memory. In addition, if too many rows are selected at once, then the rows that are returned to the SAS application might be out of date. For example, if someone else modifies the rows, you might not see the changes.

For more information about the INSERTBUFF= and READBUFF= options, see *SAS/ACCESS for Relational Databases: Reference*.

Note: In addition, the SASFILE statement enables you to store the entire Base SAS table in memory, and the table remains open until you close it because SASFILE caches the data and the open request. For more information about the SASFILE statement, see the *SAS Language Reference: Dictionary*. △

Using Threaded Reads

Most SAS/ACCESS interfaces support threaded reads. With a threaded read, the table read time can be reduced by retrieving the result set on multiple connections between SAS and a DBMS. To perform a threaded read, SAS performs these tasks:

1. Creates threads, which are standard operating system tasks that are controlled by SAS, within the SAS session.
2. Establishes a DBMS connection on each thread.
3. Causes the DBMS to partition the result set and reads one partition per thread. To cause the partitioning, SAS appends a WHERE clause to the SQL so that a single SQL statement becomes multiple SQL statements, one for each thread.

Threaded reads only increase performance when the DBMS result set is large. Performance is optimal when the partitions are similar in size. In most cases, threaded reads should reduce the elapsed time of the SAS job. However, threaded reads generally increase the workload on the DBMS. For instance, threaded reads for DB2 under z/OS involve a trade-off, generally reducing job elapsed time but increasing DB2 workload and CPU utilization.

Threaded reads are most effective on new, faster computer hardware running SAS, and with a powerful parallel edition of the DBMS. For example, if SAS runs on a fast uniprocessor or on a multiprocessor machine and your DBMS runs on a high-end SMP server, you will receive substantial performance gains.

For information about how to turn the threaded read function on or off for a DBMS library, see "Setting LIBNAME Options That Affect Performance of SAS/ACCESS Databases" on page 234.

For information about threaded reads, see *SAS/ACCESS for Relational Databases: Reference*.

Building Cubes from Star Schemas

A cube loads more efficiently when a star schema is used as the input data source. A star schema is a table in which a single fact table is connected to multiple dimension tables. This structure is visually represented in a star pattern.

The fact table is the central table in a star schema. It contains the individual facts that are being stored in the database as well as the keys that connect each particular fact to the appropriate value in each dimension. Each dimension table contains fields for each level of each hierarchy that is included in the dimension.

You can use SAS Data Integration Studio's Target Designer wizard to define star schemas. The Target Designer wizard enables you to select column metadata from various tables.

Note: Query performance is affected by the composition of the star schema. A cube that is built from a star schema that is composed of the same type of data (all SAS tables or all Oracle tables or all DB2 tables) provides better query performance than a cube that is built from a star schema that is composed of different types of data (a mixture of SAS tables, Oracle tables, and DB2 tables). △

For more information about star schemas, see the *SAS OLAP Server: Administrator's Guide*.

Validating SPD Engine Hardware Configuration

The SPD Engine automatically determines the optimal process to use to evaluate observations for qualifying criteria specified in a WHERE statement. WHERE statement efficiency depends on such factors as whether the columns in the expression are indexed. A SAS configuration validation program that measures I/O scalability with respect to WHERE processing can help you determine whether your system is properly configured for performing WHERE processing with the SPD Engine. The program performs these tasks:

1. It creates a table with two numeric columns.
2. It repeatedly reads the entire table, each time doubling the number of threads used until the maximum number is reached. The maximum number of threads is determined by the CPUCOUNT= SAS system option and is specified when SAS is started.

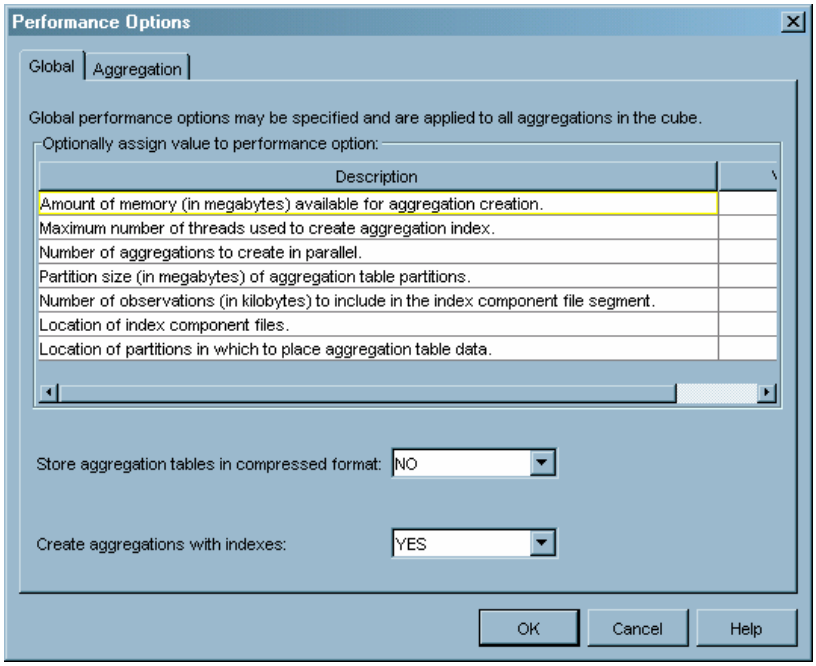
The resulting log file shows timing statistics for each cycle. You can examine this information to determine whether your system is configured correctly. The program is available at **`http://support.sas.com/rnd/scalability/spde/valid.html`**.

Building Optimized Cube Aggregations

There are global and aggregation-specific options that might improve cube loading and query performance. You set these options in the Performance Options dialog box, which is available from the Generated Aggregations window of the Cube Designer wizard. You launch the Cube Designer wizard from SAS Data Integration Studio or from SAS OLAP Cube Studio.

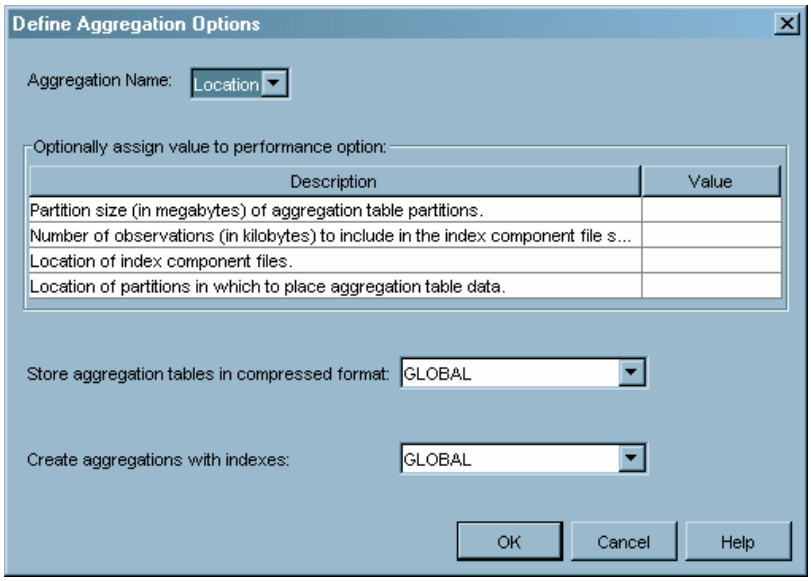
In the Performance Options dialog box, the **`Global`** tab enables you to set options that apply to all aggregations in the cube.

Display 12.6 The Global Tab in the Performance Options Dialog Box



From the **`Aggregation`** tab, you open the Define Aggregation Options dialog box, from which you can set aggregation-specific options that override any global settings.

Display 12.7 The Define Aggregation Options Dialog Box Available from the Aggregation Tab in the Performance Options Dialog Box



Here is a description of each option:

Amount of memory (in megabytes) available for aggregation creation
specifies the amount of memory in megabytes that will be available when aggregations are being created. The default is the system's available memory. This is available only as a global option.

Maximum number of threads used to create aggregation index
specifies the maximum number of threads that are used to asynchronously create the aggregation indexes. The processing engine calculates how many threads are needed based on the number of indexes being created and the amount of memory available for aggregation creation. This option sets a limit on the number of threads regardless of the number calculated by the processing engine. However, if the processing engine determines that fewer than the maximum number of threads is needed, then only the calculated number of threads are used. The default is the value of the SAS system option SPDEMAXTHREADS or 0. If the value is 0, then the processing engine determines the number of threads based on the number of indexes that are created plus the available memory. The maximum value is 65536 threads. This is available only as a global option.

Number of aggregations to create in parallel
specifies the number of aggregations to create in parallel. This option does not apply to the NWAY, which is always built first (unless you are creating the cube without an NWAY aggregation). The default is a maximum of 2, based on the results of a special algorithm that takes into consideration the number of aggregations being created and the number of processors available. The algorithm assumes that CPU resources should be saved for creating aggregation indexes.

Even if you have many CPUs, it is not recommended that you set this value above the default. This is because indexes on the tables are built concurrently and there is one index per hierarchy in each aggregation. So, if you are building two aggregations concurrently and each aggregation has 4 hierarchies, then you are building 8 indexes concurrently. Any increase above the default could dramatically decrease the memory assigned for each index build and, as a result, decrease index building performance.

This is available only as a global option.

Partition size (in megabytes) of aggregation table partitions
specifies the partition size in megabytes of the aggregation table partitions (the .dpf files) and their corresponding index components (the .idx and .hyb files). The default is 128 megabytes. The minimum is 16 megabytes. To return query results from an NWAY or aggregation, the SAS OLAP Server opens all partitions at the same time. Optimally, limit the partitions to 10 per data path.

Number of observations (in kilobytes) to include in the index component file segment
specifies the number of observations (table rows) in kilobytes to include in the index component file segment. The minimum size is 1 kilobyte (1024 rows), so the value of this option is a multiple of 1024 as expressed in kilobytes. The segmented indexes are used to optimize WHERE-expression processing. Each parallel thread is given a segment of the table to evaluate that is equal to the specified value.

Location of partitions in which to place aggregation table data

specifies the location of one or more partitions (.dpf files) in which to place aggregation table data. The data is distributed by cycling through each partition location according to the partition size. Separate multiple paths with a comma and enclose each path within quotation marks. For example, if you specify **'c:\data1','d:\data2'**, then the first partition of each aggregation table is placed into directory **c:\data1**, the second partition of each table is placed into directory **d:\data2**, the third partition of each table is placed into **c:\data1**, and so on. It is also possible to have aggregation tables that use less than the specified number of partitions. For example, your cube might contain an aggregation table that fits entirely into **c:\data1**. The default is the cube subdirectory of the path that you entered on the General window in the Cube Designer wizard.

As a best practice, use multiple paths. The optimal number of data paths is one per I/O controller. The maximum number of paths is 2 * the number of CPUs. Reserve disk drives exclusively for table storage. For best performance, the data area should be configured as a stripe-set of multiple disks (RAID 0). Mirroring is recommended.

Note: RAID (redundant array of independent disks) is a type of storage system that comprises many disks and which implements interleaved storage techniques that were developed at the University of California at Berkeley. RAIDs can have several levels. For example, RAID 0 combines two or more hard drives into one logical disk drive. Various RAID levels provide various levels of redundancy and storage capability. △

Location of index component files

specifies the locations of the index component files (the .idx and .hyb files) that correspond to each aggregation table partition. Indexes are not created for aggregations that have fewer than 1024 records. The default is the cube subdirectory of the path that you entered on the General window in the Cube Designer wizard. Separate multiple paths with a comma and enclose each path within quotation marks.

As a best practice, the index area should be configured as a stripe-set of multiple disks (RAID 0). Also, plan for redundancy, such as RAID 5. Disk space considerations include cardinality plus the number of indexed columns.

Store aggregation tables in compressed format

specifies whether to store the aggregation tables in a compressed format on disk. When you are setting global options, the default is no compression. When you are setting aggregation-specific options, the default is the GLOBAL setting. To store the aggregation tables in a compressed format, select **YES** from the pull-down list. However, since hierarchy members are stored in an internal numeric representation, little or no compression will take place. It is recommended that you accept the default of no compression.

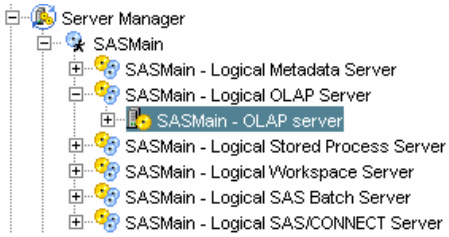
Create aggregations with indexes

specifies whether to create the aggregations with indexes. For faster cube creation and updates, you can select **NO** from the pull-down list; however, the lack of indexes will adversely affect query performance. When you are setting global options, the default is to create aggregations with indexes. When you are setting aggregation-specific options, the default is the GLOBAL setting.

Optimizing Performance of a SAS OLAP Server

During a planned installation, a SAS OLAP Server is defined and configured. The server definition includes options that affect query performance. To modify or view the default settings, you edit advanced options for the SAS OLAP Server definition in SAS Management Console. To access the Advanced Options dialog box, complete these steps:

1. In the SAS Management Console navigation tree, select **Environment Management ▶ Server Manager**.
2. Select a SAS application server that contains a SAS OLAP Server.
3. Select the name of the SAS OLAP Server, which appears beneath the logical server name.



4. Select **File ▶ Properties**.
5. In the OLAP Server Properties dialog box, select the **`Options`** tab.
6. Click **`Advanced Options`** to open the Advanced Options dialog box.

You change the advanced settings on the **`Cache`**, **`Server`**, and **`Performance`** tabs.

Setting Caching Options for the SAS OLAP Server

You set caching options on the **`Cache`** tab in the Advanced Options dialog box.

The **`Cube Cache`** is the maximum number of cube metadata registrations that you want the server to store in memory. The metadata contains information necessary to parse and plan a multidimensional expressions (MDX) query to the cube. The number of cubes that you cache is directly related to how fast your queries are processed. The default is 20 cached cube metadata registrations. For faster response times, increase the number of cubes cached. To save memory resources, decrease the number of cubes cached.

Note: Decreasing the number of cubes cached will results in slower response times. △

As new cubes are cached, older cubes are removed according to their usage.

The **`Data Cache`** is enabled by default. The data cache controls the number of cube aggregations that are stored in memory as the result of queries to the cube. Before processing any queries, the SAS OLAP Server first checks this cache to see if there is sufficient information to answer the current query. If there is, then the SAS OLAP Server fulfills the request by using the in-memory data rather than by accessing the cube.

The data cache is initially set at 16 megabytes. As a best practice, the cache should be set to use no more than 10 percent of your system's virtual memory. Also assume that queries will be running against more than one cube in one server session. Plan to provide space for multiple aggregations across multiple cubes.

For more information about how to determine the right size for your data cache, see the *SAS OLAP Server: Administrator's Guide*.

Setting Server Options for the SAS OLAP Server

A typical MDX query is executed as multiple sub-queries. Executing these sub-queries in parallel can improve performance. You use the **Maximum number of region execution threads** option to control the number of threads available to handle the sub-queries. As a best practice, do not set the number of threads to less than 2. To derive a reasonable maximum range for your system, multiply the number of processors on your system by 2. (This is how the default maximum setting is derived.)

The flattened cube options on the **Server** tab are used to manage system resources when a client application (such as the SQL Pass-Through Facility for OLAP) requests the cube data in a two-dimensional form. These options control the maximum number of flattened rows that can be processed in a request and the amount of memory that can be used to process the request. The defaults are 300,000 flattened rows and 268,435,456 bytes of memory. If the SAS OLAP Server will be processing a lot of two-dimensional queries, then you can adjust these numbers upward.

Note: For information about using the SQL Pass-Through Facility for OLAP, see the *SAS OLAP Server: Administrator's Guide*. △

The buffer size options are used to control the size of the buffer that is used to move information from the server to the client. The cellsets are the actual data values. The rowsets are metadata about the cube's members. The defaults cannot be changed in SAS 9.1.

Setting Performance Options for the SAS OLAP Server

The default (and minimum) amount of **Memory available for group by operations** is 256 megabytes. As a best practice, you should allot at least 64 megabytes for each thread spawned to process each MDX query.

The **Number of threads to spawn** is the number of threads that can be used for processing each MDX query. If the number of threads is set to 0 (the default), then an algorithm, which is based on the number of available CPUs, is calculated in order to produce a value from 1 through 8. As a best practice, if you expect a lot of concurrently running MDX queries, then set the value to less than 8.

The **Maximum number of tuples in a set** is used during the query analysis (when the SAS OLAP Server parses a query to check its validity). A tuple is a data object that contains two or more components. In OLAP, it is a selection of members (or cells) across dimensions in a cube. If the number of tuples that the query will generate exceeds this number, then the query is not processed. The default is 1 million tuples, which should accommodate most queries. You can reduce this number to block the processing of queries with large result sets that might exceed a client application's capabilities or overload your network.

When evaluating a WHERE expression for processing with indexes, the **Maximum Segment Ratio percentage value** controls whether or not to perform segment candidate pre-evaluation. It is not recommended that you change the default of 75.

Capturing SAS OLAP Server Performance Information

Server performance is recorded and analyzed using the Application Response Measurement (ARM) system. On the **Performance Logging** tab in the Advanced Options dialog box, you can specify a log file in which to save the information recorded by the ARM.

Note: For information about the ARM system, see "Monitoring Performance Using Application Response Measurement (ARM)" in *SAS Language Reference: Concepts*. △

Note: For instructions on how to access the Advanced Options dialog box, see "Optimizing Performance of a SAS OLAP Server" on page 231. △

The ARM options are

OLAP Session for each OLAP server, this option records how long each user was logged on. This ARM option is the default if you specify that you want to keep a log file.

MDX Query for each query, this option records the cube name and size of the result set (in cells).

Data Query for each data retrieval, this option records whether the data was retrieved from stored cube aggregations or from the data cache.

MDX String this option records the actual MDX query string.

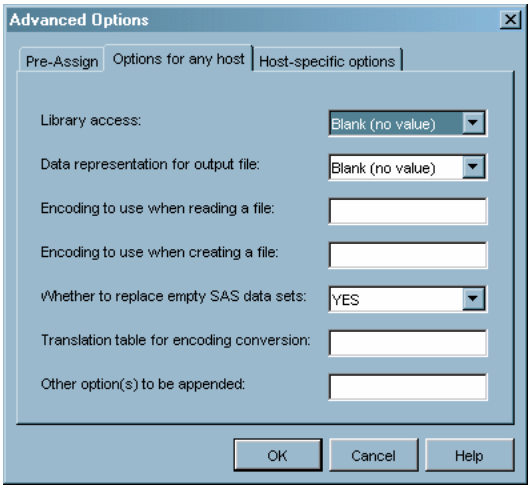
Setting LIBNAME Options That Affect Performance

When you use SAS Management Console to define a library, there are options available for the library definition that correspond to the LIBNAME options for the selected engine. Some of those options can be used to optimize use of the tables within the libraries.

Setting LIBNAME Options That Affect Performance of SAS Tables

You can set LIBNAME options that might affect performance of the Base SAS engine. You set these options when you use the New Library wizard to register a Base SAS engine library in the metadata repository. The LIBNAME options are available on the **Options for any host** tab and the **Host-specific options** tab in the Advanced Options dialog box. To access the Advanced Options dialog box, click the **Advanced Options** button on the Library Options window of the New Library wizard.

Display 12.8 The Options for Any Host Tab in the Advanced Options Dialog Box for a Base SAS Library



Here are some examples of options that might affect performance:

Data representation for the output file (OUTREP=)	For all operating environments, you can specify the data representation for the output file. Specifying this option enables you to create files within the native environment by using a foreign environment data representation. For example, an administrator who works in a z/OS operating environment might want to create a file on an HFS system so that the file can be processed in an HP UNIX environment. Specifying HP_UX_64 as the value for this option forces the data representation to match the data representation of the UNIX operating environment that will process the file. This method of creating the file can enhance system performance because the file does not require data conversion when being read by an HP UNIX machine.
Input/output block size (BLKSIZE=)	For Windows, UNIX, and z/OS environments, you can specify the number of bytes that are physically read during an I/O operation. The default is 8 kilobytes, and the maximum value is 1 megabyte.
Number of page caches to use for each open member (CACHENUM=)	For VMS, you can specify the number of page caches to use during I/O operations. The number of caches can potentially reduce the number of I/Os that are required to access the data. You also can set the size of each cache (CACHESIZE= option).

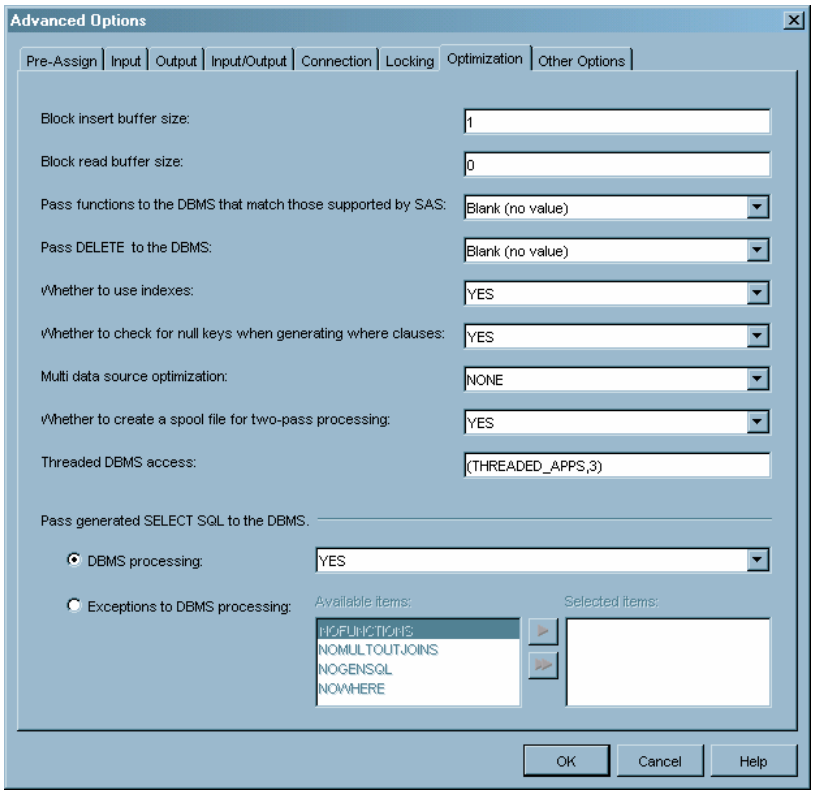
The **Other option(s) to be appended** field can be used to specify LIBNAME options such as COMPRESS= (see "Compressing Data" on page 220).

For information about each of the LIBNAME options in the Advanced Options dialog box, click the **Help** button.

Setting LIBNAME Options That Affect Performance of SAS/ACCESS Databases

The following LIBNAME options can be used to tune performance of the SAS/ACCESS engines. You can set these options when you use the New Library wizard to register the database libraries in the metadata repository. To access the Advanced Options dialog box, click the **Advanced Options** button on the Library Options window of the New Library wizard. For example, here are the **Optimization** tab default settings for DB2 libraries for UNIX and PC.

Display 12.9 The Optimization Tab in the Advanced Options Dialog Box for a DB2 Library for UNIX and PC



The tabs that are available in the Advanced Options dialog box, as well as the options on each of the tabs, vary between database management systems. Here are descriptions of the options on **`Optimization`** tab for DB2 libraries for UNIX and PC.

Block insert buffer size (INSERTBUFF=) specifies the number of rows in a single insert operation. See "Buffering Data" on page 225.

Block read buffer size (READBUFF=) specifies the number of rows of DBMS data to read into the buffer. See "Buffering Data" on page 225.

Pass functions to the DBMS that match those supported by SAS (SQL_FUNCTIONS=) when set to ALL, specifies that functions that match functions supported by SAS should be passed to the DBMS. The functions that are passed are: DATE, DATEPART, DATETIME, TIME, TIMEPART, TODAY, QRT, COMPRESS, SUBSTR, DAY, SECOND, INDEX, TRANWRD, HOUR, WEEKDAY, LENGTH, TRIMN, MINUTE, YEAR, REPEAT, MOD, MONTH, BYTE, and SOUNDEX. Use of this option can cause unexpected results, especially if used for NULL processing and date/time/timestamp handling. Exercise care when using this option.

Pass DELETE to the DBMS (DIRECT_EXE=) specifies that a SQL delete statement is passed directly to the DBMS for processing. Selecting this option improves performance because SAS does not have to read the entire result set and delete one row at a time.

Whether to use indexes (DBINDEX=) specifies whether SAS uses indexes that are defined on DBMS columns to process a join. Valid values are YES or NO. For more information about indexes, see "Indexing Data" on page 221.

Whether to check for null keys when generating where clauses (DBNULLKEYS=)
specifies whether the WHERE clause should detect NULL values in columns. Valid values are YES or NO. YES is the default for most interfaces and enables SAS to prepare the statement once and use it for any value (NULL or NOT NULL) in the column.

Multi data source optimization (MULTI_DATASRC_OPT=)
when processing a join between two tables, specifies whether an IN clause should be created to optimize the join. Valid values are NONE and IN_CLAUSE. IN_CLAUSE specifies that an IN clause containing the values read from a smaller table will be used to retrieve the matching values in a larger table based on a key column designated in an equi-join.

When processing a join between a SAS table and a DBMS table, the SAS table should be smaller than the DBMS table for optimal performance.

Whether to create a spool file for two-pass processing (SPOOL=)
specifies whether to create a utility spool file during transactions that read data more than once. In some cases, SAS processes data in more than one pass through the same set of rows. Spooling is the process of writing rows that have been retrieved during the first pass of a data read to a spool file. In the second pass, rows can be re-read without performing I/O to the DBMS a second time. In cases where the data needs to be read more than once, spooling improves performance. Spooling also guarantees that the data remains the same between passes. Valid values are YES or NO.

Threaded DBMS access (DBSLICEPARM=)
specifies the scope of DBMS threaded reads and the number of threads. If this option is set to the default, then PROC SQL will not use threading to read, for example, data for a Web report. To force a specified number of threads for a threaded read from the DBMS server, change the default to (ALL,*number-of-threads*).

Note: If PROC SQL attempts implicit pass-through, then threading will be disabled, regardless of the **Threaded DBMS access** setting. To disable implicit pass-through, set the **Pass generated SELECT SQL to the DBMS - DBMS** processing option to **NO**. △

For more information about threaded reads, see "Using Threaded Reads" on page 226.

Pass generated SELECT SQL to the DBMS - DBMS processing (DIRECT_SQL=) specifies whether generated SQL is passed to the DBMS for processing. Valid values are YES or NO.

Pass generated SELECT SQL to the DBMS - exceptions to DBMS processing (DIRECT_SQL=) if the value for the previous option is YES, then this option specifies *how* generated SQL is passed to the DBMS for processing. For example, NOWHERE prevents WHERE clauses from being passed to the DBMS for processing.

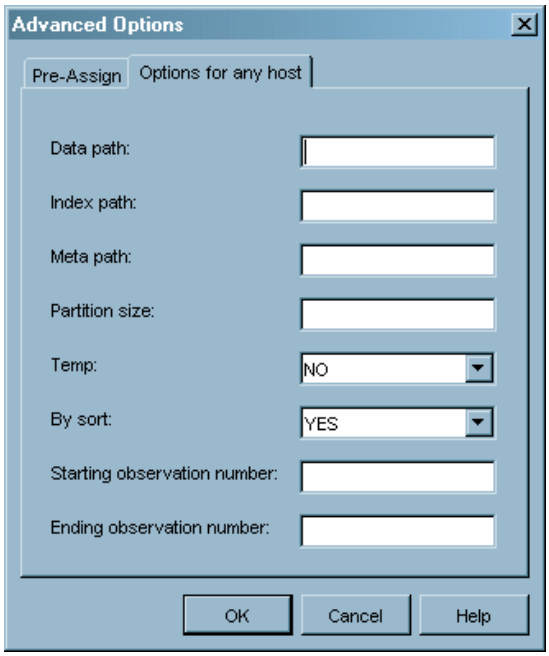
The **`Other Options`** tab, which is available for all database management systems, can be used to specify LIBNAME options such as COMPRESS= (see "Compressing Data" on page 220).

For information about each of the LIBNAME options in the Advanced Options dialog box, click the **`Help`** button. For information about all SAS/ACCESS LIBNAME options, see *SAS/ACCESS for Relational Databases: Reference*.

Setting LIBNAME Options That Affect Performance of SPD Engine Tables

The following LIBNAME options can be used to tune performance of the SPD Engine. You can set these options when you use the New Library wizard to register a SPD Engine library in the metadata repository. The LIBNAME options are available on the **`Options for any host`** tab in the Advanced Options dialog box. To access the Advanced Options dialog box, click the **`Advanced Options`** button on the Library Options window of the New Library wizard.

Display 12.10 The Options for Any Host Tab in the Advanced Options Dialog Box for a SPD Engine Library



Data path (DATAPATH=)
specifies a list of paths in which to store partitions (.dpf files) for an SPD Engine table. The engine creates as many partitions as are needed to store all the data. The size of the partitions is set using the PARTSIZE= option. Partitions are created in the specified paths in a cyclic fashion. The data path area is best configured as multiple paths. Allot one I/O controller per data path to provide high I/O throughput, which is the rate at which requests for work are serviced by a computer system. The data path area is best configured for redundancy (RAID 1).

Index path (INDEXPATH=)
specifies a path or a list of paths in which to store the two index component files (.hbx and .idx) that are associated with an SPD Engine table. Additional specified paths accept the overflow from the immediately preceding path. The index path area is best configured as multiple paths. Use a volume manager file system that is striped across multiple disks (RAID 0) to enable adequate index performance, both when evaluating WHERE clauses and creating indexes in parallel. Redundancy (RAID 5 or RAID 10) is also recommended.

Meta path (METAPATH=)
specifies a list of overflow paths in which to store metadata component (.mdf) files for an SPD Engine table. The metadata component file for each table must begin in the primary path. When that primary path is full, the overflow is sent to the specified METAPATH= location. The metadata path area is best configured for redundancy (RAID 1) so that metadata about the data and its indexes is not lost.

Partition size (PARTSIZE=)
specifies the size (in megabytes) of the data component partitions when an SPD Engine table is created. By splitting the data portion of an SPD Engine table at fixed-size intervals, you may gain a high degree of scalability for some operations. For example, the SPD Engine can spawn threads in parallel, up to one thread per partition for WHERE evaluations.

Temp (TEMP=)
specifies whether to create a temporary subdirectory of the directory specified in the Path field on the Library Properties wizard window. The directory is used to temporarily store the metadata component files associated with table creation. It is deleted at the end of the SAS session.

By sort (BYSORT=)
specifies that the SPD Engine should perform an automatic implicit sort when it finds a BY statement for processing data in the library (unless the data is indexed on the BY column). Valid values are YES (perform the sort) and NO (do not perform the sort). The default is YES.

Starting observation number (STARTOBS=)
specifies the number of the starting observation in a user-defined range of observations that are qualified with a WHERE expression. By default the SPD Engine processes all observations in the table.

Ending observation number (ENDOBS=)
specifies the number of the ending observation in a user-defined range of observations that are qualified with a WHERE expression. By default the SPD Engine processes all observations in the table.

In addition to the LIBNAME options, there are also table and system options that can be used to tune SPD Engine performance. For example, the SPDEUTILLOC=

system option allots space for temporary files that are generated during SPD Engine operations. This area is best configured as multiple paths. Use a volume manager file system that is striped across multiple disks (RAID 0) to reduce out-of-space conditions and improve performance. Redundancy (RAID 5 or RAID 10) is also recommended since the loss of the work area could stop the SPD Engine from functioning.

The *SAS Scalable Performance Data Engine: Reference* includes a "Quick Guide to the SPD Engine Disk-I/O Set-Up" that helps you

- determine the amount of space that needs to be allocated to the data, metadata, index, and work areas
- evaluate the advantages and disadvantages of different RAID groups for each of the different types of areas.

For information about table and other system options for the SPD Engine, see **`http://support.sas.com/rnd/scalability/spde/syntax.html`**. For information about each of the LIBNAME options in the Advanced Options dialog box, click the **`Help`** button.

PART 4

Application Administration

CHAPTER

13

Administering SAS Data Integration Studio

Overview of Administering SAS Data Integration Studio

SAS Data Integration Studio is a visual design tool that enables data integration specialists to consolidate and manage enterprise data from a variety of source systems, applications, and technologies. This software enables these users to create process flows that accomplish the following tasks:

- extract, transform, and load (ETL) data for use in data warehouses and data marts
- cleanse, migrate, synchronize, replicate, and promote data for applications and business services.

SAS Data Integration Studio enables users to integrate information from any platform that is accessible to SAS and from any format that is accessible to SAS.

Note: SAS Data Integration Studio was formerly named SAS ETL Studio. △

For information on how to build warehouses and marts, data integration specialists should consult the *SAS Data Integration Studio: User's Guide*, which is available at **`support.sas.com`**. To access the book, go to **`http://support.sas.com`**. Then select **Documentation ▶ Products & Solutions**. In the **`Select a Product`** drop-down list on the Documentation for SAS Products and Solutions page, select **`SAS Data Integration Studio`** and click **`Go`**.

There are also a number of administrative tasks that you must perform in support of these users. For example, in addition to installing the product, you might need to perform the following tasks:

Table 13.1 Administrative Tasks for SAS Data Integration Studio

Administrative Task	Purpose of Task
Connecting to SAS Servers	Make sure that your data integration specialists can connect to the necessary SAS servers. For instance, each user must be able to connect to the metadata server to register data sources and other objects.
Connecting to Data Servers	Make sure that your data integration specialists can connect to the necessary data servers.
Setting Up Change Management	Set up a change-management system, which enables individual users to check objects out of a foundation repository and place them in a private repository (called a *project repository*) where the users can test changes.
Setting Up Multi-Tier Environments	Set up SAS Data Integration Studio to work in a multi-tier configuration. Almost all configurations have multiple tiers.
Using Custom-Tree Folders for Security	Set up a folder structure for metadata objects that enables you to control access to those objects.
Importing and Exporting Generated Transformations	Control access to SAS Data Integration Studio generated transformations, which are a class of user-written transformation.
Importing and Exporting Metadata	Import metadata from data modeling tools such as the AllFusion ERwin Data Modeler.
Administering Message Queues	Install third-party message queue software, define queues, and register queues for interprocess communication.
Providing Support for Web Services	Install SAS BI Web Services for Java or .NET so that Web services can invoke store processes.
Enabling the Bulk Loading of Data	Enable workspace servers to execute external commands so that jobs can rapidly load data into a DBMS.
Testing the Platform Computing Scheduling Server	Test the servers (and clients) that enable your users to schedule sets of SAS Data Integration Studio jobs.
Setting Up a SAS Data Quality Server	Set up the infrastructure necessary for your users to employ data-quality transformations.
Redirecting Output and Logging Information to a File	Specify alternative destinations for the SAS log and SAS output.
Enabling Status Code Handling	Enable status code handling.
Enabling the External File Wizards to Retrieve Files Using FTP or HTTP	Enable the external file wizards to retrieve files using FTP or HTTP.

All of these topics are covered in the remaining sections of this chapter.

Connecting to SAS Servers

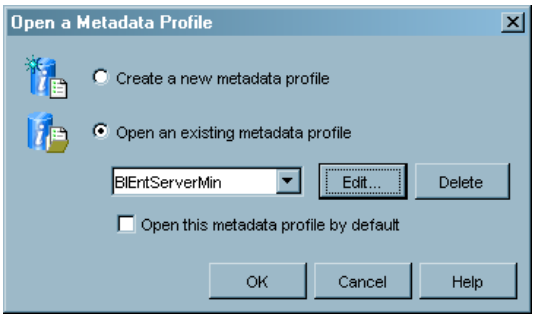
In order to use SAS Data Integration Studio, data integration specialists must be able to connect to your system's SAS Metadata Server and to one or more workspace servers. Access to the metadata server is necessary because the objects with which users interact directly—libraries, tables, and jobs—are represented by metadata objects. Without access to the metadata server, these users cannot do any work. Similarly, users need access to workspace servers for many ordinary tasks, such as registering a data source in a metadata repository and running jobs that extract, transform, and load data.

Connecting to a Metadata Server

The easiest way to ensure that data integration specialists can connect to a running metadata server is to use SAS Data Integration Studio to open a *metadata profile*, which contains information about the metadata server, a metadata repository, and a user. When SAS Data Integration Studio opens this profile, it attempts to connect to the metadata server. If it connects successfully, SAS Data Integration Studio will complete its initialization, and the object trees in the interface—such as the Inventory tree—will be populated with selected objects from a metadata repository.

There are two cases that might confront you. If you ran the SAS Configuration Wizard (during installation) on the machine from which you will perform the test, a metadata profile will already exist, and you can use it. Otherwise, you will have to create a metadata profile and then open it.

In either case, you need to go to a machine where SAS Data Integration Studio has been installed and start the application. You do this by selecting **Start ▶ Programs ▶ SAS ▶ SAS Data Integration Studio 9.1**. As the application starts, you will see the Open a Metadata Profile dialog box.



If a metadata profile has been created on this machine, you will see its name listed in the **`Open an existing metadata profile`** list box. The profile will have the same name as your configuration directory and will have been defined to connect to the metadata server using the SAS Administrator account. In this case, click **`OK`** in the dialog box. You will be prompted for a password. Enter the password for the SAS Administrator account, and click **`OK`**. If SAS Data Integration Studio is able to connect to the metadata server, it will read metadata from a repository and display a set of metadata objects in its tree views.

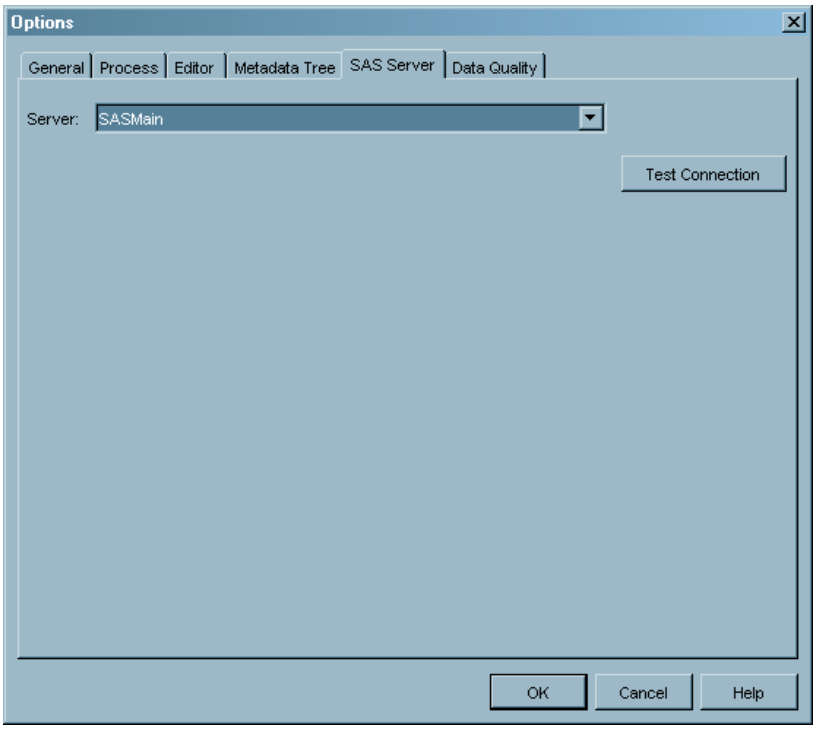
If no metadata profile has been created, you will have to define one and use it to connect to the metadata server. In this case, the **`Create a new metadata profile`** radio button will be selected when the Open a Metadata Profile dialog box appears. Click **`OK`** to start the Metadata Profile Wizard. Use the wizard and the on-line help, if necessary, to create a profile. (Do not save your password in the profile.) When you

have finished defining the profile, SAS Data Integration Studio will automatically try to use the profile to connect to the metadata server.

Connecting to a Workspace Server

When you execute a job in SAS Data Integration Studio, the application submits generated SAS code to a workspace server, which executes the code. Therefore, it is imperative that an object spawner be up and running and that SAS Data Integration Studio be able to use the spawner to start a workspace server. To test a connection to a workspace server, follow these steps:

1. Select **Tools ▸ Options**. An Options dialog box appears.
2. In the Options dialog box, select the **`SAS Server`** tab. The Options dialog box should now look like this:



3. Select from the list the name of the application server to which the workspace server belongs.
4. Click **`Test Connection`**. You might be prompted for a user name and password. If you are, enter credentials that will allow you to be authenticated on the host where the workspace server is running. If all goes well, you will see an Information dialog box that says, “Connection to the server was successful.”

Connecting to Data Servers

After you have established that your users can connect to the metadata server and the system’s workspace servers (as described in “Connecting to SAS Servers” on page 246), it is a good idea to make sure that users can get to the data sources that will provide the input to SAS Data Integration Studio jobs. Here are some examples of these sources:

- DB2 tables
- Sybase tables
- Teradata tables
- ODBC data sources
- Oracle tables
- SAS data sets
- SAS Scalable Performance Data Engine tables

The general procedure for performing this test is to follow these steps:

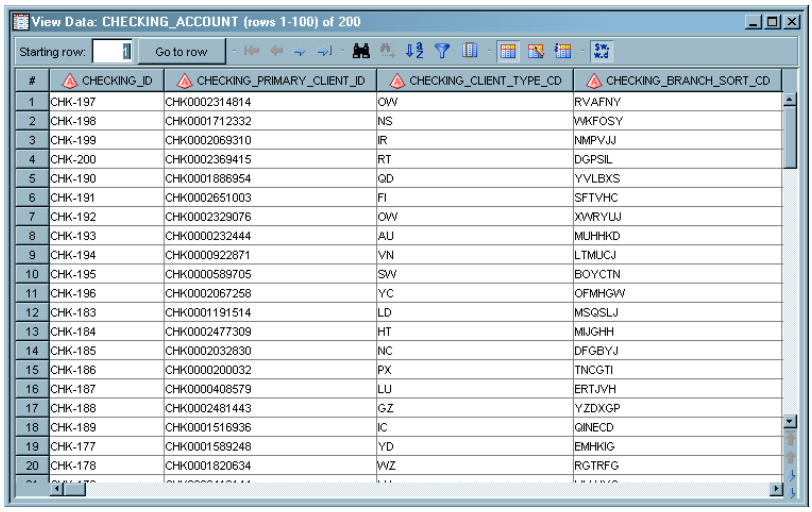
- Register your data sources in your foundation metadata repository as explained in Chapter 9, "Connecting to Common Data Sources," on page 167.
- Use SAS Data Integration Studio's View Data feature to make sure that you can read data from your different data sources.

After you have registered your SAS tables and DBMS tables in the metadata, these tables will appear in the Tables folder of SAS Data Integration Studio's Inventory tree.

To determine whether your data integration developers will be able to read data from a particular data server, perform these steps:

1. In the Inventory tree, select a table that is managed by that server.
2. Select **View ▸ View Data**.

A View Data window appears and shows you the data in the table.



#	CHECKING_ID	CHECKING_PRIMARY_CLIENT_ID	CHECKING_CLIENT_TYPE_CD	CHECKING_BRANCH_SORT_CD
1	CHK-197	CHK0002314814	OW	RVAFNY
2	CHK-198	CHK0001712332	NS	WKFOSY
3	CHK-199	CHK0002069310	IR	NMPVJJ
4	CHK-200	CHK0002369415	RT	DGPSIL
5	CHK-190	CHK0001886954	QD	YVLBXS
6	CHK-191	CHK0002651003	FI	SFTVHC
7	CHK-192	CHK0002329076	OW	XWRYUJ
8	CHK-193	CHK0000232444	AU	MUHHKD
9	CHK-194	CHK0000922871	VN	LTMUCJ
10	CHK-195	CHK0000589705	SW	BOYCTN
11	CHK-196	CHK0002067258	YC	OFMHGW
12	CHK-183	CHK0001191514	LD	MSQSLJ
13	CHK-184	CHK0002477309	HT	MIJGHH
14	CHK-185	CHK0002032830	NC	DFGBYJ
15	CHK-186	CHK0000200032	PX	TNCGTI
16	CHK-187	CHK0000408579	LU	ERTJVH
17	CHK-188	CHK0002481443	GZ	YZDXGP
18	CHK-189	CHK0001516936	IC	QINECD
19	CHK-177	CHK0001589248	YD	EMHKIG
20	CHK-178	CHK0001820634	WZ	RGTRFG

Note: You can quickly determine the type of a table in the Tables folder by bringing up the table's Properties dialog box and selecting the **`Physical Storage`** tab. The **`DBMS`** list box will contain a value such as SAS, Oracle, or Sybase. △

Setting Up Change Management

SAS Data Integration Studio contains a change management feature that enables data integration developers to check metadata objects out of a foundation repository into a work repository, called a project repository. There, a developer can modify the checked-out objects (and create new objects). Meanwhile, the corresponding objects in the foundation repository are locked. When the developer finishes his or her work and

checks in any changes, these locks are released. The best practice in this area is for each data integration specialist to have his or her own project repository. This arrangement prevents developers from making changes to the same metadata objects at the same time.

The following list *summarizes* the administrative tasks you must perform to set up change management in the case where there is one project repository per developer. (The sections that follow the list provide details where necessary about how to perform these tasks.)

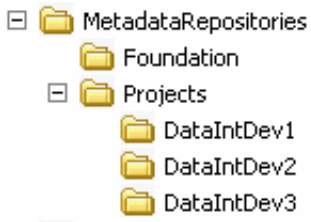
1. Set up an operating system user account for each data integration specialist. On Windows systems, you must give the user—or a group to which the user belongs—the user right "Log on as a batch job." You should already have performed this task.
2. Use the User Manager plug-in to SAS Management Console to create a metadata object that represents the user in the foundation repository. You should already have performed this task as well.
3. Create a new directory in the file system that will hold the contents of the new project repository. Make sure that only the user who invokes the metadata server has write access to this directory. For details on how to perform this step, see "Creating a Repository Directory" on page 249.
4. Use SAS Management Console to create the new project repository. For details on how to perform this step, see "Creating a Project Repository" on page 250.
5. In SAS Management Console, set the permissions on the foundation and project repositories so that the data integration specialist cannot write data directly to the foundation repository, but can write to his or her project repository. For details on how to perform this step, see "Setting Metadata Permissions for Your User" on page 251.
6. Have the data integration specialist create a metadata profile that he or she can use to connect to the metadata server and to specify which project repository he or she will be working in. For details on how to perform this step, see "Creating a Metadata Profile" on page 252.
7. Have the data integration specialist use the metadata profile to make sure that change management is working. For details on how to perform this step, see "Using the Metadata Profile" on page 253.

As previously mentioned, you should already have performed steps 1 and 2. If you have not performed those steps, perform steps 1 and 2 now. The other steps are explained in the sections that follow.

Creating a Repository Directory

On the machine where you installed your metadata server, go to the directory *SAS-config-dir*`\Lev1\SASMain\MetadataServer\MetadataRepositories`. There, you will see a directory for your foundation repository. Add a new directory called **`Projects`**. Then change directories to the **`Projects`** directory, and create a directory named for the data integration specialist. This directory will contain the user's project repository. In addition, set the ownership and permissions on this new directory so that only the user who starts the metadata server has read and write access to the directory. On UNIX systems, the owner should be **`sas`**, and the permissions should be set to **`700`**. On Windows systems where the SAS servers are running as services, **`SYSTEM`** should have full control of the directory. (On Windows systems where the servers are started using scripts, the user who runs the script **`startMetadataServer.bat`** should have full control of the directory.)

When you have created a project repository directory for each data integration developer, your directory structure should look something like this:



Creating a Project Repository

Before you begin this step, make sure that you have created a metadata object for the user and the directory that will hold the contents of the new repository. The wizard that you use to create the repository prompts you for information about these items.

Perform these steps to create the repository in SAS Management Console:

1. In the left pane, expand the **Metadata Manager** portion of the tree, and select the **Active Server**. Then, select **Actions ▶ Add Repository**. A wizard that guides you through the process of creating a metadata repository starts.
2. In the Select Repository Type window, select the **Project** radio button. Then click **Next**.
3. In the General Information window, enter a name for the new repository in the **Name** text box. Entering a description of the repository in the **Description** text box is optional. Click **Next**.
4. In the Definition of Data Source window, you are prompted for three pieces of information: an Engine, a Path, and Options.
   a. Engine—Generally, you should accept the default value, **Base**. This setting is appropriate whenever you are storing your metadata in SAS data sets (the default). If you are using a third party DBMS for repository storage, select **DB2** or **Oracle**, as appropriate.
   b. Path—Enter a full path to the directory that will hold the contents of the repository, or use the available **Browse** button to specify this directory.
   c. Options—Do not enter any options.

   Click **Next**.
5. In the Define Repository Dependencies window, specify that your project repository will depend on the foundation repository. This means that the user will check metadata objects out of the foundation repository into the user's project directory to work on the metadata objects. The user will then check these objects (and any new objects) back into the foundation repository.

   To specify this relationship between repositories, select **Foundation** from the list of repositories on the left, and then click the right arrow. The foundation repository icon will move to the **Repository will depend on** list on the right. Then click **Next**.
6. In the Choose Repository Owner window, select the user for whom you are creating the project repository, and then click **Next**.
7. Finally, in the Current Settings window, review the data you have entered; then click **Finish**.

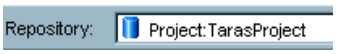
In SAS Management Console, you will see an icon that represents the new repository displayed in the Metadata Manager section of the tree in the left pane.

Setting Metadata Permissions for Your User

After you have created a project repository for a user, you need to use SAS Management Console to set (in metadata) the permissions for that user to access the foundation repository and the permissions for that user to access his or her project repository. The user needs the ReadMetadata and CheckInMetadata permissions for the foundation repository, and ReadMetadata and WriteMetadata permissions for the project repository. The following steps explain how to set up permissions for the owner of the project repository to access that repository. (You set the user's permissions to access the foundation repository in a similar way.)

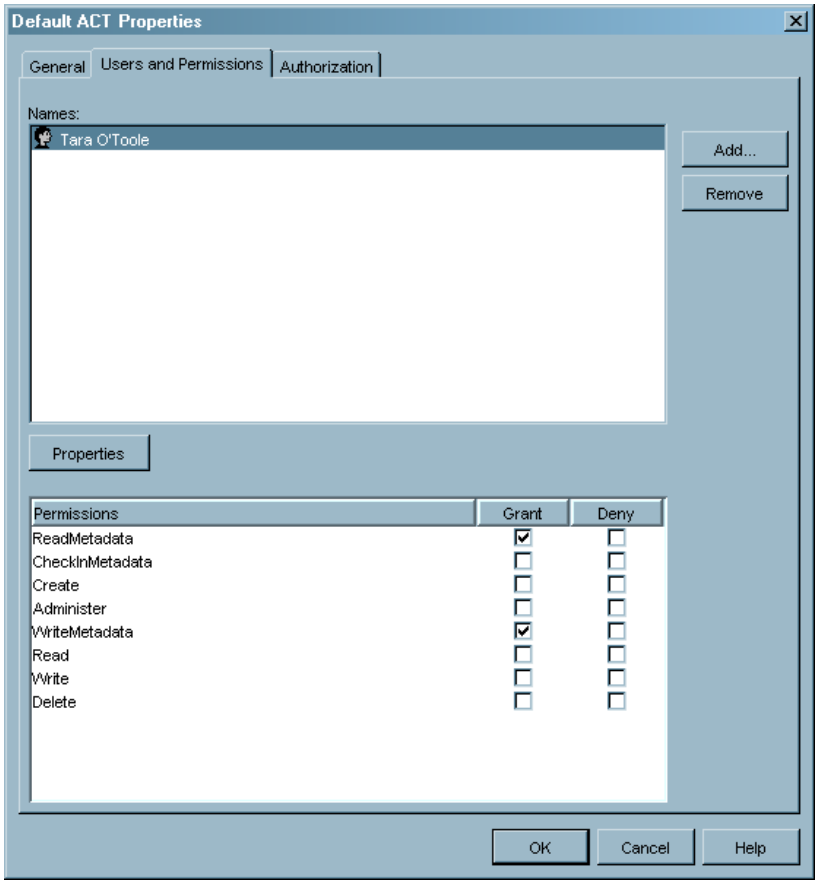
Note: When you perform this step, you must be logged on to SAS Management Console as an unrestricted user. For example, you could be logged on as the SAS Administrator (sasadm). To make another user an unrestricted user, you must add that user's ID to the file **`adminUsers.txt`** and prepend an asterisk to that ID. △

1. Select the user's project repository from the **`Repository`** list box, as shown in the following display.

   (This list box is located just below the button bar near the top of the window.)
2. In the left pane, expand the **`Authorization Manager`** section of the tree and, within that section, the **`Access Control Templates`** section. You will see an icon named Default ACT (Access Control Template). Select this icon and select **File ▶ Properties**. The Default ACT Properties dialog box appears.
3. In this dialog box, click the **`Users and Permissions`** tab. In the top half of the dialog box, you see a list of users and groups that have permissions defined for this repository. In the lower half of the dialog box, you see the permissions for the currently selected user or group.
4. On the **`User and Permissions`** tab, remove all of the existing user and group names. Then add the name of the owner of the project repository.
5. Set the user's permissions as follows:
   - Grant ReadMetadata
   - Grant WriteMetadata.

   As this point, your Default ACT Properties dialog box should look something like this:



Note: When you are setting the user's permissions to access the foundation repository, just add the new user and set his or her permissions. Do not remove the existing users and groups. △

6 Click **OK**. Now only Tara will be able to work in her project directory.

Creating a Metadata Profile

A metadata profile enables a SAS Data Integration Studio user to connect to a metadata server and to specify a default metadata repository (typically the user's project repository). You could set up metadata repositories for all of your data integration specialists, but we recommend that you give users the information necessary to create a profile and let the users create the profiles.

You will need to give each user the following information:

- the full name of the machine on which the metadata server is running—for example, server1.na.sas.com
- the port on which the metadata server is listening (8561 by default)
- the name of the user's project repository

If you also want to provide instructions about how to create a metadata profile, here they are:

1 Start SAS Data Integration Studio. A dialog box named Open a Metadata Profile appears.

2 In the Open a Metadata Profile dialog box, select the **`Create a new metadata profile`** radio button; then click **OK**. The Metadata Profile Wizard starts.

3 In the Metadata Profile Wizard window, click **Next**. The only purpose of this window is to explain what the wizard does.

4 In the Metadata Profile Wizard window, enter a name for your metadata profile in the **Name** text box. You also have the option of selecting the **Open this metadata profile by default** check box. If you will always be working in the same project repository, you should select the check box so that you will not be prompted to select a metadata profile each time that you start SAS Data Integration Studio. If you are working on multiple projects, do not select the check box, so that you can select the profile that you need each time you start the application. After you have supplied this data, click **Next**.

5 In the Connection Information window, fill in the following text boxes:

- Machine—Enter the full name of the machine on which the metadata server is running. (This information is supplied by the administrator.)
- Port—Enter the number of the port on which the metadata server is listening. (This information is supplied by the administrator.)
- Username—Enter your user name. On Windows systems, this name should be of the form *domain\user-name* or *hostname\user-name*.
- Password—Enter your password.

We recommend that you do not select the **Save username and password in this profile** check box. If you do, any user can connect to the metadata server by simply starting SAS Data Integration Studio on your workstation.

Click **Next**.

6 In the Repository Selection window, select your project repository, and click **Next**. (Your administrator will give you the name of this repository.) If no project repository is displayed, see below.

7 In the Finish window, click **Finish**.

Using the Metadata Profile

When the user finishes running the Metadata Profile wizard, SAS Data Integration Studio will automatically connect to the SAS Metadata Server and read the appropriate metadata objects. The user will know that he or she is set up correctly for change management if, when the SAS Data Integration Studio interface appears, there is a **Project** tab at the bottom of the tree pane. If the user clicks this tab, he or she will see an icon representing his or her project repository.

Note: If this **Project** tab does not appear, see "Responding When No Project Repositories Are Displayed" on page 253. △

Responding When No Project Repositories Are Displayed

The most likely reason that a **Project** tab or project repository is not displayed is that the login (user name and password) that is specified in the user's metadata identity is different from the login that is used to connect to the metadata server. You might need to ask your administrator what the correct login is. Other troubleshooting steps you can take include the following:

- Verify that the instructions in "Setting Up Change Management" on page 248 were followed.
- On a Windows platform, verify that the Windows network domain is specified for the login that is part of the user's metadata identity and for the login that is used to connect to the metadata server.

- Verify that operating system security enables the user to read the directory where the Project repository is located.

Note: For information about how to work in a change-managed environment, see the *SAS Data Integration Studio: User's Guide*. △

Setting the Default umask on UNIX Systems

When SAS Data Integration Studio writes a target data set, that data set is owned by the user who ran the job that created the data set, and the permissions on the data set file are set like this:

```
rwxr-xr-x
```

These permissions are determined by the workspace server's default umask, 022.

This setup can lead to problems if you want multiple data integration specialists to be able to update the same data set. If Developer 1 runs a job that creates a target data set and then goes on vacation, Developer 2 will not be able to run the same job. The job will fail with an error indicating that Developer 2 does not have sufficient authorization to update the target file because the developer does not have write access to the file. To circumvent this problem, you can put your data integration specialists in a group and have the workspace server use a umask that sets file permissions so that any member of the group can read and write a file that is created by a member of the group. A umask setting of 002 will enable your developers to share files.

To change the workspace server's default umask, perform the following steps:

1. Create an operating system group, and make this group the primary group for your data integration specialists.
2. Edit the file *SAS-config-dir*/**Lev1/SASMain/sas.sh** so that the script performs the following actions:
   a. Obtains the user's primary group number.
   b. If the user's primary group is the data-integration-specialist group, sets the umask to 002.

The code below shows the lines that you might add to the **sas.sh** file on an AIX system:

```
CMD=/usr/bin/id
CURR_GID=eval $CMD -g
GID=201
if [$CURR_GID -eq $GID]; then umask 002 fi
```

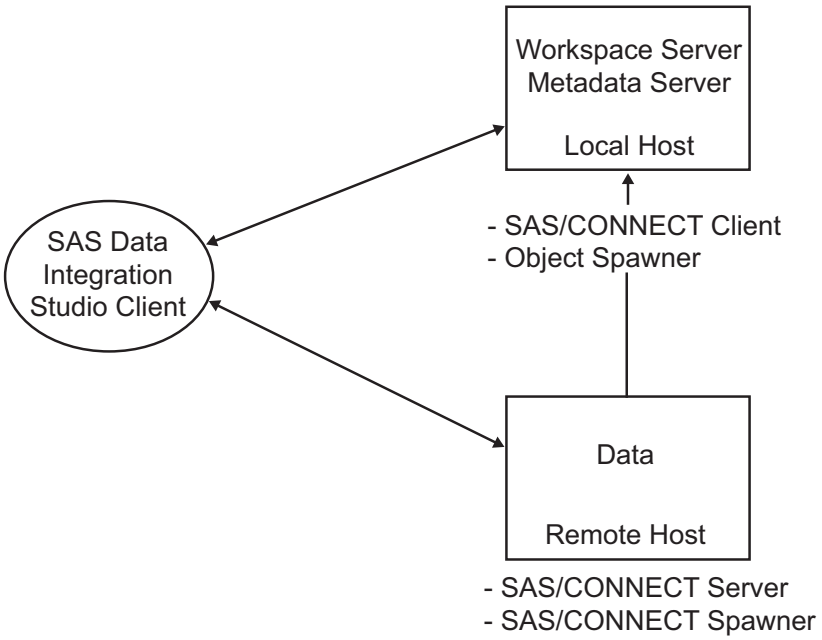
Setting Up Multi-Tier Environments

The procedures in this section explain how to enable your SAS Data Integration Studio users to access data on a host that is remote from the workspace server by installing SAS/CONNECT software on the workspace server host and configuring a SAS/CONNECT server on the remote-data host.

Accessing Data Remotely

In order to access data that is remote from your workspace server, you must run SAS/CONNECT software on your workspace server machine (the *local host*) and also on

the machine where your data resides (the *remote host*). To understand this setup, consider the following figure.



In this network, we have three machines:

- the machine running the workspace server and the metadata server (the local host)
- a separate machine with data (the remote host)
- the SAS Data Integration Studio client machine

Setting Up a Local Host

These procedures assume that SAS software has been installed on this machine. Furthermore, the local host machine needs to be set up as follows:

- The local host runs the SAS Workspace Server.
- The application server of which the workspace server is a component needs to be set as the default application server for SAS Data Integration Studio. This is done from the SAS Data Integration Studio client machine. See "Setting Up a SAS Data Integration Studio Client Machine" on page 257.
- SAS/CONNECT software needs to be installed on the local host machine. This should have been done at installation time.

Setting Up a Remote Host

Base SAS and SAS/CONNECT software must be installed on the remote host machine, and a SAS/CONNECT server must be configured. If you have not already installed SAS software, follow the instructions for a Software Index installation in the *SAS Intelligence Platform: Installation Guide*. After installation, run the Configuration Wizard to configure the SAS/CONNECT software. This will create various scripts, including the **`InstallConnectServer.bat`** script. You use that script to start the SAS/CONNECT spawner as a service. Running as a service, the spawner is able to start server processes as needed. You can find **`InstallConnectServer.bat`** in the *path-to-config-dir***`Lev1\SASMain\ConnectServer`** directory. In a Windows

environment, the **InstallConnectServer.bat** script creates a service. A different script named **StartConnectServer.bat** starts the server.

You can make other configuration changes to the remote host from any machine running SAS Management Console.

In the next two sections, you will define the SAS/CONNECT server and your SAS library in the metadata managed by the metadata server.

Setting Up a Remote SAS/CONNECT Server

Use SAS Management Console to define an application server that contains a SAS/CONNECT component that will run on the remote host machine. This server is a second application server, in addition to the application server running on the local machine (SASMain). This new server is necessary because SASMain does not have access to the remote library; the new application server does. Furthermore, by detecting the existence of this new server, SAS Data Integration Studio knows to generate the code necessary to download the data from the remote server. Follow these steps to set up the server:

1. Start SAS Management Console.
2. Right-click **Server Manager** and select **New Server**. The New Server Wizard appears.
3. Select **SAS Application Server**. Click **Next**.
4. Enter a name for the new server. Click **Next**.
5. Accept the default values for the Server Properties. Click **Next**.
6. Specify SAS/CONNECT Server as the type of server to add to the application server.
7. Select the Basic configuration setting method, and enter the following values:
   - Authentication Domain. When your system was originally set up, a single authentication domain (DefaultAuth) was created. The workspace server was placed in this domain, and each SAS Data Integration Studio user (in the metadata) was given permission to start the workspace server. In your new multi-tier setup, you can continue to use DefaultAuth if the user name and password that are currently in the metadata are also valid on the host where the SAS/CONNECT server is running. If those credentials aren't valid on that host, you must create a new authentication domain. You must place the SAS/CONNECT server in that domain, and create a second login (in the metadata) for each SAS Data Integration Studio user (see "Setting Up a SAS Data Integration Studio Client Machine" on page 257).
   - Host Name: Enter the name of the remote host machine.
   - Port Number: Enter **7551**.
8. Click **Finish**.

Defining a SAS Library

From SAS Management Console, define a SAS library on the remote host machine. These instructions are appropriate for SAS data sets:

1. Expand **Data Library Manager**.
2. Right-click **SAS Libraries** and select **New Library**.
3. Select **SAS Base Engine Library** and click **Next**.
4. Enter a name for the new library and click **Next**.
5. Enter the library properties:
   - **Libref**— Enter an appropriate libref name.

- **Engine**— Select **Base**.
- **Path specification**— Select a path that you have created. Click **Next**.

6 Select your application server from the list. The library is assigned to the server that you select.

7 Click **Finish**.

Setting Up a SAS Data Integration Studio Client Machine

From the SAS Data Integration Studio client machine, you need to (1) define a default application server, (2) define tables in the metadata for the remote host, and (3) test your setup. In addition, if you have created a new authentication domain for your remote server, you need to create new logins for your SAS Data Integration Studio users. These logins must contain a user ID and password that are valid on the SAS/CONNECT host. See "Setting Metadata Permissions for Your User" on page 251.

Defining Tables in the Metadata for the Remote Host

Follow these steps to define tables in the metadata for the remote host:

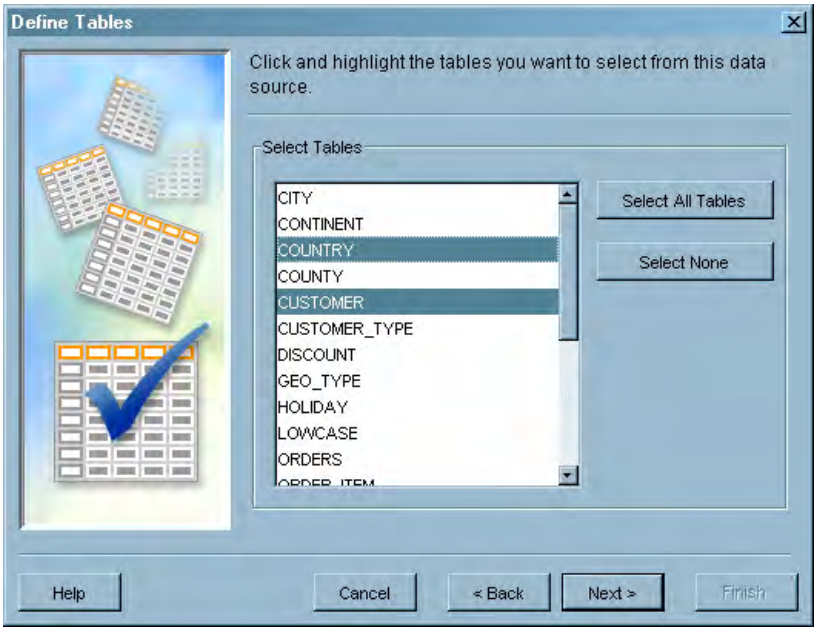
1 Open SAS Data Integration Studio. If you already have a metadata profile, select it. Otherwise, create a new metadata profile as described at "Creating a Metadata Profile" on page 252.

2 Click **Source Designer**.

3 When prompted for the source type that you want to import metadata about, select **Sources** ► **SAS**. Click **Next**.

4 On the Select a SAS Library screen, select the library you created earlier. Click **Next**.

5 Select one or more tables from the library to register, as in the following screen:



6 Select your project repository, click **Next**.

7 The Select Group window gives you the opportunity to place the tables. Skip this option and click **Next**.

8 Click **Finish** in the Wizard Finish window.

Testing Your Setup Using View Data

Select **Repositories ▶ Foundation ▶ Libraries**. Find and expand your remote library. Right-click one of the tables, and select **View Data**. The table contents appear as shown below:

View Data: CLIENT_CHECKING_ACCOUNT (rows 1-100) of 400

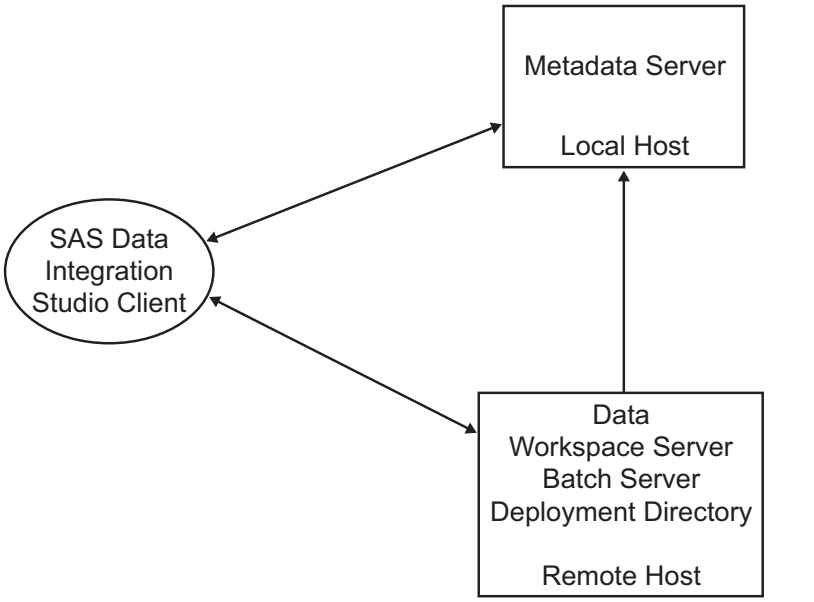
Starting row: 1 Go to row

#	CLIENT_ID	CHECKING_ID	INVOLVEMENT_TYPE_CD
1	0001120358	CHK-113	Y
2	0001621100	CHK-117	Y
3	0001218552	CHK-151	Y
4	0002371066	CHK-138	N
5	0000179858	CHK-8	N
6	0002353883	CHK-107	N
7	0001657610	CHK-37	N
8	0000565835	CHK-23	N
9	0001959476	CHK-43	Y
10	0002112704	CHK-53	Y
11	0001235906	CHK-120	Y
12	0001474992	CHK-191	N
13	0002063250	CHK-71	N
14	0000032744	CHK-180	N
15	0000157738	CHK-48	Y
16	0002504205	CHK-151	N
17	0002504205	CHK-96	Y
18	0002311390	CHK-105	Y
19	0002371066	CHK-131	N
20	0001934490	CHK-37	N

This test verifies that you can access your remote library.

Processing Jobs Remotely

At times you might want to process one or more SAS Data Integration Studio jobs on a large amount of data on a remote machine and then save your results to that remote machine. In such an instance, it might be more efficient to move the job itself to the remote machine. A multi-tiered configuration can make that possible.



- SAS Integration Technologies

To move a job to a remote machine, you need to have a workspace server on the remote host. The following software must be installed on the remote host machine:

- Base SAS software
- SAS Integration Technologies software

If you have not already installed this software, follow the instructions for a Software Index installation in the *SAS Intelligence Platform: Installation Guide*. After installation, run the Configuration Wizard to configure the following:

- SAS Object Spawner
- SAS Workspace Server
- SAS Data Step Batch Server

The Configuration Wizard will create a file named **`instructions.html`** that will give you specific manual instructions to follow in order to complete the process. The steps in that file will be tailored to your own setup and will be based upon your answers to prompts during installation. Below is a generic summary of those steps. Note that in a few cases involving the name of the authentication domain, you might need to enter a different value from that prescribed in **`instructions.html`**.

Starting the SAS Management Console

SAS Management Console is used to manage your SAS environment and will be used to complete many of the following steps.

The SAS Configuration Wizard will create a new metadata profile and start SAS Management Console. If SAS Management Console does not start automatically, you can start it manually.

Defining Your Application Server and Workspace Server

Using SAS Management Console, you need to define your servers in the metadata. To do this, you use the Server Manager plug-in.

1. Right-click **`Server Manager`** and select **`New Server`**.
2. Under **`SAS Servers`**, select **`SAS Application Server`**. Click **`Next`**.

3 Enter a name for the server. (Description is optional.) Click **Next**.
4 Accept the default values for software versions. Click **Next**.
5 Select **Workspace Server**.
6 Select **Basic Configuration**, and enter the following values:
   - Authentication Domain: In some cases this will be your default authentication domain **DefaultAuth**. However, in certain circumstances (described in "Setting Up a Remote SAS/CONNECT Server" on page 256), you might need to enter a different authentication domain.
   - Host Name: Enter the name of your remote host.
   - Port Number: Enter **8591**.
7 Review settings and select **Finish**.

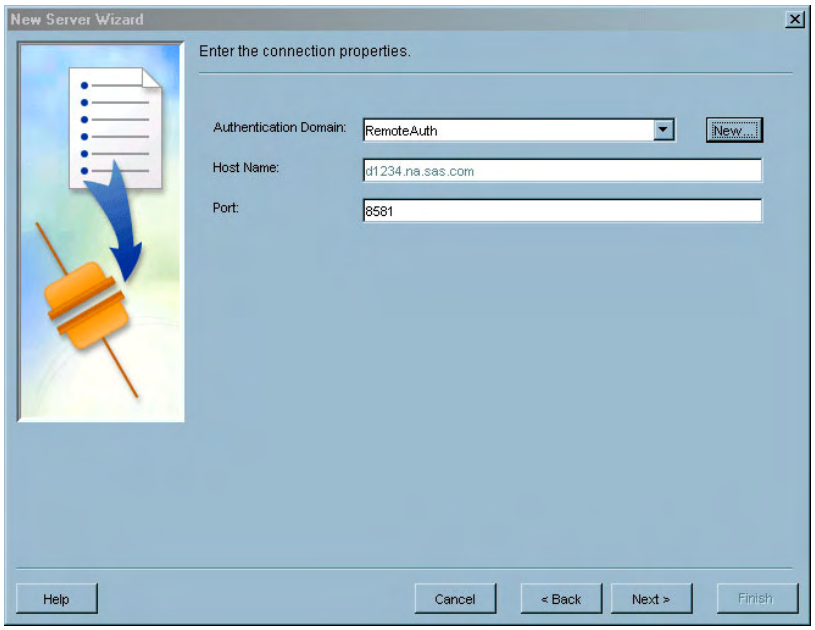
Editing the SAS Command for the Workspace Server

1 Select **+** to expand the **Server Manager** node. Fully expand all three levels of the application server you created previously.
2 Highlight the **Workspace Server** (lowest) level. Using the right mouse button, select **Properties**.
3 On the **Options** tab, replace the existing text in the **Command** field with the command provided in **instructions.html**.
4 Click **OK** to save changes.

Defining Your SAS Object Spawner

The SAS Object Spawner is used to start workspace server processes.

1 Right-click **Server Manager** and select **New Server**.
2 Under **SAS Servers - Spawners**, select **Object Spawner**. Click **Next**.
3 Enter a name for the spawner. (Description is optional.) Click **Next**.
4 Accept the default values for software versions. For the **Associated Machine**, select the name of the remote host machine.
5 Accept the default values for **Operator Login** and **Log file**. Click **Next**.
6 In the list of servers, move the name of the remote workspace server to the Selected list. Click **Next**.
7 Select **Operator Connection**.
8 Enter the following connection properties:
   - Authentication Domain: In some cases this will be your default authentication domain **DefaultAuth**. However, in certain circumstances (described in "Setting Up a Remote SAS/CONNECT Server" on page 256), you might need to enter a different authentication domain.
   - Host Name: Enter the name of the remote host.
   - Port Number: Enter **8581**.



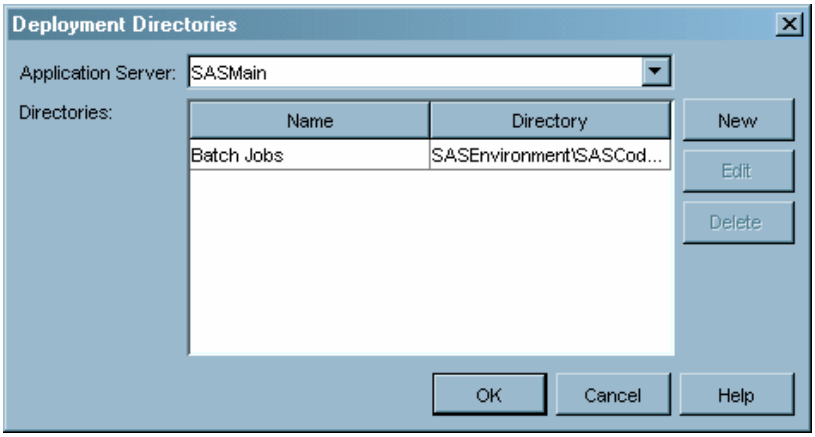
9 Review settings and select **Finish**.

Defining Your SAS Data Step Batch Server

1 Select the top level of the application server you created in the previous procedure.
2 Using the right mouse button, select **Add Application Server Component**.
3 Select **SAS Data Step Batch Server**. Click **Next**.
4 Enter a name for the server. (Description is optional.) Click **Next**.
5 Enter the following server properties:
   - Associated Machine: Enter the name of your remote host.
   - Command Line: Enter the command line provided by **instructions.html**.
   - Logs Directory: Enter the directory provided by **instructions.html**.
6 Review settings and select **Finish**.

Defining Your Batch Job Deployment Directory

1 Highlight **Schedule Manager**. Using the right mouse button, select **Deployment Directories**.
2 With the name of the new application server selected, select **New**.
3 Enter the following directory properties:
   - Name: Enter **Batch Jobs**.
   - Directory: Enter **SASEnvironment\SASCode\Jobs**.



4 Select **OK** to create the new directory definition.
5 Select **OK** to save your changes.

Verifying Your Object Spawner

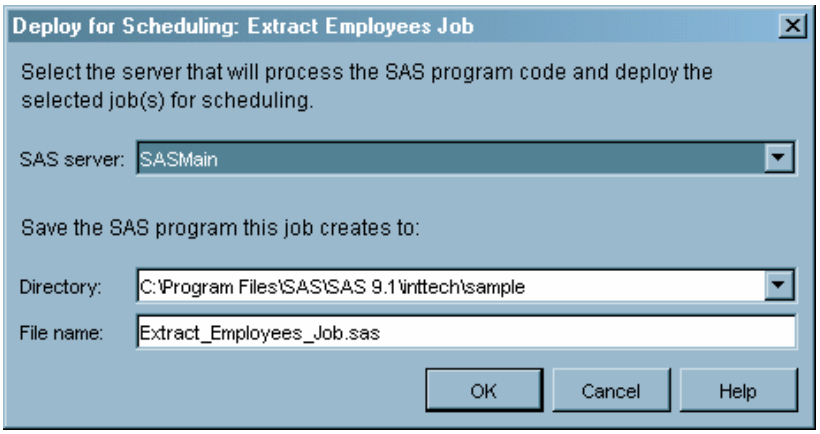
To verify the status of the Object Spawner, view the log file, which is located in a path on your system similar to the following:
C:\SAS\EntBIServer\Lev1\SASMainonD9576\ObjectSpawner\logs\objspawn.log.

Verifying That the Object Spawner Is Working Properly

1 In SAS Management Console, select **+** to expand the **Server Manager** node. Fully expand all three levels of the application server created previously.
2 Highlight the **Workspace Server** (lowest) level.
3 In the right-hand view, you should see a single connection defined. Highlight that connection name and use the right mouse button to select **Test Connection**.
4 Enter a valid user ID and password (for example, *d1234\sasdemo*) and select **OK**. You should see a "Test Connection Successful" message.

Deploying a Job Remotely Using SAS Data Integration Studio

Follow these steps to test the setup you have created:
1 Start SAS Data Integration Studio.
2 Select **Repositories ▶ Foundation ▶ Jobs** in the Inventory tree.
3 Right-click the job you that want to deploy and select **Deploy for Scheduling**.



4 In the **SAS Server** drop-down list, select the application server containing the servers on the remote host.
5 Click **OK** to deploy the job.

The deployed code will be automatically copied to the remote host and will be ready for scheduling or running at any time. You must have a Platform Computing scheduling server installed before you can schedule a job.

Using Custom Tree Folders for Security

When your data integration developers are working in SAS Data Integration Studio, by default the left pane in the main application window displays an Inventory tree. This tree contains a set of folders, each of which corresponds to a type of metadata object that the data integration developer will use routinely in his or her work. The Inventory folder contains the following folders:

- Cubes
- Documents
- External Tables
- Jobs
- Libraries
- Notes
- OLAP Schema
- Tables

By opening a folder, the data integration developer can see the objects of a particular type that are stored in the metadata repository that is specified in his or her metadata profile.

The developer also has the option of creating a Custom tree, which contains user-defined folders. This option has two advantages, one for the data integration developer and one for the administrator.

- The Custom tree enables developers to organize their metadata objects in any manner they see fit. For example, they might want to create a folder that contains all of the metadata objects that are related to a particular set of jobs.
- The Custom tree also displays in the SAS Management Console's **Authorization Manager**. From the **Authorization Manager**, you can set permissions on folders in the Custom tree to specify who can access the objects in the folders and what permissions those users have.

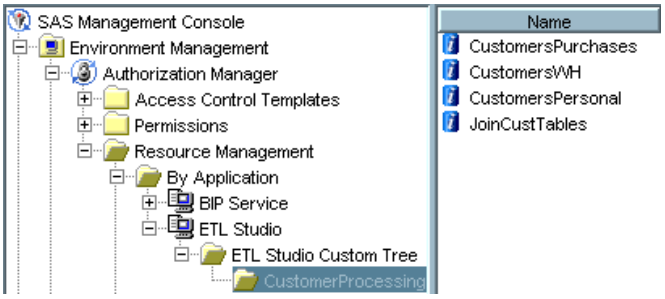
Setting Permissions on Custom Tree Folders

Assume that a SAS Data Integration Studio developer has set up a group (a folder) in the Custom tree to hold the metadata objects with which he or she is working. This section explains how you can use the Authorization Manager in SAS Management Console to set permissions on the group so that only that developer can work with those metadata objects. Perform these steps to set up this type of security:

1 In SAS Management Console, expand the following sections of the SAS Management Console tree, in this order:
   - **Authorization Manager**
   - **Resource Management**

- **By Application**
- **ETL Studio**
- **ETL Studio Custom Tree**

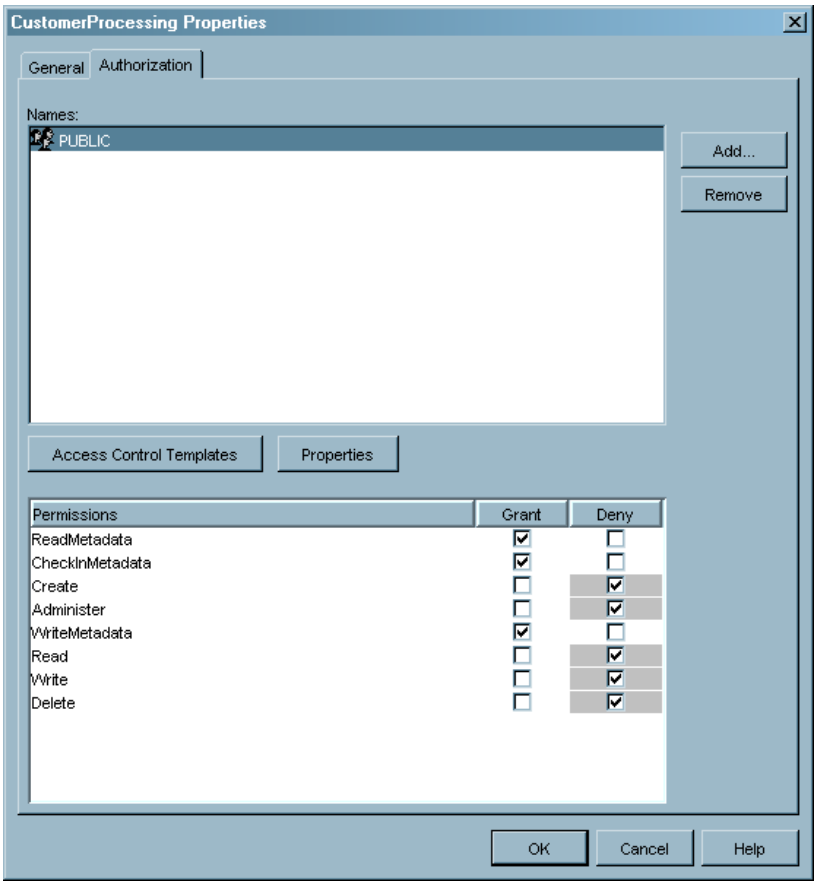
At this point, you will see the top-level groups in the Custom tree, and if you select a group/folder, you will see the objects in that folder in the pane to the right of the tree structure.



In this case, a user has created a group/folder called CustomerProcessing, and this group contains several tables and a job in SAS Data Integration Studio.

2 Open the Properties dialog box for the group by right-clicking the group and selecting **Properties** from the pop-up menu.

3 In the Properties dialog box, select the **Authorization** tab. The exact settings in this box depend on the Default ACT in use. An example is shown in the following display:



4 To make the developer who created the group the only person who is able to access the objects in the group, first deny access to users you do not want to access the objects (such as the members of the group PUBLIC). Then, add your user to the list of names and grant that person appropriate access. If you have set up change management—which is a good idea—you should give the user the permissions ReadMetadata and CheckInMetadata. Otherwise, give the user the permissions ReadMetadata and WriteMetadata.

In a few cases, objects do not inherit permissions from their folder. For example, an external table does not inherit permissions in this way. For these objects, you must set the permissions on the object rather than on the folder. The following table indicates which objects inherit permissions from a group.

Table 13.2 Custom Tree Object Types

Object Type	Description	Inherits from Folder
Cubes	OLAP cubes including dimensions, hierarchies, and levels.	Yes
Documents	Web documents that contain information about another object defined in your metadata repository, such as a table or a job.	Yes
External Tables	Flat files.	No
Jobs	Processes that create output. In SAS Data Integration Studio, the job is illustrated by a process flow diagram.	Yes
Libraries	Data libraries for either SAS data sets or DBMS tables.	Yes
Notes	Text objects that contain information about another object that is defined in your metadata repository, such as a table or job.	Yes
OLAP Schemas	OLAP schemas associated with OLAP cubes.	No
Tables	SAS data sets and DBMS tables.	Yes

Note: Be sure to read the next section about the multiple inheritance of access controls. If you do not understand how multiple inheritance works, users might be able to access resources to which you denied them access in the Custom tree. △

Administering Multiple Inheritance of Access Controls

As is explained in "Inherited Access Controls" in the "Understanding the Metadata Authorization Layer" section in the *SAS Intelligence Platform: Security Administration Guide*, a single resource can inherit permissions from more than one set of parents. For instance, a SAS data set in a custom tree group inherits permissions not only from its group and the group's parents, but also from a SAS library and its parents.

In this scenario, you might have specified that User A cannot access the contents of the custom tree group; however, if User A has access to the SAS library that contains the data sets in the group, User A will also be able to access the data sets in the library (unless the permissions on the data sets themselves deny access to this user).

To make sure that this situation does not prevent you from setting up your access controls properly, we recommend the following practice. Create a separate SAS library

or database library for each set of tables that should be accessed only by a particular user or group of users. In this way, you can set up the same permissions for a custom tree folder and a corresponding library.

Importing and Exporting Generated Transformations

SAS Data Integration Studio ships with a set of transformations that you can use to perform common transformations such as joining data from a set of input tables in a single target table. Some of the transformations use Java code to transform the data, and other transformations use SAS code. The former class are called Java plug-ins, and the latter are called generated transformations. Data integration specialists can write additional Java plug-ins and generated transformations. However, this section deals only with generated transformation, because it is these transformations that you might need to secure.

SAS Data Integration Studio users can use a Transformation Generator wizard to create generated transformations. They can then export these transformations. In their exported state, the transformations are stored as XML files. Other SAS Data Integration Studio users can import these files, at which point those users can employ the generated transformations in their jobs.

The potential problem with this arrangement is this: suppose that multiple data integration specialists are designing these transformations and exporting them to a central repository. Periodically, all of the users import the transformations from the repository to pick up any new transformations. If one user makes a change to an existing generated transformation and other users import that transformation, any jobs written by the other users that make use of that transformation will be changed without their knowledge.

For this reason, it is important for you to set up a central repository (a directory) to which only you have write access. This way, all data integration users can import generated transformations, but only you can write such transformations to the repository. You can disallow changes to existing transformations or let all users know if such a change takes place.

Importing and Exporting Metadata

A common requirement for data integration developers is the ability to do the following:

- design a data warehouse using a data modeling tool such as the AllFusion ERwin Data Modeler
- import metadata that describes the data warehouse into a SAS Intelligence system's metadata repository

You can use either SAS Data Integration Studio or SAS Management Console to import metadata of this type. Both applications enable you to export metadata as well.

Managing the Supported Metadata Formats

The SAS Metadata Server enables you to import metadata from a variety of sources (and to export it in a variety of formats). The server supports the Object Management Group's Common Warehouse Metamodel/XML Metadata Interchange (CWM/XMI) format, the industry standard for data warehouse metadata integration. In addition, by

installing Meta Integration Model Bridge (MIMB) software, you can import metadata from market-leading design tool and repository vendors.

Meta Integration is a SAS software partner. For information about obtaining and installing MIMB software, see **`www.metaintegration.net/Products/MIMB/Description.html`**. You can also request an evaluation license key from this location.

Importing Metadata

The SAS Data Integration Studio Metadata Import Wizard uses converters installed in the **`plug-ins`** directory (of SAS Data Integration Studio or SAS Management Console) to import metadata. By default, you have a converter that handles metadata stored in a CWM 1.0/XMI document. If you want to import metadata that is stored in other formats, you must install the appropriate MIMB software, as described in "Managing the Supported Metadata Formats" on page 266.

The Metadata Import Wizard enables you to import relational data— that is, data from a SAS library or a DBMS schema. The import process ignores any non-relational data. The following list shows the object types you can import:

- CWMRDB.Schema
- CWMRDB.Table
- CWMRDB.View
- CWMRDB.Column
- CWMRDB.SQLDistinctType
- CWMRDB.PrimaryKey
- CWMRDB.UniqueKey
- CWMRDB.ForeignKey
- CWMRDB.SQLIndex

Do either of the following to start the import wizard:

- In SAS Data Integration Studio, connect to the metadata server by using a metadata profile that specifies the repository into which you want to import metadata, and select **Tools ▸ Metadata Importer**.
- In SAS Management Console, right-click the repository into which you want to import metadata (in the Metadata Manager portion on the tree in the left pane), and select **`Import Metadata`** from the pop-up menu.

Both procedures start the same wizard.

After you have started the wizard, consult the *SAS Management Console: User's Guide* for information about how to run the wizard. This document contains step-by-step instructions on how to import metadata.

Exporting Metadata

To export data from an Open Metadata Architecture metadata repository to a file, you use the Metadata Export Wizard. By default, the Open Metadata Architecture enables you to export metadata to CWM/XMI files. As with the import feature, you can export metadata in other formats by installing the appropriate MIMB software in your SAS Data Integration Studio or SAS Management Console **`plug-ins`** directory.

There are two restrictions on the export function:

- You can export only relational data, such as data from a SAS library or a DBMS schema. The section "Managing the Supported Metadata Formats" on page 266 lists the types of objects that you can export.

- If you are exporting metadata from a dependent repository—such as a project directory—metadata is not retrieved from the parent(s) of the repository. For example, only tables that use library definitions in the exported repository are exported. Tables that use library definitions in a parent repository are not exported.

Do either of the following to start the export wizard:

- In SAS Data Integration Studio, connect to the metadata server using a metadata profile that specifies the repository from which you want to export metadata, and select **Tools ▶ Metadata Exporter**.
- In SAS Management Console, right-click the repository from which you want to export metadata (in the Metadata Manager portion of the tree in the left pane), and select **`Export Metadata`** from the pop-up menu.

Both procedures start the same wizard.

After you have started the wizard, consult the *SAS Management Console: User's Guide* for information about how to run the wizard. This document contains step-by-step instructions on how to export metadata.

Administering Message Queues

Message queues are collections of data objects that enable asynchronous communication between processes. One application writes a message to a queue. Another application reads messages from the queue to begin the next step in a process. Verification processes guarantee that all messages are transmitted without error.

Queue managers handle message transmission and verification.

About the Third-Party Messaging Software

Fundamental message queue technology is provided in the following third–party software:

- Microsoft MSMQ
- IBM WebSphere MQ

You need to install and configure at least one messaging software package before you register metadata and run SAS messaging jobs. To learn more about the third-party messaging software, see "Application Messaging Overview" in the *SAS Integration Technologies: Developer's Guide* at **`http://support.sas.com/rnd/itech/doc9/dev_guide/messageq/index.html`**.

About Message Queues in the SAS Intelligence Platform

In the SAS Intelligence Platform, you register metadata for queues and queue managers by using SAS Management Console after you install and configure third-party messaging software.

After registering metadata, you use the following transformations to create messaging jobs in SAS Data Integration Studio:

- Microsoft Queue Reader
- Microsoft Queue Writer
- Websphere Queue Reader
- Websphere Queue Writer

SAS Data Integration Studio also enables impact analysis and reverse impact analysis on queues, which you can use to track data flow into and out of queues.

The following data can be written to message queues using the queue writer transformation:

- text of length up to 32,767 bytes
- rows from SAS tables, one row per message
- external files

The queue reader transformation can be configured to read a specified number of rows from the message queue.

Requirements for Message Queues

To implement message queueing, you need to install and configure Microsoft MSMQ or WebSphere MQ. Start the installation process by downloading and installing third-party software on specified application servers. In general you can plan to install a queue manager on each host that runs a workspace server that will run message queue jobs. If you choose WebSphere MQ, you can install queue managers on hosts other than workspace servers.

Hosts that receive third-party messaging software also need to receive Base SAS and SAS Integration Technologies software, release 9.1.3, Service Pack 4 or later.

SAS Intelligence Platform clients that run SAS Data Integration Studio need to install release 3.4 or later. As part of the installation of SAS Data Integration Studio 3.4, you are required to use SAS Management Console to run **Tools ▶ Update Metadata for SAS Data Integration Studio in SASMC**. If you are using change management at your site, all metadata needs to be checked in before you can run the metadata update. This generally happens as a normal part of the SAS Data Integration Studio installation process.

Installing and Configuring the Microsoft MSMQ Message Queue Software

Complete the following steps to install the MSMQ software on a workspace server host that will run message queue jobs:

1. Ensure that the host meets the requirements for message queues, as described in "Requirements for Message Queues" on page 269.
2. In Windows, select **`Add/Remove Programs`**.
3. In the shortcut bar of the Add/Remove Programs window, select **`Add/Remove Windows Components`**, which opens the Windows Components Wizard.
4. In the Windows Components Wizard, select the **`Message Queueing`** check box and click the **`Details`** button, which displays the Message Queueing dialog box.
5. In the Message Queueing dialog box, select all four subcomponent check boxes, and then click **`OK`** to return to the Windows Components Wizard.
6. Click **`Next`** to install the MSMQ software, and then close the Windows Components Wizard.
7. Use the MSMQ software to create the queues and queue managers that are required on this particular host. You need to create at least one queue for each queue manager. The same object names will be used in SAS, so you might want to include the host name as part of the object names.
8. Use the Microsoft MSMQ software to start the queue managers.

9 Use SAS Management Console to register metadata for queues and queue managers, as described in "Registering Metadata for Queues" on page 270.

Installing and Configuring the WebSphere MQ Message Queue Software

Follow these steps to install and configure the WebSphere MQ software:

1 Download the software and documentation. Start with the following Web site: **http://www-306.ibm.com/software/integration/wmq/index.html**. Work with installation representatives from IBM and SAS as needed. At minimum, you need to install the WebSphere MQ Client software on all workspace server hosts that will run message queue jobs. Make sure that you meet the requirements for message queueing that are specified in "Requirements for Message Queues" on page 269.

   Note that if you install only the WebSphere MQ Client software, then you need to configure the environment variable MQSERVER to communicate with the WebSphere MQ Server software.

2 Use the WebSphere MQ software to create queues and queue managers on their respective hosts. For each queue manager you need to define at least one queue. For reference in SAS, you might want to include the machine name in the names of the queue managers.

3 Use the WebSphere MQ software to start the queue managers on their respective hosts.

4 Use SAS Management Console to register metadata for queues and queue managers, as described in the next section.

Registering Metadata for Queues

Complete the following steps to register metadata for a queue and a queue manager:

1 In SAS Management Console, right-click **Server Manager** and select **New Server**, which opens the New Server Wizard.

2 Under **Queue Managers**, select either **MSMQ Queue Manager** or **Websphere Queue Manager** and click **Next**.

3 Enter a name and description for the new queue manager. The name needs to be exactly the same as the name that is defined in the third-party messaging software. The names are case-sensitive. When you are finished, click **Next**.

4 Check the queue type and SAS version number, and click **Next**.

5 Specify an authentication domain. Click **New** to define a new authentication domain.

6 Specify a full network name for the host.

7 Specify a port number.

8 Register metadata for at least one queue. One queue is required to be registered for each queue manager. Click **New** to display the New Queue dialog box.

9 In the New Queue dialog box, enter a queue name, description, and queue manager name. If you are registering metadata for a queue that already exists in the third-party software, the queue name and the queue manager name must match exactly the names that are defined in the third-party messaging software. The names are case-sensitive. Click **OK** to display the queue in the **Selected items** list. Click **Next** to apply your input.

10 Review your entries and click **`Finish`**.

Be sure to register metadata for all of the queues and queue managers that will be used in SAS jobs.

To begin running messaging jobs, make sure that the queue managers are running on their respective hosts.

Managing Queues and Queue Managers

To modify the metadata definitions of queues and queue managers, use the New Server Wizard in SAS Management Console to delete the existing definitions and create new definitions.

Before you delete metadata, you might want to run impact analysis and reverse impact analysis in SAS Data Integration Studio to examine the related jobs. To run impact analysis, right-click the queue in the **`Inventory`** tab and select **`Impact Analysis`** or **`Reverse Impact Analysis`**. The resulting analytical report enables you to display the properties of tables, jobs, and transformations.

To delete metadata for queues and queue managers, right-click the object in the Server Manager of SAS Management Console and select **`Delete`**. Use the third-party software to delete the physical queue objects.

Providing Support for Web Services

Web Services

Beginning in version 3.4, SAS Data Integration Studio supports the creation of stored processes that can be invoked through a Web service. For complete information about the procedure for creating these special stored processes, see the SAS Data Integration Studio help. The interfaces to these stored procedures are published in WSDL (Web Services Description Language), and remote clients both discover the available services and execute services using SOAP requests.

To provide Web services to clients, in addition to having a metadata server and a stored process server, you are required to install and configure the SAS BI Web Services for Java or the SAS BI Web Services for .NET. These products handle service requests from remote clients. If you do not have one of the two products installed, you must install one of them, as explained in the next section.

SAS BI Web Services

SAS BI Web Services for Java Versus SAS BI Web Services for .NET

You can use either SAS BI Web Services for .NET or SAS BI Web Services for Java to handle requests for Web services. SAS BI Web Services for .NET is supported on the following platforms: Windows 2000, Windows XP Professional, and Windows 2003. SAS BI Web Services for Java is supported on any operating system that can host one of the SAS Intelligence Platform's supported Java application servers. For information on what Java application servers are supported, see the Third-Party Software Downloads page at **`http://support.sas.com/thirdpartysupport`**. Click the Web Application Servers link for your version of the SAS Intelligence platform to see the list of supported servers.

For more information about the differences between the two versions of the SAS BI Web Services, see the section "Deciding Between .NET and Java" in the *SAS Integration Technologies: Developer's Guide*, at **`http://support.sas.com/rnd/itech/doc9/dev_guide/websrvcs/decide.html`**.

Install SAS BI Web Services for .NET

To install and configure SAS BI Web Services for .NET, perform the following steps.

1. Install and configure the following prerequisite software (in this order):
   - Microsoft Internet Information Services (IIS)
   - Microsoft .NET Framework
2. If necessary, add the contents of the CD containing the SAS BI Web Services for .NET to your SAS Software Depot.

   Note: You can find detailed instructions for performing this step and the following steps in the *SAS Intelligence Platform: Installation Guide*. △
3. If you received a new SAS Installation Data (SID) file when you ordered the SAS BI Web Services for .NET, put that SID file in your project directory.
4. Install the BI Web Servics for .NET using the SAS Software Navigator's Software Index option. For details about how to perform this type of installation, see "Install Third-Party Software Using Software Index" in the chapter "Installing Third-Party Products" in *SAS Intelligence Platform: Installation Guide*. (You add a SAS product to an existing system in the same way that you install some third-party products.)
5. Use the SAS Configuration Wizard to configure the SAS BI Web Services for .NET. For information about using this wizard to configure a single product, see "Configure a Newly Added Product" in the chapter "Software Index Installations" in *SAS Intelligence Platform: Installation Guide*.

Install SAS BI Web Services for Java

SAS BI Web Services for a Java is a J2EE Web application that runs on a Java application server. It also has a dependency on the SAS Web Infrastructure Kit. Therefore, unless you already have one or more of these products, you must install the following items:

- a Java 2 SDK
- a supported servlet container or Java application server
- the SAS Web Infrastructure Kit
- the SAS BI Web Services for Java

To install and configure these products, perform the following steps:

1. Install the Java 2 SDK.
   a. Go to the Third-Party Software Downloads page (**`http://support.sas.com/thirdpartysupport`**).
   b. Locate the set of links that is appropriate for your version of the SAS Intelligence Platform, and click the Java Development Kits link. You will see a list of kits organized by operating system.
   c. Search for the row that has your operating system in the Platform column, and click the corresponding Download link. This will take you to a third-party Web site where you can download the Java 2 SDK.
   d. Download the SDK, and run the installation program.
2. Install a servlet container or Java application server.

a If you have not already done so, determine which product you want to use. To see what products are supported, go to the Third-Party Software Downloads page (**`http://support.sas.com/thirdpartysupport`**), locate the set of links that is appropriate for your version of the SAS Intelligence Platform, and click the Web Application Server link. You will see a table that lists the supported servers.

Note: SAS BI Web Services for Java is not currently supported on WebSphere. △

b Obtain the server from the vendor. You can download Tomcat from the Apache Web site for free by using the link on the Third–Party Software Downloads site. You can purchase WebLogic from BEA Systems (**`http://www.bea.com`**).

c Install the server. Installation instructions for installing WebLogic are available from the Web page mentioned in step a.

3 If necessary, add the contents of the CDs containing the SAS Web Infrastructure Kit and SAS BI Web Services for Java to your SAS Software Depot.

Note: You can find detailed instructions for performing this step and the following steps in the *SAS Intelligence Platform: Installation Guide*. △

4 If you received a new SAS Installation Data (SID) file when you ordered the products mentioned in the previous step, put that SID file in your project directory.

5 Install the SAS Web Infrastructure Kit and the SAS BI Web Services for Java using the SAS Software Navigator's Software Index option. For details about how to perform this type of installation, see "Install Third-Party Software Using Software Index" in the chapter "Installing Third-Party Products" in *SAS Intelligence Platform: Installation Guide*. (You add SAS products to an existing system in the same way that you install some third-party products.)

6 Use the SAS Configuration Wizard to configure the SAS Web Infrastructure Kit and SAS BI Web Services for Java. For information about using this wizard to configure these products, see "Configure a Newly Added Product" in the chapter "Software Index Installations" in *SAS Intelligence Platform: Installation Guide*. The SAS Configuration Wizard will guide you through the process of defining the necessary metadata objects and deploying your Web applications to your Java application server.

After you have installed and configured the SAS BI Web Services for Java, a README file (**`xmla_readme.html`**) will be available in the directory *SAS-install-dir***`\Web\WebServicesforJava\1.0`**. Read this document for further information about the product.

Calling Web Services

There are a couple of sources of information about writing clients that can access your Web services. One source is the *SAS Integration Technologies: Developer's Guide*. See the section of the document called "SAS BI Web Services" (**`http://support.sas.com/rnd/itech/doc9/dev_guide/websrvcs/index.html`**). If you have installed the SAS BI Web Services for Java, you can also find information in the README file *SAS-install-dir***`\Web\WebServicesforJava\1.0\xmla_readme.html`**.

Enabling the Bulk Loading of Data

Bulk Loading Data into DBMS Tables

When data integration specialists use the SQL Join transformation to write data to tables in a DBMS, they have the option of specifying that they want to use the DBMS's bulk-loading capabilities. When this option is used, entire rows of data are written to the database at once, which can significantly improve performance.

By default, however, your object spawner is configured so that operating system commands cannot be executed from a spawned workspace server, which is a requirement for using the bulk-load feature. To reconfigure your object spawner, follow the directions in the next section. Also, note the caution at the end of the procedure.

Enable the Workspace Server to Execute Operating System Commands

To enable a workspace server to execute operating system commands, you must add the **-allowxcmd** spawner invocation option to the command that starts the object spawner. Use the following directions, as appropriate for your operating system.

On Windows systems, perform these steps:

1. In a Command Prompt window, change directories to *SAS-config-dir***\Lev1\SASMain\ObjectSpawner**.
2. Stop the object spawner by entering the command **ObjectSpawner stop**.
3. If the object spawner is set up to run as a service, remove the service. You can do this by entering the command **ObjectSpawner remove**.
4. Edit the script that is used to start the object spawner: *SAS-config-dir***\Lev1\SASMain\ObjectSpawner\ObjectSpawner.bat**. Add the spawner invocation option **-allowcmd** to the appropriate line. If you will run the object spawner as a service (which is recommended), append the option to the line following the label **:install**. Otherwise, append the option to the line following the label **:start2**.
5. Save your changes.
6. If you want to run the object spawner as a service, install the service. You can do this by entering the command **ObjectSpawner install**.
7. Start the object spawner by entering the command **ObjectSpawner start**.

On UNIX systems, perform these steps:

1. Change directories to *SAS-config-dir***/Lev1/SASMain/ObjectSpawner**.
2. Stop the object spawner by entering the command **ObjectSpawner.sh stop**.
3. Edit the script **ObjectSpawner.sh** so that the command used to start the object spawner contains the option **-allowxcmd**. Add the option to the line that begins with **$CMD**, just before the first **>** sign.
4. Save your changes.
5. Restart the object spawner by entering the command **ObjectSpawner.sh start**.

CAUTION:

After you have loaded your data, you should remove the -allowxcmd option from your object-spawner script and restart the spawner. There are security risks associated with allowing system commands to be executed from a client application. Clients can use host commands to perform potentially harmful operations, such as file deletion. △

Testing the Platform Computing Scheduling Server

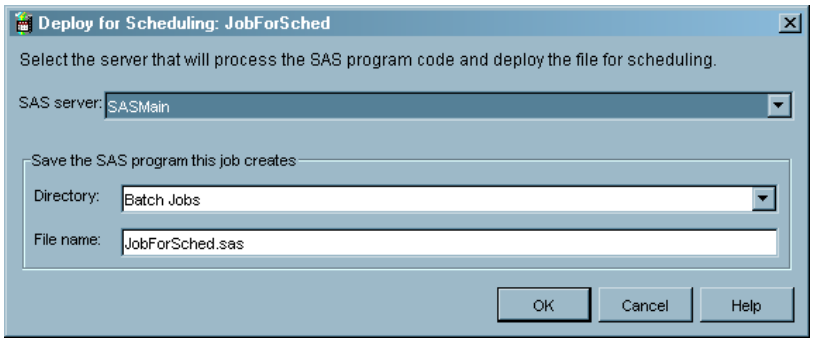
Data integration specialists can create sets of SAS Data Integration Studio jobs, called flows. Each job within a flow can be scheduled to execute as follows:

- at a certain time
- at a time that depends on the state of the file system
- at a time that depends on the status of another job within the flow

The software that enables this scheduling includes the Schedule Manager plug-in to SAS Management Console and a product from Platform Computing: a Platform scheduling server. During the initial installation of your system, you will have installed these products and configured the necessary batch and scheduling servers. Now you need to test the scheduling system.

Here is how to quickly make sure that the scheduler is working correctly.

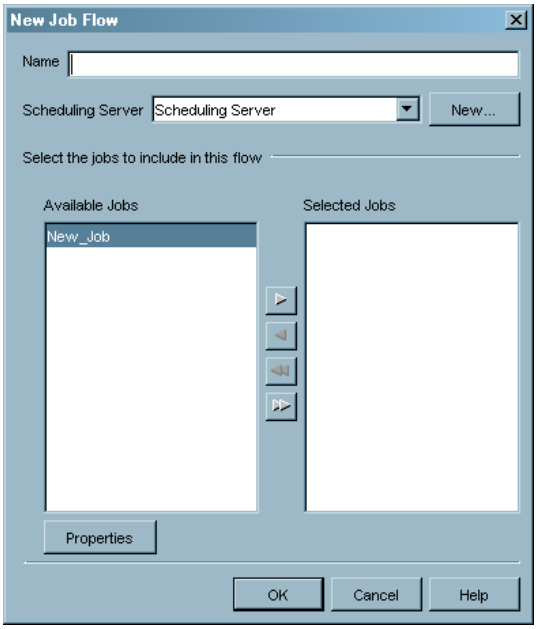
1. Create an empty job in SAS Data Integration Studio. You can do this by starting the New Job Wizard, entering a name (such as JobForSched) in the **Name** text box, and clicking **Finish**. A new job appears in the Jobs folder of the Inventory tree.
2. Deploy the new job for scheduling by right-clicking the icon for the new job and selecting **Deploy for Scheduling**. The Deploy for Scheduling dialog box appears.



   From the **SAS server** list box, select the application server that contains your batch server, and from the **Directory** list box, select the name that you specified earlier (**SASEnvironment\SASCode\Jobs** by default) for your deployment directory. Use the default value for **File name**. Then click **OK**.

3. In SAS Management Console, use these steps to create a job flow (a set of related jobs):

   a. Right-click the Schedule Manager, and select **New Flow** from the pop-up menu. The New Job Flow dialog box appears.



b Enter a unique name for the job flow in the **Name** text box.
c From the **Scheduling Server** list box, select the name of the Platform Job Scheduling Server that you created earlier.
d Move the empty job you just created from the list of available jobs to the list of selected jobs.
e Click **OK**.

A job flow icon appears beneath the Schedule Manager.

4 In SAS Management Console, schedule the new job flow that you want to run:

a Right-click the Job Flow icon, and select **Schedule Flow** from the pop-up menu that appears. A Schedule Flow dialog box appears.
b In the Schedule Flow dialog box, leave the value of the **Trigger** list box set to **Run Once** and click **OK**.

You should see a message indicating that the flow has been scheduled to run.

5 To verify that the job ran successfully, use the Flow Manager application (part of the Platform scheduling server).

Setting Up the SAS Data Quality Server Software

By installing the SAS Data Quality Server software and configuring a SAS application server to read a Quality Knowledge Base, you enable data integration developers to use the data quality transformations Create Match Code and Apply Lookup Standardization. Those developers can also make use of the data quality functions that are provided in the Expression Builder, which is available in many of the data integration transformations.

This section explains the installed configuration of the SAS Data Quality Server software, and how to perform a simple test to ensure that the system is working. This section also covers several administrative tasks associated with data quality, including the following:

- downloading new locales

- creating new schemes
- setting data quality options in SAS Data Integration Studio

Note: If you are unfamiliar with the subject of data quality and the terminology used in this section, see the *SAS Data Quality Server: Reference*. This document is available in the SAS Help and Documentation and on the SAS OnlineDoc CD-ROM. △

Understanding the Data Quality Configuration

If you have installed the SAS Foundation software and the SAS Data Quality Server software and have run the SAS Configuration Wizard, everything should be set up for your data integration developers to use the data quality transformations and functions. This automatic setup is convenient; however, you need to understand how things are set up in case you need to make changes later.

During installation, the SAS Data Quality Server software is installed in *sas-root***dquality**. A typical default path might be **C:\Program Files\SAS\SAS 9.1\dquality**.

Locales and schemes are located in directories subordinate to **dquality**: **sasmisc\content\locale** and **sasmisc\content\scheme**.

A data quality configuration file is used to specify the location of the Quality Knowledge Base. When you start a local interactive SAS session, the data quality configuration is referenced by default at *sas-root***dquality\sasmisc\dqsetup.txt**.

When you use a client application such as SAS Data Integration Studio to start a SAS workspace server session on a SAS application server, the data quality configuration file that is referenced is located by default at *app-server-root***Lev1\SASMain\dqsetup.txt**. A typical default path might be **C:\SAS\EGServers\Lev1\SASMain\dqsetup.txt**, where SASMain is the name of the SAS application server.

On SAS application servers, the location of the data quality configuration file is specified in the SAS configuration file *app-server-root***Lev1\SASMain\sasv9.cfg**. In the SAS configuration file, the line that specifies the location of the setup file (by default) is as follows:

```
-dqsetuploc "dqsetup.txt"
```

In that same SAS configuration file, the server's default data quality locale is specified as:

```
-dqlocale (ENUSA)
```

You can change values in the SAS configuration file without restarting the SAS application server. Any Data Integration Studio clients will need to be restarted in order to use any new values.

You can use the Properties windows of the data quality transformations to override the server defaults for data quality. You can also use the **Data Quality** tab of the Options dialog box in SAS Data Integration Studio to override server defaults in metadata profiles. To display the Options dialog box, select **Tools ▶ Options**.

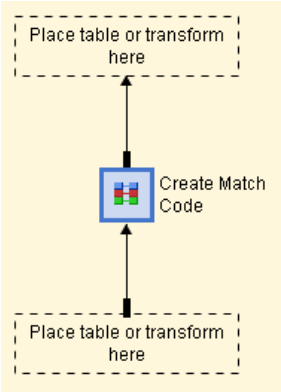
Testing the SAS Data Quality Server Software

A simple procedure to verify that your SAS Data Quality Server software is working is to create a job in SAS Data Integration Studio that contains a Create Match Code transformation. Follow these steps to create such a job.

1. From the SAS Data Integration Studio desktop, select **Tools ▶ Process Designer** to start the New Job Wizard.

2 In the New Job Wizard, enter a name for the job—such as Create Database Match Codes—in the **Name** field. Then, click **Finish**. A new Process Designer window appears on the right side of your workspace.

3 From the **Process Library** tree, select and drag the Create Match Code template into the Process Designer. You should see the following template in the designer:



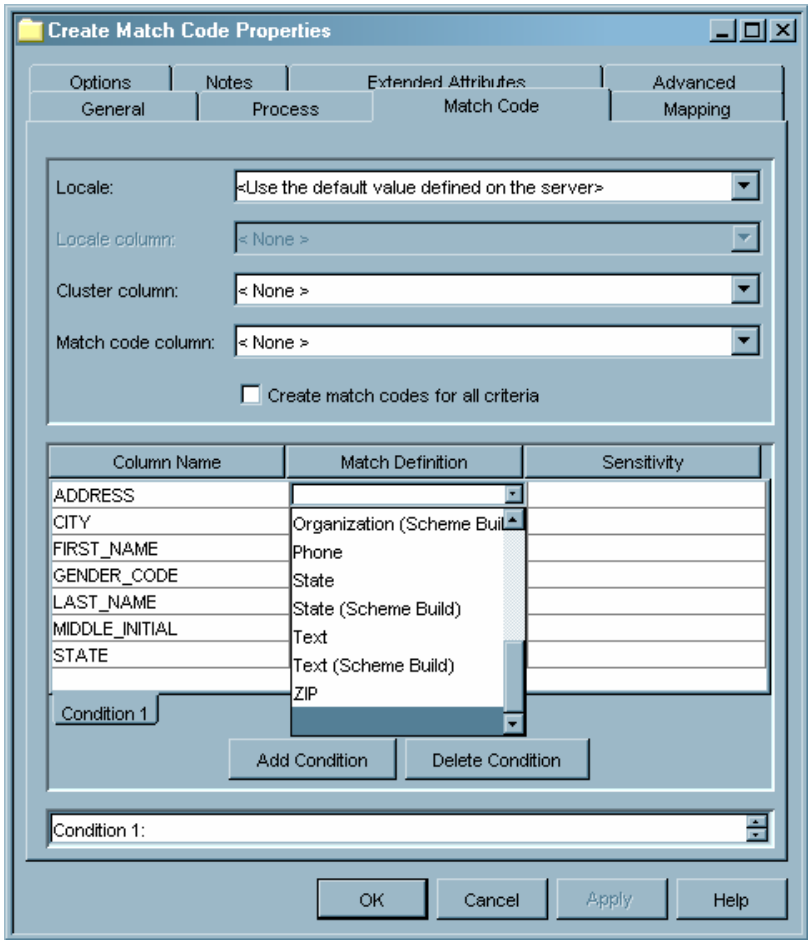
4 From the **Inventory** tree, or another tree view, select and drag the metadata object for any table to the source drop zone.

5 From the **Inventory** tree, or another tree view, select and drag the metadata object for any table to the target drop zone. Both a Loader and the target table will be added to the graphical representation of the job.

6 Double-click the **Create Match Code** transformation to display the **Properties** menu.

7 As the Properties window opens, a dialog box indicates that match definitions are being loaded from the Quality Knowledge Base. This indicates that the SAS Data Quality Server software has been properly installed and configured.

8 To see a list of the match definitions that were loaded, specify a source table for the job and display the **Match Code** tab of the Properties window. Double-click the **Match Definitions** column to show the list of match definitions, as shown in the following display:



Downloading Locales

When initially installed, the Quality Knowledge Base contains a single locale (English/USA). You can obtain additional locales from DataFlux, a SAS Company, at the following Web address: **`http://www.dataflux.com/QKB`**. DataFlux regularly adds new locales for various regions and national languages.

If you install additional locales, you need to update your data quality setup file accordingly, as indicated in the documentation that is provided with each locale. Information on locating and editing the setup file is also provided in *SAS Data Quality Server: Reference*.

You can also create new locales (and edit existing ones) using the dfPower Customize software from DataFlux, a SAS Company.

Creating Schemes

A *scheme* is a reusable collection of match codes and standardization values that is applied to input character values for the purposes of transformation or analysis. Schemes can be created in Blue Fusion Data (BFD) format or SAS format (NOBFD). Before your data integration developers can use the Apply Lookup Standardization template in SAS Data Integration Studio, you must specify a scheme repository and a scheme repository type. You also need to create schemes by using the SAS Data Quality

Server or dfPower Studio software. For information on how to create schemes, see the relevant product documentation.

Scheme repositories should be separated based on scheme type. BFD schemes and NOBFD schemes should be stored separately to ensure that standardization jobs use the appropriate schemes. Two scheme repositories are provided in the default installation. On the SAS application server, the default scheme directory is *path-to-config-dir***\Lev1\SASMain\SASEnvironment\QltyKB\scheme**. This directory contains a number of BFD schemes that are supplied with the SAS Data Quality Server software.

A second default scheme directory is provided for interactive SAS sessions that are started on the local host: *sas-root***\dquality\sasmisc\content\scheme**.

In SAS Data Integration Studio, the scheme repository and scheme repository type are specified in the **Data Quality** tab of the Options dialog box (select **Tools ► Options**). The default scheme repository type is **dfPower Scheme (BFD)**. No default scheme repository is specified. When you specify a scheme repository in SAS Data Integration Studio, a full or explicit path is recommended. Relative paths must be specified relative to the SAS application server.

If you change an existing value in the fields **Scheme Repository Type** or **Scheme Repository**, then you need to replace any existing instances of the Apply Lookup Standardization transformation in any existing jobs. Replacement is required because scheme metadata is added to these jobs when they are run for the first time. To update a job to use a different scheme repository, add a new Apply Lookup Standardization transformation to the job, configure the new transformation, delete the old transformation, and move the new transformation into place.

Setting Data Quality Options for SAS Data Integration Studio

You can set several options related to data quality by using the **Data Quality** tab in the Options dialog box in SAS Data Integration Studio (select **Tools ► Options**). The following table defines the available options.

Table 13.3 Data Quality Options in Data Integration Studio

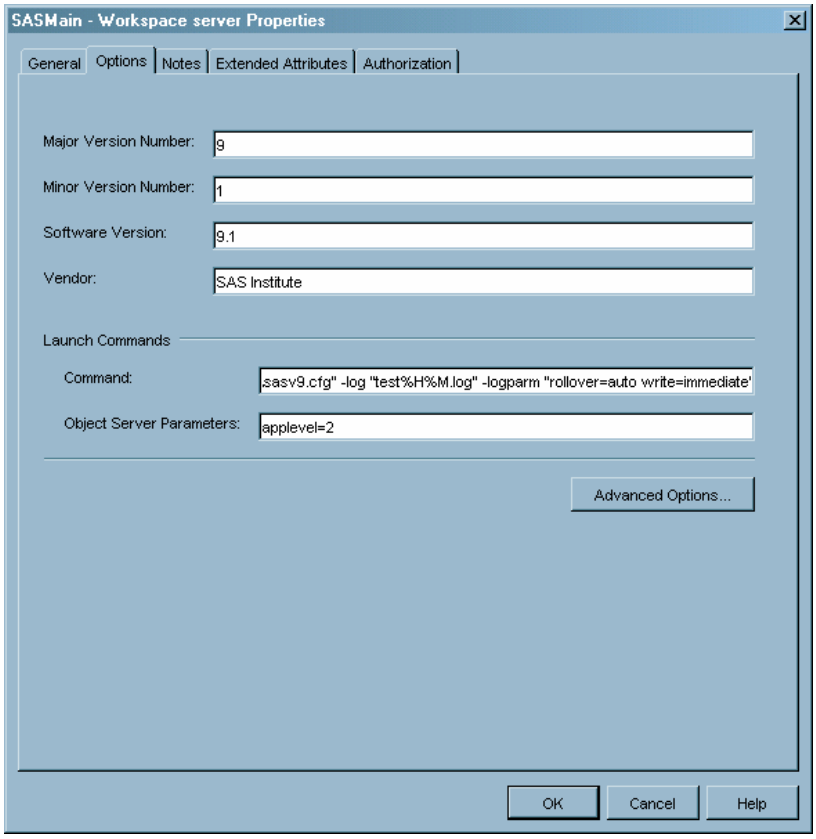
Field	Explanation
Default Locale	By default, this value is set to **<Use the default value defined on the server>**. Unless you have edited the **sasv9.cfg** file in your configuration directory, the default locale is ENUSA. You can change the default locale by selecting a different value from the drop-down list.
DQ Setup File Location	By default, this value is also set to **<Use the default value defined on the server>**. Unless you have edited the **sasv9.cfg** file in your configuration directory, the setup file **dqsetup.txt** will be located in the same directory as the **sasv9.cfg** file. Use the keyboard, drop-down list, or **Browse** button to specify another setup file.
Scheme Repository Type	By default, the repository type is **dfPower scheme (BFD)**. Any schemes that you create in SAS format (NOBFD) need to be stored in a separate repository. Use the drop-down list to switch between the two repository types.

Field	Explanation
Scheme Repository	Specifies the path of the BFD or NOBFD scheme repository. BFD repositories need to be specified as SAS libraries. NOBFD repositories need to be specified with a path. Use the keyboard, drop-down list, or **Browse** button to specify a different repository. Always store BFD and NOBFD schemes in separate repositories.
Enable launcher	Enables the field **DataFlux installation folder**, displays the dfPower Profile Launcher icon in the Shortcuts bar, and adds the dfPower Profile Launcher option to the **Tools** menu.
DataFlux installation folder	Specifies the location of the DataFlux installation directory. The default value is **C:\Program Files\DataFlux\dfPower Studio***release-number*. Use the keyboard, drop-down list, or the **Browse** button to specify a different installation folder.

Redirecting Output and Logging Information to a File

SAS Data Integration Studio jobs generate SAS code when they are executed. It is sometimes useful to designate destinations for the outputs and logs that are associated with this code. To specify alternative destinations for the SAS log and SAS output, add the following options to the **sas** command that starts the SAS Workspace Server.

1. In SAS Management Console, expand the **Server Manager**; then, expand the **SASMain - Logical Workspace Server**. You will see a tree node that represents the physical workspace server.
2. Right-click the icon for the physical workspace server, and select **Properties** from the pop-up menu. A Workspace Server Properties dialog box appears.
3. Click the **Options** tab. You will see the information that is shown in the following display.



4 Edit the text in the **Command** text box. By default, this text is set to

```
sas -config "path-to-config-dir\Lev1\SASMain\sasv9.cfg"
```

To route the SAS log to a file, edit the command to make it look something like this:

```
sas -config "path-to-config-dir\Lev1\SASMain\sasv9.cfg"

  -log log-file-name%H%M.log -logparm "rollover=auto"
```

For routing the SAS output to a file, the command should look like this:

```
sas -config "path-to-config-dir\Lev1\SASMain\sasv9.cfg"
  -print print-file-name.lst
```

5 If you are redirecting the log, enter the following parameter in the **Object Server Parameters** text box:

```
applevel=2
```

This will allow an appropriate level of log information to be routed to the file.

CAUTION:
Do not add the options to the configuration file specified in this command. Setting values in **Lev1\SASMain\sasv9.cfg** affects every server that is launched. This includes the metadata server, the OLAP server, and every workspace server and stored process server. △

6 Click **OK** in the Workspace Server Properties dialog box.

Enabling Status Code Handling

When a data integration developer executes a job in SAS Data Integration Studio, notification of the job's success or failure can be e-mailed to a person, can be written to a file, or can cause the execution of an autocall macro. A **Status Handling** tab is included in the property windows for jobs and for some transformations. Users can select options from a list of code conditions and actions on this tab. For example, a user can select a code condition such as **successful** and associate it with an action such as Send Email.

Before data integration developers can use some of the actions, you must set up the environment properly. Such setup is required for the following actions:

- E-mail actions—You must set SAS system options for e-mail for the SAS application server that is used to execute jobs.
- Custom actions—You must make a SAS macro autocall library accessible by the SAS application server that is used to execute jobs.
- Send Entry to a Data Set—You must pre-assign the library that contains the data set to an application server before the job or transformation executes.

For information on how to support the first two actions listed above, see the subsections below. For information on how to support the last action, see Chapter 10, "Assigning Libraries," on page 197.

Supporting the E-mail Action

Setting up the e-mail action is simple. Just add the appropriate SAS system options for e-mail to the configuration file *path-to-config-dir***Lev1\SASMain\sasv9.cfg**. For example, if you are using the SMTP e-mail interface, you would add to this file the following options:

```
-emailsys smtp
-emailhost email-server
```

In this case, the value of *email-server* specifies the SMTP server that supports e-mail access for your site.

Note: The e-mail system options are documented in SAS Help and Documentation. △

Supporting the Custom Action

Using the Custom action, a data integration developer can execute a macro in a macro autocall library in order to provide user-defined status handling for a job or transformation. On the administrative side, all you need to do is to make sure that the autocall library is known to the application server. You do this by editing the file *path-to-config-dir***Lev1\SASMain\sasv9.cfg**. By default, the file contains a line similar to this:

```
-sasautos ("SASEnvironment/sasMacro" SASAUTOS)
```

Add the library to the list of libraries in parentheses. The path to the library can be a full path or a path relative to *path-to-config-dir***Lev1\SASMain**.

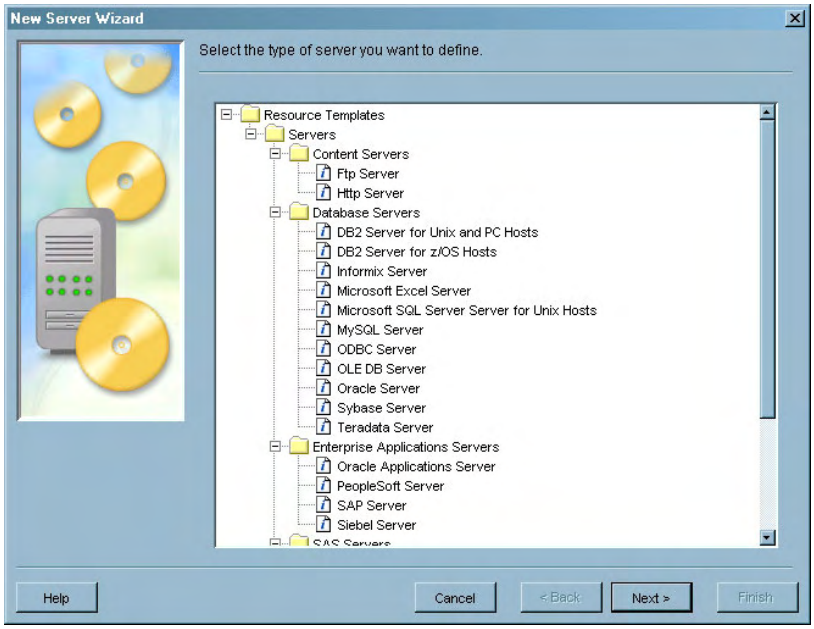
Enabling the External File Wizards to Retrieve Files Using FTP or HTTP

External files are often found on FTP or HTTP servers. In order to access these files, you must resolve the physical path to the server. You define an FTP or HTTP server with the Server Manager plug-in to SAS Management Console. To launch the New Server wizard from SAS Management Console, complete these steps:

1 Use your metadata profile to log on to the SAS Metadata Server that contains the SAS Metadata Repository connection that you want to use.

2 In the SAS Management Console navigation tree, select **Environment Management ▶ Server Manager**.

3 To display the first wizard window, select **Actions ▶ New Server**.



4 Select either the FTP or HTTP server type, then click **`Next`** to continue.

5 When defining the server in the New Server wizard, specify these properties:

Server type	Enter **`FTP`** or **`HTTP`**.
Name	Enter a name.
Software Version	Enter the software version.
Vendor	Enter the name of the vendor.
Application Server Type (HTTP only)	Enter the type of application server.
Base Path(s) (HTTP only)	Enter the full path to the server.
Authentication Domain	Enter your default authentication domain (DefaultAuth).
Application Protocol (HTTP only)	Enter either **`HTTP`** or **`HTTPS`**.

Host Name	Enter the host name of the machine on which the server runs.
Port Number	Enter 21 (80 for HTTP).
Proxy URL (optional, HTTP only)	Secure Socket Layer (SSL) support is necessary to use a proxy server. You need to download the SAS Secure Software package from **`http://www.sas.com/apps/demosdownloads/setupintro.jsp?listing=cat=none.`**

For more information about how to define a server, see the *SAS Management Console: User's Guide*.

CHAPTER

14

Administering SAS Information Map Studio

Introduction to Administering SAS Information Map Studio

This chapter explains how to perform the administrative tasks that are specific to SAS Information Map Studio. The following table outlines those administrative tasks.

Table 14.1 Administrative Tasks for SAS Information Map Studio

Administrative Task	Purpose of Task
Managing storage of information maps	Simplifies the process of managing access controls for information maps and ensures that information maps are available to SAS Web Report Studio.
Adding content for use by information map creators	Makes tables, SAS OLAP cubes, stored processes, and imported information maps available to users of SAS Information Map Studio.
Controlling access to information maps	Restricts access to information maps in accordance with your security goals.

Administrative Task	Purpose of Task
Setting up users for SAS Information Map Studio	Enables users to log on to SAS Information Map Studio, and ensures that each client is using appropriate encryption.
Understanding your SAS Information Map Studio configuration	Enables you to troubleshoot, modify, or update a user's configuration.

Managing Storage for Information Maps

Proper storage of information maps is important for these reasons:

- If you organize storage of information maps appropriately for your environment, then the management of access controls will be easier.
- If you do not store an information map in an appropriate location within the repository, then that information map will not be available as a data source for SAS Web Report Studio.
- If you change the storage location of an information map, then any reports that were created using the information map in its original location will fail.

Information maps are stored only in a SAS Metadata Repository; there are no corresponding entities for information maps in the file system or in a WebDAV content server. To ensure that each information map can be saved to an appropriate folder in the metadata repository, complete these steps:

1. Verify that the top-level storage container for information maps exists within the report storage structure in the metadata repository.
   a. Log on to SAS Management Console.
   b. Select the foundation repository.
   c. Navigate to **Environment Management ► BI Manager ► BIP Tree ► ReportStudio**.

      Note: You might have stored information maps in other locations as well. The documentation specifies the ReportStudio location because that is where SAS Web Report Studio expects to find information maps. △

   d. Verify that a **`Maps`** folder exists beneath the **`ReportStudio`** folder. "Standard Storage Containers for Reporting" on page 297 documents all of the folders in the reporting storage structure and explains how they are created.
2. If you want to control access to sets of information maps, rather than to each individual information map, then create subfolders beneath the **`Maps`** folder. For instructions, see "Adding Folders to Your Report Storage Structure" on page 300.

   For example, you might add a subfolder for Human Resources information maps. You can then give a Human Resources user group exclusive access to the Human Resources information map folder. Those access controls will be inherited by all of the information maps within that folder.
3. Explain to users of SAS Information Map Studio that any information map that will be used in SAS Web Report Studio must meet these requirements:
   - The information map must be stored in the **`Maps`** folder or in one of the subfolders of the **`Maps`** folder.

     CAUTION:
     Only those information maps that are stored in the Maps folder (or in a subfolder of the Maps folder) can be used in SAS Web Report Studio. △

- The storage location of the information map must not change. When a report is created using an information map, the report definition specifies the location of the information map. If the information map or the folder that contains the information map is moved or renamed, the report definition cannot reference the information map, and the report cannot be rendered.

CAUTION:
Deleting, moving, or renaming folders that contain information maps can break metadata associations and can result in reports that cannot be rendered. △

Adding Content for Use by Information Map Creators

The resources that can be used as inputs to SAS Information Map Studio are data sources, stored processes, and imported information maps. SAS Information Map Studio uses the metadata server to access these resources, so these resources must be registered in the metadata repository.

Making Data Sources Available to SAS Information Map Studio

Each information map must have at least one data source. The data sources for an information map can consist of *either* of these things:

- one or more tables that are registered in the metadata repository and are associated with a library definition. The tables can be any combination of SAS data sets and third-party relational database tables.
- one SAS OLAP cube that is registered in the metadata repository and is associated with a SAS OLAP schema definition.

Users of SAS Information Map Studio must have appropriate permissions to access the data sources that they will include in their information maps. For example, in the metadata authorization layer, the following requirements apply:

- In order to include a particular data source in an information map, a user must have ReadMetadata permission for that data source.
- In order to test a query against a SAS OLAP data source, a user must have both ReadMetadata and Read permissions for that data source.
- In order to test a query against a table data source that is accessed with the metadata LIBNAME engine, a user must also have Read permission for the data source.

For information about registering data sources in the metadata repository, see Chapter 9, “Connecting to Common Data Sources,” on page 167.

Making Stored Processes Available to SAS Information Map Studio

Each information map can include no more than one stored process. A stored process that is associated with an information map is executed before any queries that are generated against that information map. This processing order enables you to use SAS tools such as the DATA step or the macro language to process the data that will be used as input for the information map. Preprocessing data often involves subsetting or updating the data on a per-user basis (for example, when a user is prompted to enter parameters). For this reason, it is often helpful to use the WORK library of the workspace server to store temporary copies of data that has been preprocessed for a particular user.

To make a stored process available to users of SAS Information Map Studio, follow these steps:

1. Write and execute a SAS program that modifies existing data and creates custom data sets. Store the custom data sets in a permanent library. The custom data sets must contain header information about the columns and their attributes; rows are not required.
2. Use either SAS Management Console, SAS Data Integration Studio, or the metadata LIBNAME engine to register the custom data sets and their associated library in the metadata.
3. Insert a *PROCESSBODY statement as the first line of the program.
4. Insert a LIBNAME statement that references the location of the source data sets. The stored process will read the data from this permanent location.
5. Insert another LIBNAME statement that references the Work library of the workspace server. The stored process will write temporary, user-specific copies of data to this location. For example:

   ```
   *processbody;
   libname custom (work); /* Custom library and datasets must already be */
                          /* registered in metadata */
   libname source 'path-to-source-data';

   data custom.result_set;
    set source.data1;
     /* more code */
   run;
   ```

   Note: Using the Work library prevents the original data from being overwritten and enables each concurrent user of a stored process to have access to a temporary, private version of the resulting data. △
6. Save the program.
7. Register the program in the metadata as a stored process (using either SAS Management Console or SAS Enterprise Guide). Specify execution on a workspace server and an output type of NONE.

   Note: The stored process must run on a *workspace server*; a stored process that executes on a stored process server cannot be used by an information map. △
8. In SAS Management Console, navigate to the stored process and verify that your SAS Information Map Studio users have ReadMetadata access to the stored process.

A user of SAS Information Map Studio can now associate the stored process with an information map by completing these steps:

1. Create an information map that includes the data in the custom library.
2. From the menu bar, select **Tools ▸ Stored Processes**, and then select the stored process.
3. Save the information map. The stored process is now associated with the information map.

Note: To learn more about stored processes, see "SAS Stored Processes" in *SAS Integration Technologies: Developer's Guide* at **support.sas.com/rnd/itech/doc9/dev_guide/stprocess/**. △

Making User-Defined Formats Available to SAS Information Map Studio

You must import any user-defined formats that have been created for a data set in order to view the data. More information is provided at "Establishing Connectivity to a Library of SAS Data Sets" on page 168.

Importing Information Maps

Importing an information map is the process of retrieving the XML file that defines an information map from an external location and then adding that XML file to the current metadata repository as an imported information map. If the information map that you want to import is stored in another SAS Metadata Repository, then you must begin by exporting the information map from the source repository.

Note: When you export an information map, the associated access control settings are not exported △

Before you import an information map into a target repository, you should ensure that all of the data sources and stored processes that the information map uses are registered and available in the target metadata repository. To minimize the amount of XML editing that you must do after you import an information map, follow these guidelines when you register the stored process and the data sources in the target metadata repository:

- Use names and labels that are identical to the names and labels that are used in the source metadata repository.
- Use folders and paths that are identical to the folders and paths that are used in the source metadata repository.

The Help for SAS Information Map Studio provides instructions for importing and exporting information maps.

Considerations for Setting Up Access Control to Information Maps

Information maps should be secured because they reference physical data sources. Information maps are stored only in the metadata, so you must use the metadata authorization layer to manage access to these resources. When you work with metadata access controls for information maps, consider these things:

- The only relevant permissions for information maps and information map folders are Read, ReadMetadata, and WriteMetadata. The Write, Create, Delete, CheckInMetadata, and Administer permissions have no effect on these objects. For more information, see "Which Actions are Controlled by Each Permission?" in the "Understanding Authorization" section in the *SAS Intelligence Platform: Security Administration Guide*.

  Note: Beginning with Service Pack 4, Read permission for an information map is required in order to access data through that information map. For details, see "Access Requirements for Information Maps" in the chapter "Access Guidelines and Requirements" in the *SAS Intelligence Platform: Security Administration Guide*. △

- The effective permissions for an information map folder are inherited by all of the information maps within that folder. For more information, see "Where Can

Permissions Be Set?" in the "Understand Authorization" section in the *SAS Intelligence Platform: Security Administration Guide*.

- The ability to view or work with an information map can be affected by access to each of the information map's underlying components. For information about the metadata layer requirements for working with information maps and information map folders, see "Access Requirements for Information Maps" in the chapter "Access Guidelines and Requirements" in the *SAS Intelligence Platform: Security Administration Guide*.
- Administrators can use SAS Management Console to set access controls. To access the **`Authorization`** tab for an information map folder or an individual information map, navigate in SAS Management Console under **Environment Management ► BI Manager ► BIP Tree ► ReportStudio**.

  Note: You might have stored information maps in other locations as well. The documentation specifies the ReportStudio location because that is where SAS Web Report Studio expects to find information maps. △
- Information map creators can use SAS Information Map Studio to set permissions on individual information maps. Any user who has WriteMetadata access to an information map can change the access controls for that information map.
- Information map creators can also use SAS Information Map Studio to move an information map from one folder to another. This can affect access to the information map because information maps inherit access controls from their parent folders. A user can move an information map from one folder to another folder if the user has WriteMetadata permission to the information map and to both of the folders.

Setting Up Users for SAS Information Map Studio

The process for adding users to a deployment is documented in "How to Add a User" in the chapter "User and Group Management" in *SAS Intelligence Platform: Security Administration Guide*. These aspects of setting up users are specific to SAS Information Map Studio:

- Users cannot log on to SAS Information Map Studio as PUBLIC. In order to log on to SAS Information Map Studio, a user must have an individual metadata identity. If the user's only metadata identity is as a member of the PUBLIC group, then the user cannot log on to SAS Information Map Studio. For more information, see "Authentication Overview" in the chapter "Understanding Authentication in the SAS Intelligence Platform" in *SAS Intelligence Platform: Security Administration Guide*.
- SAS Information Map Studio can cache and reuse the credentials that a user supplies in an interactive log on. This can reduce (or in some cases eliminate) the need to store the user's password in the metadata repository. For more information, see "Reuse of Credentials That are Cached from an Interactive Log" in the chapter "Understanding Authentication in the SAS Intelligence Platform" in the *SAS Intelligence Platform: Security Administration Guide*.
- Every time you install the SAS Information Map Studio client, you must implement encryption for that client in accordance with the encryption that is used throughout your deployment. For instructions, see "Enabling Encryption" in the chapter "Securing a Deployment" in the *SAS Intelligence Platform: Security Administration Guide*.

Understanding the SAS Information Map Studio Configuration

The administrative and configuration files for SAS Information Map Studio are located on each client machine.

The following administrative and configuration files are included:

`IMSApplicationProperties.properties`
Contains user preferences and configuration settings for SAS Information Map Studio. This file is located in each user's Windows Profile directory. Do not edit this file directly. Instead, use the Options dialog box in SAS Information Map Studio to make changes.

`mapstudio.ini`
Contains configuration settings for SAS Information Map Studio. For example, you can change the amount of available memory by adding or modifying arguments in this file. In general, you should modify this file only if SAS Technical Support directs you to do so. The default location for this file is the installation directory for SAS Information Map Studio.

`IMS_Log.log`
Contains logging information for SAS Information Map Studio. The default location for this file is the user's Windows Profile directory.

By default, this file contains error information only. You can use the Diagnostic Settings dialog box within SAS Information Map Studio to make changes.

Note: This log can get large very quickly (depending on the logging level that you select). △

CHAPTER

15

Administering SAS Web Report Studio

Introduction to Administering SAS Web Report Studio

This chapter explains how to perform the administrative tasks that are specific to SAS Web Report Studio. The following table outlines those administrative tasks.

Table 15.1 Administrative Tasks for SAS Web Report Studio

Administrative Task	Purpose of Task
Setting up storage of reports and report-related objects	Ensures that resources are stored in appropriate locations and simplifies the process of managing access controls.
Adding content for use by report creators	Makes data sources (information maps), stored processes, images, fonts, and imported reports available to users of SAS Web Report Studio.
Setting up users for SAS Web Report Studio	Enables users to log on to SAS Web Report Studio, ensures that users are assigned to appropriate roles, and explains how you can modify the authentication process.
Managing access to reports	Restricts access to reports in accordance with your security goals.
Configuring the SAS Web Report Studio log files	Helps you to track and audit user actions for performance and security reasons.

Administrative Task	Purpose of Task
Improving performance of SAS Web Report Studio	Helps you to manage memory, take advantage of server pooling capabilities, and understand the performance trade-offs between content servers.
Understanding your SAS Web Report Studio configuration	Helps you to troubleshoot, modify, or update the configuration files and the settings in the BI Manager and Web Report Studio plug-ins to SAS Management Console.
Customizing Reports	Enables you to add disclaimer text to reports and to add custom report styles.
Setting up batch processing and scheduling of reports	Enables you to pre-generate reports so they will render quickly.

Setting Up Storage for Reporting

Proper storage of reports and report-related objects is important for these reasons:

- Storage of reports (and some report-related objects) must always be synchronized between your metadata repository and your external content server.
- Moving information maps or stored processes from one location to another breaks any reports that are based on those objects in their original locations.
- Information maps and stored processes must be registered in a particular location in the metadata repository in order to be available to users of SAS Web Report Studio.
- If you organize storage of reports appropriately for your environment, then controlling access to reports will be easier.

Standard Storage Containers for Reporting

These are the standard storage containers for reports and report-related objects:

`ReportStudio`
The top-level storage container for the reporting environment.

`ReportStudio/BannerImages`
The location where SAS Web Report Studio looks for banner images when building a report.

`ReportStudio/ConditionalHighlightingImages`
The location where SAS Web Report Studio looks for conditional highlighting images when building a report.

`ReportStudio/Maps`
The top-level container for information maps. SAS Web Report Studio users can access only those information maps that are stored in the **`Maps`** folder or in a subfolder of the **`Maps`** folder.

`ReportStudio/Shared`
The top-level container for report-related objects that will be accessed by multiple users. Although saving reports in this container is an easy way to make the reports widely available, this practice can create clutter that is difficult to secure and to navigate.

ReportStudio/Shared/Images
The top-level container for images that can be included in reports (other than banner images and conditional highlighting images).

ReportStudio/Shared/Reports
The top-level container for reports that will be accessed by multiple users. It is recommended that you store reports in subfolders beneath this container.

ReportStudio/Shared/Reports/StoredProcesses
The initial location where SAS Web Report Studio looks for stored processes when you attempt to insert a stored process section into a report.

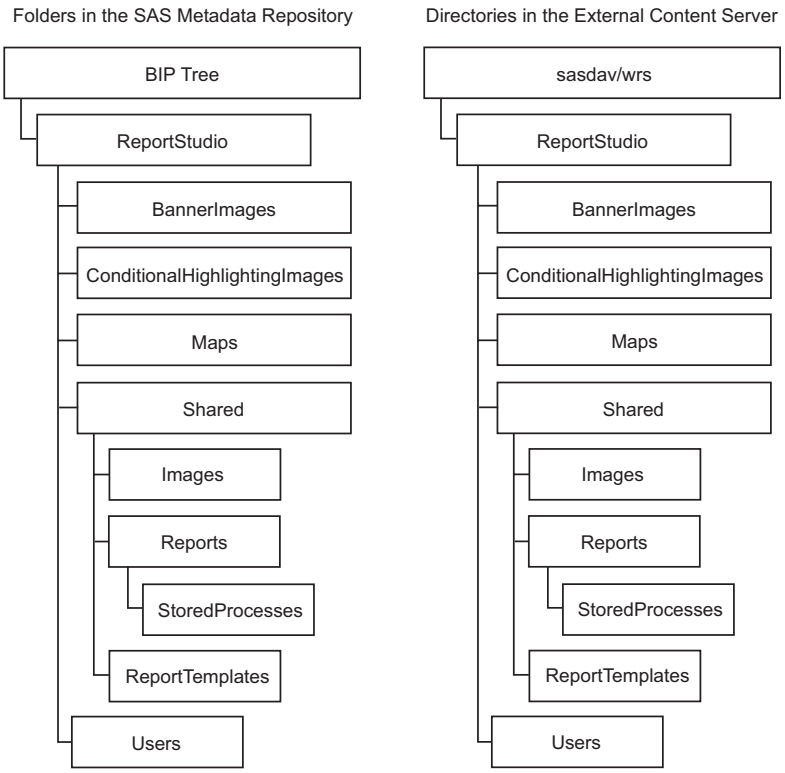
ReportStudio/Shared/ReportTemplates
The container for the templates that are used when you create reports with custom layouts in SAS Web Report Studio. This folder is created automatically when templates are used.

ReportStudio/Users
The storage container for the personal folders in which each user can store private reports. A personal folder is created for each user when the user first logs on to SAS Web Report Studio. By default, only the user who is associated with a folder can read and write metadata in that folder. These default permissions are set only when a user folder is created automatically. For this reason, you should not manually create user folders.

Note: This **Users** folder is used only by SAS Web Report Studio. The SAS Information Delivery Portal has its own **Users** folder in a different location. △

This set of storage containers for SAS Web Report Studio must exist in both the foundation SAS Metadata Repository *and* the external content server (third-party WebDAV server or file system). The following figure depicts the parallel storage structures that are required by SAS Web Report Studio. The figure also depicts the relationship between the top-level report storage containers. The **BIP Tree** folder in the metadata repository corresponds to the **sasdav/wrs** directory in the content server. These top level containers should be created during the installation process.

Figure 15.1 Parallel Structure for Report Storage Containers



The parallel storage structures are necessary because reports and some report-related objects (such as images) have both a metadata component and a content component. For example, for each report that is saved in SAS Web Report Studio, two objects are stored:

- A metadata object that describes the report is stored in your metadata repository. The report metadata object contains information such as time stamps, authorship, access controls that provide security for the report, and other report- and application-specific properties.
- A report definition file is stored in your content server. The report definition file is an XML file that contains information about how the report is presented and what data is included in the report. The report definition is constructed according to the SAS Report Model, which is an XML specification for business reports. Reports that comply with the SAS Report Model can be created, viewed, and modified by a variety of SAS Business Intelligence applications.

In order to display a report, SAS Web Report Studio must retrieve both of these components. The parallel storage structures in the metadata repository and the content server facilitate this two-part retrieval.

CAUTION:

You must keep the report content files synchronized with their corresponding metadata objects. Using SAS applications to manage your reports preserves the synchronization. Using the external content server or interacting directly with the report content can leave the corresponding report metadata objects in an inconsistent state. For example, using Xythos WebFile Server to delete or rename files or directories will break the synchronization. Similarly, using a text editor to make changes to a report definition XML file without also updating the corresponding report metadata object will result in a nonfunctional report. △

During the first use of SAS Web Report Studio, all of the standard storage containers are created for you in both the metadata repository and your external content server. These containers should be in place at the end of the installation process, because the installation includes a verification step in which someone logs on to SAS Web Report Studio. For each additional report creator who logs on to SAS Web Report Studio, one additional personal report storage container is added under the **Users** container.

If your deployment does not include SAS Web Report Studio, then you must use the BI Manager plug-in to create all of your report storage containers. For instructions, see "Adding Folders to Your Report Storage Structure" on page 300.

Verifying Your Reporting Storage Structure

To view the report storage containers in your metadata repository, complete these steps:

1. Log on to SAS Management Console and access the foundation repository. It is recommended that you log on as an unrestricted user such as the SAS Administrator (sasadm) for this task. This ensures that you will not be prevented from seeing a folder because of access controls in the metadata layer.
2. Navigate to **Environment Management ▶ BI Manager ▶ BIP Tree ▶ ReportStudio**.

   Note: If the BI Manager is not listed under Environment Management, then this plug-in has not been installed on the computer where you are working. The BI Manager plug-in is added to SAS Management Console when you install SAS Foundation Services. △
3. Verify that the **BannerImages**, **Maps**, **Shared**, and **Users** folders exist beneath the **ReportStudio** folder. Also verify that the subfolders inside the **Shared** folder exist.

To verify that the same set of storage containers exists in your external content server, complete these steps:

1. In SAS Management Console, navigate to **Environment Management ▶ BI Manager ▶ BIP Tree ▶ Properties** to see the content server and content path that your deployment of SAS Web Report Studio has been configured to use.
2. In your external content server, navigate to the content path that is specified in metadata for the BIP Tree, and verify that the **BannerImages**, **Maps**, **Shared**, and **Users** directories exist under a **ReportStudio** directory.

   For example, if you are using Xythos WebFile Server as your content server and you used the standard ports and names, then you would open a Web browser to **<machine-name>:8300/sasdav/wrs** and log on as the xythos administrator.

Adding Folders to Your Report Storage Structure

In the metadata repository, each report, information map, and stored process inherits permissions from the folder in which it is stored. For example, the access controls that you set on a report folder are inherited by all of the reports within that folder. It is easier to manage permissions on folders than on individual resources, so you might want to define additional subfolders within the standard structure. For example, within the **ReportStudio/Shared/Reports** container you could create separate report subfolders for Human Resources, Finance, and Sales. Then, within the **Sales** container, you could create an additional subfolder for each sales region.

Each new subfolder inherits the effective access controls of its parent folder. You can change the permission settings for any folder by navigating to the folder in SAS

Management Console under **Environment Management ► Authorization Manager ► Resource Management ► By Application ► BIP Tree ► ReportStudio** and accessing the folder's **`Authorization`** tab. This structure enables you to store reports that everyone will access in the **`ReportStudio/Shared/Reports`** folder and to limit access to each of the subfolders so that only certain user groups can see or modify the folder and its contents.

You should use the BI Manager in SAS Management Console to add folders to the report storage structure. For each folder that you create using the BI Manager, a corresponding directory is also automatically created in your content server. In this way, the BI Manager preserves the necessary synchronization between the folders in the metadata repository and the content server.

SAS Web Report Studio and SAS Information Map Studio also enable users to add folders. However, users cannot set permissions on folders from within these applications. You can prevent users from creating subfolders by denying WriteMetadata permission to the parent folder. However, this will prevent users from adding resources to the parent folder. The inherited denial of WriteMetadata permission from the parent folder will also prevent users from modifying or deleting the resources in that folder (unless you set a direct grant of WriteMetadata permission on each resource).

Adding Content for Use by Report Creators

The resources that can be used as inputs to SAS Web Report Studio are information maps, stored processes, banner images, fonts, and existing reports from other locations. SAS Web Report Studio uses the metadata server to access these resources, so these resources must be registered in the foundation metadata repository.

Making Data Sources Available to SAS Web Report Studio

In SAS Web Report Studio, users do not interact directly with SAS data sets, relational database tables, or SAS OLAP cubes. Instead, users interact with information maps that provide a business view of the underlying data. In SAS Web Report Studio, the term "data source" refers to an information map. Each report that is created in SAS Web Report Studio can use no more than one information map.

In order for an information map to appear in the list of data sources in SAS Web Report Studio, the information map must meet all of these criteria:

- The information map must exist in the same foundation repository that the user of SAS Web Report Studio is accessing.
- The information map must be stored in the main **`Maps`** folder or in one of the subfolders of that folder.
- The user of SAS Web Report Studio must have both Read and ReadMetadata permission to the information map.

  Note: Beginning with Service Pack 4, Read permission for an information map is required in order to access data through that information map. For details, see the "Access Management" chapter of the *SAS Intelligence Platform: Security Administration Guide* document. △

Making Stored Processes Available to SAS Web Report Studio

Including a stored process in a report section is one of the ways in which to obtain data for the report. Each report section can contain multiple stored processes. When

the report section is rendered, the output of each included stored process is displayed. Users can also run stored processes directly from within SAS Web Report Studio. The output of the stored process is rendered in the user's Web browser.

You cannot use SAS Web Report Studio to modify the query that is generated from the stored process, but you can use SAS Web Report Studio to add layout elements such as headers, footers, images, and text that are independent of the stored process output.

Note: Stored process reports that were created by using SAS Enterprise Guide do not support any layout design. △

You can convert existing SAS programs into stored processes for use in SAS Web Report Studio. The programs can be parameterized, which enables users to input data in response to prompts. Prompted parameter values are transferred to the stored process as macro variables. To convert an existing program to a stored process, complete these steps:

1. Insert a *PROCESSBODY statement.
2. Insert a %STPBEGIN statement prior to a section of the code that produces output.
3. Insert a %STPEND statement after a section of the code that produces output.

For example, to alter this SAS program:

```
%let year=2002;
title "Sports & Outdoors Sales &year";
proc print data=sashelp.orsales;
                where year=&year;
run;
```

to become a stored process, change the code to look like this:

```
%global year;
*processbody;
%stpbegin;

title "Sports & Outdoors Sales &year";
proc print data=sashelp.orsales;
                where year=&year; /* &year is a parameter from a user prompt */
run;

%stpend;
```

Stored process output that will be included in a report must be generated through the Output Delivery System (ODS). Output that is generated in other ways, such as with PUT statements, is not accessible from SAS Web Report Studio. The %STPBEGIN and %STPEND macros in the stored process code ensure that ODS is used to generate the output.

The ODS output type for each stored process is determined by the manner in which the stored process is registered and executed. The ODS output type cannot be controlled by making changes to the stored process code (neither by setting the value of the stored process input parameter _RESULT, nor by explicit ODS statements). The following table indicates how the style is determined for stored process output.

Table 15.2 Style of Stored Process Output

Type of Output	Example	Style of Output
ODS text output	PROC PRINT listing	The style is determined by the user's preferences in SAS Web Report Studio. For example, "Seaside."
ODS graphical output	PROC GCHART graphs	The default is an ActiveX device.* **`GOPTIONS DEVICE=ACTIVEX;`**

* By default, the ACTIVEX device driver is used for graphs in stored process output. This format requires users to install a graph control on their local system in order to render the graph. However, to maintain a zero footprint on the client, SAS Web Report Studio does not require this installation. Therefore, when the stored process is run in SAS Web Report Studio, the ACTXIMG device driver is substituted so that a static image is created. Similarly, if the JAVA device driver is specified, then the JAVAIMG device driver will be substituted automatically.

To make a stored process available to users of SAS Web Report Studio, complete these steps:

1. Register the stored process in the metadata by using either SAS Management Console or SAS Enterprise Guide. The stored process must be registered in the foundation repository that will contain the reports that use the stored process. Within that repository, the stored process must be registered in the **`BIP Tree/ReportStudio/`** folder structure.
2. In SAS Management Console, navigate to the stored process under **Environment Management ▶ Authorization Manager ▶ Resource Management ▶ By Application ▶ BIP Service ▶ BIP Tree ▶ ReportStudio**. On the **`Authorization`** tab for the stored process, verify that your SAS Web Report Studio users have ReadMetadata access to the stored process.

Note: To learn more about stored processes, see "SAS Stored Processes" in *SAS Integration Technologies: Developer's Guide* at **`support.sas.com/rnd/itech/doc9/dev_guide/stprocess/`**. △

Making Images Available to SAS Web Report Studio

Each report that is created in SAS Web Report Studio can include one or more images. The types of images that report creators can use are described in the following table.

Table 15.3 Images for SAS Web Report Studio

Type of Image	Details and Defaults
Banner images	Any report can include a banner image in the header and footer of the report. Banner images make it easier for report consumers to identify the report and to distinguish the report from other reports. Banner images are stored in **`ReportStudio/BannerImages`**. By default, the **`BannerImages`** folder is empty. You should use the BI Manager in SAS Management Console to manage your banner images.
Conditional highlighting images	A report that includes tables can use images to draw attention to items that might be of particular interest to report consumers. A report creator can define conditions and, for each condition, select an image that will be displayed in every table cell where the condition is met. Conditional highlighting images are stored in **`ReportStudio/ConditionalHighlightingImages`**. You should use the BI Manager in SAS Management Console to manage your conditional highlighting images. (The initial set of images do not appear in the **`ConditionalHighlightingImages`** folder until after the Conditional Highlighting dialog box has been accessed from within SAS Web Report Studio.)
Other images	Any report can include additional images for decorative or other purposes. These images are stored under the **`ReportStudio/Shared/Images`** folder. By default, the **`Images`** folder is empty. You should use SAS Web Report Studio to add images to this folder. For instructions, see the Help for SAS Web Report Studio.

To make a banner image or conditional highlighting image available to users of SAS Web Report Studio, complete these steps:

1. If you are using a WebDAV server, make sure that the server is running.
2. In SAS Management Console, navigate to the appropriate images folder beneath **Environment Management ▶ BI Manager ▶ BIP Tree ▶ ReportStudio**.
3. From the menu bar, select **Actions ▶ Add Content From External File**.
4. From the Specify a Source File dialog box, select the file (or files) that you want to import and then click **`Open`**.

   Note: If you select a folder, the folder and its contents are recursively imported. However, SAS Web Report Studio does not detect banner images or conditional highlighting images that are stored in sub-folders of the **`BannerImages`** and **`ConditionalHighlightingImages`** folders. △

5. In the **`Enter description`** text box, enter the description that you want to be displayed for the graphic in SAS Web Report Studio. Image descriptions should be fewer than 20 characters.
6. Click **`OK`** to close the Enter Description text box.
7. The imported images are available in SAS Web Report Studio within 10 minutes. To makes the images available immediately, restart the Web application server.

If an existing image is later modified, you can reimport the new image by using the previous instructions. SAS Web Report Studio will detect and use the updated image.

To delete an image so it is no longer available to users of SAS Web Report Studio, complete these steps:

1. In SAS Management Console, navigate under **Environment Management ▶ BI Manager ▶ BIP Tree ▶ ReportStudio** and select the image that you want to delete.
2. From the menu bar, select **Edit ▶ Delete**.

Note: The minimum screen resolution that is supported for clients (browsers) is1024 x 768. △

Making Fonts Available to SAS Web Report Studio

You can customize the fonts that are available for tables and graphs in the report. SAS Web Report Studio (and SAS Web Report Viewer, if it is installed) uses the default fonts that are loaded from the following files:

- The **`ServerFonts.xml`** file lists fonts that are rendered on the server. These are the fonts that are available for graphs in a report. The fonts that are listed in this file should be installed on the middle-tier server where SAS Web Report Studio is deployed.
- The **`ClientFonts.xml`** file lists the fonts that are rendered on the client (user's) system. These fonts are available for tables, headers, and other text. These fonts should be installed on the client system where the browser is running.

To supply additional fonts, edit the following files:

- **`LocalServerFonts.xml`**
- **`LocalClientFonts.xml`**

You can create these files from the **`LocalServerFonts.xml.sample`** and **`LocalClientFonts.xml.sample`** files that reside in the **`customer`** subdirectory of your SAS Web Report Studio installation. To create the files:

1. Open **`LocalServerFonts.xml.sample`** and save it using the name **`LocalServerFonts.xml`**.
2. Open **`LocalClientFonts.xml.sample`** and save it using the name **`LocalClientFonts.xml`**.

Each sample file contains information on adding fonts. Here is the general format for the font information in the file:

```
<?xml version="1.0" encoding="UTF-8"?>
  <fonts>
    <font actualfont="Arial" displayfont="Arial" />
    <!-- more fonts -->
  </fonts>
```

In the previous code sample, the **`actualfont`** attribute is the font name that is stored in the report. The value for this attribute should match the name of the font on the system. If they differ, a font substitution can occur. The **`displayfont`** attribute is the font name that is displayed to users.

Note: XML tags, such as **`<fonts>`**, are case-sensitive, and should be specified exactly as shown. △

If you plan to use SAS Web Report Viewer to render reports, create and edit a **`LocalServerFonts.xml`** and **`LocalClientFonts.xml`** file for SAS Web Report Viewer in a similar way. Make a copy of each respective sample file in the SAS Web Report Viewer **`customer`** subdirectory, and add your fonts to the copy.

SAS Web Report Studio (and SAS Web Report Viewer, if applicable) must be reconfigured and redeployed after the custom font files are created or modified. For details, see "Redeploy SAS Web Report Studio" on page 337.

Importing Reports that Conform to the SAS Report Model

In addition to enabling users to create new reports, SAS Web Report Studio enables users to work with reports that were created elsewhere. Importing a report is the process of retrieving the XML file that defines a report and then adding that report to your report storage structure. The retrieved XML file is written to the appropriate directory within your content server, and a corresponding metadata object is created and stored in a parallel location in the metadata repository.

A report that you import into a new metadata repository will render properly only if all of the report's underlying components (such as an information map, a stored process, and the data sources) are available in the appropriate locations in the new repository's report storage structure. To import a report, complete these steps:

1. Log on to SAS Management Console with a metadata profile that connects to the metadata server into which you will import the report.
2. Navigate to the BI Manager and select the folder into which you will import the report.
3. From the menu bar, select **Actions ► Import**.
4. Select the report to import.
5. Select **`Open`**.

Importing Legacy Reports

You can use the Output Delivery System (ODS) to make legacy SAS reports available in a SAS Intelligence environment. For example, you might have a collection of legacy reports that were created using a SAS program editor, SAS Enterprise Guide, or SAS IntrNet. You can use ODS to write those reports directly to the **`ReportStudio`** storage structure. SAS Web Report Studio treats the ODS output as a report, allowing a user to display, move, rename, and delete the output as with any other report. However, this type of report cannot be edited from SAS Web Report Studio.

To write ODS output directly to the report storage structure, use the SAS Report XML tag set and the SASXPRRP access method on the FILENAME statement. When you use the SASXPRRP access method on the FILENAME statement, a SAS Business Intelligence Protocol (SBIP)* URL identifies the external file that you want to write to. If your process generates multiple files in the same location, the SBIP URL should refer to a directory rather than to a specific file. A trailing slash in the SBIP URL is required when specifying a directory. If the specified file or directory already exists, it is overwritten.

The following options to the SASXPRRP access method are required unless otherwise indicated:

USERID="*user ID*"
: The userID to access the server.

PASSWORD="*password*"
: The password to access the server.

DOMAIN="*domain*"
: The domain name for the server.

OMRHOST="*host*"

* SBIP is a proprietary protocol for specifying the location of resources in a SAS Metadata Repository. For example, this path **`SBIP://Foundation/BIP Tree/ReportStudio/Shared/Reports/MyReport.srx`** specifies the location of a report named **`MyReport`** within a repository named **`Foundation`**.

The network name of the machine hosting the metadata repository.

OMRPORT="*nnnn*"
The port number for accessing the repository.

OMRUSER="*user ID*"
The user ID to access the repository. This can be the same as the server user ID, or it can be different.

OMRPASSWORD="*password*"
The password to access the repository. This can be the same as the server password, or it can be different.

OMRREPOSNAME="*name*"
The name of the repository.

For example, the following code outputs SAS Report XML to the specified report storage container:

```
filename dest sasxprrp "SBIP://RepName/Bip Tree/ReportStudio/Users/xyz/Reports"
   userid="xyz" password="bip2004" domain="thisDomain"
   OMRHost="bipsvrxyz.na.sas.com" OMRPort="9999" OMRUser="xyz"
   OMRPassword="bip2004" OMRReposName="RepName"
   ;

option noovp;
ods sasreport file="myreport.xml" path=dest;
proc print data=sashelp.class;
run;
ods sasreport close;
```

Setting up Users for SAS Web Report Studio

The process for adding users to a deployment is documented in "How to Add a User" in the "User and Group Management" section in *SAS Intelligence Platform: Security Administration Guide*. These are the aspects of setting up users that are specific to SAS Web Report Studio:

- It is recommended that you designate a metadata identity to serve as a surrogate for SAS Web Report Studio users who do not have their own metadata identities. *This is a requirement if you are using Web authentication.*
- You can use roles to control access to SAS Web Report Studio functionality.
- You can configure SAS Web Report Studio to authenticate users on the Web application server.

The following sections provide instructions and details for each of these topics.

Designate a Surrogate Metadata Identity

By default, SAS Web Report Studio users who do not have their own individual metadata identities use the PUBLIC group's metadata identity. It is recommended that you instead use the SAS Guest User metadata identity as a surrogate for these public-only users. This is a requirement if you are using Web authentication.

To designate the SAS Guest User as a surrogate identity for public-only users of SAS Web Report Studio, complete these steps:

1. Open the **`LocalProperties.xml`** file. For instructions on creating a **`LocalProperties.xml`** file, see "Create a LocalProperties.xml File" on page 325.
2. In the wrs.pfs section, edit the properties to specify these values:
   - Set the allowPublicUsers property to **`true`**.
   - Set the publicUserSurrogate.activate property to **`true`**.
   - Set the publicUserSurrogate.uid property to **`sasguest`**.
   - Set the publicUserSurrogate.pw property to the password that your site is using for the sasguest account.
   - Set the publicUserSurrogate.domain property to **`web`** (if you are using Web authentication) or **`DefaultAuth`** (if you are not using Web authentication).

   If your **`LocalProperties.xml`** file does not contain the wrs.pfs section, then you can add it to the file by copying the section block from the **`WebReportStudioProperties.xml.orig`** file.
3. Redeploy SAS Web Report Studio. For instructions, see "Redeploy SAS Web Report Studio" on page 337.

After you make these changes, the access controls and logins that are defined for the SAS Guest User are applied to every public-only user. For example, all of the resources that are available to the SAS Guest User are also available to all public-only users. The report folders that belong to public-only users function as shared folders; these folders are not protected.

Note: If you are not using Web authentication, you can choose to configure SAS Web Report Studio to require each user to have an individual metadata identity in order to log on. This is controlled by the wrs.pfs.allowPublicUsers property. △

SAS Web Report Studio Roles

Everyone who can log on can view reports. In addition, each user of SAS Web Report Studio can be assigned to one or more standard roles. A user's role assignments determine which SAS Web Report Studio menu items are available to that user. These roles facilitate administration in a couple of ways. First, by default, all users have full permissions. When you are ready to lock down your environment, you need only limit roles to particular groups of users that you have defined in metadata. In addition, there is a small number of roles for you to manage. Each role can be assigned multiple groups of users.

The roles correspond to user groups in metadata. You can use the User Manager in SAS Management Console to add or remove users and other groups. For instructions on using the User Manager, select User Manager in the SAS Management Console navigation pane and then select Help from the menu bar.

The roles are created for you during installation. If you need to recreate a role, then you must specify the exact role name as specified in your application properties files. This documentation assumes that the default role names are used.

The next table describes a group of roles that have the following characteristics:

- By default, all SAS Web Report Studio users implicitly have the role. However, if you explicitly assign any members to the role, then only the explicitly-assigned members will have the role. This enables you to start using SAS Web Report Studio immediately after installation, yet still have the ability to restrict user access when locking down your deployment.
- Each role is a superset of the preceding role. For example, members of the "WRS Report Author" role have all the permissions that apply to the "WRS Report Consumer".

- Once you explicitly assign members to a role, you must explicitly assign members to each superset role. For example, if you assign members to the "WRS Report Author" role, then all of the subsequent superset roles (in this example, "WRS Advanced User") also become explicitly-assigned roles. The reason is that WRS Advanced User is a superset of WRS Report Author.

Table 15.4 User Roles That Apply to All SAS Web Report Studio Users by Default

Role (Default Group Name in Metadata)	**Capabilities**
WRS Report Consumer	Users who have this role can view reports and manipulate report data in the View Report view. Users can copy, move, save, rename, or delete reports. Users cannot create new reports.
WRS Report Author *	In addition to the abilities assigned to WRS Report Consumers, users who have this role can create reports with the report builder or report wizard. Users can also schedule reports.
WRS Advanced User	In addition to the abilities assigned to WRS Report Authors, users who have this role can distribute reports. Users cannot create or delete recipient lists that are used for report distribution.

* By default, WRS Report Authors can schedule reports, though you can change the default behavior and limit the scheduling feature to WRS Advanced Users. To do this, in your **`LocalProperties.xml`** file, specify true for the **`schedulingRequiresAdvancedUserRole`** property. For instructions on creating a **`LocalProperties.xml`** file, see “Create a LocalProperties.xml File” on page 325.

The following table describes user roles that do not have any members by default. You must explicitly add members to these roles.

Table 15.5 User Roles That Have No Members by Default

Role (Default Group Name in Metadata)	Capabilities
WRS Administrator	Users who have this role can perform all tasks that are associated with SAS Web Report Studio, including the ability to create and delete recipient lists that are used for report distribution. This role provides full permissions to SAS Web Report Studio and should be safeguarded accordingly. This role provides application level administrator functionality. However, this role has no effect on metadata access (authorization) rights to report data.
WRS Prohibited	Users who have this role cannot log on to SAS Web Report Studio. Regardless of the user's membership in any of the previous roles, if the user attempts to log on, the logon page displays the following error message: "This user is not allowed to access SAS Web Report Studio. Please contact your administrator." Some organizations might apply this role for users who are allowed to access some SAS applications but not SAS Web Report Studio. Alternatively, if an organization has multiple Web Report Studio installations, this role can be used to restrict some users from specific instances.

Trusted Web Authentication

During a planned installation, SAS Web Report Studio is configured to use the metadata server's authentication provider to verify credentials that users submit. You might choose to change to Web authentication if you want to take advantage of user accounts that are already established with an authentication provider for a Web application server, or if you want to minimize the number of user accounts that you have to create on the metadata server.

Note: If you configure Web authentication for SAS Information Delivery Portal, then you should also configure Web authentication for SAS Web Report Studio. If you configure Web authentication for SAS Web Report Studio, then you must also designate a surrogate metadata identity for SAS Web Report Studio public-only users. See "Designate a Surrogate Metadata Identity" on page 307. △

For instructions on configuring Web authentication, see "Setting Up Web Authentication" in the *SAS Web Infrastructure Kit: Administrator's Guide* at **`http://support.sas.com/rnd/itech/library/toc_portaladmin.html`**.

For more information about authentication, see the *SAS Intelligence Platform: Security Administration Guide* document.

Additional Authentication for SAS Web Report Studio Users

In addition to accessing the SAS Metadata Server, users of SAS Web Report Studio often use one or more of the servers that are listed in this table:

Table 15.6 Additional Authentication for SAS Web Report Studio Users

Server	Interaction
SAS Workspace Server	Access resources such as tables.
SAS OLAP Server	Access cube data and process MDX queries.
SAS Stored Process Server	Execute stored processes and collect resulting output.

The authentication processes and requirements for these servers are documented in detail in "Understanding Authentication in the SAS Intelligence Platform" in the *SAS Intelligence Platform: Security Administration Guide*. Key points to consider for a default configuration of SAS Web Report Studio are:

- In the simplest case, all of the SAS servers use the same host authentication provider. For example, in a single-machine deployment with a default configuration, a user of SAS Web Report Studio needs only a local (or network) account in the operating system. Similarly, in a multi-machine deployment where all servers use the same host authentication provider, a user of SAS Web Report Studio needs only a network account with that host authentication provider.
- In a more diverse environment, additional accounts, logins, and authentication domains are required. For example, if your stored process server is running on UNIX and your other servers are using Windows host authentication, then each user also needs an (individual or shared) account on the UNIX server and an additional (individual or group) login in the metadata. The additional login must include the credentials for the UNIX account. The additional login must be associated with the stored process server's authentication domain.

Changes that you make to the default SAS Web Report Studio configuration can affect your user's ability to access servers. For example:

- If you modify your deployment to use pooled workspace servers, then it is no longer necessary for your users to have accounts or logins for the purpose of accessing the workspace server. Instead, each SAS Web Report Studio user must be a member of at least one group that is associated with a puddle within the pool. In the simplest case, the PUBLIC group is associated with the only puddle, so all users can access the pooled workspace servers. In a more restricted environment, the puddle might be associated with a user-defined group that includes only those users who should access the pooled workspace servers.
- If you modify your deployment to use Web authentication, then additional logins are required. For example, if you are using Web authentication with an LDAP authentication provider, you could choose to meet the additional authentication requirements as follows:
  - To enable users to access the SAS Workspace Server, use a pooled configuration (with a single puddle that is associated with the PUBLIC group).
  - To enable users to access the SAS OLAP Server, use the same LDAP authentication provider for the SAS OLAP Server as you are using for the Web application server.
  - To enable users to access the SAS Stored Process Server, give each user an individual or shared account in the host operating system and an additional individual or group login in the metadata. The additional login must include the credentials for the operating system account. The additional login must be associated with the stored process server's authentication domain.

Note: In all of these scenarios, if you set up a surrogate public user, then the surrogate user's logins are available to users who do not have their own metadata identities. △

Managing Access to Reports

The following table summarizes the basic security considerations for reports.

Table 15.7 Report Security Considerations

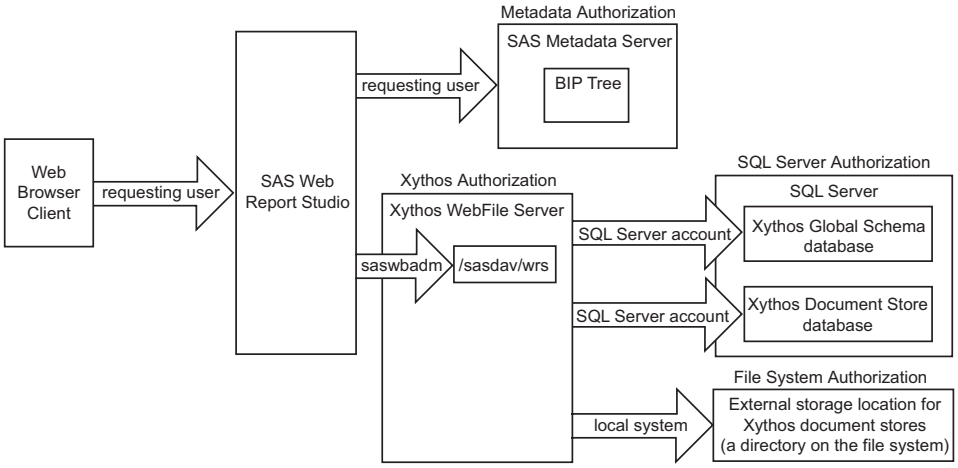
In order to protect	You must secure
Report definitions	The metadata objects that are associated with the reports
	The physical storage location of the report definitions
Underlying report data	The metadata objects that are associated with the report data
	The physical storage location of the report data
	The information maps that reference the report data
	The stored processes that reference the report data
	The report definition (if the report includes embedded data)
	The generated report (if the report is a batch report)

Different types of reports require different security measures. For example, if there is no embedded data of a sensitive nature in a report definition, then the report definition can be considered secure if the report's underlying data, information maps, stored processes, and output are secure. However, batch reports (and some reports that are created through ODS) can include embedded data, so these reports must be protected with access controls that parallel the access controls on the underlying information maps and stored processes.

CAUTION:
Do not rely on restricting access to the underlying information maps or stored processes to ensure that batch reports are viewed only by the appropriate users. △

Access to a report can be affected by multiple layers of controls. For example, the following figure depicts the authorization layers that affect access to reports in a deployment that is using a Xythos WFS content server with the Xythos document store located in a directory in the file system.

Figure 15.2 Authorization Layers for SAS Web Report Studio



In the figure, the requesting user's access to reports is subject to controls in the metadata, Xythos, SQL server, and file system authorization layers. However, the only layer in which the requesting user's permissions matter is the metadata layer, because this is the only layer in which the requesting user's identity is known. In the metadata layer, each user's access to reports is based on the user's individual identity and group memberships. When you work with metadata access controls for reports, consider these points:

- The **`ReportStudio`** folder structure includes appropriate metadata access controls for the **`Shared`** folder and the **`Users`** report folders. Additional steps should be taken in the metadata layer to protect your reports.
- The only relevant permissions for reports are ReadMetadata and WriteMetadata. The Read, Write, Create, Delete, CheckInMetadata, and Administer permissions have no effect on these objects. For more information, see "Which Actions are Controlled by Each Permission?" in the "Understanding Authorization" section in the *SAS Intelligence Platform: Security Administration Guide*.
- The effective permissions for a report folder are inherited by all of the reports within that folder. For more information, see "Inherited Access Controls" in the "Understanding the Metadata Authorization Layer" section in the *SAS Intelligence Platform: Security Administration Guide*.
- The ability to view or work with a report can be affected by access to each of the report's underlying components. For information about the metadata layer requirements for working with reports and report folders, see "Access Requirements for Reports" in the "Access Guidelines and Requirements" section in the *SAS Intelligence Platform: Security Administration Guide*.
- If your organization uses publication channels to deliver reports, the reports can also be protected by controlling access to the publication channels. A user must have ReadMetadata access to a SAS Publication Channel in order to self-subscribe to that channel. In addition, you might need to edit the policy file to add or remove permissions for the folders that correspond to the channels. To learn how to set up a publication channel, see "Adding SAS Publication Channels" in *SAS Web Infrastructure Kit: Administrator's Guide* at **`http://support.sas.com/rnd/itech/doc9/portal_admin/content/ag_saschannels.html`**.
- To access the **`Authorization`** tab for a report folder or an individual report, navigate in SAS Management Console under **Environment Management ▶ BI Manager ▶ BIP Tree ▶ ReportStudio**.

As the preceding figure depicts, SAS Web Report Studio uses only one account to connect to the external storage location, so you cannot make access distinctions between individual users by setting operating system access controls on specific items within the external storage location. However, you should set operating system controls that allow only the identity under which the Xythos process is running (local system in this example) to access to this physical file location.

Similarly, SAS Web Report Studio uses only one account (saswbadm) to communicate with the WebDAV content server, so you cannot make access distinctions between individual users by setting access controls in Xythos WFS. However, you should use this layer to protect your SAS report content, as described in "Protecting Report Content in the WebDAV Server" on page 314.

Security Considerations for Pre-generated Batch Reports

When a batch report is generated, the content of the report reflects the access that the generating user ID has to objects such as data sources and stored processes. For example, if a batch report is configured to use an identity named BATCH, then the report can include anything that BATCH is able to access. Regardless of who actually views the report, the report content is always based on the access controls that apply to BATCH. This means that any user who has ReadMetadata permission for a batch report can view that report, even if other metadata access controls deny the user access to the report's underlying components (such as data sources and stored processes). For this reason, you must give careful consideration to the identity that each batch report uses for generation, and you must secure the batch reports that you create.

Note: Once a user refreshes the report data, that users sees only the content that he or she has permission to see. △

Considerations for Row-level Security

If you implement row-level access to data, it is recommended that you configure a pooled workspace server that is dedicated for use by SAS Web Report Studio. You need a pooled workspace server to prevent the workspace server processes from running under the accounts of requesting users. Pooled workspace servers run under one or more designated accounts that are called puddle accounts, or puddle logins. You need a dedicated workspace server to isolate the row-level security puddle account from applications that do not fully enforce row-level security.

For instructions on setting up the workspace server, see "Restricted Workspace Server Pooling for SAS Web Report Studio (Enforcing Row Level Permissions)" on page 393.

For more information about row-level security, see *SAS Intelligence Platform: Security Administration Guide*.

Protecting Report Content in the WebDAV Server

On a publicly accessible WebDAV server, the area where SAS report content is stored should be protected against access by components that do not enforce SAS metadata permissions. For example, applications from other vendors and the DAV navigator portlet should not be able to access content in this area.

The recommended approach is to have SAS Web Report Studio use an area named **`sasdav/wrs`** on the content server, and to use the content server's access controls to give the SAS Web Administrator account (saswbadm) exclusive access to that area. The

user ID and password of the SAS Web Administrator are available to SAS Intelligence applications through the SAS metadata repository. Applications that are not aware of SAS metadata do not have access to the saswbadm user ID and password, so these applications cannot access the **sasdav/wrs** area of the content server.

For example, if you are using Xythos WFS as your content server, you can verify that these protections are in place by completing these steps:

1. In SAS Management Console, navigate to **Environment Management ▶ BI Manager ▶ BIP Tree ▶ Properties** and verify that your deployment of SAS Web Report Studio has been configured to use the SAS Web Administrator (saswbadm) to access the **sasdav/wrs** area in your content server.

   Note: If you have installed the SAS Foundation Services 1.2, then you should use the BI Manager plug-in instead of the Business Report Manager plug-in. Starting with that release, the BI Manager replaces the Business Report Manager, and is the recommended SAS Management Console plug-in for administering reports. △

2. Access the Xythos administration console by opening a Web browser to **<machine-name>:8300/xythosadmin** and logging on as the Xythos administrator.
3. Select **File System ▶ Directory & File Admin**.
4. Select the permissions icon for the **sasdav/wrs** directory.
5. On the access permissions page for the **sasdav/wrs** directory, verify that the SAS Web Administrator has exclusive, full access to this directory (and to this directory's subdirectories and files).
6. Verify that the SAS Web Administrator can access the **sasdav/wrs** directory by completing these steps:
   a. Open a Web browser to **<machine-name>:8300/sasdav/wrs** and use the credentials of the SAS Web Administrator to log on.
   b. On the index page for **sasdav/wrs**, select Launch Web Folder.
   c. Drag and drop a local file onto the page to add it to the folder.
   d. Delete the file that you added to the folder.
7. Verify that other users cannot directly access the **sasdav/wrs** area by completing these steps:
   a. Delete the credentials that were cached for the SAS Web Administrator on the machine where you are working.
   b. Open a Web browser to **<machine-name>:8300/sasdav/wrs** and use the credentials of the SAS Demo User to log on.
   c. Instead of seeing the index page for the **sasdav/wrs** directory, you should see a "Page Not Found" message.

These are the only Xythos layer access controls that you should set in the **sasdav/wrs** content area, because these are the only Xythos layer access controls that are meaningful for SAS Web Report Studio. Administration of the SAS Information Delivery Portal requires you to use Xythos layer access controls to manage access for other content areas within the WebDAV server.

Protecting Data in the SAS Web Report Studio Temporary Files

SAS Web Report Studio writes temporary files that might contain data that should be protected. These temporary files are stored in the following locations:

- In the **tmpnull** and **tmpuser** subfolders within the folder where SAS Web Report Studio is deployed. For example, if you are using Tomcat, this location might be **C:\Tomcat4.1\work\Standalone\localhost\SASWebReportStudio\sas.wrs**.

- In the Java temporary folder on the server where SAS Web Report Studio is running. The location of this folder is defined by the Java property java.io.tmpdir.

To protect the data in these temporary files, you should do these things:

- Place the computer on which SAS Web Report Studio is deployed in a physically secure location.
- Use operating system protections to limit access to the computer on which SAS Web Report Studio is deployed.
- Set additional operating system protections on the folders that contain the temporary files. Only system administrators who require access to all folders should be able to access these folders.

Configuring the SAS Web Report Studio Logs

Overview of SAS Web Report Studio Log Files

You can use the SAS Web Report Studio log files to help you manage performance, track security enforcement, and analyze specific situations. SAS Web Report Studio records events in two log files. By default, both log files are created in the *SAS-config-dir*`\Lev1\web\Deployments\WebReportStudio\logs` directory.

The following table summarizes the log files:

Table 15.8 Log Files

Log Context and Default File Name	Description
General Purpose Log (WebReportStudio.log)	Logs events such as serious errors, application restarts, and users logging on.
Key User Action Log (WebReportStudio_KeyActions.log)	Logs events such as application use, failed attempts to log on, report access, and batch report activities. For a list of all events, see "Understanding Key User Action Log Output" on page 318.

Note: There are similar log files for SAS Web Report Viewer in the directory. △

Change the General Purpose Log File's Logging Level

For the General Purpose Log, you can change the amount of information that is recorded. The log has four levels of warnings: DEBUG, INFO, WARN, and ERROR. By default, the log level is set to WARN, which means that only WARN and ERROR messages are recorded. In large-scale deployments, the size of the log file can grow rapidly when INFO messages are enabled. However, you might want to enable the INFO messages during the development and testing phases. (You can also set the level to DEBUG, but the amount of output is very high, and this can negatively affect performance and consume file space. If you need to debug a problem, it is recommended that you dynamically change the log output temporarily. See "Configure Debug Logging Dynamically" on page 317.)

To enable INFO level messages in the General Purpose Log:

1. Open the `DefaultLoggerProperties.xml.orig` properties file, which is located in *SAS-install-dir*`\SASWebReportStudio\3.1\config`.

2 In the **<LoggingContext>** block for "com.sas.apps.citation," change the priority value from **WARN** to **INFO**.

3 If you are using SAS Web Report Viewer to render reports, then make similar changes for SAS Web Report Viewer. Open the SAS Web Report Viewer's **DefaultWRVLoggerProperties.xml.orig** properties file , and edit the priority value.

4 After you make these changes and save the files, you must redeploy SAS Web Report Studio and, if applicable, SAS Web Report Viewer before your changes take effect. See "Redeploy SAS Web Report Studio" on page 337.

Note: It is recommended that you make a copy of the **DefaultLoggerProperties.xml.orig** file. Subsequent installation or upgrade activity can overwrite this file. △

Configure Debug Logging Dynamically

The previous topic discusses how to change the logging level for debugging. However, the procedure that was specified requires you to redeploy and restart SAS Web Report Studio. As an alternative, you can configure debug logging dynamically without restarting SAS Web Report Studio. You can do the following:

- activate a one-line notification for every action that occurs. This one-line message can be useful for providing debugging information about events.
- change the log level for events that are logged for **com.sas.apps.citation** or for one of its descendant contexts.

To implement this functionality, you manually edit the URL for SAS Web Report Studio in your browser. You must be logged on as a WRS Administrator (member of the WRS Administrator group) to use this functionality.

There is no browser-based feedback for this debugging feature. All relevant information about events is placed in the General Purpose Log file.

To edit the URL, do either or both of the following:

- To activate the one-line notification , add the following string to the end of the URL:

  ```
  debugLog.do?LogAllActions=true
  ```

  Here is an example:

  ```
  http://localhost:8080/SASWebReportStudio/debugLog.do?LogAllActions=true
  ```

  Here is an example one-line message that might appear in the log file:

  ```
  WRS 16:25:50,825 WARN report.OpenReportManagerAction
  [da8ff705996908f9:14eaec9:107ed50f82f:-7fea]- DEBUG
  logging of action requested: OpenReportManagerAction
  ```

  When you have finished debugging, to suppress the one-line notification, change

  ```
  debugLog.do?LogAllActions=true
  ```

  to

  ```
  debugLog.do?LogAllActions=false
  ```

- To change the logging level, add the following to the end of the URL:

  ```
  debugLog.do?LogName=log.context&LogLevel=level
  ```

  Provide the following values:

  - Replace *log.context* with the log context that you want to debug. You can specify **com.sas.apps.citation**, or any context under

com.sas.apps.citation. The **com.sas.apps.citation** context is the highest level, and represents the entire SAS Web Report Studio subsystem.

- Replace *level* with the log level that you want. You can specify any one of the following: DEBUG, INFO, WARN, ERROR

Here is an example:

```
http://localhost:8080/SASWebReportStudio/
debugLog.do?LogName=com.sas.apps.citation&LogLevel=DEBUG
```

Manage the Key User Action Log File

For the Key User Action Log, SAS uses a rollover mechanism to manage the size and age of the log. SAS periodically archives the current log and creates a new one. To archive a log, SAS saves the log with a new name that includes the current date and time. SAS archives the current log based on configurable settings related to the size of the file and the duration since the last archive. SAS also can delete files after the number of archived files reaches a particular limit.

To manage the Key User Action Log:

1. Edit the **<rollover>** element in the **<sas.wrs.keyUserActionLog>** block of your **LocalProperties.xml** file. The **<rollover>** element contains attributes that enable you to specify the maximum number of rollovers, the maximum size of the log file, and the schedule for performing rollovers.

   If your **LocalProperties.xml** file does not contain the **<sas.wrs.keyUserActionLog>** element block, then you can add it to the file by copying the block from the **WebReportStudioProperties.xml.orig** file. For instructions on creating a **LocalProperties.xml** file, see "Create a LocalProperties.xml File" on page 325.
2. If you are using SAS Web Report Viewer to render reports, then make similar changes for SAS Web Report Viewer. Open the SAS Web Report Viewer's **LocalProperties.xml** file , and edit the **<rollover>** element.
3. Periodically move or delete outdated archived log files.

Understanding Key User Action Log Output

Events are logged to the **WebReportStudio_KeyActions.log** file in an XML format. Each event has a numeric code value that uniquely identifies the event.

The following table lists the events and their respective codes.

Table 15.9 Log Events and Their Codes

Event	Code
User logged on.	0
User attempted to log on but failed.	1
User logged off.	2
User saved a report.	3
User opened a report.	4
User deleted a report.	5
User moved a report.	6
User copied a report.	7

Event	Code
User renamed a report.	8
User started a system.	9
User scheduled a report. If the user scheduled a folder of reports, then the log file lists the folder.	10
User distributed a scheduled report.	11

Here are some entries from a sample log file:

```
<event><javaDate>1124136823696</javaDate><date>8/15/05</date><time>4:13PM</
time><code>9</code><description>System Startup</description></event>

<event><javaDate>1124136826633</javaDate><date>8/15/05</date><time>4:13PM/
</time><user>saswbadm</user><code>0</code><description>Logon</description>
</event>

<event><javaDate>1124136878587</javaDate><date>8/15/05</date><time>4:14PM/
</time><user>dolson</user><code>0</code><description>Logon</description></event>

<event><javaDate>1124136923432</javaDate><date>8/15/05</date><time>4:15PM/
</time><user>dolson</user><code>4</code><description>Open</description>
<report>/ReportStudio/Shared/Reports/Deanna/Bursting/Orion 2 level bygroup -3
</report></event>

<event><javaDate>1124136977261</javaDate><date>8/15/05</date><time>4:16PM/
</time><user>dolson</user><code>3</code><description>Save</description>
<report>/ReportStudio/Shared/Reports/Deanna/Bursting/testReport</report></event>

<event><javaDate>1124136992808</javaDate><date>8/15/05</date><time>4:16PM/
</time><user>dolson</user><code>2</code><description>Logoff</description>
</event>
```

Suggested Procedure for Reporting Events in the Key User Action Log

The information in the Key User Action Log can be imported into SAS data sets and presented in reports. Here is a suggested procedure for reporting the data:

1. Import the **`WebReportStudio_KeyActions.log`** data into a SAS data set. Following are the main steps:

   a. Assign a libref to the XML file, and specify the XML engine.
      Here is an example:
      ```
      libname myxml xml 'C:\My Files\XML\MyFile.xml';
      ```

   b. Use the SAS DATASETS procedure to import the XML file into a SAS data set.
      Here is an example:
      ```
      proc datasets library=myxml;
      ```
      For more information, see SAS Help and Documentation.

2. In SAS Information Map Studio, create an information map based on the data set that you created in the previous step. For the information map, you might want to provide the ability to filter based on the event code (<code> tag), the user name (<user> tag), the report name (<report> tag), or the date. For information about using SAS Information Map Studio, see the product Help.

3 In SAS Web Report Studio, define a report based on the information map that you created in the previous step. You can optionally define the report to be refreshed manually, and then schedule the report to run at regular intervals.

Improving the Performance of SAS Web Report Studio

Suggestions for Improving the Performance of SAS Web Report Studio

To optimize the performance of SAS Web Report Studio, you should do these things:

- Tune your SAS servers as recommended in "Configuring a Workspace Server for SAS Web Report Studio" on page 381 and "Workspace Server Pooling for SAS Web Report Studio and SAS Information Delivery Portal" on page 384.
- Configure your middle tier as recommended in Chapter 19, "Best Practices for Configuring Your Middle Tier," on page 417. This chapter includes information about setting up your J2EE application server to use the correct Java Virtual Machine options, creating a cluster of servers, and using an HTTP server to handle requests for static pages.
- Make appropriate use of pre-rendered reports such as manually refreshed reports and batch reports.
- Use report scheduling to control when batch reports are generated. For example, you can schedule reports to be generated on a nightly, weekly, or monthly basis.
- Use the query cache, which is enabled by default. For more information, see "Overview of the Query Cache" on page 320.
- Consider the performance, security, and flexibility trade-offs between using a file-based content server and using a WebDAV content server. Using a file-based content server can result in faster response times because the report read and write requests do not pass through an HTTP/DAV server. Although several variables influence how much of an improvement can be realized, a performance increase of 10–15% is typical.

  However, this performance increase comes at the cost of flexibility. A DAV-based content server provides access to content without direct operating system support or shared network areas. This type of access is especially important if an installation requires that content be accessible by tools or applications running on several machines or on widely dispersed machines. In such a diverse environment, a DAV-based content server is most likely a necessity. If content is accessed only from one machine or a very small number of machines, sharing the content space may not be an issue, and a file-based content server with the resulting performance increase is the better choice.

  Note: If you use Xythos WFS as your WebDAV content server, then you can improve the performance by changing the document store location to external storage in a file system location. The SAS installation instructions for Xythos WFS follow this recommended approach. △

Overview of the Query Cache

By default, SAS Web Report Studio (and SAS Web Report Viewer) use a large query cache to improve performance. For reports that contain more than one data-driven object, this cache maximizes efficiency. The query cache builds a temporary common data table that can fulfill the needs of all data-driven objects in the report. When the

query cache is used, complex queries that include functions such as joins and filters are run only once (to build the common data table). Each data-driven object in the report can then run simple extraction queries against the common data table.

Note: The use of the cache is determined on a per-report basis, depending on the content of each report. In the current release, cache optimization is used only for reports that are based on relational data. △

During installation, the query cache is enabled and is associated with a SAS library. After installation, you can optionally do the following:

- change the location of the query cache library
- disable the query cache

Using the query cache will likely increase performance if any of your reports has any of the following characteristics:

- a large number of joins from many tables
- many BY groups
- many report objects
- data sources other than SAS (for example, Oracle or DB2)
- formatted data values from data sources other than SAS

Conversely, using the query cache will *not* increase performance if your report has all of the following characteristics:

- few joins from few tables
- few BY groups
- few report objects
- only SAS data table(s) as a source, or non-formatted data values from data sources other than SAS

There is no performance penalty for using the query cache unless the report uses a large native SAS table with report-ready data.

Change the Location of the Query Cache Library

The default location for the library that is used for the query cache is *SAS-config-dir*`\Lev1\SASMain\Data\wrstemp`. After installation, you can specify a different location for this library. For performance purposes, the library should be created on a dedicated fast drive that has plenty of disk space (on the order of 100GB, but the needed size will vary based on your system's use). Backups are unnecessary because the cache files are temporary.

Note: Do not use the Work or Saswork library for this feature. The query cache won't function correctly if you use Work or Saswork. △

For clustered environments, the folder for this library needs to be exported to all nodes in the cluster (and you should specify the network address to this folder, not the local machine address). For non-clustered environments, or for a cluster that is restricted to a single physical machine, this folder does not need to be exported.

To change the query cache library, follow these steps:

1. Create a library where the query cache can temporarily store common data tables. To create the library, follow these steps:
   a. Create an `autoexec.sas` file that assigns the library. For example, the file might contain the following:

```
/* Libname assigned for temporary tables for the query cache */
libname optlib 'C:\SASServers\wrstemp';
```

b Save the **autoexec.sas** file on the workspace server machine.

c In SAS Management Console, navigate under the Server Manager to the lowest level of the workspace server definition.

d Select **Properties ▶ Options**.

e In the **Command** field, append a pointer to the **autoexec.sas**. file that you saved on the server. For example, add this string to the command:

```
-autoexec c:\servers\autoexec.sas
```

f Click **OK**.

Note: The SAS Web Report Studio user must be granted all operating system permissions for the directory that is associated with the library. For example, if you have assigned the SAS library optlib to **C:\SASServers\wrstemp**, then you must grant the users all permissions on the **C:\SASServers\wrstemp** folder. On UNIX systems, the SAS Web Report Studio user would need Read/Write/Execute permissions on the folder.

With pooled workspace servers, a single user group is granted rights to use the puddles in the pool. Only the pool administrator and the SAS Web Administrator require access to the puddles. If the query cache is enabled, then make sure the pool administrator and the SAS Web Administrator are members of the puddle user group. Otherwise, the query cache will not be used.

If you have not configured pooling, then each requesting user's individual (or shared) account will access the temporary tables. △

2 Edit the following values in the **wrs.config** file:

$RENDERER_OPTIMIZER_LIBNAME$=

$RENDERER_OPTIMIZER_SERVER$=

For example:

$RENDERER_OPTIMIZER_LIBNAME$=optlib

$RENDERER_OPTIMIZER_SERVER$=Pooled Workspace Server - Logical Workspace Server

3 Redeploy SAS Web Report Studio and SAS Web Report Viewer (if it is installed). For instructions, see "Redeploy SAS Web Report Studio" on page 337.

Disable the Query Cache

To disable the query cache, in the **LocalProperties.xml** file, set the **<wrs.activateReportOptimizer>** property to **false**. You might need to add the property to the file. If you don't have a **LocalProperties.xml** file, you can create one. See "Create a LocalProperties.xml File" on page 325.

Note: If you are using SAS Web Report Viewer to render reports, then configure SAS Web Report Viewer in a similar way. In the **LocalProperties.xml** file for SAS Web Report Viewer, set the **<wrs.activateReportOptimizer>** property to **false**. △

You might want to delete the library that is associated with the query cache if you are certain that you will not use the query cache in the future. If there's a chance that you will re-enable the query cache , then you should leave the library in place.

SAS Web Report Studio Configuration

Load Initial Metadata That Enables Interaction with Other SAS Applications

Interaction between SAS Web Report Studio and other SAS products, such as SAS Enterprise Guide, requires installation of metadata into your metadata repository. When you installed and configured SAS Web Report Studio, you ran a SAS program called **`LoadDefaultPreferences.sas`** that enables this interaction. During installation, you should also have been instructed to create a SAS user profile in metadata.

If you didn't run **`LoadDefaultPreferences.sas`**, or if you didn't create the profile, an error message is added to your log file when you log on to SAS Web Report Studio. The context for this error message is **`com.sas.apps.citation.model.pfs.PFSHelper`**. The error is not serious, but you can eliminate the error message by doing the following:

- run **`LoadDefaultPreferences.sas`**

  Run **`LoadDefaultPreferences.sas`** on the same system on which SAS Web Report Studio is installed. **`LoadDefaultPreferences.sas`** can be executed only one time, but it is safe to issue a run command multiple times (the program simply aborts if you try to rerun it). For instructions on running **`LoadDefaultPreferences.sas`**, see the **`deployment.html`** file.

- create the SAS profile

To create the SAS profile, follow these steps:

1. In SAS Management Console, navigate under **Foundation Services Manager ► Query and Reporting ► BIP Core Services** and select **Platform User Service**.
2. From the menu bar, select **File ► Properties**.
3. Select **Service Configuration ► Edit Configuration ► Profiles** and select **Add**. The New Profile dialog box is displayed.
4. The following table contains values that you should enter in this dialog box:

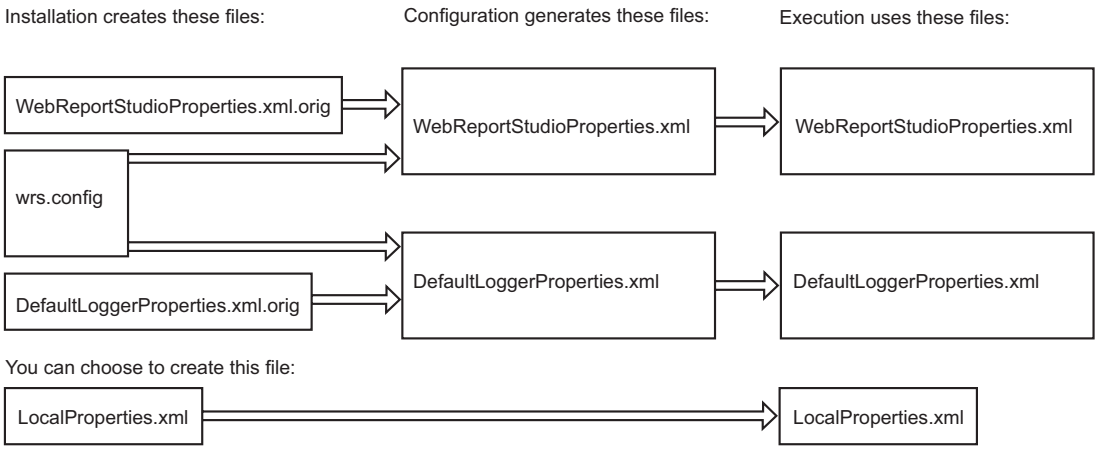
Table 15.10 Fields in the New Profile dialog box

Field	Value
Application	Enter the following: **`SAS`**
Domain URL	Copy and paste the URL from an existing profile.
Class	Enter the following: **`com.sas.preferences.SASProfile`**
Type	Leave this field blank.
Filter	Leave this field blank.

Administrative Files for SAS Web Report Studio

The configuration and properties files for SAS Web Report Studio are located on the middle-tier server on which SAS Web Report Studio is deployed. The following figure depicts the files that an administrator should be familiar with.

Figure 15.3 SAS Web Report Studio Configuration and Properties Files



The following list describes the depicted files:

- When SAS Web Report Studio is installed, these files are created:

  `WebReportStudioProperties.xml.orig`
  Contains most application properties. This file is located in *SAS-install-dir***`SASWebReportStudio\3.1\config\`**. Changes that you make to this file take effect after you deploy an new WAR file. This file can be overwritten by subsequent installation activities. For these reasons, the preferred method for setting site-specific application properties is to use a **`LocalProperties.xml`** file, as described below.

  `wrs.config`
  Contains attributes such as connection parameters, authentication providers, and location of the application log. This file is located in *SAS-install-dir***`SASWebReportStudio\3.1\`**. Changes that you make to this file take effect after you deploy a new WAR file.

  `DefaultLoggerProperties.xml.orig`
  Contains attributes such as the format, locations, names, and contents of the SAS Web Report Studio logs. This file is located in *SAS-install-dir***`SASWebReportStudio\3.1\config\`**. Changes that you make to this file take effect after you deploy a new WAR file. This file can be overwritten by subsequent installation activities.

- It is recommended that you store application properties settings in a centralized location that is not affected by subsequent installation activities. To do this, create and use a **`LocalProperties.xml`** file. For instructions, see "Create a LocalProperties.xml File" on page 325. Settings in the **`LocalProperties.xml`** file override conflicting settings in the **`WebReportStudioProperties.xml`** file. Changes that you make take effect after you restart your Web application server.

- Each time you configure SAS Web Report Studio to create a new application WAR file, these files are written to the **`WEB-INF`** directory of your Web application server:

  `WebReportStudioProperties.xml`
  Contains most application properties. You should modify this file only for the purposes of making temporary changes, because this file is overwritten by subsequent installation and configuration activities. To make changes in a clustered environment, you must locate and modify multiple instances of this file.

`DefaultLoggerProperties.xml`
Contains log configuration settings. You should modify this file only for the purposes of making temporary changes, because this file is overwritten by subsequent installation and configuration activities. To make changes in a clustered environment, you must locate and modify multiple instances of this file.

- When SAS Web Report Studio executes, it uses the properties files that are in the **`WEB-INF`** directory (and your local properties file if you chose to create one).

Create a LocalProperties.xml File

The **`WebReportStudioProperties.xml`** file can be overwritten by subsequent installation activities. It is recommended that you store application properties settings in a centralized location that is not affected by subsequent installation. To do this, create and use a **`LocalProperties.xml`** file.

Follow these steps to create a **`LocalProperties.xml`** file:

1. Locate the sample file in the **`customer`** subdirectory of the installation directory. The sample file is named **`LocalProperties.xml.sample`** .
2. Make a copy of this file in the **`customer`**directory, and name the copy **`LocalProperties.xml`**.

   Note: If you plan to use SAS Web Report Viewer to view reports, create a **`LocalProperties.xml`** file for SAS Web Report Viewer in a similar way. Locate the **`LocalProperties.xml.sample`** file in the SAS Web Report Viewer **`customer`** subdirectory, make a copy of the file, and name your new file **`LocalProperties.xml`**. △

Settings in the**`LocalProperties.xml`** file override conflicting settings in the **`WebReportStudioProperties.xml`** file. Changes that you make to **`LocalProperties.xml`** take effect after you restart your servlet container.

BI Manager Plug-In

The BI Manager in SAS Management Console enables administrators to perform these tasks:

- specify the account that SAS Web Report Studio uses to connect to your WebDAV content server.
- coordinate report storage by associating a top-level reporting folder in the metadata repository with a top-level report content area in the external content server.
  - In the metadata repository, the default name for the top-level reporting folder is **`ReportStudio`**. You can have more than one top-level reporting folder. The top-level reporting folder is sometimes referred to as the root folder.
  - In the content server, the top-level report content area is a directory location. If you are using the file system as your content server, the directory is simply a file system address. If you are using a WebDAV server, the directory is a content base path (/sasdav/wrs).
- manage the report storage containers. However, the BI Manager does not enable you to control access to the report folders. To set permissions on a reporting folder, navigate to the folder in SAS Management Console under **Environment Management ► Authorization Manager ► By Application ► BIP Tree ► ReportStudio** and then access the **`Authorization`** tab in the properties dialog box for that folder.

- import reports and other report content into the metadata repository.
- distribute reports among different metadata repositories by exporting and then importing a BI package that contains the reports along with their metadata definitions.
- deploy reports as jobs for scheduling.

This plug-in is added to your local copy of SAS Management Console when SAS Foundation Services is installed on the computer where you are working. For detailed instructions on using this plug-in, select **Environment Management ▶ BI Manager**and then select Help from the main menu bar in SAS Management Console.

Report Studio Configuration Plug-In

The Report Studio Configuration plug-in to SAS Management Console provides a graphical interface to the **`wrs.config`** file, which is documented in "Administrative Files for SAS Web Report Studio" on page 323. This plug-in is added to your local copy of SAS Management Console when SAS Query and Reporting Services is installed on the computer where you are working.

Set Maximum Values for Report Filters

Two values in the **`LocalProperties.xml`** file determine the following:

- The maximum number of filter values that can be displayed when report creators define a filter.

  Here is the corresponding element block, along with the default value:

  ```
  <webreportstudio.max.filter.choices>1000
  </webreportstudio.max.filter.choices>
  ```

- The maximum number of filter values that can be displayed when report viewers query for available filter values.

  Report creators can configure their reports to prompt for filter values that are generated dynamically when the report is rendered. To enable this feature, report creators choose the **`Prompting users to select values from a list`** and the **`allow users to query for values`** options in the Create New Filter dialog box. When the report is rendered, report viewers click a **`Get Values`** button to load the values that are available for the filter.

  You can configure the maximum number of prompt values that can be loaded when report viewers click the **`Get Values`** button. The default value is 1,000.

  Here is the corresponding element block, along with the default value:

  ```
  <webreportstudio.max.prompt.choices>1000
  </webreportstudio.max.prompt.choices>
  ```

To configure the maximum number of prompt values, in the **`LocalProperties.xml`** file, specify the number that you want for the appropriate element. You might need to add the element to the file. If you don't have a **`LocalProperties.xml`** file, you can create one. See "Create a LocalProperties.xml File" on page 325.

For more information about dynamic prompt values, or for instructions on creating a filter, see the product Help and Documentation.

Customizing Reports

Add Disclaimer Text to Graphs and Tables

SAS Web Report Studio enables you to add disclaimer text to graphs and tables. You can use the disclaimer text to provide a copyright statement or some general disclaimer of usage.

To add disclaimer text to graphs and tables, follow these steps:

1. Open the **LocalProperties.xml** file. For instructions on creating the **LocalProperties.xml** file, see "Create a LocalProperties.xml File" on page 325.
2. In **LocalProperties.xml**, add or edit the following element block:

```
<wrs.disclaimer.tableAndGraph>
MyDisclaimer
</wrs.disclaimer.tableAndGraph>
```

   In the above block, replace *MyDisclaimer* with your own disclaimer text.

   The text will wrap automatically; you cannot specify separate lines by inserting a carriage return or new line. Graphics are not supported in disclaimer text.
3. Save your changes.
4. Restart the servlet container.

The disclaimer text does not affect existing reports. The text will appear beneath the tables and graphs of all new reports.

Provide Custom Report Styles

Report styles affect the colors, fonts, and other elements that are used in tables and graphs. By default, report viewers can select one of the following styles for a report: Meadow, Seaside, or Festival. You can add your own custom styles to the list of available styles.

Note: The ability to apply custom styles is currently available only for applications that run with a U.S. locale. △

SAS Web Report Studio relies on cascading style sheets (CSS) to render styles. To supply a custom style, follow these steps:

1. Create a CSS file and define the formats that you want for the style. For details about the supported formats, see "CSS Formats for Custom Report Styles" on page 329. For information about CSS files in general, consult the W3C organization's Web site at **http://www.w3.org/TR/CSS21/**.

   A sample CSS file is available to help you develop your own custom styles. The file **Seaside_CSS.css** was copied to the **customer** folder when you installed and then configured SAS Web Report Studio. This CSS is based on the built-in Seaside style.
2. In your **LocalProperties.xml** file, provide information that SAS Web Report Studio needs in order to locate and render the style.

To provide information about the style, follow these steps:

1. Open the **LocalProperties.xml** file in a text editor. If you don't have this file, then you can create it. See "Create a LocalProperties.xml File" on page 325.

2 In **LocalProperties.xml**, add the following element block if it's not already there (you can copy and paste from the **WebReportStudioProperties.xml.orig** file):

```
<sas.wrs.style>
<css></css>
<schemelist>
Seaside,Festival,Meadow
</schemelist>
<defaultscheme></defaultscheme>
</sas.wrs.style>
```

3 In the element block, modify the elements to specify your CSS file and style scheme. The following table describes the elements:

Table 15.11 Report Style Elements in LocalProperties.xml

Element	Description
<css>	Provides the fully qualified path to one or more external CSS files from which style schemes will be read. If you specify multiple files, separate them with a comma. If you remove a file name from this element, then any report that has been created with the corresponding style might not render correctly. The rendering behavior is undefined if the CSS file has been removed.
<schemelist>	Specifies the list of styles that are available to SAS Web Report Studio users. You must add your custom style name to the list in order for that style to be available for use. The name must match exactly the name of a CSS file in the <css> list (but without the file path or CSS file name extension). Any mismatches cause the name not to be available in SAS Web Report Studio. If you specify multiple styles, separate the style names with a comma. Default styles are Meadow, Seaside, and Festival. If you remove any of these names from the list, then the corresponding styles will no longer be available to users. However, existing reports that reference the styles will continue to render properly because these styles are built in and inherently known by SAS Web Report Studio.
<defaultscheme>	Defines the default styles that will be applied to new reports. If no style is specified, then the default style is Seaside.

This example shows how these properties can be specified:

```
<sas.wrs.style>
<css>C:\styles\CustomScheme1.css,C:\styles\CustomScheme2.css</css>
<schemelist>
Seaside,Festival,Meadow,CustomScheme1,CustomScheme2
</schemelist>
<defaultscheme>CustomScheme1</defaultscheme>
</sas.wrs.style>
```

4 Save your changes.

5 If you intend to use SAS Web Report Viewer to render reports, then you must make similar changes for SAS Web Report Viewer. Open the **LocalProperties.xml** file for SAS Web Report Viewer, and repeat the previous steps.

6 You must restart the servlet container before your changes take effect.

CSS Formats for Custom Report Styles

About CSS Formats

In order to provide custom report styles, you create one or more CSS files. A CSS file enables specified formats (CSS rule sets) to be available for end users to modify in SAS Web Report Studio.

Here are the elements that can be modified by users in SAS Web Report Studio:

- tables, both list and crosstabulation
- graphs
- text objects
- headers and footers
- containers for synchronized objects
- display filters

In the CSS file, lines that start with < or - are considered comments. These lines are ignored by SAS Web Report Studio.

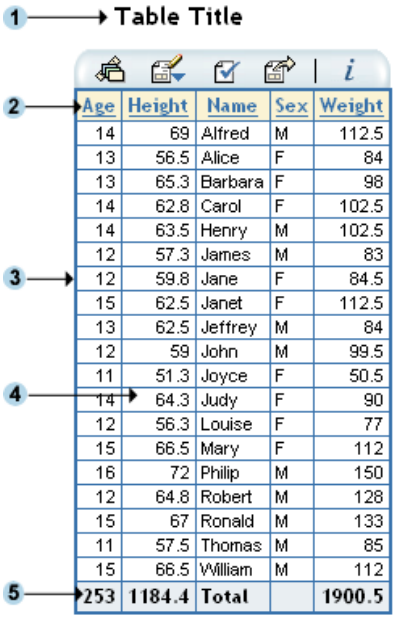
SAS Web Report Studio does not support at-rules, such as @import. Such directives are ignored.

A sample CSS file is available to help you develop your own custom styles. The file **`Seaside_CSS.css`** was copied to the **`customer`** folder when you installed and then configured SAS Web Report Studio. This CSS is based on the built-in Seaside style.

For instructions on making the CSS formats available to SAS Web Report Studio, see "Provide Custom Report Styles" on page 327. For information about CSS files in general, consult the W3C organization's Web site at **`http://www.w3.org/TR/CSS21/`**.

Tables

The following figure shows a sample list table.



Age	Height	Name	Sex	Weight
14	69	Alfred	M	112.5
13	56.5	Alice	F	84
13	65.3	Barbara	F	98
14	62.8	Carol	F	102.5
14	63.5	Henry	M	102.5
12	57.3	James	M	83
12	59.8	Jane	F	84.5
15	62.5	Janet	F	112.5
13	62.5	Jeffrey	M	84
12	59	John	M	99.5
11	51.3	Joyce	F	50.5
14	64.3	Judy	F	90
12	56.3	Louise	F	77
15	66.5	Mary	F	112
16	72	Philip	M	150
12	64.8	Robert	M	128
15	67	Ronald	M	133
11	57.5	Thomas	M	85
15	66.5	William	M	112
253	**1184.4**	**Total**		**1900.5**

Here are the supported style formats for elements in the list table.

Table 15.12 CSS Formats for List Tables

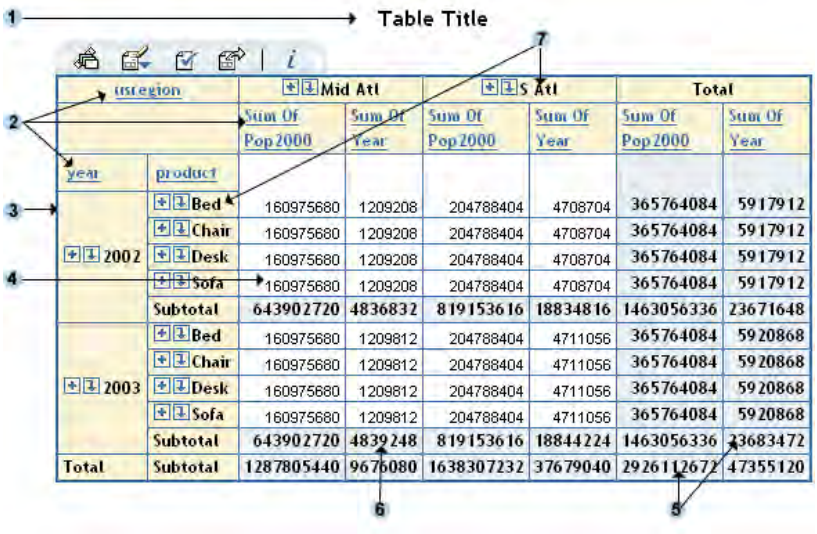
Callout Number	Selector	Supported Property Types
❶ (title)	Table Caption	text
❷ (headings)	Table Column Label	text cell border
❸ (border)	Table	border
❹ (cells)	Table Column Cell	text * cell border
❺ (totals)	Table Rows Summary	text cell border

* The alignment (text-align property) for cells is overridden based on data type (numeric vs. text).

Note: In the CSS file, you must define the Table format before you define any of its descendant formats, such as Table Caption or Table Column Label. △

For more details about the supported property types, see "Supported Properties" on page 336.

The following figure shows a sample crosstabulation table.



		usregion					
		Mid Atl		S Atl		Total	
		Sum Of Pop2000	Sum Of Year	Sum Of Pop2000	Sum Of Year	Sum Of Pop2000	Sum Of Year
year	product						
2002	Bed	160975680	1209208	204788404	4708704	365764084	5917912
	Chair	160975680	1209208	204788404	4708704	365764084	5917912
	Desk	160975680	1209208	204788404	4708704	365764084	5917912
	Sofa	160975680	1209208	204788404	4708704	365764084	5917912
	Subtotal	643902720	4836832	819153616	18834816	1463056336	23671648
2003	Bed	160975680	1209812	204788404	4711056	365764084	5920868
	Chair	160975680	1209812	204788404	4711056	365764084	5920868
	Desk	160975680	1209812	204788404	4711056	365764084	5920868
	Sofa	160975680	1209812	204788404	4711056	365764084	5920868
	Subtotal	643902720	4839248	819153616	18844224	1463056336	23683472
Total	Subtotal	1287805440	9676080	1638307232	37679040	2926112672	47355120

Here are the supported style formats for elements in the crosstabulation table.

Table 15.13 CSS Formats for Crosstabulation Tables

Callout Number	Selector(s)	Supported Property Types
❶ (title)	Table Caption	text
❷ (headings)	Table Rowgroup Label	text
	Table Rowgroup Row Label	cell
	Table Columngroup Label	border
	Table Columngroup Column Label	
❸ (border)	Table	border
❹ (cells)	Table Rowgroup Row Cell	text *
	Table Columngroup Column Cell	cell
		border
❺ (totals)	Table Rows Summary	text
	Table Columns Summary	cell
		border
❻ (subtotals)	Table Rowgroup Rows Summary	text
	Table Columngroup Columns Summary	cell
		border
❼ (subheads)	Table Rowgroup Values	text
	Table Columngroup Values	cell
		border

* The alignment (text-align property) for cells is overridden based on data type (numeric vs. text).

Note: In the CSS file, you must define the Table format before you define any of its descendant formats, such as Table Caption or Table Column Label. △

For more details about the supported property types, see "Supported Properties" on page 336.

Graphs

Like tables, graphs support styles for different aspects of their rendering. However, when subgroups are used in a graph, you should specify a unique format for each subgroup value in order to distinguish between the values. Since subgrouping is data dependent (one subgroup might have three values, whereas the same subgroup on different data might have nine values), SAS Web Report Studio supports a flexible collection of rules called *graph data styles*. A report scheme can consist of up to 12 specified graph data styles. Each graph data style can in turn be used for a particular subgroup of data.

The following example shows three sample graph data styles:

```
Graph GraphDataStyle1
{
        color : red;
        marker-symbol : DIAMONDFILLED;
        marker-size : 10px;
        line-thickness : 2px;
}
```

```
Graph GraphDataStyle2
{
        color : green;
        marker-symbol : DIAMONDFILLED;
        marker-size : 10px;
        line-thickness : 2px;
}

Graph GraphDataStyle3
{
        color : blue;
        marker-symbol : DIAMONDFILLED;
        marker-size : 10px;
        line-thickness : 2px;
}
```

This method enables you to define graph schemes that supply common formats across different types of graphs. Not all the graph data styles are used for each graph.

Note: The progressive bar chart and the geographical chart do not support the GraphDataStyle*n* formats. The supported formats for these charts are described later in this section. △

The following figure shows a sample graph, followed by a list of the supported formats for elements in the graph.

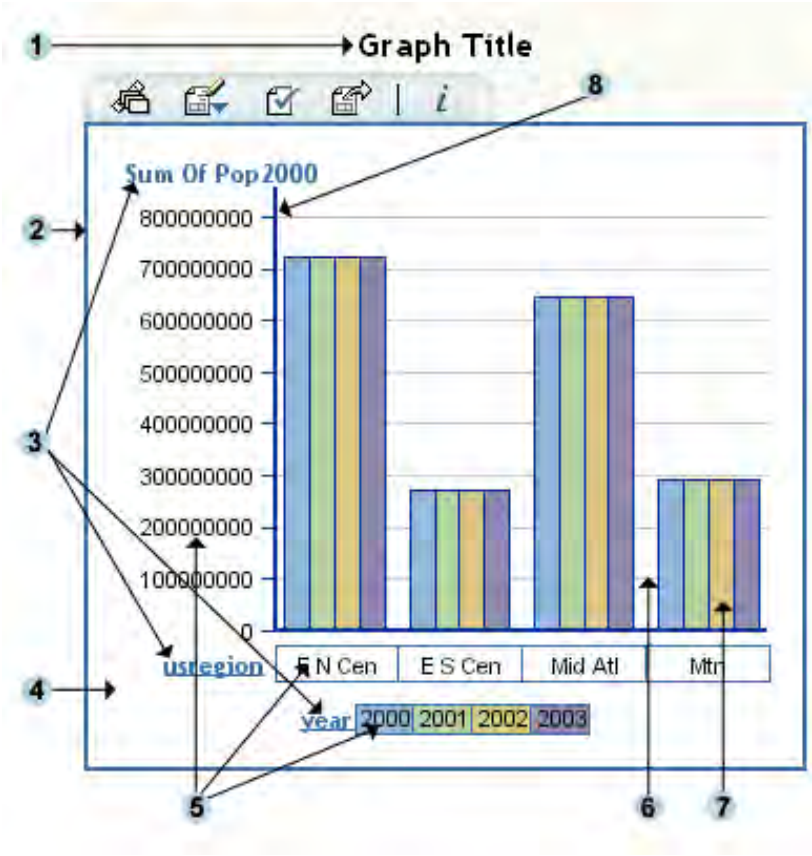


Table 15.14 CSS Formats for Graphs

Callout Number	Selector	Supported Properties and Property Types
❶ (title)	Graph TitleText	text
❷ (border)	Graph BorderLines	line-color property
❸ (axis and legend labels)	Graph LabelText	minimal text
❹ (background)	Graph BackFill	fill-color property
❺ (axis and legend values)	Graph ValueText	minimal text
❺ (axis and legend values)	Graph LegendFill	fill-color property
❻ (grid lines)	Graph GridLines	line-color property
❼ (data)	Graph GraphDataStyle*n*	graph data styles
❽ (horizontal and vertical axis)	Graph AxisLines	line-color property line-thickness property

For more details about the supported property types, see "Supported Properties" on page 336.

The geographical (ESRI) chart supports only the border style.

The progressive bar chart does not support the GraphDataStyle*n* formats. Instead, the chart uses three different formats for its data styles. These formats are unique to the progressive bar chart.

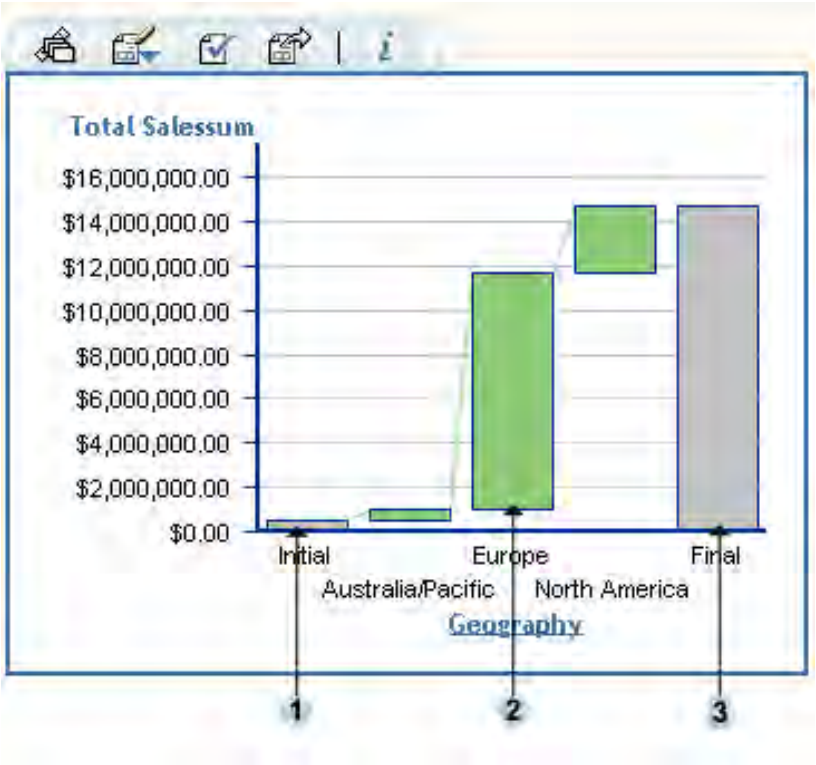


Table 15.15 CSS Formats for Progressive Bar Charts

Callout Number	Selector(s)	Supported Properties
❶ (initial bar)	Graph InitialDataStyle	fill-color
❷ (positive/negative bars)	Graph ThreeColorRamp	fill-gradient-start-color
	Graph ThreeColorAltRamp	fill-gradient-end-color
❸ (final bar)	Graph FinalDataStyle	fill-color

Text

Text elements, including headers and footers, use the **`text`** property type, and support all text formats.

Synchronized Objects Container

The following illustration shows a container for synchronized objects, followed by a list of the supported formats.

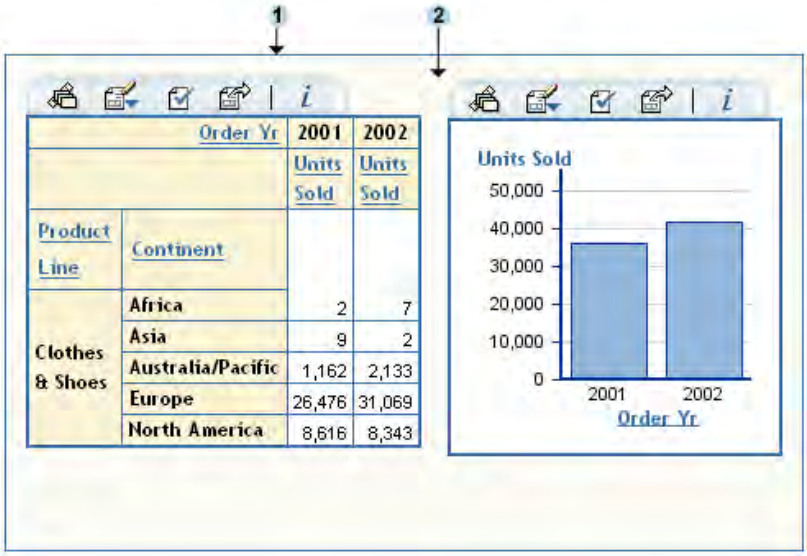


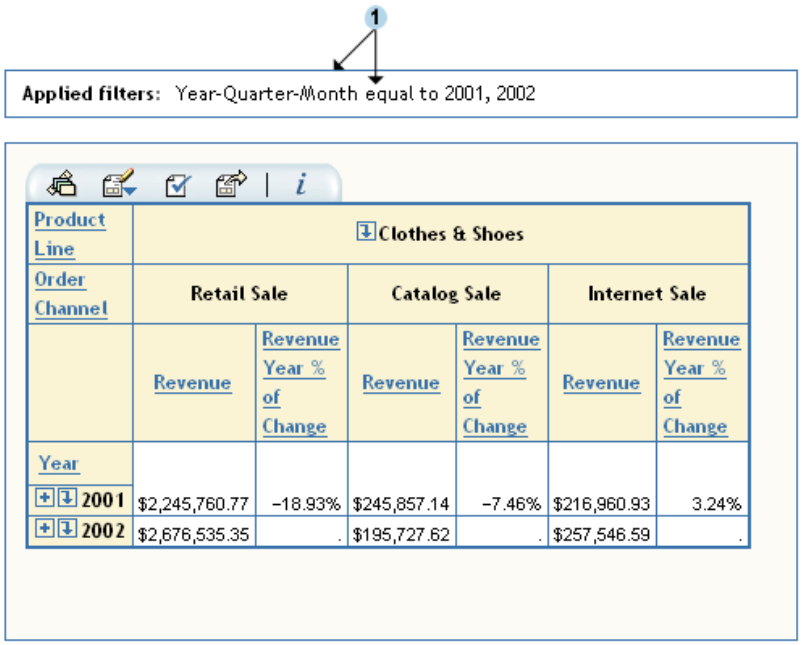
Table 15.16 CSS Formats for a Synchronized Objects Container

Callout Number	Selector	Supported Properties or Property Types
❶ (container border)	LinkedContainer	border
❷ (container)	LinkedContainer	background-color property
		padding property

For details about the supported property types, see “Supported Properties” on page 336.

Display Filters

Display Filters must be specified individually for graphs, tables, and the containers for synchronized objects. The following figure shows display filters for a table. Display filters are similar for graphs and synchronized object containers.



Here are the supported style formats for display filters. When the tables and graphs are synchronized objects, then the LinkedContainer selector must be used, because the filters are displayed for the container that holds the tables and graphs.

Table 15.17 CSS Formats for Display Filters

Callout Number	Selector	Supported Property Types and Properties
❶ (filter)	Graph DisplayFilter	text border margin-bottom
❶ (filter)	Table DisplayFilter	text border margin-bottom
❶ (filter)	LinkedContainer DisplayFilter	text border margin-bottom

Note: For each of the formats in the table, you must define the parent format before you define any of its descendants in the CSS file. For example, you must define a LinkedContainer format before you define a LinkedContainer DisplayFilter format. △

Supported Properties

This table lists the properties that are supported for the property types that are mentioned in the previous sections.

Table 15.18 Supported Properties for CSS Formats

Property Type	Supported Properties
text	font-family
	font-weight
	color
	background-color
	text-align
	font-size
	text-decoration
	font-style
minimal text	font-family
	font-weight
	text-color
	font-size
	font-style
border	border
	border-color
	border-top-color
	border-bottom-color
	border-right-color
	border-left-color
	border-width
	border-top-width
	border-bottom-width
	border-right-width
	border-left-width
	border-style
	border-top-style
	border-bottom-style
	border-right-style
	border-left-style

Property Type	Supported Properties
cell	padding
	padding-top
	padding-bottom
	padding-left
	padding-right
graph data styles	color
	marker-symbol*
	marker-size
	line-thickness

* Possible values are: TRIANGLEFILLED, SQUAREFILLED, STARFILLED, HEXAGONFILLED, CIRCLEFILLED, CROSSFILLED, FLAGFILLED, CYLINDERFILLED, PRISMFILLED, X, SPADEFILLED, DIAMONDFILLED, HEARTFILLED, CLUBFILLED, POINT, NONE.

Redeploy SAS Web Report Studio

After initial installation, if you make configuration changes, then you might be instructed to redeploy SAS Web Report Studio. All of the procedures in the documentation explicitly state when you must redeploy SAS Web Report Studio.

Follow these steps to redeploy SAS Web Report Studio:

1. Create a new **SASWebReportStudio.war** file by running **sas.wrs.config.sh** (UNIX) or **sas.wrs.config.bat** (Windows). These scripts are located in *sas-install-dir***\SASWebReportStudio\3.1**.
2. If you are using WebLogic, run **sas.wrs.weblogic.prepare.bat** to explode the new WAR file into a specified directory. For example, from a command prompt, change to the SAS Web Report Studio installation directory and enter this command:

   ```
   sas.wrs.weblogic.prepare.bat C:\bea\webapps SASWebReportStudio SASWebReportStudio.war
   ```

   CAUTION:
   The contents of the target directory are deleted by this script, so do not reference a directory that includes files for other Web applications. For complete configuration instructions, see *sas-install-dir***\SASWebReportStudio\3.1\deployment.html**. △
3. Deploy the new WAR file. For deployment instructions, see *sas-install-dir***\SASWebReportStudio\3.1\deployment.html**.

   Note: If you are deploying into Tomcat, you will need to run a second script that disables Tomcat's serialization feature, that provides an explicit context for SAS Web Report Studio, and that explodes the WAR file. See the **deployment.html** file for details. △
4. Restart the servlet container or J2EE application server.

For some procedures, such as configuring trusted Web authentication, you might also be instructed to redeploy SAS Web Report Viewer. The deployment steps are similar. Create a new **SASWebReportViewer.war** file by running **sas.wrv.config.sh** or **sas.wrv.config.bat** located in *sas-install-dir***\SASWebReportViewer\3.1**. Then deploy the new WAR file. For detailed instructions, see **<install>\SASWebReportViewer\3.1\deployment.html**.

Scheduling and Distributing Pre-generated Reports

Overview of Pre-generated Reports

SAS Web Report Studio enables you to run queries against reports and provide the results to users. One advantage of providing pre-generated results is improved performance. It takes less time to view the report because the queries have already been processed and the results have already been generated. However, because these reports are non-interactive, they cannot accept input from a requesting user. When a pre-generated version of a report includes prompts (questions that require user input), the prompt values that were provided when the report was generated are used. The non-interactive nature also has important implications for security, which are described in "Security Considerations for Pre-generated Batch Reports" on page 314.

Pre-generated reports can be provided to users in three ways:

- You can save any report as a static report in PDF format. The report is saved in your repository and is made available to authorized users. Users cannot interact with a report that has been saved in PDF format.
- You can create manually refreshed reports. Manually refreshed reports are saved reports that contain data from a pre-generated query. The report is stored in your repository, and the results can be viewed in SAS Web Report Studio or SAS Web Report Viewer. When users open the report, they can manually refresh the report data.

  There are two ways to create a manually refreshed report:

  - You can select **`Data can be manually refreshed`** from the Save As dialog box when you save the report. This type of report typically requires a manual refresh each time it is opened.
  - You can schedule saved reports to be run at a specified time. This feature enables you to refresh report data at specified intervals or times. You can also specify an archive in order to maintain older versions of the report.

  The following list explains how users interact with manually refreshed reports:

  - A user cannot interact with a manually refreshed report until the user refreshes the report.
  - If a user saves changes to the live version of a report, then the original, static version of the report is deleted.
  - If a user saves changes to the live version of a report and specifies that the report can be manually refreshed, then a new static version of the report is generated and saved (along with the revised live report).
  - If a user saves changes to the live version of a report using a different name for the report, then the original version of the report is preserved. In this case, a version of the revised report is not generated.
- The third way to provide a pre-generated report is to create a snapshot of report data, and then distribute the static results to recipients.

  The snapshot that you distribute might include all or part of the original report. For example, suppose that your organization has sales teams in different countries, and you want to provide high-level sales information to the team managers in each country. When you set up a distribution, you can group the report based on the country, and then specify which managers should receive the sales information for each country. SAS Web Report Studio e-mails the appropriate version of the report to the specified recipients in either PDF or HTML format. Recipients cannot refresh or interact with the report that they receive.

The distributive nature of this feature presents some inherent risks for an organization. For more information about those risks, see "Security Considerations for Pre-generated Batch Reports" on page 314.

The remainder of this topic describes scheduling manually refreshed reports and distributing static snapshot reports.

Required Permissions for Scheduling and Distributing Reports

Users must be assigned to the following roles in order to schedule or distribute reports:

Table 15.19 User Roles for Scheduling and Distributing Tasks

Task	Role Requirement
Schedule a report *	WRS Report Author
Schedule a folder of reports	WRS Advanced User
Specify an archive for reports and control the size of the archive	WRS Advanced User
Distribute a report	WRS Advanced User
Create or delete a recipient list for distributed reports	WRS Administrator

* You can specify that all users must have WRS Advanced User permissions in order to schedule a report by changing the value of **`schedulingRequiresAdvancedUserRole`** to **`false`** in the **`LocalProperties.xml`** file. If you don't have a **`LocalProperties.xml file`**, then you can create one. See "Create a LocalProperties.xml File" on page 325.

For more information about these user roles, see " SAS Web Report Studio Roles" on page 308.

Note: In addition to the above role requirements, you must have configured a *trusted user* (sastrust by default) in order to schedule or distribute reports. You configured the trusted user during installation. The SAS Trusted User is used to establish a trust relationship with the metadata server. The **`OutputManagementConfigTemplate.xml`** file contains the user ID and password for the SAS Trusted User. The **`OutputManagementConfigTemplate.xml`** file resides in the installation directory. △

Prerequisites for Scheduling and Distributing Reports

The scheduling and distribution features use trusted authentication and rely on the SAS Query and Reporting Services, which were configured during installation.

In addition to this initial configuration, you must satisfy the following requirements before users can schedule or distribute reports:

- Before users can schedule or distribute reports, you must have installed Platform LSF (Load Sharing Facility) and SAS scheduling software, as described in Chapter 20, "SAS Scheduling," on page 455. In addition, you must complete the configuration steps that are documented in "Enabling the Scheduling of Reports" on page 462.
- Before users can distribute a report, you must create a library in the metadata repository. The library will contain your distribution recipient list tables. For more information, see "Create a Library for Recipient Lists" on page 341.

- Before users can distribute a report, you must create a recipient list for the report. For more information, see "Creating a Recipient List for Report Distribution" on page 341.

Methods for Scheduling and Distributing Reports

Once you have enabled scheduling (as described in "Prerequisites for Scheduling and Distributing Reports" on page 339), you can schedule or distribute reports using either of these methods:

- In SAS Web Report Studio, report creators can create reports and control when those reports are updated or distributed. For instructions, see the Help for SAS Web Report Studio.

  Administrators can limit access to this scheduling functionality by assigning only selected users to the WRS Advanced User role. For details, see the discussion of user roles in "Setting up Users for SAS Web Report Studio" on page 307.

- In SAS Management Console, administrators can use BI Manager to create a job for scheduling and then use Schedule Manager to schedule the jobs. For instructions, see the Help for each of these plug-ins.

How Report Scheduling Differs From Report Distribution

The processes that you use to schedule and distribute reports are similar in several ways. Both tasks use wizards, and both rely on the **`outputgen.exe`** executable to create output. (The **`outputgen.exe`** tool replaces the **`batchgen.exe`** tool that was included with previous releases.)

However, there are several differences between scheduling and distribution. The following table summarizes these differences:

Table 15.20 Differences Between Scheduling and Distributing Reports

Report Scheduling	Report Distribution
Reports are generated and stored in a repository.	Reports are generated and e-mailed to recipients that you specify in a recipient list. See "Creating a Recipient List for Report Distribution" on page 341.
Reports can be pushed to a publication channel. Publication channels are specified in the Schedule Report wizard.	Reports can be pushed to a publication channel. Publication channels are specified in the recipient list.
The full report is generated.	The full report can be distributed. Alternatively, the report can be divided by group breaks so that each recipient gets a subset of the report.
Users can schedule a folder for report generation. All reports in the folder are generated.	Users can set up distribution for only one report at a time.
Users can enable archiving for a report. If archiving is enabled, the latest report is archived when a new one is generated.	Reports cannot be archived.

Report Scheduling	Report Distribution
Scheduling does not include the ability to preview a schedule.	Users can run a test to preview the distribution of a report. The test returns a recipient list either in the user interface or via an e-mail.
Users who receive the report can refresh and interact with the report.	Users cannot refresh or interact with the report.

Note: A publication channel is an information repository that has been established by using the SAS Publishing Framework in SAS Management Console and which can be used to publish information to users and applications. If you publish your report to a publication channel, then authorized users and applications can access your report by subscribing to the channel. For example, the SAS Information Delivery Portal can list the content of a publication channel. △

Create a Library for Recipient Lists

Depending on how you installed SAS Web Report Studio, you might have been prompted to specify the name of a library to contain your recipient data. Regardless of whether you were prompted for the name, a library path was created on your host machine. You must define that library in the metadata repository before the library can become available for use by SAS Web Report Studio.

Perform the following steps in SAS Management Console:

1. Double-click the **Data Library Manager**. Right-click the **SAS Libraries** icon. Then, select the **New Library** option to access the first screen of the **New Library Wizard**.
2. Select **SAS Base Engine Library** from the list of **SAS Libraries**. Click **Next** to access the next screen of the wizard.
3. In the **Name** field, enter the name of the library that was specified during installation. This value is specified in the **<libname>** element in the **WebReportStudioProperties.xml** file, which is found in the WEB-INF directory of your deployment. Optionally, supply a description of the library, and then click **Next** to access the next screen of the wizard.
4. Provide a libref value for the library, and specify *Base* for the engine. Then provide a path for the library. The path should match the library path that is found in *sas-config-dir***\Lev1\SASMain\appserver_autoexec.sas**. The default path is *sas-config-dir***\Lev1\SASMain\Data\wrsdist**. Click **Next** to access the next screen of the wizard.
5. Select one or more SAS servers. The library is assigned to the server or servers that you select from this list. Click **Next** to access the next screen of the wizard.
6. Examine the finish screen of the wizard to ensure that the proper values have been entered. Click **Finish** to save the settings.

Creating a Recipient List for Report Distribution

Understanding How Recipient Lists Enable Report Distribution

Suppose that you want to distribute an employee salary graph to human resources (HR) representatives in different locations around the world. The following report summarizes salary information for men and women based on company location:

This sample report was created with a group break on a variable named Company. The result is a separate report page for each value of Company (each main division or corporate office location).

In order to distribute the relevant report to each HR representative, you must create a recipient list that maps each Company value to one or more recipients. A recipient list is a SAS table that contains one or more group break values along with e-mail addresses and publication channels. After you create the recipient list, you can schedule the report to be generated and distributed to the specified recipients.

The following SAS data set illustrates a sample recipient list for a report that has a group break on Company:

	Company	EMAIL	CHANNEL
1	Board of Directors		
2	Concession		
3	Logistics		
4	Marketing		
5	Orion Australia		
6	Orion Belgium		
7	Orion Denmark		
8	Orion France		
9	Orion Germany		
10	Orion Holland		
11	Orion Italy		
12	Orion Spain		
13	Orion UK		
14	Orion USA		
15	Purchasing		

As shown in the data set, the EMAIL and CHANNEL columns are empty. You must provide e-mail and channel information for recipients of the distributed report. For details, see "Create a Recipient List" on page 343.

Note: You can create recipient lists for reports that have more than one group break. For more information about nested group levels, see "Considerations for Creating Recipient Lists" on page 346. △

Create a Recipient List

To create a recipient list, you first use SAS Web Report Studio to create an initial list that includes your group breaks. After you create the list, you use Base SAS to provide e-mail addresses and publication channels for that list.

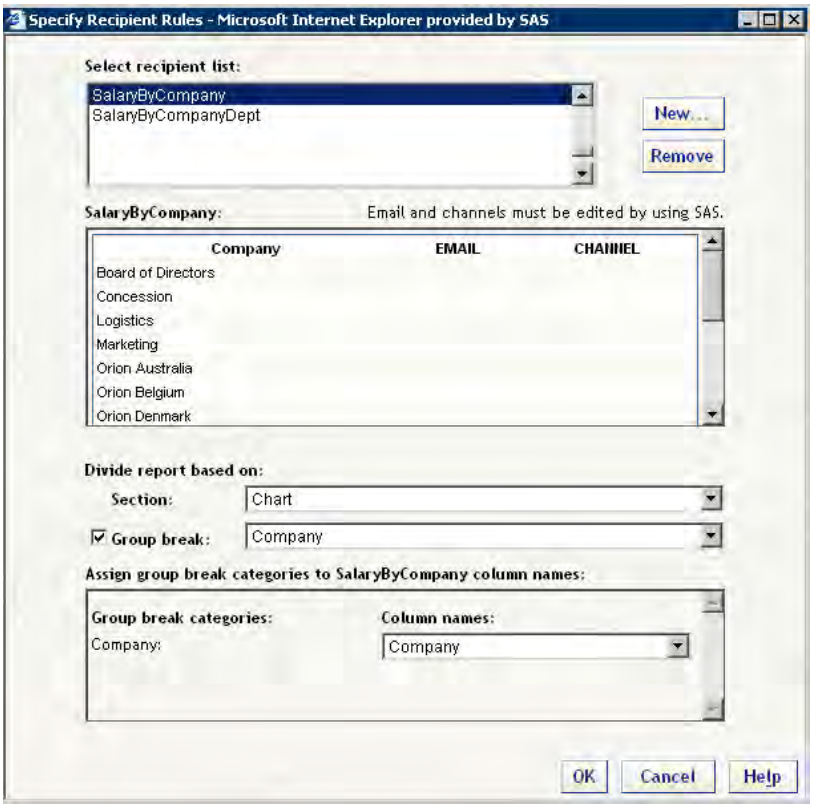
To create a recipient list, follow these steps:

1. Log on to SAS Web Report Studio as someone who has the WRS Administrator role.
2. In SAS Web Report Studio, select the report that you want to distribute.
3. Start the Distribution wizard, and proceed as though you were creating a distribution.

   Note: This documentation does not include instructions for starting the wizard or for creating a distribution. For instructions, see the SAS Web Report Studio online Help. △

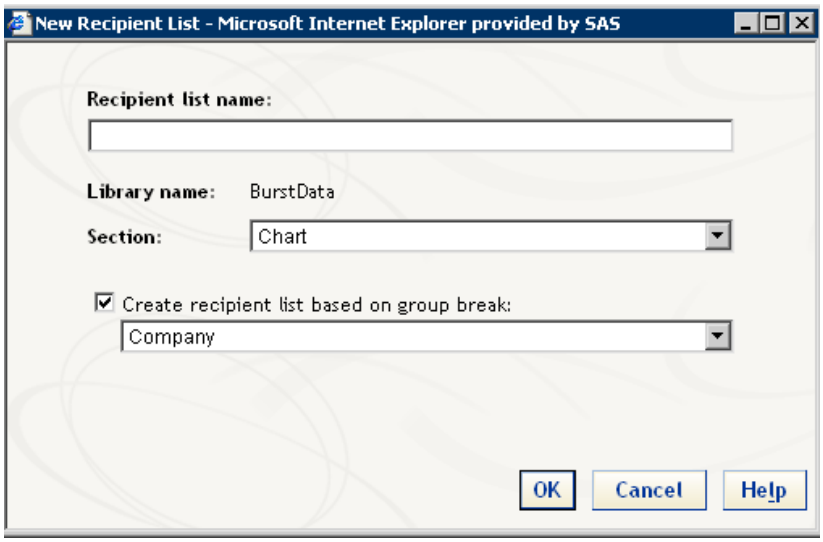
   Here is a summary of the steps that you take in the wizard:

   a. In step 1 of the wizard, specify a date and time.
   b. In step 2 of the wizard, click **`Specify Recipient Rules`**.
4. In the Specify Recipient Rules dialog box, select **`New`**.



The **New** and **Remove** buttons are available only if you are logged on as a member of the WRS Administrator role. Be aware that if you remove a list, any distribution that references the list becomes nonfunctional.

5 In the New Recipient List dialog box, provide information about the recipient list that you want to create.



The following table explains the fields in this dialog box:

Table 15.21 Fields in the New Recipient List dialog box

Field	Description
Recipient list name	The name that you specify for the list of recipients. This name must be a valid SAS data set name.
Library name	The name of the library that you created in metadata for recipient lists. You will need this library name when you subsequently edit the list to add e-mail or channel recipients.
Section	The section that you want to use to subset the report by specifying a group break. This field is available only if the report contains more than one section.
Create recipient list based on group break	The check box indicates whether you want to subset the report. When the check box is selected, you can select the group break that you want to use to subset the report. This field is available only if the section contains a group break.

6 Click **OK**.

7 You can either cancel out of the wizard or continue defining a distribution. Either way, the recipient list has been created as a SAS data set within the specified library.

Next, you must specify the actual recipient e-mail addresses and/or publication channels.

8 In Base SAS, open the data set that corresponds to the recipient list that you just defined. The data set will reside in the library that corresponds to the **Library name** value from the New Recipient List dialog box. You must run SAS from a

system that can access the library location, and then you must assign the library by using the LIBNAME command.

Note: You should run SAS with the VALIDVARNAME= system option set to ANY. The SAS system option VALIDVARNAME controls the type of column names that can be used during a SAS session. For more information, see SAS Help and Documentation.

Note also that the column sizes are fixed. The group break columns are 256kb, and the EMAIL and CHANNEL columns have a fixed name and a fixed size of 1024kb. Do not change the name of the EMAIL and CHANNEL columns. △

9 In the data set, provide the e-mail addresses and/or publication channels that are appropriate for the recipients.

Note the following about e-mail addresses and publication channels:

- The e-mail address must be in the form **`John.Doe@abc.com`**. Use this format also if you specify a distribution list (for example, **`hr.mylist@abc.com`** as opposed to **`My List`**) .
- SAS Web Report Studio does not check the validity of the e-mail address that you provide. You are responsible for ensuring that every e-mail address is valid.
- To specify multiple e-mail addresses in a single row, delimit the e-mail addresses with a comma.
- To specify multiple publication channels, delimit the channel names with a comma.

If you specified a group break for the report, then the data set will be grouped according to the level that is associated with the group break. In the example that is shown in "Understanding How Recipient Lists Enable Report Distribution" on page 341, you would specify the e-mail address and/or the publication channel for a recipient in each Company.

You can later return to the recipient data set and make additional changes. Note, however, that on UNIX, the data set assumes its permissions from the administrator who created the data set. Often, the administrator's permission mask is set to restrict "group" and "other" from write permission. If someone other than the file's creator wants to edit the data set, then that person might not have the required permissions. The person can either use the same account name as the data set creator, the creator can change the mask, or the creator can change the permissions that are assigned to the data set.

As an alternative to creating the recipient list in SAS Web Report Studio, you can create a recipient table in Base SAS, for example by using a DATA step or PROC SQL. You might want to do this if you already have a list of e-mail addresses in a mail directory, and you want to import those addresses into the table. The next section illustrates a sample program.

Alternative Example: Create a Recipient List Using PROC SQL

As an alternative to creating the recipient list in SAS Web Report Studio, you can create a table manually in Base SAS. This example illustrates one way to create a table manually using PROC SQL. After you create the table, you must import the table into your metadata repository by using the Data Library Manager in SAS Management Console.

The example uses a library that is named OMData, a SAS table named Burst, and a group break variable named Year. You will need to change these values as applicable for your environment. For the library name, use the value that is found in the output-generation configuration file (**`OutputManagementConfigTemplate.xml`**) that you

created when you configured the scheduling of reports, as instructed in "Prerequisites for Scheduling and Distributing Reports" on page 339.

```
libname OMData ''\\server\c$\DataSources\SAS\OMData'';
proc sql;
create table OMData.Burst (Year num, EMAIL char(256), CHANNEL char(256));
insert into OMData.Burst
values (2000, 'email1@abc.com', 'channelname')
values (2001, 'email2@abc.com', 'channelname')
values (2002, 'email3@abc.com', 'channelname')
;
quit;
```

After you have created the SAS table for your environment, you must import the table into metadata.

Here is a summary of the steps to import the table into metadata:

1. In SAS Management Console, expand the Data Library Manager until you reach the library that you specified in the LIBNAME instruction (in this example, OMData).
2. Right-click the library name and choose **Import Tables**.
3. Follow the prompts to provide information about the table that you created (in this example, Burst).

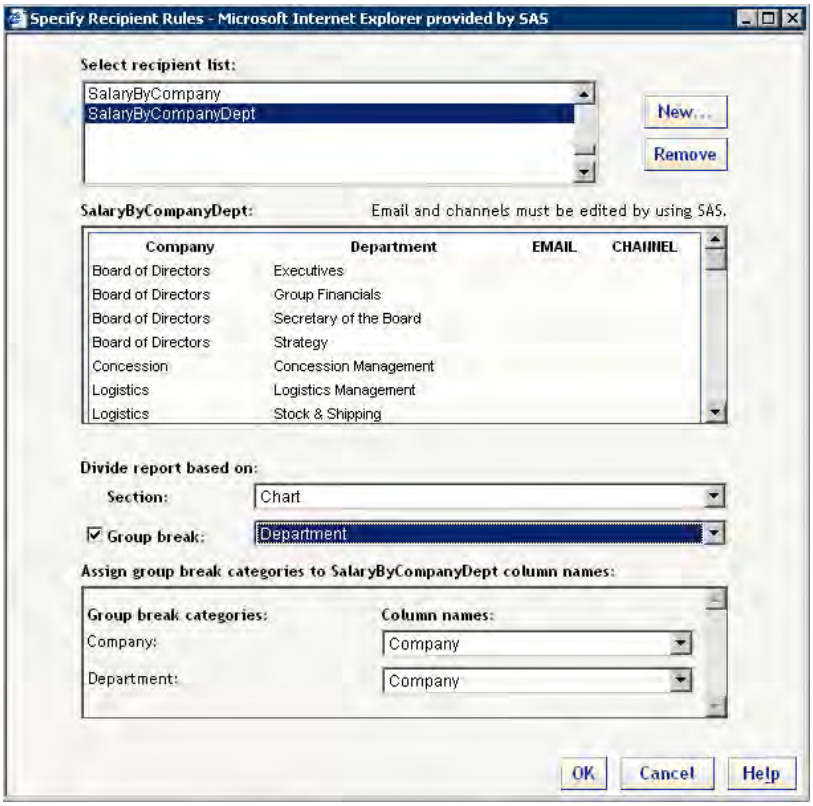
For complete instructions on importing a table, see the Help for the Data Library Manager.

Considerations for Creating Recipient Lists

Here are some things to consider when you create recipient lists:

- Although each recipient list is based on a report, a single recipient list can be used to distribute more than one report. When you use a recipient list for multiple reports, you reduce the overall number of recipient lists that must be created and maintained.

  For example, the sample recipient list shown earlier could have been created with multiple group break levels, as seen here:



When you open the table in SAS, you see something like this:

	Company	Department	EMAIL	CHANNEL
8	Marketing	Marketing		
9	Orion Australia	Administration		
10	Orion Australia	Engineering		
11	Orion Australia	Sales		
12	Orion Australia	Sales Management		
13	Orion Belgium	Administration		
14	Orion Belgium	Engineering		
15	Orion Belgium	Sales		
16	Orion Belgium	Sales Management		
17	Orion Denmark	Administration		
18	Orion Denmark	Engineering		
19	Orion Denmark	Sales		
20	Orion Denmark	Sales Management		
21	Orion France	Administration		
22	Orion France	Engineering		

In this example, the recipient list can still be used for the report that is grouped by Company. In addition, the list can be used for any report that is grouped by Company and then by Department. In this example, you would specify the e-mail address and/or the publication channel for a recipient in each department.

- There is no limit to the number of group levels that you can use in a distribution. Note, however, that each new level exponentially increases the number of recipient e-mails to define.

  You can also have a recipient list with no group break column. This means that you will distribute the entire report to all recipients that are specified in the EMAIL or CHANNEL columns.

- You can leave recipient EMAIL or CHANNEL cells empty for some or all of the group breaks. When a row has empty EMAIL and CHANNEL cells, the corresponding group break report is not generated.

- Users will select the proper recipient list when they create a distribution. For that reason, you should provide descriptive names for your recipient lists. SAS Web Report Studio doesn't validate the relationship between group break columns in a recipient list and group breaks in the corresponding report.
- SAS Web Report Studio stores all the lists in a single library that you defined during installation. Use a naming convention that makes sense for your organization and that prevents collisions in the event that multiple administrators create lists.

Additional Documentation for SAS Web Report Studio

- Chapter 19, "Best Practices for Configuring Your Middle Tier," on page 417 contains information that is associated with middle-tier administration.
- SAS Web Report Studio online Help provides task instructions and information about the user interface.
- *SAS Web Report Studio: User's Guide*, which is available from within SAS Web Report Studio (the Help link).
- *SAS Web Infrastructure Kit: Administrator's Guide* at **`http://support.sas.com/rnd/itech/library/toc_portaladmin.html`** includes information to help you integrate SAS Web Report Studio with the SAS Information Delivery Portal.

CHAPTER

16

Administering SAS Web OLAP Viewer for Java

Introduction to Administering SAS Web OLAP Viewer for Java

SAS Web OLAP Viewer for Java provides a Web interface for viewing and exploring OLAP data. SAS Information Delivery Portal invokes SAS Web OLAP Viewer for Java when a portal user attempts to view OLAP data. SAS Web OLAP Viewer for Java can also be accessed by itself directly from a browser.

This documentation assumes that you have successfully installed and configured SAS Web OLAP Viewer for Java. There are two methods by which you might have performed your installation:

- A planned installation (personal or advanced) uses information from a planning document as input to the SAS Software Navigator and the SAS Configuration

Wizard. For this type of installation, you should follow all of the post-installation steps that are provided in the **instructions.html** file that is generated by the SAS Configuration Wizard. In addition, you must deploy SAS Web OLAP Viewer for Java using the instructions that are provided in the **config.pdf** file. The **config.pdf** file resides in your installation directory.

- An installation using the Software Index is performed without the use of plans and SAS project directories. For this type of installation, you should follow all of the post-installation configuration and deployment steps that are provided in the **config.pdf** file.

For a comprehensive overview of installation, see the *SAS Intelligence Platform: Installation Guide*. For instructions on using the SAS Web OLAP Viewer for Java interface, see the product's online Help.

Requirements for Viewing OLAP Cubes in SAS Web OLAP Viewer for Java

SAS Web OLAP Viewer for Java does not render SAS OLAP cubes directly. Instead, SAS Web OLAP Viewer for Java renders multi-dimensional SAS Information Maps that have been created from OLAP cubes. Information maps can be created in two ways:

- You can create information maps in SAS Information Map Studio.
- If a user attempts to view an OLAP cube directly from the SAS Information Delivery Portal, then SAS Web OLAP Viewer for Java generates an information map for that cube at run time.

In order for an information map to be displayed in SAS Web OLAP Viewer for Java, the information map must meet all of these criteria:

- The OLAP cube and the information map must exist in the same foundation repository that the user of SAS Web OLAP Viewer for Java is accessing.
- The user of SAS Web OLAP Viewer for Java must have ReadMetadata and Read permission for the information map.

  Note: Beginning with Service Pack 4, Read permission for an information map is required in order to access data through that information map. For details, see "BI Row-Level Permissions" in the "Access Management" section in the *SAS Intelligence Platform: Security Administration Guide* document. △

- If you upgraded from a version earlier than SAS Web OLAP Viewer for Java 3.1, then you might need to upgrade your existing information maps. For details, see "Upgrade Information Maps to the SAS Information Map Studio 3.1 Format" on page 350.

Upgrade Information Maps to the SAS Information Map Studio 3.1 Format

If you have OLAP information maps that were created with SAS Web OLAP Viewer for Java 1.2 or SAS Information Map Studio 1.0.1, then those information maps must be updated to the new SAS Information Map Studio 3.1 format.

Updates can occur in two ways:

- If the source cube is more recent than the existing information map, then SAS Web OLAP Viewer for Java automatically generates an updated map from the source cube.

- If the source cube is as old as the existing information map, then you must manually update the information map.

To update one or more information maps manually, follow these steps:

1. Open SAS Information Map Studio 3.1 and connect to the server and repository that contains the information maps that you want to modify.
2. From the menu bar, select **Tools ▶ Administrative Tools**.
3. Select the **`Metadata Format`** tab, and browse to the metadata repository folder that contains the information maps that require updates.
4. Click **`Run`**. The tool will be run on all of the information maps that you have permission to access in the specified folder and its subfolders.

Change the HTTP Session Timeout Interval

By default, SAS Web OLAP Viewer for Java uses the session timeout interval that is specified in your servlet container configuration. You can specify a different timeout interval for SAS Web OLAP Viewer for Java by modifying the **`web.xml`** file. This file is located in **`SASWebOLAPViewer/WEB-INF`** under the root installation directory for SAS Web OLAP Viewer for Java.

To specify a session timeout interval for SAS Web OLAP Viewer for Java, modify the following code in the **`web.xml`** file:

```
<session-config>
<session-timeout>timeout-interval</session-timeout>
</session-config>
```

In the previous code, *timeout-interval* specifies the timeout interval in minutes. As a recommendation, the number should be no smaller than 5.

Configure Logging

You can use SAS Web OLAP Viewer for Java log files to help manage performance, track security enforcement, and analyze specific situations. You can record events such as application use, failed attempts to log on, and other events.

Various contexts and outputs are created by default. You can control the level of logging messages by changing the logging priority for a particular context. Priority levels include DEBUG, INFO, WARN, ERROR, and FATAL. The default level is WARN. The DEBUG level is useful for troubleshooting, but is also very verbose and is not recommended for a production environment.

To configure logging, perform these steps:

1. In SAS Management Console, navigate to **Foundation Services Manager ▶ SAS Web OLAP Viewer Local Services ▶ BIP Core Services**.
2. Select **`Platform Logging Service`**.
3. From the menu bar, select **File ▶ Properties**.
4. In the **`Service Configuration`** tab, click **`Edit Configuration`**. The Logging Service Configuration dialog box is displayed. There are three possible predefined logging contexts:
   - The **`RootLoggingContext`** context processes all logging messages for a selected priority.
   - The **`com.sas.services`** context processes service-related logging messages.

- The **com.sas.webapp.dataexplorer** context processes all SAS Web OLAP Viewer for Java specific messages.

The Outputs list boxes show the available and the selected outputs for the logging contexts. By default, the output for all contexts is to the console. A log file context is also available, but is not selected by default.

5 To edit a context, select the context, click **Edit**, and make your changes. For example, you can do the following:
   - Select a different priority for the context
   - Enable log file output for the context by moving the appropriate output from the **Available** to the **Selected** column.

6 To change an output, in the **Outputs** tab select the output that you want to change, and then click **Edit**. For file outputs, you can specify the path to the logging file. You can also control the format of your logging. For more information, click **Help**.

7 You can also modify the contexts by assigning new priority levels, adding new outputs, or changing other properties. For more information, click **Help** in the properties dialog box.

Trusted Web Authentication

By default, SAS Web OLAP Viewer for Java relies on the SAS Metadata Server to authenticate users. However, you can configure trusted Web authentication instead. When you configure Web authentication, SAS Web OLAP Viewer for Java obtains authentication information from an HTTP server, a J2EE application server, or a servlet container.

Note: If you configure Web authentication for SAS Information Delivery Portal, then you should also configure Web authentication for SAS Web OLAP Viewer for Java. △

For instructions on configuring Web authentication, see "Changing from Host to Trusted Web Authentication" in the *SAS Web Infrastructure Kit: Administrator's Guide* at **http://support.sas.com/rnd/itech/library/toc_portaladmin.html**.

For more information about authentication, see the *SAS Intelligence Platform: Security Administration Guide*.

Enabling ESRI Maps

The ESRI map component is a feature of SAS Web OLAP Viewer for Java that enables you to plot your OLAP data onto an interactive ESRI map. With an ESRI map, you can zoom, subset, expand the map regions, and get detailed values.

For information about enabling and administering ESRI maps, see Chapter 23, "Configuring the ESRI Map Component," on page 503.

Customizing the SAS Web OLAP Viewer for Java Display

You can configure SAS Web OLAP Viewer for Java with custom properties for tables, plots, charts, and ESRI maps, column layout, header and footer styles, and more. You can also configure the interface to display a default information map or data exploration.

You customize SAS Web OLAP Viewer for Java by modifying the **WebOLAPViewerConfig.xml** file, which is located in the **SASWebOLAPViewer\WEB-INF** directory of the installation. Instructions for customizing that file are presented here. Instructions can also be found as comments within the file.

Applying Your Changes

After you modify the **WebOLAPViewerConfig.xml** file, your changes won't take effect until you update and redeploy SAS Web OLAP Viewer for Java. For instructions, see "Redeploy SAS Web OLAP Viewer for Java" on page 360.

Specify a Default Data Source

You can specify a default data source that will be displayed when SAS Web OLAP Viewer for Java opens. The default data source can be either an information map or a data exploration.

Note: If you specify both a default data exploration and a default information map, the data exploration is used. △

Specify an Information Map

To display a default information map, customize the **pathURL** attribute of the **<InformationMap>** element.

Use the following format to specify a fully-qualified path to the information map:

```
SBIP://repository-name/path-to-map
```

You can optionally specify a custom query for the information map by defining **<DataItem>** elements within the **<Rows>**, **<Columns>**, and **<Slicer>** elements. If you do not specify a custom query, a default query is generated for the information map.

You can also specify one or more predefined filters for the information map by using the **<Filters>** element (located outside of the **<InformationMap>** element).

For example, the following code specifies a default information map, a custom query, and a filter:

```
<InformationMap pathURL=
  "SBIP://Foundation/BIP Tree/ReportStudio/Maps/SampleMap" emptyQuery=false>
     <Rows>
        <DataItem label="Geography"/>
        <DataItem label="Sum of Sales"/>
        <DataItem label="Average Sales"/>
     </Rows>
     <Columns>
        <DataItem label="Product"/>
     </Columns>
     <Slicer></Slicer>
</InformationMap>
<Filters>
     <Filter label="myFilter">
</Filters>
```

The **emptyQuery** property enables you to specify that an empty query will be displayed in the user interface. When you change the value to **true**, a default query will not be generated for the information map. End users will have to create their initial query in the user interface. Note, if **emptyQuery** is set to true and a query is also

defined, the query will be ignored; the **emptyQuery** property overrides the specified query.

Specify a Data Exploration

To specify a default data exploration, modify the **activeDataExplorationPathURL** attribute of the **<DataExplorations>** element.

Use the following format to specify a fully-qualified path to a data exploration:

```
SBIP://repository-name/path-to-exploration
```

You can optionally specify a particular bookmark from the data exploration by using the **activeBookmarkName** attribute. If you do not specify a bookmark, the default bookmark for the data exploration is displayed.

The following sample code specifies a default data exploration and bookmark:

```
<DataExplorations activeDataExplorationPathURL=
  "SBIP://Foundation/BIP Tree/Users/saswbadm/SampleDE"
         activeBookmarkName="SampleBookmark">
</DataExplorations>
```

After you change the **WebOLAPViewerConfig.xml** file, you must redeploy SAS Web OLAP Viewer for Java.

Specify Common Public Data Explorations

To specify data explorations that will be available to all users in the Bookmarks panel, modify the **<DataExploration>** element within the **<DataExplorations>** element.

Use the following format to specify a fully-qualified path to a data exploration:

```
SBIP://repository-name/path-to-exploration
```

For example, the following code specifies two data explorations:

```
<DataExplorations>
     <DataExploration pathURL=
       "SBIP://Foundation/BIP Tree/Users/saswbadm/SampleDE1">
     <DataExploration pathURL=
       "SBIP://Foundation/BIP Tree/Users/saswbadm/SampleDE2">
</DataExplorations>
```

In this example, all users will be able to access SampleDE1 and SampleDE2 from the Public Data Explorations group in the Bookmarks panel.

After you change the **WebOLAPViewerConfig.xml** file, you must redeploy SAS Web OLAP Viewer for Java.

Customize the Default Display for Viewers

The **<Viewers>** element determines which viewers will be displayed initially in the content area. The **layoutStyle** attribute controls the layout of these viewers. In addition, the number of visible rows and columns can be defined for the table viewers. If a data exploration has been specified, then its viewers will take precedence over those defined here. By default only the table is displayed, and the default layout style is one column.

To specify the default display for the data viewers, modify the elements that correspond to each viewer:

Table 16.1 Viewer Elements

Element	Viewer
<Table>	table viewer
<BarChart>	bar chart viewer
<ColorMappedTable>	color-mapped table viewer
<PieChart>	pie chart viewer
<ScatterPlot>	scatter plot viewer
<LineChart>	line chart viewer
<BarLineChart>	barline chart viewer
<TileChart>	tile chart viewer
<ESRIMap>	ESRI map viewer
<AppliedFilters>	applied filters viewer
<DrillPath>	drill path viewer

The data viewer properties are located within the **<Viewers>** element, and the **<AppliedFilters>** and **<DrillPath>** elements are located outside of the **<Viewers>** element.

For all of the viewer elements, the **visible** attribute specifies whether the viewer is displayed or hidden by default. If the value for **visible** is **true**, then the viewer is displayed by default. If the value for **visible** is **false**, then the viewer is hidden by default.

For the table and color-mapped table, you can also the specify number of rows and columns that are displayed by default.

For example, the following code specifies that the color-mapped table displays 10 columns and 25 rows. The bar chart viewer, color-mapped table viewer, applied filters viewer, and drill path viewer are displayed, and the other viewers are hidden.

```
<Viewers>
     <Table numberOfColumns="5" numberOfRows="20" visible="false"/>
     <BarChart visible="false"/>
     <ColorMappedTable numberOfColumns="10" numberOfRows="25"
         visible="false"/>
     <PieChart visible="false"/>
     <ScatterPlot visible="false"/>
     <LineChart visible="false"/>
     <BarLineChart visible="false"/>
     <TileChart visible="false"/>
     <ESRIMap visible="false"/>
</Viewers>

<AppliedFilters visible="true"/>
<DrillPath visible="true"/>
```

After you change the **WebOLAPViewerConfig.xml** file, you must redeploy SAS Web OLAP Viewer for Java.

Specify the Column Layout

You can specify the column layout for your data viewers by setting the **`layoutStyle`** attribute of the **`<Viewers>`** element. Specify one of the following values:

Table 16.2 Column Layout Attributes

Attribute	Description
ONE_COLUMN_LAYOUT	specifies that the viewers are arranged in a single column
TWO_COLUMN_LAYOUT	specifies that the viewers are arranged in two columns

After you change the **`WebOLAPViewerConfig.xml`** file, you must redeploy SAS Web OLAP Viewer for Java.

Specify the Default Panel

The **`<StartPanel>`** element controls which panel is initially displayed. The panel is the portion of the application that is displayed along the left side of the browser. By default, the Query Panel is displayed.

You can specify the default panel by using the **`<StartPanel>`** element. Specify one of the following values:

Table 16.3 <StartPanel> Element Attributes

Attribute	Description
BOOKMARK_PANEL	specifies that the Bookmarks panel is displayed by default
NAVIGATION_PANEL	specifies that the Navigator panel is displayed by default
QUERY_PANEL	specifies that the Query panel is displayed by default

Note: An invalid value will result in no panel being displayed. △

After you change the **`WebOLAPViewerConfig.xml`** file, you must redeploy SAS Web OLAP Viewer for Java.

Customize the Header and Footer Styles

By default, the header consists of the SAS banner. There is no default footer.

Customize the Main Header

You can customize the default page header by editing the **`<Header>`** element. You can edit the following elements within **`<Header>`**:

Table 16.4 <Header> Elements

Element	Description
<BackGroundImage>	The background image for the header. The default image is a blue SAS background.
<Image>	A logo image that is displayed in the top right corner of the header. The default image is a SAS logo. You can specify either a URL or the name of an image that is stored in the images subdirectory of your SAS Web OLAP Viewer for Java deployment.
<Text>	The main text for the header. This element specifies any static text that you want to show in the banner. The default text is "SAS Web OLAP Viewer."
<SecondaryText>	Secondary text that follows the value of **`<Text>`**. This text can display the name of the information map or data exploration that is currently open. The **`useDynamicText`** attribute specifies whether the value of **`SecondaryText`** is generated from the name of the data source that you are viewing.
<TemplateFileName>	An HTML file that provides the structure of the header. The template file must reside in the templates subdirectory of the installation. If no file is specified, the default template is VisualDataExplorerHeader.html.

Note: Images are stored in the **`images`** subdirectory under the SAS Web OLAP Viewer for Java deployment in the servlet container. △

After you change the **`WebOLAPViewerConfig.xml`** file, you must redeploy SAS Web OLAP Viewer for Java.

Specify a Main Footer

You can specify a page footer by customizing the **`<Footer>`** element. There is no default footer.

You can edit the following elements:

Table 16.5 <Footer> Elements

Element	Description
<BackGroundImage>	The background image for the footer.
<Image>	A logo image that is displayed in the top right.
<Text>	The main text for the footer.

Element	Description
<SecondaryText>	Secondary text that follows the value of **<Text>**. The useDynamicText attribute specifies whether the value of **<SecondaryText>** is generated from the name of the data source that you are viewing.
<TemplateFileName>	An HTML file that provides the structure of the footer. The template file must reside in the templates subdirectory of the installation. If no file is specified, then the default template is **VisualDataExplorerFooter.html**.

Specify an Alternate Header and Footer for Exporting and for Printing

You can specify an alternate header and footer that are used when you export to Excel or create a PDF file for printing.

To specify an alternate header and footer, customize the following elements within the **<AlternateHeader>** and **<AlternateFooter>** elements:

Table 16.6 Alternate Header and Footer Elements

Element	Description
<Image>	An image for the alternate header or footer.
<SecondaryImage>	A second image for the alternate header or footer.
<Text>	The main text for the alternate header or footer.
<SecondaryText>	Secondary text for the alternate header or footer. The **useDynamicText** attribute is only available in the **<AlternateHeader>** element, and it specifies whether the value of **<SecondaryText>** is generated from the name of the data source you are viewing. The **newLine** attribute specifies whether the value of **<SecondaryText>** is displayed on a new line beneath the value of **<Text>**.

For the previous elements, you can specify the following attributes:

Table 16.7 Attributes for Alternate Header and Footer Elements

Attribute	Description
hAlignment	Specifies the horizontal alignment for the item. Specify either left, center, or right.
vAlignment	Specifies the vertical alignment for the item. Specify either top, middle, or bottom.
column	Specifies which column the item is placed in.
row	Specifies which row the item is placed in.

Change the Personal Data Explorations Folder

To change the root folder in metadata for personal data explorations, edit the **<PersonalDataExplorationFolder>** element in the **WebOLAPViewerConfig.xml** file. The root folder that you specify will contain subdirectories for each user, in the following format: /user-name/Data Explorations.

Here is the default value:

```
SBIP://repository-name/BIP Tree/Users
```

After you change the **WebOLAPViewerConfig.xml** file, you must redeploy SAS Web OLAP Viewer for Java.

Customize the Open Dialog Box

You can customize the behavior of the Open dialog box by using the **<FileOpen>** element.

Specify a Default Initial Path

You can specify a default initial path by using the **<InitialPath>** element in the **WebOLAPViewerConfig.xml** file. The default initial path is used when you are not currently viewing a data source.

Use the following format to specify the initial path:

```
SBIP://repository-name/path-name(Folder)
```

The **(Folder)** string at the end of the path is required.

For example, the following code fragment specifies a default initial path:

```
<FileOpen>
     <InitialPath>
       SBIP://Foundation/BIP Tree/ReportStudio/Maps (Folder)
     </InitialPath>
     . . .
</FileOpen>
```

Customize Available File Types

You can specify the file types that are available from the Open dialog box by using the **<FileTypes>** element.

The **<Cubes>** element specifies whether cubes are available, and the **<OLAPMaps>** element specifies whether information maps are available. For each element, the display attribute specifies whether the data type is available from the Open dialog box.

Note: Data explorations are always available from the Open dialog box. △

For example, the following code specifies that information maps are available and that cubes are not available:

```
<FileOpen>
   <InitialPath></InitialPath>
   <FileTypes>
       <Cubes display="false"/>
       <OLAPMaps display="true"/>
    </FileTypes>
</FileOpen>
```

After you change the **WebOLAPViewerConfig.xml** file, you must redeploy SAS Web OLAP Viewer for Java.

Enable the Theme That Is Used for the Current Portal User

A theme is a collection of specifications (for example, colors, fonts, and font styles) and graphics that control the appearance of an application. By default, SAS Web OLAP Viewer for Java uses the default theme that is used by SAS Information Delivery Portal. You can configure SAS Web OLAP Viewer for Java to use theme resources that are in effect for the particular user who is logged on.

To enable the theme for the current user's profile, in the **WebOLAPViewerConfig.xml** file, change the **<SASThemes enabled="false"/>** element to **<SASThemes enabled="true"/>**.

After you make this change, you must redeploy SAS Web OLAP Viewer for Java before the change takes effect.

Disable the Logoff Link

By default, SAS Web OLAP Viewer for Java displays a **Log Off** link in the banner, but you can disable the link. You might want to disable the logoff link when SAS Web OLAP Viewer for Java is accessed from SAS Information Delivery Portal, and you prefer that users log off from the portal. You could disable the logoff link and replace it with a link that returns the user to the portal.

To disable the logoff link, in the **WebOLAPViewerConfig.xml** file, change the **<LogoffButton visible="true" url="logoff.do"/>** element to **<LogoffButton visible="false" url="logoff.do"/>**.

After you make this change, you must redeploy SAS Web OLAP Viewer for Java before the change takes effect.

Improving the Performance of SAS Web OLAP Viewer for Java

There are a few administration tasks that you can perform to improve overall performance. Most administration tasks are performed in conjunction with other SAS Web applications, such as SAS Information Delivery Portal.

To optimize the performance of SAS Web OLAP Viewer for Java, you can do the following:

- Configure the application's static content to be served from an HTTP server, such as Apache HTTP Server. This will reduce the number of requests made to the J2EE application server. For more information, see "Configuring Apache HTTP Server to Serve Static Content for SAS Web Applications" on page 440.
- Deploy the application as part of a cluster configuration. See "Configuring a Cluster of J2EE Application Servers" on page 439.

For more information about middle-tier deployment configurations and performance enhancements, see Chapter 19, "Best Practices for Configuring Your Middle Tier," on page 417.

Redeploy SAS Web OLAP Viewer for Java

After initial installation and configuration, if you make changes to your configuration, then you must re-create and redeploy SAS Web OLAP Viewer for Java.

For example, changes that you make to **`WebOLAPViewerConfig.xml`** require that you redeploy SAS Web OLAP Viewer for Java.

To redeploy SAS Web OLAP Viewer for Java, follow these steps:

1. Run the configure script (**`configure.bat`** or **`configure.sh`**) that is located in your root installation directory. This updates the **`SASWebOLAPViewer.war`** file.
2. Deploy the updated WAR file into your J2EE application server or servlet container. Follow the deployment instructions for your platform that are described in the **`config.pdf`** file.

CHAPTER

17

Preparing SAS Enterprise Miner for Use

Overview of Preparing SAS Enterprise Miner for Use

SAS Enterprise Miner 5.2 is the first and only data mining solution that addresses the entire data mining process. Combined with SAS data warehousing and OLAP technologies, SAS Enterprise Miner helps companies reveal trends, explain known outcomes, predict future outcomes, and identify factors that can produce a desired effect.

There are two ways to deploy SAS Enterprise Miner:

- *Personal workstation.* In this deployment, Java files, configuration files, and documentation are stored locally on the client computer. The client communicates directly with the SAS Workspace Server and the SAS Metadata Server and must remain connected for the duration of a model training event. The personal workstation installation is appropriate for single-user configurations.
- *Enterprise client and Analytics Platform server.* The SAS Enterprise Miner enterprise client uses only the Java files that are needed for display on the client

computer. The other components of Enterprise Miner run within an Analytics Platform server, to which Enterprise clients can connect. All configuration is managed by the shared Analytics Platform installation, thus eliminating the need for any client configuration. This installation also facilitates multiple users working on projects collaboratively. Users can work in the same project, disconnect and reconnect to model training processes, and share mining results packages without experiencing resource conflicts.

As a best practice, you can deploy the SAS Enterprise Miner client using Java Web Start, which enables you to deliver enterprise client files on demand from a link in a Web document. Java Web Start automatically downloads the most recent versions of required files from an application server and stores them on the client computer. If the Analytics Platform is configured to run its embedded HTTP server, then the Java Web Start feature is enabled automatically.

Note: For more information about how to deploy SAS Enterprise Miner, see the SAS Enterprise Miner 5.2 Help. △

During a planned SAS installation, a SAS Metadata Server, an object spawner, and a SAS Workspace Server are defined and configured to be available for use by SAS Enterprise Miner. The deployment process also creates a foundation metadata repository and some initial users.

In addition, if this is not a personal workstation installation, you must perform the following tasks:

- create additional SAS Enterprise Miner users
- secure the metadata definitions for projects and models that are created by SAS Enterprise Miner users

Note: You might need to configure the Apache Tomcat HTTP Server for use with the SAS Enterprise Miner model repository. △

Configuring SAS Enterprise Miner

Enterprise Miner provides some configuration tools and files that allow you to customize the behavior of the application. All configuration is managed centrally. In the case of a personal desktop installation, this means that all of the configuration is managed on your desktop system. In the case of a multi-user installation, Enterprise Miner configuration is managed on the server hosting the remote Enterprise Miner components via the SAS Analytics Platform server. Clients connecting to an Analytics Platform server do not require individual configuration.

Enterprise Miner configuration can be divided into two parts:

1. configuration of Enterprise Miner features
2. configuration of the SAS Analytics Platform, which provides the infrastructure for Enterprise Miner, and other analytical applications

Normally, the SAS Analytics Platform will already be configured after installation has completed. However, there may be cases in which it is necessary to reconfigure it using its own configuration wizard. For more information about the SAS Analytics Platform, see the *SAS Analytics Platform Administrator's Guide*. For more information about how to customize some metadata objects for SAS Enterprise Miner, see "Customizing SAS Enterprise Miner Metadata by Using the Enterprise Miner Plug-in for SAS Management Console" on page 367.

Enterprise Miner Macro Variables

Enterprise Miner can be further customized by the use of some SAS macro variables in the Enterprise Miner project startup code.

Table 17.1

Macro Variable	Default Value	Description
EM_DECMETA_MAXLEVELS	32	Controls the maximum number of levels for a target variable for which a target profile can be built.
EM_TRAIN_MAXLEVELS	512	Controls the maximum number of levels for a targert variable when building a Data Mining Database (DMDB).
EM_PMML	N	Controls the generation of Predictive Modeling Markup Language (PMML) score code. When this macro variable is set to Y (or YES) the PMML score code is generated by nodes that support it (Association, Clustering, Tree, Regression, Neural, Autoneural).
EM_EXPLOREOBS_MAX		Controls the maximum number of observations downloadable for visualization. When this value is blank, the maximum number of rows downloaded is based on the record length.
EM_EXPLOREOBS_DEFAULT		Controls the default number of observations downloadable for visualization. When this value is blank, the default number of rows downloaded is based on the record length.
EM_DISPLAY_UDF	N	If set to Y, then any user-defined formats found in the format search path are downloaded whenever the project is open, and are subsequently used when displaying data for tables that reference those formats. Otherwise, only raw data values are displayed. Once downloaded, the formats are used for the remainder of your Enterprise Miner session, even though you may later set this macro variable to N. However, if the macro variable is set to Y, the download occurs every time you open the project in order to ensure that the downloaded copies are up to date.

Enterprise Miner Install Directory

The Enterprise Miner install directory is located in the SAS Analytics Platform *applications* directory. If the Analytics Platform is installed at the location **`C:\Program`**

Files\SAS\SASAPCore, then the Analytics Platform applications directory is **C:\Program Files\SAS\SAS\APCore\apps** and the Enterprise Miner install location is **C:\Program Files\SAS\SASAPCore\apps\EnterpriseMiner**.

Java Virtual Machine Properties

A number of properties can be used to customize some aspects of Enterprise Miner behavior. These properties are passed to the Java Virtual Machine as options to the Java command used to launch Enterprise Miner. If you are launching Enterprise Miner using the provided Java Web Start link available from the Analytics Platform status page, you will need to set these properties in the Enterprise Miner configuration file (see the following section). Otherwise, you can edit either the em.ini file (on Windows) or the em launch script (UNIX) to add the properties to the Java command. Either file can be found in the Enterprise Miner *executables* directory called **bin**, which is an immediate sub-directory of the Enterprise Miner install directory.

Table 17.2 Java Virtual Machine Properties

Property	Default Value	Description
em.maxlines	20,000	Maximum number of lines of text fetched from the SAS server before truncation occurs. This value is used, for example, in the log and output windows of the results viewers to avoid out-of-memory conditions that may otherwise occur.
em.appserver.url	none	The default Analytics Platform host (normally used for Java Web Start deployments — see the following section).

The Enterprise Miner Configuration File

Located in the install directory is a file called app.config that controls some properties of Enterprise Miner for all clients.

You can edit the app.config file by using a text editor to customize the following properties.

Java Web Start Properties

These properties control how clients are launched from the Analytics Platform status page using Java Web Start. The embedded HTTP server for the Analytics Platform must be enabled in order to provide Java Web Start as a deployment option.

Table 17.3 Java Web Start Properties

Property	Description
webstart.client.initial.heap.size	Controls the initial amount of heap memory allocated to clients
webstart.client.max.heap.size	Controls the maximum amount of heap memory allocated to clients
webstart.client.vm.properties	Initial properties that can be passed to the Java Virtual Machine (JVM) when clients are launched. The properties are specified using the following syntax: *-Dproperty-name=property-value* For example: -Dem.maxlines=10000

By default the property em.appserver.url is set as follows:

```
-Dem.appserver.url=rmi://$RMI_HOST$:$RMI_PORT$
```

Whenever a request is made from a web browser to launch Enterprise Miner using Java Web Start, this property is updated with the host name and port of the Analytics Platform server serving the request, to form a valid RMI-based URL. For example, if the Analytics Platform server is called myhost.mydomain.com and the RMI registry port is 5099, then the server resolves this property as follows:

```
-Dem.appserver.url=rmi://myhost.mydomain.com:5099
```

RMI_HOST and RMI_PORT are replaced with the actual host name and port. This property informs the Enterprise Miner client of the location of the RMI registry to contact in order to establish a link with the Analytics Platform server.

Customizing SAS Enterprise Miner Metadata by Using the Enterprise Miner Plug-in for SAS Management Console

New with SAS Enterprise Miner 5.2 is the introduction of a plug-in for the SAS Management Console. This plug-in enables you to browse and customize some of the metadata objects used by Enterprise Miner.

Server Extensions

You can use metadata extensions to Logical Workspace Server definitions in order to customize the behavior of Enterprise Miner in the following ways:

- set the default locations for new projects on a particular server, and specify whether these locations can be modified by users when creating new projects.
- set the maximum number of concurrent tasks allowed on a particular server. This setting affects the extent of parallel processing applied to sibling branches running in a data mining process flow, so that you can tune your SMP environment for optimal performance.
- specify SAS initialization code to be run whenever a project is opened, or whenever a process flow is run to generate diagram-level results or exportable model results

packages. This is similar to the Project Start-up Code feature of Enterprise Miner projects, except that it is specified on a server so that you can initialize all SAS sessions created for all projects associated with a particular server.

- provide an alternate command to launch the SAS/CONNECT sessions that are used for each node running in a data mining process flow.

Model Management Configuration

The Enterprise Miner plug-in also lets you configure an optional WebDAV server to act as the model repository storage center, so that mining results packages registered to the model repository from your Enterprise Miner projects are copied to a WebDAV server location. These packages can then be accessed by the model consumers throughout your organization.

You can also use the plug-in to create groups to help you categorize your various classes of models.

Note: The configuration features for model management provided by this plug-in replace the functionality found in the Enterprise Miner configuration wizard that was bundled with earlier releases of this product. △

Other features

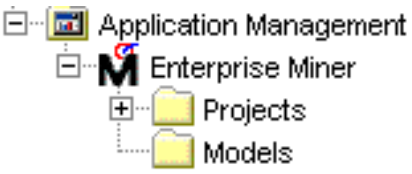
The Enterprise Miner plug-in provides some basic browsing functionality to help you navigate through the list of projects and mining result packages registered to the Foundation repository.

Getting Started

In order to use the Enterprise Miner plug-in, you must have first started an Enterprise Miner client at least once, so that the metadata repository can be initialized with Enterprise Miner metadata. Once you have successfully logged on to Enterprise Miner, you can log off and start SAS Management Console. It is not necessary to have defined or opened any Enterprise Miner projects at this stage.

The plug-in installs into your locally installed SAS Management Console directory when you install Enterprise Miner. Once it is installed, you will see the Enterprise Miner application icon appear under the Application Management section in the left pane of the Management Console window.

Expand **`Enterprise Miner`** to display the **`Projects`** and **`Models`** folders, as shown in the following display:



The Projects Folder

This folder contains a list of all Enterprise Miner projects registered in the Enterprise Miner repository. Click on the folder to see the list of all projects in the right

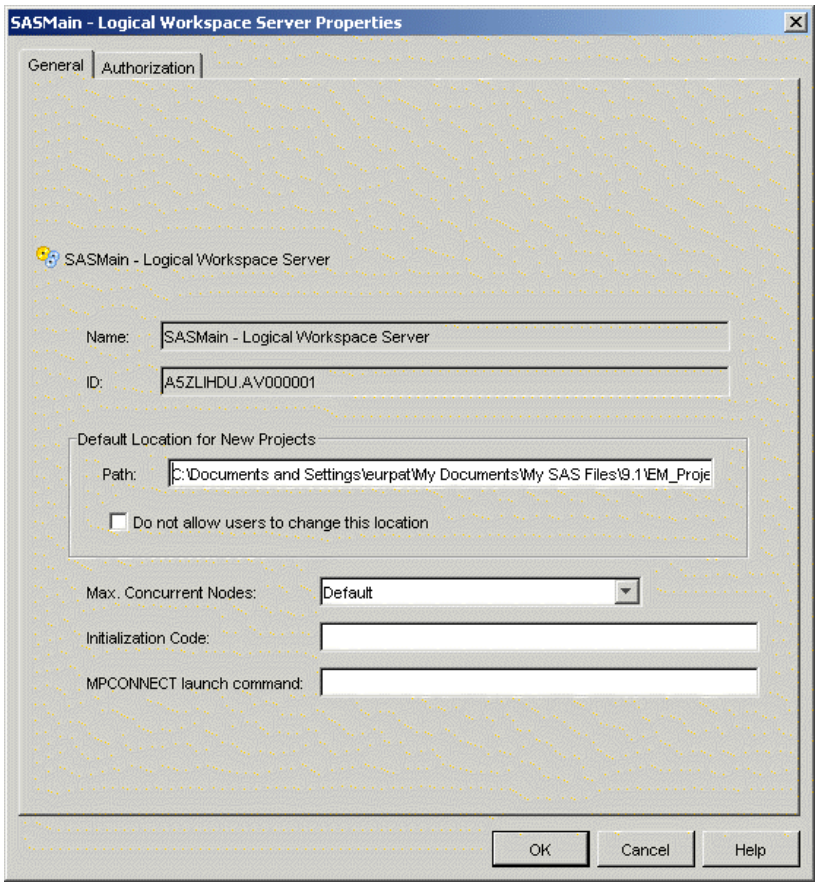
console pane. By expanding the Projects folder, you will find the projects organized by Logical Workspace Server. Clicking on a Logical Workspace Server icon in the left pane will subset the list of projects on the right, so that only those projects belonging to that server are displayed.

You do not need to have any registered projects in order to view the Logical Workspace Servers, or to customize them for Enterprise Miner. Servers for which no projects exist will have their customizations recognized whenever new projects are created on those servers.

Customize Servers for Enterprise Miner

To customize the servers for Enterprise Miner, perform the following steps:

1 Right-click a server icon under the **Projects** folder, and select **Properties** from the pop-up menu.



2 In the **Path** box, enter a default location for new Enterprise Miner projects created on this server.

3 Select **Do not allow users to change this location** if you want to prevent the user who is creating the project from changing the location.

4 From the **Max. Concurrent Nodes** drop-down list, select the maximum number of concurrently running branches in process flows.

5 In the **Initialization Code** box, enter the path (on this server) to a text file containing SAS code that you would like to run any time a project is opened, or a run or report action is performed on a process flow.

6 In the **MPCONNECT launch command** box, enter an alternate command to use for launching MPCONNECT sessions. Normally it is safe to leave this box blank, but there may be cases where you would like to modify some SAS system defaults for sessions that are used when running process flows. The following default command is used when this box is left blank:

```
!sascmdv -noobjectserver -nosyntaxcheck -noasynchio
```

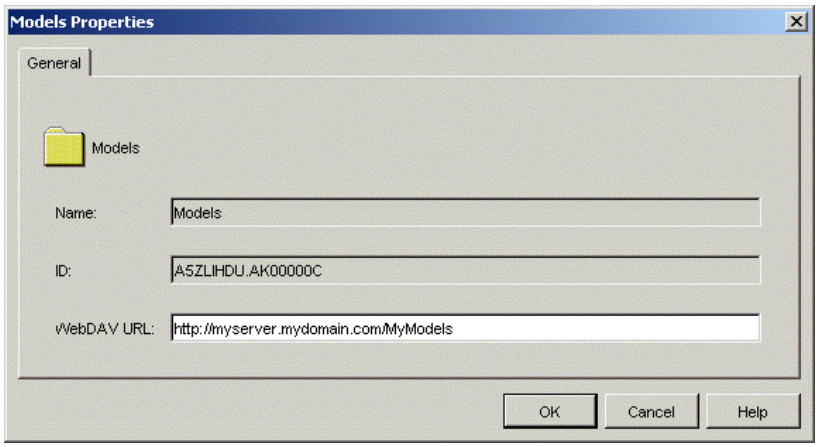
This command has the same effect as using the SAS command that was used to launch the SAS workspace session at the time the project was opened.

The Models Folder

Specify the WebDAV Server

You can use the Models folder to specify a URL to your WebDAV server. If specified, model result packages will be copied to this location at the time they are registered from the Enterprise Miner project environment. To specify the WebDAV server, perform these steps:

1 Right-click the **Models** folder and select **Properties** from the pop-up menu.



2 In the **WebDAV URL** box, enter the URL to your Web server.

Create Model Groups

To create model groups, perform the following steps:

1 Right-click the **Models** folder and select **New Group** from the pop-up menu.

2 Enter a new group name at the prompt. There is currently no support for sub-groups, and duplicate group names are not allowed.

Customizing the Apache Tomcat HTTP Server

The HTTP server recommended for use with SAS Enterprise Miner 5.2 is the Apache Tomcat 4.1.18 server with integrated WebDAV support. To enable the model registration and storage functions of SAS Enterprise Miner, you must enable the Tomcat WebDAV write functionality and define the context path for the Enterprise Miner Model Viewer application for Tomcat.

Note: Servers other than Tomcat 4.1.18 might have different instructions. △

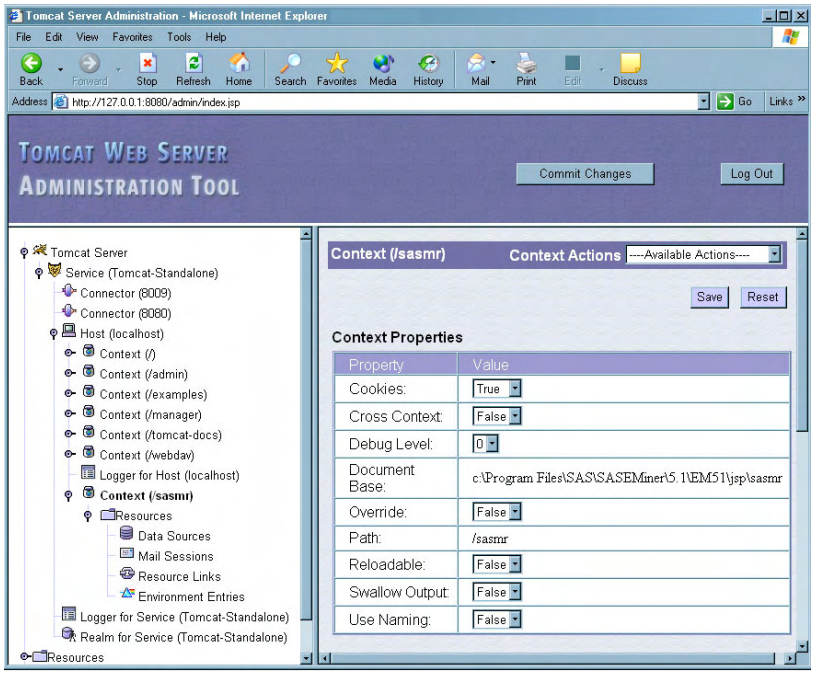
1 To enable the Tomcat write functions, remove the comment delineators from the following block of code in the WebDAV configuration XML file. The file is typically found in

c:\Program Files\Apache Group\Tomcat 4.1\webapps\webdav\WEB-INF\web.xml.

```
<!--
    <init-param>
      <param-name>readonly</param-name>
      <param-value>false</param-value>
    </init-param>
-->
```

2 As a best practice, use the Tomcat Web Server Administration Tool to define the SAS Enterprise Miner Model Viewer to Tomcat. On Windows, you can launch the tool by selecting **Start ▶ Programs ▶ Apache Tomcat 4.1 ▶ Tomcat Administration**.

   a In the navigation tree on the left, select **Tomcat Server ▶ Service ▶ Host**.
   b In the display area to the right of the navigation tree, select **Create New Context** from the **Host Actions** list.
   c In the **Document Base** field, enter the path to the viewer. Typically, the path is
   C:\Program Files\SAS\SASAPCore\apps\EnterpriseMiner\webapps\sasmr
   d In the **Path** field, enter the context path. Typically, the path is **/sasmr**.
   e Click **Save**. The context is added to the selected Host.



If you do not want to use the Tomcat Web Server Administration Tool, you can manually add the following context coding beneath the **`<Host ...>`** tag in the Tomcat server configuration XML file. The file is typically found in

`c:\Program Files\Apache Tomcat 4.1\conf\server.xml`.

```
<!-- EM Model Viewer -->
    <Context path=''/sasmr''
      docBase="C:\Program Files\SAS\SASAPCore\apps\EnterpriseMiner\webapps\sasmr"
      crossContext="false"
      debug="0"
      reloadable="false">
    </Context>
```

Note: If your SAS Enterprise Miner installation path varies from the one shown above, change the location of the Document Base (**`docBase=`** *value*). △

If the SAS Enterprise Miner Model Viewer is installed on a Tomcat server on which Tomcat security is enabled, the installer will need to grant permissions to Catalina for the model viewer. To accomplish this, you need to add the following item to the Catalina permissions file, typically located at

`C:\Program Files\Apache Group\Tomcat 4.1\conf\catalina.policy` or

`C:\Tomcat 4.1\conf\catalina.policy`. Append these lines to the end of the module:

```
grant codeBase
"file:C:/Program Files/SAS/SASAPCore/apps/EnterpriseMiner/webapps/sasmr/-"
{
permission java.security.AllPermission;
      };
grant {
  permission java.security.AllPermission  "com.sas.*";
};
```

where **`C:/Program Files/SAS/SASAPCore/apps/EnterpriseMiner/webapps/sasmr/`** (or its UNIX counterpart) is the docBase path to the **`sasmr`** application on the local disk.

Note: Use forward slashes even for Windows installations. △

3 Stop and restart the Tomcat HTTP server to apply the changes.

Note: To install the link into your Start Menu, you need to access the server administration tool at http://localhost:8080/admin. △

Securing SAS Enterprise Miner Metadata

For thin-client installations, to secure access to the metadata objects that represent SAS Enterprise Miner projects, you grant or deny access to individual users or groups by using the **`Authorization`** tab for the following:

- the **`SAS Enterprise Miner`** folder
- the **`Projects`** folder
- individual projects
- the SAS application server that contains the logical SAS Workspace Server that is associated with the SAS Enterprise Miner projects

Note: For stand-alone installations, there is no need to secure the metadata. △

Note: Because all SAS Intelligence Platform applications use the metadata server when accessing resources, permissions that are enforced by the metadata server provide the strongest protections that are available in the metadata authorization layer. For more information, see the "Using the Metadata Authorization Layer" section in the *SAS Intelligence Platform: Security Administration Guide*.. △

Note: Currently, you cannot secure SAS Enterprise Miner models. △

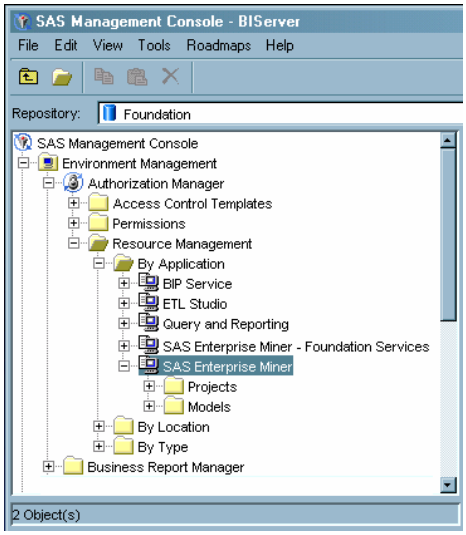
Securing Access at the SAS Enterprise Miner Folder Level

To access the **`Authorization`** tab for the **`SAS Enterprise Miner`** folder in the SAS Management Console navigation tree, complete these steps:

1 Use your metadata profile to log on to the SAS Metadata Server that contains the SAS Metadata Repository connection that you want to use.

2 Select **Environment Management ► Authorization Manager ► Resource Management ► By Application ► SAS Enterprise Miner**.

3 Select the **`SAS Enterprise Miner`** folder.



4 Select **File ▶ Properties**.

5 Click the **`Authorization`** tab.

Note: For help on using the **`Authorization`** tab, click **`Help`**. △

For example, if you deny ReadMetadata access to the **`SAS Enterprise Miner`** folder to User A, then User A will not be able to see any projects in SAS Enterprise Miner, unless you explicitly grant ReadMetadata permission to UserA on the Projects folder.

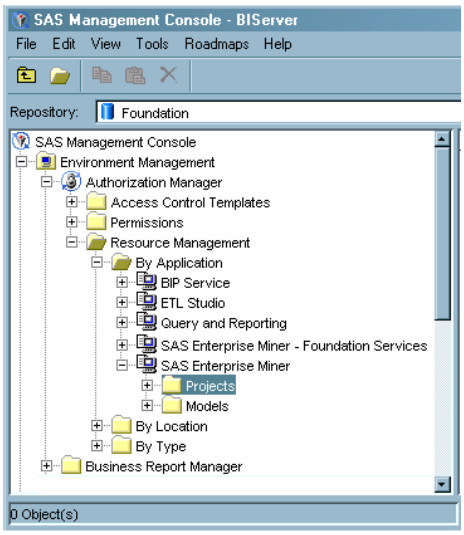
Securing Access at the Projects Folder Level

To access the **`Authorization`** tab for the **`Projects`** folder in the SAS Management Console navigation tree, complete these steps:

1 Use your metadata profile to log on to the SAS Metadata Server that contains the SAS Metadata Repository connection that you want to use.

2 Select **Environment Management ▶ Authorization Manager ▶ Resource Management ▶ By Application ▶ SAS Enterprise Miner**.

3 Select the **`Projects`** folder.



4 Select **File ▶ Properties**.

5 Click the **Authorization** tab.

Note: For help on using the **Authorization** tab, click **Help**. △

For example, if you deny ReadMetadata access to the **Projects** folder to User A, then User A will not be able to see any projects in SAS Enterprise Miner.

Permissions that you explicitly set on the **Projects** folder will override permissions set on the **SAS Enterprise Miner** folder.

Securing Access at the Individual Project Level

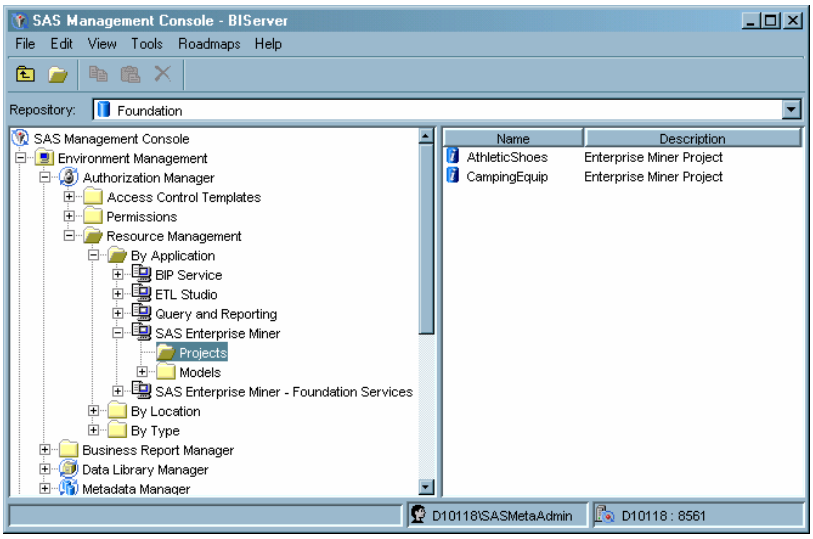
To access the **Authorization** tab for an individual project in the SAS Management Console navigation tree, complete these steps:

1 Use your metadata profile to log on to the SAS Metadata Server that contains the SAS Metadata Repository connection that you want to use.

2 Select **Environment Management ▶ Authorization Manager ▶ Resource Management ▶ By Application ▶ SAS Enterprise Miner**.

3 Select the **Projects** folder.

4 In the display area to the right of the navigation tree, select the project that you want to secure.



5 Select **File ▶ Properties**.

6 Click the **Authorization** tab.

Note: For help on using the **Authorization** tab, click **Help**. △

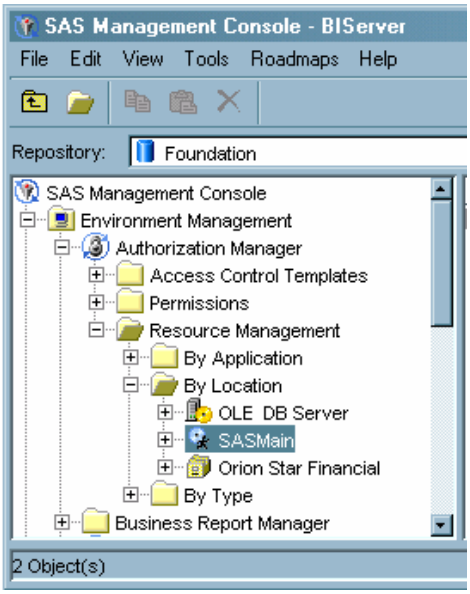
For example, if you deny ReadMetadata access to the **CampingEquip** project to User A, then User A will not be able to see the **CampingEquip** project in SAS Enterprise Miner. If you grant ReadMetadata but deny WriteMetadata, then User A will be able to open the project but will not be able to delete it.

Note: An explicit grant of ReadMetadata permission on the individual project does not override an inherited denial of ReadMetadata from the **Projects** or the **SAS Enterprise Miner** folders. △

Securing Access at the SAS Workspace Server Level

To access the **Authorization** tab for a SAS application server in the SAS Management Console navigation tree, complete these steps:

1. Use your metadata profile to log on to the SAS Metadata Server that contains the SAS Metadata Repository connection that you want to use.
2. Select **Environment Management ▶ Authorization Manager ▶ Resource Management ▶ By Location**.
3. Select the SAS application server that contains the SAS Workspace Server that is associated with your SAS Enterprise Miner projects.



4. Select **File ▶ Properties**.
5. Click the **Authorization** tab.

Note: For help on using the **Authorization** tab, click **Help**. △

For example, if you deny ReadMetadata access to the **SASMain** application server to User A, then User A will not have access to any projects associated with that server.

PART 5

Advanced Topics

CHAPTER

18

Configuring Your SAS Servers

Overview of Configuring Your SAS Servers

This chapter explains how you can configure your workspace and stored process servers to improve the performance of your SAS applications. Here are the specific areas that we will be looking at:

- If you are running SAS Web Report Studio at your site, it is a good idea to change the SAS system options that are used to start the workspace server processes that the application uses. This recommended setup enables the application to serve more clients. For information on how to tune your workspace server for use with SAS Web Report Studio, see "Configuring a Workspace Server for SAS Web Report Studio" on page 381.
- If you have a large number of users who are running SAS Web Report Studio or SAS Information Delivery Portal, you can improve the performance of the application by setting up a pooling workspace server. Without pooling, each user session must establish a connection to a workspace server process, and establishing this connection takes time. If you set up a pooling workspace server, an application can create a set of connections from which users can simply get a reference to a connection. This setup greatly increases performance. For information on how to set up a pooling workspace server, see "Workspace Server Pooling for SAS Web Report Studio and SAS Information Delivery Portal" on page 384
- If your users are using SAS Web Report Studio and you have a requirement to support BI row-level permissions, see "Restricted Workspace Server Pooling for SAS Web Report Studio (Enforcing Row Level Permissions)" on page 393.
- If you need to support a large group of SAS Data Integration Studio users, you might want to set up load balanced workspace servers. Because each workspace server is a single-user server, each client request causes a new server process to be created. If too many clients require the use of workspace servers simultaneously, the performance of programs executing on the workspace server host will degrade. You can address this problem by
  1. setting up a workspace server on a second host
  2. configuring the server to be a part of your existing logical workspace server
  3. converting the logical workspace server to load balancing.

  For step-by step instructions on how to perform these tasks, see "Load Balancing Workspace Servers for Desktop Applications" on page 400.
- Because stored process servers support MultiBridge connections—which means that an object spawner can direct requests to any one of a set of multi-user server processes—it is possible to implement load balancing on a single host. This is, in fact, how the stored process server is set up during a planned installation. Three MultiBridge connections are set up so that the object spawner can start up to three stored process server processes, and the object spawner balances the workload across these processes. For more information about this initial configuration, see "Overview of the Initial Load Balancing Setup for Stored Process Servers" on page 406.

  To some extent, you can scale your system by adding MultiBridge connections to the existing stored process server. However, at some point, you will need to add a

second stored process server running on a second host to improve performance. For information on how to perform this task, see "Load Balancing Stored Process Servers on Multiple Hosts" on page 407.

Note: All of the servers in a load balanced cluster must belong to the same SAS authentication domain. △

Configuring a Workspace Server for SAS Web Report Studio

Tuning a Workspace Server for Best Performance

To obtain the best performance from SAS Web Report Studio, you should tune the workspace server that is being used by that product as described in this section. The changes you should make include specifying the following:

- an appropriate work folder
- a buffer size for writing files to the work area
- a limit on the total amount of memory that SAS uses at any one time

Note: In addition to tuning your workspace server by following the directions in this section, you should convert your workspace server to pooling as discussed in "Workspace Server Pooling for SAS Web Report Studio and SAS Information Delivery Portal" on page 384. △

The following table lists the system options that you should set and recommends values to use with those options. "Add System Options to the Workspace Server Launch Command" on page 382 explains how to add the system options to the command that starts the workspace server.

The arguments to the system options that are shown in the table are values that are useful on a system with the following characteristics:

- four CPUs, 2.0 GHz
- 3.5 GB RAM
- Windows Server 2003

Table 18.1 System Options for the Workspace Server

System Option	Explanation
-RSASUSER	Opens the SASUSER library in read-only mode. Declaring this library read only makes the workspace server much faster for SAS Web Report Studio.
-work *work-folder*	Specifies the pathname for the directory that contains the Work data library. This directory should reside on a disk that emphasizes fast write performance (not, for example, on a RAID-5 device).
-ubufsize 64K	Specifies a buffer size for writing files to the work area.
-memsize 192M	Specifies a limit on the total amount of memory that SAS uses at any one time.
-realmemsize 128M	Indicates the amount of RAM available to a process before it begins to page. Keeping this number low limits the amount of RAM consumed by a SAS server in order to reduce paging activity.

System Option	Explanation
-sortsize 800M	Limits the amount of memory that can be used temporarily for sorting. Larger sort sizes reduce the use of the **work** folder, but increase the possibility of paging.
-cpucount 2	Specifies the number of processors that thread-enabled applications should assume will be available for concurrent processing. This setting maximizes the effectiveness of the SAS Web Report Studio sorting algorithm.

Note that the arguments to these options will be site and job specific. Take care in choosing these values, and consult a SAS representative if necessary.

Configuring the Workspace Server for the Hebrew Locale

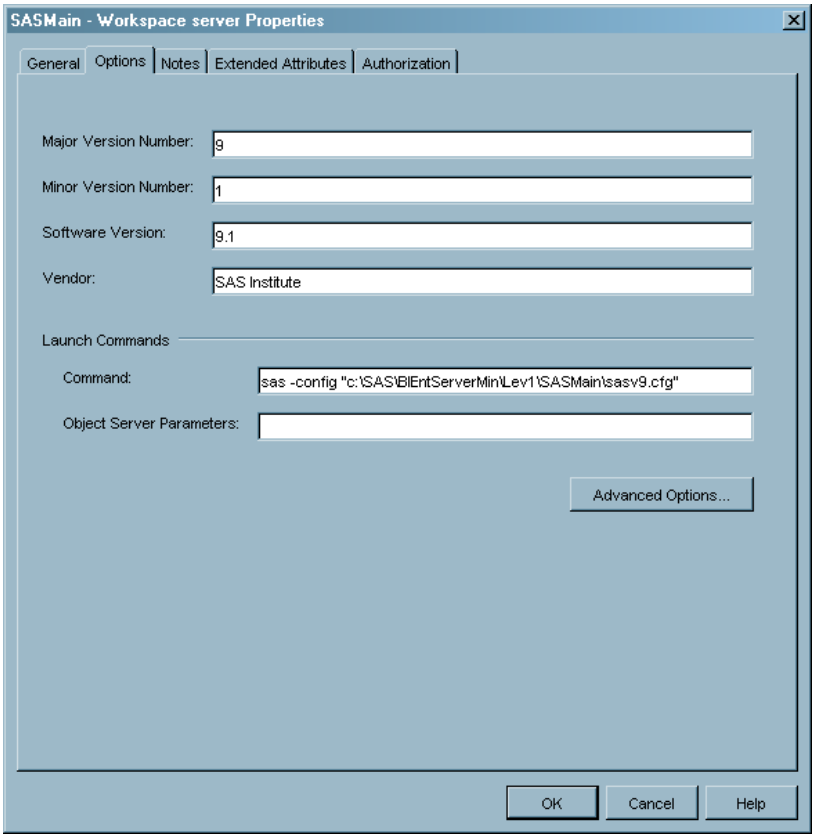
When SAS Web Report Studio is configured to run under the Hebrew locale, the workspace server is started incorrectly. Because of this issue, if you intend to provide SAS Web Report Studio in Hebrew, it is recommended that you add the **-nosyntaxcheck** option to the start-up command for the workspace server. "Add System Options to the Workspace Server Launch Command" on page 382 explains how to add system options to the start-up command.

Note: Alternatively, you can create a new SAS application server (parallel to SASMain) and a new workspace server within the new application server context. You could then change the command (using the **-nosyntaxcheck** option) for the new workspace server, without affecting the workspace server under SASMain. You would then assign selected libraries to the new application server, including any libraries that you intend to query for reports. See "Restricted Workspace Server Pooling for SAS Web Report Studio (Enforcing Row Level Permissions)" on page 393 for information about creating a new SAS application server and workspace server. △

Add System Options to the Workspace Server Launch Command

After you have determined the system options that you want to use to start your workspace server, follow the directions in this section to edit the **sas** command that starts the server.

1. In SAS Management Console, expand the **Server Manager** node, then expand the **SASMain---Logical Workspace Server** node. You will see a tree node that represents the physical workspace server.
2. Right-click the icon for the physical workspace server, and select **Properties** from the pop-up menu. A Workspace Server Properties dialog box appears.
3. Click the **Options** tab. You will see the information that is shown in the following display.



4 Edit the text in the **Command** text box, which by default is set to

```
sas -config "path-to-config-dir\Lev1\SASMain\sasv9.cfg"
```

For example, here is a command with options that improve performance:

```
sas -config "path-to-config-dir\Lev1\SASMain\sasv9.cfg" -rsasuser
  -work work-folder -ubufsize 64K -memsize 192M -realmemsize 128M
  -sortsize 800M -cpucount 2
```

CAUTION:
Do not add the options to the SASMain\sasv9.cfg. Setting values in **Lev1\SASMain\sasv9.cfg** affects every server launched. This includes the metadata server, the OLAP server, and every workspace server and stored process server. △

5 Click **OK** in the Workspace Server Properties dialog box.

Note: At the end of this procedure, you will have optimized your workspace server for use with SAS Web Report Studio. If you are using other applications and these applications would benefit from a differently configured workspace server, you must create a new logical workspace server (under SASMain) and add a workspace server to it. △

Workspace Server Pooling for SAS Web Report Studio and SAS Information Delivery Portal

This section explains how to set up pooling for your workspace server for use with SAS Web Report Studio and SAS Information Delivery Portal. The result of this configuration is that the users of these Web applications will see much better performance than they would if they were connecting to a standard workspace server.

Before we explain how to perform this configuration, it is important that you understand a couple of concepts. For example, what does it mean to set up a pooling workspace server? If a workspace server has not been converted to pooling, each time a SAS Web Report Studio or SAS Information Delivery Portal user starts a session, a workspace server process must be created and the user must establish a connection to this process. This can be a time-consuming sequence of events. When you set up a pooling workspace server, a pool (or set) of connections to workspace servers are opened when SAS Web Report Studio or SAS Information Delivery Portal makes its first request for a workspace server. A user can then obtain a preexisting connection from the pool instead of having to establish the connection.

Another important concept is that of a *puddle*. A puddle is a subset of the connections in a pool. Setting up puddles enables you to associate a different set of users with different puddles. Typically, the reason for setting up different groups of users is to give the different groups different levels of access.

One of the advantages of using a pooling workspace server is to enable SAS Web Report Studio and SAS Information Delivery Portal users to obtain an existing connection to a workspace server. Other advantages include the following:

- You can limit the number of clients that can connect to workspace servers simultaneously. As a result, you can ensure acceptable response times for all connected clients.
- You can add server hosts to the pool to accommodate increased demand.
- You can grant certain groups greater access to the pool than others.
- You can divide the pool into puddles of connections that have unique login credentials and access to server-side resources.

Note: In this section, we assume that you are starting with a standard workspace server. The instructions are applicable both in the case where the users of your Web applications are being authenticated by the metadata server (the default) and in the case where you Java application server is authenticating those users (Web authentication). If you plan to use Web authentication, you should set that up before configuring your workspace server for pooling. For information on how to set up Web authentication for users of SAS Web Report Studio, see "Trusted Web Authentication" on page 310, and for information on setting up Web authentication for the SAS Information Delivery Portal, see "Setting Up Web Authentication" in the *SAS Web Infrastructure Kit: Administrator's Guide* (**`http://support.sas.com/rnd/itech/doc9/portal_admin/installmig/ag_settrust.html`**.) △

To set up your pooling workspace server for use with SAS Web Report Studio, follow the directions in these sections:

- "Configuring the Workspace Server and Pool" on page 385
- "Adding an Authentication Domain to the SAS Trusted User's Login" on page 387
- "Configuring the Foundation Services Manager User Service for Web Report Studio" on page 389

Note: If you have configured SAS Web Report Studio to use Web authentication, you can skip the second of these sections. In this case, the SAS Trusted User is not used as the pool administrator, as this user is when you are using host authentication. △

To set up your pooling workspace server for use with SAS Information Delivery Portal, follow the directions in these sections:

- "Configuring the Workspace Server and Pool" on page 385
- "Adding an Authentication Domain to the SAS Trusted User's Login" on page 387

It is not necessary to perform the steps in "Configuring the Foundation Services Manager User Service for Web Report Studio" on page 389 because the User Service that is employed by the portal application is configured by default in a way that will work with a pool of workspace servers.

Finally, to verify that connection pooling is working, see "Verifying That Connection Pooling Is Working for SAS Web Report Studio" on page 390.

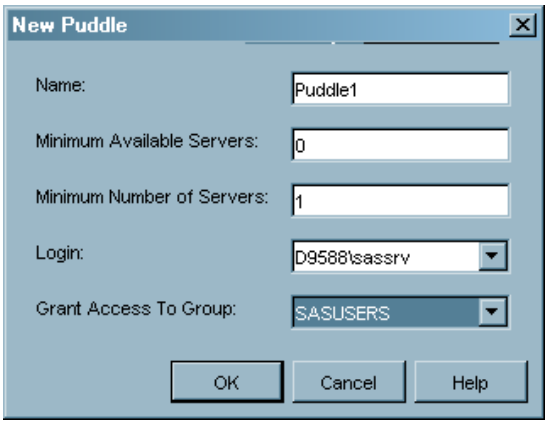
Configuring the Workspace Server and Pool

The first step in configuring connection pooling is to convert your logical workspace server to a pooling configuration and to configure a pool and puddle for this server. Follow the directions in the next two subsections.

Converting the Workspace Server to Pooling

To convert your workspace server to pooling and to define a puddle, perform these steps:

1. Log on to SAS Management Console as the SAS Administrator (**sasadm**).
2. In SAS Management Console, expand the Server Manager tree node and the node for the SASMain application server. One of the tree nodes under SASMain will be SASMain—Logical Workspace Server.
3. Right-click the icon for the logical workspace server, and select **Convert To ▸ Pooling**. You will see an Information dialog box that asks if you want to continue with the conversion. Click **Yes**. The Pooling Options dialog box appears.
4. In the Pooling Options dialog box, click **New** to indicate that you want to define a puddle. The New Puddle dialog box appears.



5. Fill out the fields in the dialog box as shown in the previous display. The following table explains what the values in the dialog box mean.

Table 18.2 New Puddle Information

Field	Explanation
Name	The name of the puddle, for example, Puddle1.
Minimum Available Servers	Specifies the number of workspace servers to start if the minimum number of servers is already in use. The goal for this value is to provide enough workspace servers to accommodate the average number of simultaneious client connections. If there are too few servers, some clients must wait while additional servers are started. If there are too many servers, all client requests can be accommodated from the puddle; however, the server host will be performing unnecessary work.
Minimum Number of Servers	Specifies the initial number of started workspace servers that are available to satisfy client connection requests from SAS Web Report Studio and SAS Information Delivery Portal users.
Login	Contains the user ID of the user who will start the connections in the pool. Enter the ID of the SAS General Server User (**sassrv**) here. Note: A login for this user was added to the group SAS General Servers when your system was first configured.
Grant Access To Group	This field specifies which group of users can use connections from the pool. In the current example, this is the SASUSERS group. If you want to allow all users to access the system, you should change this value to PUBLIC. Note: If you do not want members of PUBLIC to be able to use SAS Web Report Studio *and* you want to present such users with a clear and user-friendly error message, follow these directions. First, set the value of the **Grant Access To Group** field to SASUSERS (to control access to the pool). Then, edit the properties file **WEB-INF\WebReportStudioProperties.xml**, and change the value of the wrs.pfs.allowPublicUsers element to **false** (to provide useful application-level feedback to members of PUBLIC).

Note: If you are not logged on to SAS Management Console as the SAS Administrator (**sasadm**), you might not see **sassrv** in the **Login** drop-down list box. In this case, click **Cancel** in the New Puddle dialog box. Then, reconnect to the metadata server using the metadata profile for the SAS Administrator. △

6 Click **OK** in the New Puddle dialog box.

7 Click **OK** in the Pooling Options dialog box.

Setting Configuration Options for the Pool

Next, set the configuration options for the pool itself by following these directions:

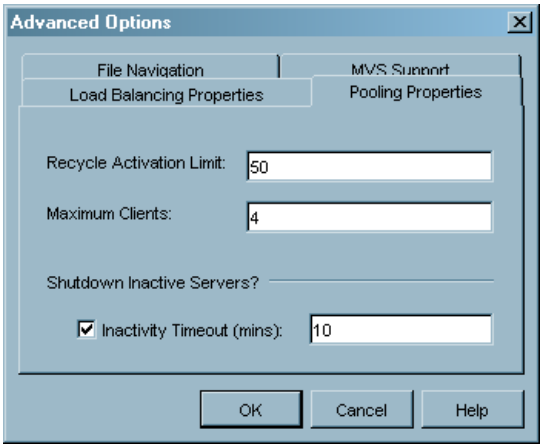
1 In SAS Management Console, expand the SASMain—Logical Workspace Server to reveal the icon for the physical workspace server.

2 Right-click the workspace-server icon, and select **Properties** from the pop-up menu. A Workspace Server Properties dialog box appears.

3 In the Workspace Server Properties dialog box, select the **Options** tab.

4 On the **Options** tab, click **Advanced Options**. The Advanced Options dialog box appears.

5 In the Advanced Options dialog box, select the **`Pooling Properties`** tab.



6 Fill the fields in the dialog box as shown in the previous display. The following table explains the meaning of each value.

Table 18.3 Pooling Properties

Field	Explanation
Recycle Activation Limit	Places a limit on how often workspace server processes are reused to satisfy puddle connections.
Maximum Clients	Each client requires a workspace server process on the workspace server host. These processes constitute the pool that will be available to SAS Web Report Studio and SAS Information Delivery Portal. Each process requires approximately 150 MB for efficient operation. As a guide, we recommend two processes per CPU. If the server is *not* also acting as the metadata server, you can add one or two to this maximum. If you have long-running queries, you can add one or two servers. This will make the system seem more responsive to short-running queries, at the expense of total throughput. A typical setting is between 4 and 6. Applications should not share the same pooled workspace configuration as this creates duplicate pools, one for each application, and an unexpectedly high workload for the server host.
Inactivity Timeout	A workspace server process can have an inactivity timeout. Having a short timeout reduces the workload on the server host, but might reduce response time for client connection requests.

7 Click **`OK`** in the Advanced Options dialog box.

8 Click **`OK`** in the Workspace Server Properties dialog box.

Adding an Authentication Domain to the SAS Trusted User's Login

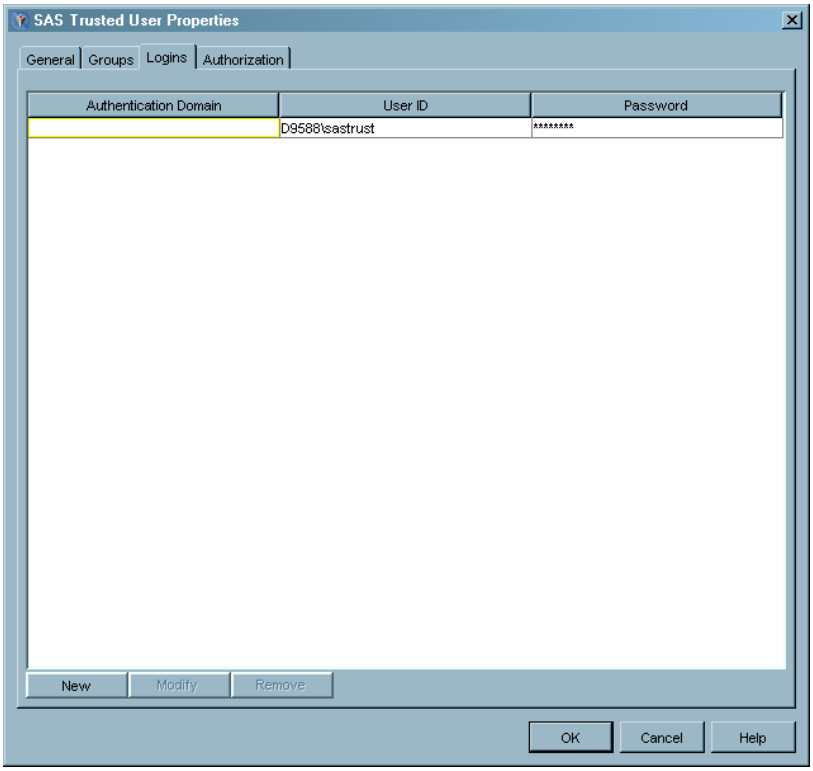
When you configure the Foundation Services Manager User Service—as described in the next section—you are asked to specify a *pool administrator*, and the configuration instructions tell you to specify the SAS Trusted User (**`sastrust`**) as this user. In order for the SAS Trusted User to be used as the pool administrator, you must add the proper authentication domain to that user's login.

To define an authentication domain for the SAS Trusted User, perform the following steps in SAS Management Console:

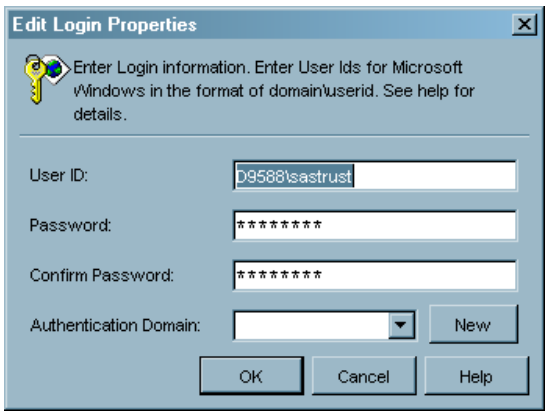
1 Select the icon for the User Manager plug-in so that a list of users and groups appears in the console.

2 Double-click the entry for the SAS Trusted User. A SAS Trusted User Properties dialog box appears.

3 In this dialog box, select the **Logins** tab.



4 Select the existing login, and click **Modify**. An Edit Login Properties dialog appears.



Notice that no authentication domain has been specified.

5 Select DefaultAuth from the **Authentication Domain** drop-down list, and click **OK**. The login for the SAS Trusted User is updated in the SAS Trusted User Properties dialog box.

6 Click **OK** in the SAS Trusted User Properties dialog box.

Configuring the Foundation Services Manager User Service for Web Report Studio

SAS Web Report Studio uses the Platform User Service, which is one of the SAS Query and Reporting Services. Configure this service so that it knows about the pool administrator by following these directions.

Note: The analogous step was performed for SAS Information Delivery Portal during the installation of that product. △

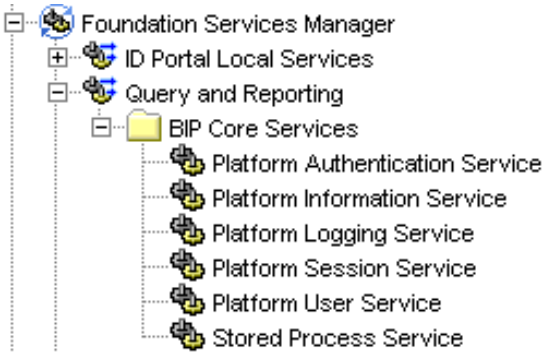
1 In SAS Management Console, expand the Foundation Services Manager. (SAS Foundation Services should have been installed when your system was first set up.) Most likely, you will see nodes for ID Portal Local Services and Remote Services. If this is the case, continue with step 2 to create the necessary query-and-reporting service deployment. (If you already have a Query and Reporting node, you can skip to step 6.)

2 Right-click the Foundation Services Manager icon, and select **Import Service Deployment** from the pop-up menu. An Import Service Deployment dialog box appears.

3 In the Import Service Deployment dialog box, click **Add**. An Open dialog box (a file-system browser) appears.

4 The service-deployment file that you need to select resides on the host where SAS Web Report Studio or SAS Web Report Viewer is deployed. Look in your J2EE application server's deployment area for the appropriate application. From there, the path to the file is **WEB-INF\pfsconfig\config***host-name***_sas_pfs_queryandreporting.xml**. Select this file and click **Open**. The name of the file will display in the Import Service Deployment dialog box.

5 Click **OK** in the Import Service Deployment dialog box. The service deployment will be imported, and if you fully expand the Query and Reporting node, you should see the following tree nodes.



6 Right-click the Platform User Service icon, and select **Properties** from the pop-up menu. A Platform User Service Properties dialog box appears.

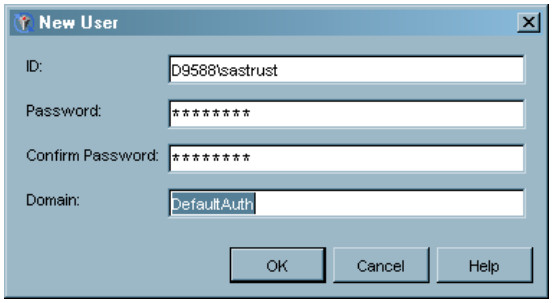
7 In the Platform User Service Properties dialog box, select the **Service Configuration** tab.

8 On the **Service Configuration** tab, click **Edit Configuration**. A User Service Configuration dialog box appears.

9 In the User Service Configuration dialog box, select the **Users** tab.

10 On the **Users** tab, click **Add**. A New User dialog box appears. This is where you specify the user ID of the pool administrator..

Note: For information about the pool administrator and how applications employ this user's ID, see "Planning the Pooling Security (IOM Bridge only)" in the *SAS Integration Technologies: Server Administrator's Guide*, which is available at **http://support.sas.com/rnd/itech/doc9/admin_oma**. △



11 If SAS Web Report Studio is configured to use host authentication, specify information about the SAS Trusted User (sastrust) in the New User dialog, as shown in the display above. Change the domain qualifier for the user ID as necessary, and note that the user ID is case sensitive.

If the application is configured to use Web authentication, perform the following steps:

a Use SAS Management Console's User Manager to create a new user called Pool Administrator. Create a login for this user with a user ID of **pooladm** and an authentication domain of **web**. No password is necessary.

b Enter information about the Pool Administrator in the New User dialog. Leave the password fields blank.

12 Click **OK** in the New User dialog.

13 Click **OK** in the User Service Configuration dialog box.

14 Click **OK** in the Platform User Service Properties dialog box.

Note: After making these changes, you should make sure that only the SAS Web Administrator (saswbadm) and the IT staff have ReadMetadata access to the Platform User Service. For details on this subject, see "Minimizing the Availability of Accounts" in the chapter "Securing a Deployment" in the *SAS Intelligence Platform: Security Administration Guide*. △

Verifying That Connection Pooling Is Working for SAS Web Report Studio

If you have not configured connection pooling correctly, SAS Web Report Studio will continue to work; however, it will not be able to take advantage of a connection pool. Therefore, it is important that you verify that your system is configured correctly. You can verify this by temporarily changing the logging level for connections, viewing a SAS Web Report Studio report, and then checking the contents of the SAS Web Report Studio log file. Detailed instructions follow:

1 Change the logging level for connections by adding the following element to the file *J2EE-server-deployment-area***\WEB-INF\DefaultLoggerProperties.xml**.

```
<LoggingContext name     = "com.sas.services.connection"
                priority = "INFO"
```

```
                  chained  = "false">
    <OutputRef outputID = "WRS" />
  </LoggingContext>
```

You can add this element directly below the existing LoggingContext element in the file.

CAUTION:
We recommend that you create a back-up copy of this XML file before editing it. An error in the XML syntax in this file can prevent SAS Web Report Studio from starting up properly. △

2 Start SAS Web Report Studio and log on.
3 View a report that causes a relational database query.
4 View the SAS Web Report Studio log file (*path-to-config-dir***\Lev1\web\Deployments\WebReportStudio\logs\WebReportStudio.log**). If pooling is working, you will see information that is similar to this information about the connection to the workspace server:

```
privileged user name: D9588\sastrust
pd#0: putting cx#8 on the available queue
rq#0 routed to pd#0
```

If pooling is not working correctly, you will see a message similar to this message:

```
request served by unshared connection #9
```

After you have confirmed that connection pooling is working, you can undo the changes that you made to **DefaultLoggerProperties.xml**.

Configuring Pooling Across Multiple Machines

If your deployment requires more than one host for pooled workspace servers, then you can configure pooling across multiple machines.

An important aspect of multi-machine workspace pooling involves the puddle login ID that you supplied in the section "Converting the Workspace Server to Pooling" on page 385. In order to implement multi-machine pooling, this login must be valid for all machines in the pool.

This topic describes a basic configuration where the pool has only one puddle. For this basic implementation, the account that you use for the login ID must be able to authenticate on every machine in the puddle. This means that the login ID must be a network account, and it must be associated with a single authentication domain in metadata (for example, NTDomain). In the section "Converting the Workspace Server to Pooling," we recommended that you use the SAS General Server User (**sassrv**) for this account. For most deployments, **sassrv** is configured as a network account.

If you require a workspace pool that consists of host servers in different authentication domains (for example, Windows and UNIX), then you would configure a pool with a separate puddle for each host. Each puddle would have its own login ID as appropriate for the host's authentication domain (for example, NTDomain or UnixAuth). Each login ID must be able to authenticate on every machine in its respective puddle. With this configuration, the pool manager software can identify which machine is used for a given process based on the puddles that are defined for the pool, and it uses the appropriate login to make a connection.

The following instructions explain how to configure workspace pooling across two machines in the same authenticaiton domain, both supporting the same puddle. Repeat these instructions as appropriate for each additional machine that you want to configure for that same puddle.

Note: These instructions assume that you have already set up and verified pooling on a single machine, as explained in the previous sections. △

To configure pooling across two machines:

1 Log on to SAS Management Console as the SAS Administrator (**sasadm**).

2 If the puddle login ID is *not* accessible to all machines in the pooling cluster, then you must create a new user. Follow these steps:

   a On the host, identify or create a network account (preferably **sassrv**). For example, on Windows the user name might be *NTDomain\sassrv*. The user must authenticate against each machine in this puddle.
   If creating the account on Windows, grant the "*Log on as a batch job*" permission/policy for the account on each Windows machine in the cluster.

   b In the SAS Metadata Console User Manager plug-in, add a login to the SAS General Servers group with the user ID and password of the user that you just created. (You might now have two login IDs defined for this group— one that is a network account, and one that is a local account.)
   This login should be associated with the server's authentication domain (for example, DefaultAuth or ServerAuth), even if you are using middle-tier trusted authentication. The login is used to start processes on the server machine rather than to authenticate against the metadata server.

   c Change the login ID that is used for the puddle to this newly established login ID. Make this change in the Edit puddle dialog box, which is accessed from the Pooled Workspace Server–Logical Workspace Server properties window under the Server Manager plug-in. For details, see "Configuring the Workspace Server and Pool" on page 385

3 In the SAS Metadata Console Server Manager plug-in, add a pooled workspace server definition for the machine that you are adding to the pool. Follow these steps:

   a Expand the SASMain application server, then right-click Pooled Workspace Server—Logical Workspace Server and select **Add Server** from the pop-up menu. A New Server wizard appears.

   b In the wizard, specify the host name for the machine being added to the pool.

   c In the wizard, click **Advanced Options**, and then select the **Pooling** tab.

   d In the **Pooling** tab, set the pooling options for this machine. The maximum number of clients can vary for each server included in the pool, and should be configured based upon the capacity of each physical server being used. For details, see "Setting Configuration Options for the Pool" on page 386.

4 In the Server Manager plug-in, define a new object spawner. Following are the steps:

   a Right-click the Server Manager and select **Add Server** from the pop-up menu. The New Server Wizard appears.

   b In the wizard, choose **Object Spawner** as the type of server, then click **Next**.

   c Provide information in the wizard as appropriate, using these guidelines: The **Associated Machine** should be the host name of the server being added to the pool. The **Operator Login** can be left as **(None)**. Select the newly created pooled workspace server as the server to be spawned by this object spawner.

5 Under SASMain, expand the logical pooled workspace server, and verify that both physical workspace servers appear under the logical pooled workspace server.

6 Install the necessary SAS software on the machine that is being added to the pool. At a minimum, you should install Base SAS, the SAS Workspace Server, and the SAS Object Spawner. The object spawner should be configured to run as a service, and should be started automatically. When configuring the object spawner, be sure

that the object spawner can connect to the metadata server (running on another machine). If the spawner cannot access the metadata server, then the object spawner can not be started.

7 Verify that pooling works for SAS Web Report Studio. To do this, first start the metadata server. Then start the object spawner on the machine being added to the pool. Finally, start or restart the middle-tier components. See "Verifying That Connection Pooling Is Working for SAS Web Report Studio" on page 390.

Note: Because the pool administrator did not change, there is no need to change the middle-tier configuration. Recall that the pool administrator (typically **sastrust**) is used by the middle-tier application to process pooling requests. You configured the pool administrator previously in the section "Configuring the Foundation Services Manager User Service for Web Report Studio" on page 389. △

Depending on the puddle configuration and the maximum number of clients allowed for each server under the logical pooled workspace server, the number of pooled workspace server processes that show up on each machine will vary.

As mentioned earlier in this section, if you require a workspace pool that consists of host servers in different authentication domains, then you would add more puddles to the pool. For each puddle, you would provide a login ID that is able to authenticate against all machines in the puddle.

The data sets to be accessed by the pooled workspace server processes must either be replicated on each machine that will execute the pooled workspace server processes or accessed via a shared location using a common shared path—for example, a UNC or NFS path. For more information on putting the data in a shared location, see "Data and Catalogs for Servers on Multiple Machines" on page 414. Alternatively, the workspace server processes can access data using some other sharing mechanism, such as a SAS/CONNECT server. All **LIBNAME** assignments will use the same path information, regardless of which physical machine the process runs on.

The object spawner process on the machine that has been added to the pool cannot be started until the metadata server is up and running. If the object spawner is configured to run as a service that starts automatically, this dependency can not be enforced via the Windows services configuration.

Restricted Workspace Server Pooling for SAS Web Report Studio (Enforcing Row Level Permissions)

After the initial installation and configuration of the SAS Intelligence Platform, most sites have a single workspace server, which is part of the default SAS application server, SASMain (or SASApp). By default, this workspace server is a standard workspace server, which means that workspace server processes are spawned on an as-needed basis. If you are running SAS Web Report Studio at your site, most likely this workspace server has been converted to pooling. After this conversion, a pool of workspace server processes are available for use by SAS Web Report Studio sessions. This setup provides a big boost in performance for the users of this application.

This section explains how to create a second pooling workspace server that you can use as part of an environment in which row-level security is enforced. For more information about this environment, see "How to Create a Secure Environment for BI Row-Level Permissions" in the chapter "BI Row-Level Permissions" in the *SAS Intelligence Platform: Security Administration Guide*.

Define the Necessary Users and Groups

The first step in setting up the new pooling workspace server is to define two accounts for users who must be authenticated by the operating system on the workspace server host, and several user and group metadata objects.

User Accounts

Create user accounts that will enable the operating system on the workspace server host to authenticate the following users. On Windows systems, these accounts can be domain accounts or local accounts:

- rpooladm - This account is for the pool administrator, the user who handles requests for processes in the workspace server pool. The password for this account should be unique.
- rpoolsrv - This is the puddle login account. Each SAS Web Report Studio user who will have access to the pool must belong to a metadata group one of whose logins contains this user ID and the associated password.

On Windows systems, grant both of these users the **Log on as a batch job** right. If you created a SAS Server Users group when you first installed the SAS Intelligence Platform, you can give these users this right by adding them to that group.

Once you have created these user accounts, you should create the metadata objects described in the next section.

User and Group Objects

In SAS Management Console, use the User Manager to create one user and two groups:

- User: Restricted Pool Administrator
- Group: Restricted Pool Puddle Login
- Group: Restricted Pool Puddle Access

To define the Restricted Pool Administrator:

1. Right-click **User Manager**, and select **New ▶ User** from the pop-up menu.
2. In the New User Properties dialog box, on the **General** tab, enter the name **Restricted Pool Administrator** in the **Name** box.
3. In the same dialog box, select the **Logins** tab.
4. Add a login to the user object by selecting **New** and entering the appropriate information in the New Login Properties dialog box. In the **Name** box, enter the name **rpooladm**. If the operating system account for this user is a Windows account, qualify the name with a domain or machine name and a backslash. Then select the authentication domain for your workspace server from the **Authentication Domain** drop-down list. It is preferable, for security reasons, not to put the password in the metadata. Click **OK** at the bottom of the dialog box.
5. Click **OK** in the New User Properties dialog box.

To define the Restricted Pool Puddle Login group:

1. Right-click **User Manager**, and select **New ▶ Group** from the pop-up menu. A New Group Properties dialog box appears.
2. On the **General** tab, enter the name **Restricted Pool Puddle Login** in the **Name** box.
3. On the **Members** tab, select the **Restricted Pool Administrator**, and click the right-arrow button to move the user to the **Current Members** list.

4 On the **Logins** tab, create a new login that contains the credentials for rpoolsrv and the authentication domain of the workspace server. (See step 4 in the preceding paragraph for details about how to create this login.)

5 Click **OK** in the New Group Properties dialog box.

Create a second group named Restricted Pool Puddle Access. Add any users or groups that you want to be able to use the restricted pool as members of this group. No logins are necessary.

Create a Restricted Workspace Server Pool

To create the restricted pooling workspace server, follow the directions below.

1 In the directory *SAS-config-dir***\Lev1\SASMain**—or **SASApp**—create a directory called **RestrictedPool**. Then, in the **RestrictedPool** directory, create a **logs** directory.

2 In the directory *SAS-config-dir***\Lev1\SASMain\RestrictedPool**, create a configuration file that will be used when the restricted workspace server is started. The way in which you perform this step depends on whether the workspace server will run on a Windows or a UNIX host. See the appropriate instructions below:

Windows

In the directory *SAS-config-dir***\Lev1\SASMain\RestrictedPool**, create a file named **sasv9.cfg**, and enter the following lines in the file:

```
-config "SAS-config-dir\Lev1\SASMain\sasv9.cfg"
/* -log "SAS-config-dir\Lev1\SASMain\RestrictedPool\logs\
  WorkspaceServer_%v.log" */
/* -logparm "rollover=session open=replaceold write=immediate" */
```

UNIX

In the directory, *SAS-config-dir***/Lev1/SASMain/RestrictedPool**, create a file named **workspaceServer.cfg**, and enter the following lines in the file:

```
-config !SASROOT/sasv9.cfg
-config sasv9.cfg
/* -log RestrictedPool/logs/WorkspaceServer_%v.log */
/* -logparm "rollover=session open=replaceold write=immediate" */
```

Later in the procedure, you will test the connection to the workspace server. If the test fails, you can uncomment the lines related to logging in order to enable logging. You can then repeat the test and check the workspace server log file for error messages.

3 Choose one of the following authentication methods for the workspace servers to use in connecting to the metadata server. Also, perform any tasks associated with the method that you choose.

- If you use the TRUSTSASPEER object server parameter in your metadata server's configuration file (the default), you can rely on that mechanism for workspace server authentication. The restricted pool workspace servers will connect to the metadata server under the rpoolsrv identity when launched by SAS Web Report Studio and will connect under the end user's identitiy when launched by a desktop application. In this mode, if you are working with an external DBMS, you must ensure that both the Restricted Pool Puddle Login group and any allowed individuals have database credentials.
- You can also use the METAUSER and METAPASS options in the restricted pool workspace server's configuration file. With this approach, the TRUSTSASPEER option is not required, and only the Restricted Pool Puddle Login group needs

database credentials. For this approach, edit the configuration file that you created in step 2 to add these lines:

```
-metauser "rpoolsrv"
-metapass "encrypted-rpoolsrv-password"
```

On Windows systems, be sure to prepend a domain or host-name qualifier to the user ID. You can encrypt the password using PROC PWENCODE.

Note: The configuration file for the restricted workspace server must be locked down at the operating system level. The pool administrator launches workspace servers under an operating system user ID of rpoolsrv. Non-pooled workspace servers executed against this pool (such as those run by the SAS Information Map Studio test dialog) run under the end user ID. You typically want to let IT staff do this. So the general guideline that the install areas be readable only by IT staff will ensure that only allowed staff and SAS Web Report Studio can launch workspace servers in this pool.

You will have to remember to change this configuration file if site policy requires periodic changes on service accounts. △

4 In SAS Management Console, define a new SAS application server, named RestrictedPool, that contains a workspace server.

a Right-click **Server Manager**, and select **New Server** from the pop-up menu. The New Server Wizard starts.

b On the wizard's first page, select **SAS Application Server** and click **Next**.

c On the wizard's second page, enter the name **RestrictedPool** in the **Name** box and click **Next**.

d On the wizard's third page, accept the default values and click **Next**.

e On the wizard's fourth page, select **Workspace Server** and click **Next**.

f On the wizard's fifth page, select the **Custom** radio button and click **Next**.

g On the wizard's sixth page, enter the values shown below in the **Command** and **Object Server Parameters** boxes. Note that the first command is appropriate for a workspace server running on a Windows host, and the second is appropriate for a UNIX host. The value that you enter in the **Object Server Parameters** field is not operating system dependent.

Command (Windows)	**sas -config "*SAS-config-dir*\Lev1\SASMain\RestrictedPool\sasv9.cfg"**
Command (UNIX)	***SAS-config-dir*/SASMain/sas.sh -config RestrictedPool/workspaceServer.cfg**
Object Server Parameters	**metaautoinit**

Then click **Next**.

h On the wizard's seventh page, select the **Bridge Connection** radio button and click **Next**.

i On the wizard's eighth page, specify the following values.

Authentication Domain	Specify the same authentication domain that you used when you defined your first workspace server. By default, this will be **DefaultAuth**.
Host Name	Specify the fully qualified name of the host on which the new workspace server will run. (This should be the same name that you used in the definition of your original workspace server.)
Port Number	Change the default value, 8591, to the number of an unassigned port, such as 9591.

j On the wizard's ninth page, click **Finish**.

Note: Do *not* select the **Test Connection** button in order to test your ability to connect to the new workspace server. You can perform this test after performing the following step. △

5 Update the metadata definition of your object spawner (SASMain - Spawner) to indicate that the spawner should start processes for the new workspace server.

a Right-click the icon representing the spawner, and select **Properties** from the pop-up menu. A SASMain - Spawner Properties dialog box appears.
b Select the **Servers** tab.
c Move **RestrictedPool - Workspace Server** from the list of **Available servers** to the list of **Selected servers**.
d Click **OK**.
e Restart your object spawner.

6 Test the connection to your new workspace server.

a In the left pane of SAS Management Console, select **RestrictedPool - Workspace Server**. Information about a connection appears in the right pane.
b Right-click the icon representing the connection, and select **Test Connection** from the pop-up menu.
c If you are logged in to SAS Management Console as an *unrestricted user*—such as sasadm—you will be prompted for the credentials of a user who can start a workspace server. Enter the credentials for a user such as sasdemo. You should see a message indicating that the test was successful.

If the connection test fails, look at the log files for the object spawner and the new workspace server. The most likely cause of the problem is that you made a mistake in editing the configuration file for the new workspace server—the configuration file in the **RestrictedPool** directory.

7 Convert the new workspace server to pooling.

a Right-click **RestrictedPool - Logical Workspace Server**, and select **Convert To ▶ Pooling** from the pop-up menu.
b You are asked whether you want to continue. Click **Yes**. The Pooling Options dialog box appears.
c In this dialog box, click **New** to bring up the New Puddle dialog box.
d In the New Puddle dialog box, supply the following values:

Table 18.4 Defining a New Puddle

Field	Value
Name	**restrictedPoolPuddle**
Minimum Available Server	**0**
Minimum Number of Servers	**0**
Login	**rpoolsrv**
Grant Access to Group	**Restricted Pool Puddle Access**

Then click **OK**.
e Click **OK** in the Pooling Options dialog box.

Assign Libraries to the New Server

You must assign each library that you plan to access from the locked-down instance of SAS Web Report Studio to the server RestrictedPool. For each such library, perform the following steps.

1. Right-click the icon for the library, and select **Edit Assignments** from the pop-up menu. The Edit Assignments dialog box appears.
2. Hold down the CTRL key and click the list entry for **RestrictedPool**. (This action should select **RestrictedPool** and leave any already selected items in a selected state.) Then click **OK**.
3. Right-click the library again, and select **Properties** from the pop-up menu. The *Library-name* Properties dialog box appears.
4. Select the **Options** tab.
5. Click the **Advanced Options** button. The Advanced Options dialog box appears.
6. Select the **Library is pre-assigned** option, and click **OK**.
7. Click **OK** in the properties dialog box.

In the future, when you create information maps that you want to access from the locked-down instance of SAS Web Report Studio, make sure that you locate relational data sources using the RestrictedPool server. Also, save these maps in a separate folder: **/BIP Tree/ReportStudio/RestrictedData/Maps**.

Create a Second SAS Web Report Studio Deployment

Follow the directions below to deploy a second instance of SAS Web Report Studio, named RestrictedDataReporting.

1. Configure the Platform User Service in the Query and Reporting service deployment in order to add a user, the pool administrator, for the restricted pool.
   a. In SAS Management Console, expand **Foundation Services Manager**, the **Query and Reporting** node, and **BIP Core Services**. You see a list of services that includes the Platform User Service.
   b. Right-click **Platform User Service**, and select **Properties** from the pop-up menu. The Platform User Service Properties dialog box appears.
   c. Select the **Service Configuration** tab.
   d. Click **Edit Configuration**. The User Service Configuration dialog box opens.
   e. Select the **Users** tab.
   f. Click **Add**. The New User dialog box appears.
   g. In the New User dialog box, enter the user ID and password for the pool administrator (rpooladm). If this is a Windows account, be sure to qualify the ID with a domain or host name. Also, enter the authentication domain associated with these credentials (DefaultAuth by default). Then click **OK**.
   h. Click **OK** in the User Service Configuration dialog box.
   i. Click **OK** in the Platform User Service Properties dialog box.

   Note: After making these changes, you should make sure that only the SAS Web Administrator (saswbadm) and the IT staff have ReadMetadata access to the Platform User Service. For details on this subject, see "Minimizing the Availability of Accounts" in the chapter "Securing a Deployment" in the *SAS Intelligence Platform: Security Administration Guide*. △

2. If you will run SAS Web Report Studio on Tomcat, you must edit your **catalina.policy** file and restart Tomcat. Add a section to the file that grants

RestrictedDataReporting (the new Web application) the same permissions as SASWebReportStudio.

3 In your SAS Web Report Studio installation directory, create two subdirectories: one to hold the configuration files for your original SAS Web Report Studio deployment and one to hold the configuration files for the restricted deployment.

   a In the directory *SAS-install-dir*`\SASWebReportStudio\3.1`, create two subdirectories: `siteStdConfig` and `siteRestrictedConfig`.
   b Copy the files `wrs.config` and `sas.wrs.config.xml` from the `3.1` directory to the subdirectories that you just created.

   Note: You will use the configuration files stored in `siteStdConfig` if you recreate and redeploy the original SAS Web Report Studio WAR file, and you will use the configuration files stored in `siteRestrictedConfig` when you create and deploy the instance of SAS Web Report Studio that will run against the restricted pool. You must copy the most recently edited versions of the configuration files to the `3.1` directory before running the configuration and deployment scripts. △

4 Edit the file `wrs.config` in the directory `siteRestrictedConfig`. The properties that you should edit and the values that you should assign to those properties are shown in the table below.

Table 18.5

Property	Value
MAP_FOLDER	`ReportStudio/RestrictedData/Maps`
$PDF_LOGICAL_WORKSPACE_SERVER$	RestrictedPool - Logical Workspace Server
LOG_FILE_PATH	*SAS-config-dir*`\Lev1\web\Deployments\WebReportStudio\logs\restrictedPool`
$POOLING_USERID$	*domain-or-host-name*\rpooladm (Windows) rpooladm (UNIX)
$RENDERER_OPTIMIZER_SERVER$	RestrictedPool - Logical Workspace Server

5 Create the directory shown as the value of LOG_FILE_PATH in the table above: *SAS-config-dir*`\Lev1\web\Deployments\WebReportStudio\logs\restrictedPool`.

6 Edit the file `sas.wrs.config.xml` in the directory `siteRestrictedConfig`. Change the `<Define>` element near the top of the file so that the warfile attribute is set to `RestrictedDataReporting.war`.

7 Copy the two configuration files that you have edited to the parent directory: *SAS-install-dir*`\SASWebReportStudio\3.1`.

8 From the `3.1` directory, execute the script `sas.wrs.config.bat` or `sas.wrs.config.sh`. This script builds a new WAR file, `RestrictedDataReporting.war`.

9 If SAS Web Report Studio will run in Apache Tomcat, execute the deployment script `sas.wrs.tomcat.deploy.bat` or `sas.wrs.tomcat.deploy.sh`. If you will deploy the application to the BEA WebLogic Server, execute the script `sas.wrs.weblogic.prepare.bat` or `sas.wrs.weblogic.prepare.sh`. The WebLogic scripts explode the contents of the new WAR file to the directory *SAS-config-dir*`\Lev1\web\webapps\exploded\RestrictedDataReporting.war`.

10 If you will run the new SAS Web Report Studio on the BEA WebLogic Server or the IBM WebSphere Application Server, deploy the application to your server using the management console that came with the product—as you did when you initially deployed SAS Web Report Studio. (See the **instructions.html** file from your initial installation for further information on how to deploy the application.)

11 Go to the area where you have deployed RestrictedDataReporting, and locate the file **WebReportStudioProperties.xml** in the **WEB-INF** directory. This file will contain the following lines:

```
<!-- begin map accessibility check section -->

<wrs.map.accessibility.check>
  <!-- are map accessibility checks performed? -->
  <enabled>false</enabled>
  <!-- if map accessibility checks are performed, which locations are -->
  <!-- secured? This property may contain a comma-delimited list of   -->
  <!-- secured root locations; if empty, default to the value of      -->
  <!-- citation.model.repository.path.maps                             -->
  <rootlocations></rootlocations>
</wrs.map.accessibility.check>
```

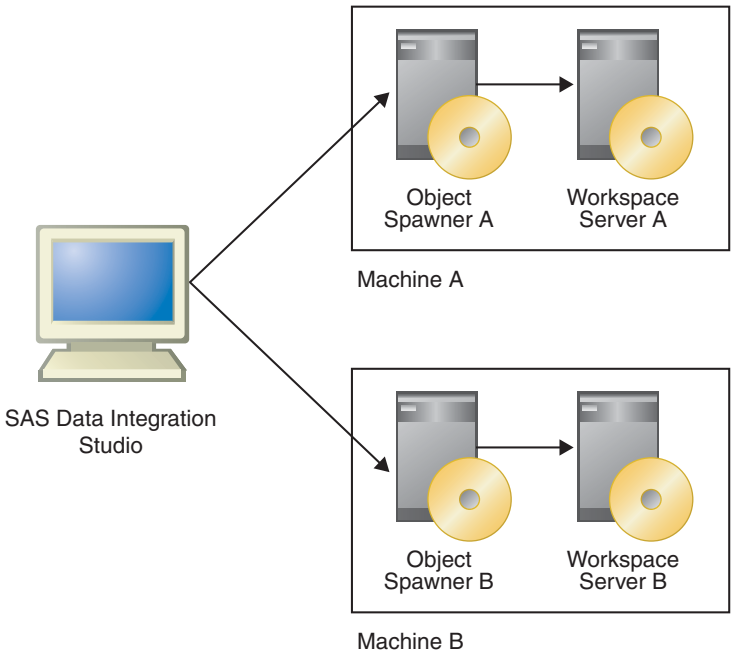
Change the value of the **<enabled>** element to **true**. Then, save your changes, and restart you Java application server.

Secure Sensitive Data Resources

Ensure that sensitive data resources are readable only by rpoolsrv (not sassrv) and the IT staff.

Load Balancing Workspace Servers for Desktop Applications

Users of desktop applications, such as SAS Data Integration Studio, can place a heavy load on a workspace server. For example, in the case of SAS Data Integration Studio, you might have a number of ETL specialists submitting long-running jobs that execute on the workspace server, and a job scheduler might be running jobs there as well. If you reach the point where you need more resources for these users, you can add a new host to your system, set up an object spawner and a workspace server on that host, and balance the workload across your new and old servers.



Each machine involved runs an object spawner that handles client requests for connections. A load balancing routine runs in the object spawners and directs client requests to the machine that is least loaded at the time that the client request is made. All the object spawners know about each other once they start up and read information from the metadata server.

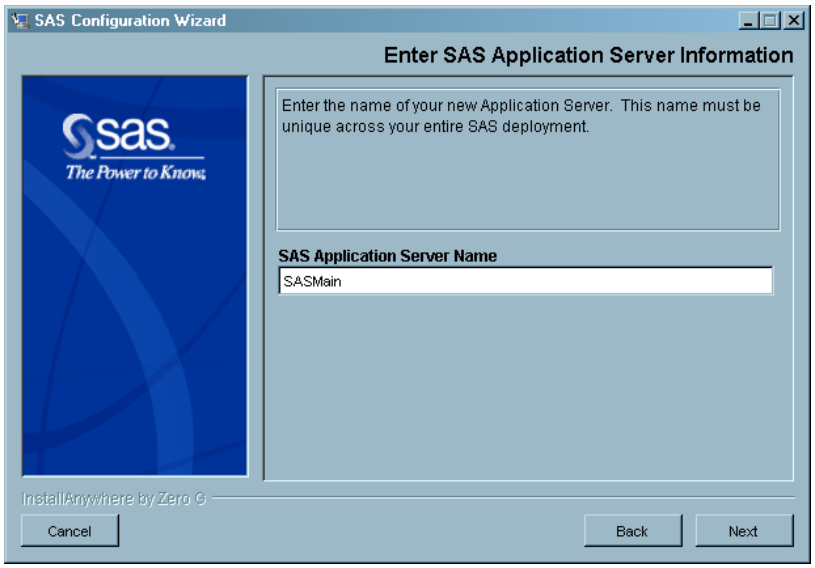
Installing the Software

As when you first set up your system, you use the SAS Software Navigator to install your software. Just let your SAS representative know that you want to add a new host to your intelligence system and that you want to run a second workspace server on that host. Your SAS representative will be able to get you the software that you need—SAS Foundation, including SAS Integration Technologies, and SAS Management Console—and to prepare the planning file that you need for the installation.

Configuring the Workspace Server and Object Spawner

After you have installed SAS Foundation and SAS Management Console, you should run the SAS Configuration Wizard on the new host. The wizard will prompt you for information about a configuration directory and about the credentials for certain users. Because you are not installing a metadata server on this machine, the wizard will also prompt you for an application server as shown in the following display.

Display 18.1 Enter SAS Application Server Information Window



The assumption behind this prompt is that your workspace server will be part of a new application server. In the case we are considering, however, this assumption is incorrect; you want to add the new server to an existing logical workspace server, which is part of an existing application server (probably called **`SASMain`**). The best practice is to enter the name of your new host computer when you are prompted for an application server. This will not have any effect—except on the HTML instructions that the SAS Configuration Wizard generates. The remaing instructions in this section assume that you have followed this best practice.

After you have provided the SAS Configuration Wizard with all of the input that it needs, it will create a configuration directory and generate a set of HTML instructions that you should follow to complete the configuration. However, the instructions will not be 100 percent correct because of the assumption mentioned previously. See the following sections for information on where you need to deviate from the generated instructions.

Note: The section titles that follow match section names in the **`instructions.html`** file. △

Start the SAS Management Console

SAS Management Console should start automatically. If it does not, start it by following the instructions that were generated by the SAS Configuration Wizard. You will need to use this application to define your workspace server and object spawner in the metadata.

Defining Your Application Server and Workspace Server

Do not follow the instructions in the section "Defining Your Application Server and Workspace Server." You have already defined an application server in your metadata —when you first installed your system—so you do not need to define one now, and you need to define your new workspace server in a manner different from the one described in **`instructions.html`**.

Note: Although you are not going to follow all of the directions in this section of **`instructions.html`**, you can cut and paste some responses from this section in order to prevent typographical errors. △

Follow these instructions instead:

1 Expand the **Server Manager** node. Fully expand all three levels of **SASMain**.

2 Highlight **SASMain - Logical Workspace Server**. Using the right mouse button, select **Add Server**. The New Server Wizard starts.

3 In the wizard's first screen, enter the name "*host-name* - **Workspace Server**," and click **Next**.

4 In the wizard's second screen, set the **Command** to **sas -config** "*SAS-config-dir***Lev1***host-name***sasv9.cfg**" and click **Next**.

5 In the wizard's third screen, select the **Bridge Connection** radio button, and click **Next**.

6 In the wizard's fourth screen, enter the following information:

- **Authentication Domain**: DefaultAuth
- **Host Name**: *new-workspace-server-host*
- **Port Number**: 8591

7 In the wizard's fifth and final screen, review the information that you have supplied, and click **Finish**. A new workspace-server icon now appears in SAS Management Console.

Define Your Object Spawner

Follow the directions in this section to define an object spawner on the new host. These directions will be correct. (If the instructions indicate that a script is available to define the object spawner, you can simply run that script instead.)

Start the Object Spawner

The directions in this section will be correct as well. However, you should not start your object spawner until after you have performed the configuration steps that are detailed in the sections "Converting the Logical Workspace Server to Load Balancing" on page 403 and "Setting Load Balancing Parameters for Each Workspace Server" on page 404.

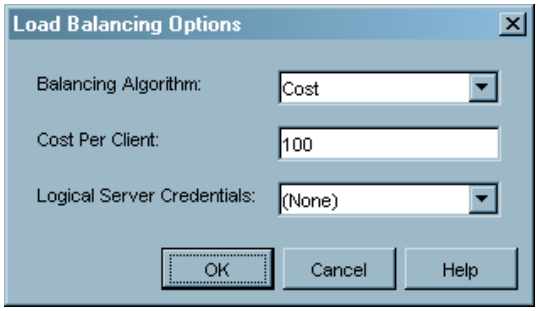
Converting the Logical Workspace Server to Load Balancing

After you have defined your new workspace server and object spawner in the metadata, you can use SAS Management Console to convert your logical workspace server to load balancing. During this process, you set some options that apply to all of the physical workspace servers in the logical workspace server.

Note: If you do not convert the logical workspace server to use load balancing, clients will be able to use only the first physical workspace server in the logical workspace server. △

To convert the logical workspace server to load balancing, perform these steps:

1 Right-click **SASMain - Logical Workspace Server**, and select **Convert To ► Load Balancing**. You will be asked whether you want to continue. Click **Yes**. The Load Balancing Options dialog box appears.



2 Set the parameters in this dialog box using the explanations in the following table, and click **OK**.

Table 18.6 Load Balancing Options

Parameter	Explanation
Balancing Algorithm	Specifies the algorithm that the object spawners should use to control load balancing. The only valid choice when you are load balancing workspace servers is Cost. The Cost algorithm specifies that client requests are processed based on the current server cost and the start-up cost of a new server. The clients' costs are added to and subtracted from the server cost as they connect and disconnect. For more information on the Cost algorithm, see the *SAS Integration Technologies: Server Administrator's Guide* at **support.sas.com/rnd/itech/doc9/admin_oma/**.
Cost Per Client	Specifies the default amount of weight (cost) that each client adds to (on connection) or subtracts from (on disconnection) the total cost of the server.
Logical Server Credentials	Shows the credentials that the object spawners on the two hosts will use to communicate about load balancing. We recommend that you use the SAS General Servers group login (**sassrv**) for this purpose. This account will be used in both directions, so it must be a network account that will be valid on both spawner hosts.

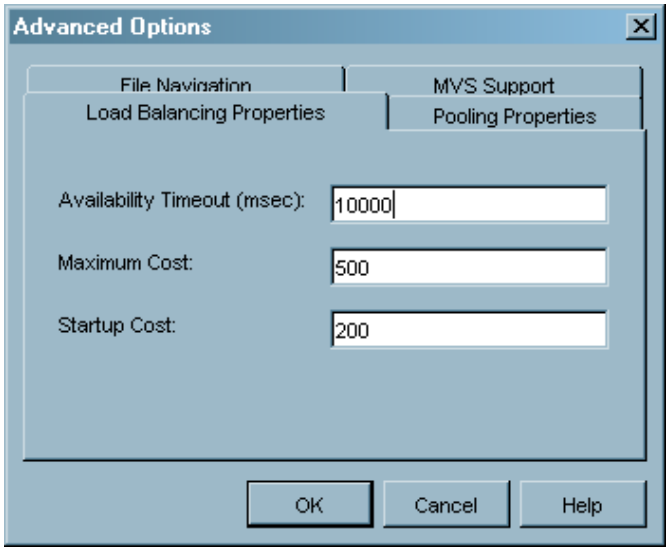
Setting Load Balancing Parameters for Each Workspace Server

There are also some load-balancing properties that you set for each of your physical workspace servers. To set these properties, perform these steps:

1 In SAS Management Console, right-click the icon for the physical server, and select **Properties** from the pop-up menu. A Properties dialog box appears.

2 Select the **Options** tab.

3 Click **Advanced Options**. An Advanced Options dialog box appears.



Select the **Load Balancing Properties** tab if it is not already selected.

4 Set the load-balancing properties using the information in the following table; then, click **OK**.

Table 18.7 Load Balancing Properties

Property	Explanation
Availability Timeout (msec)	Specifies the number of milliseconds to wait for an available server. The wait can be caused by the time that is required for a server to start or the time that is required for a running server to become available.
Maximum Cost	Specifies the maximum cost allowed on the server before requests to the server are denied. Use the value of the Cost Per Client field on the logical server to determine this value based on the number of client connections allowed.
Startup Cost	Specifies the start-up cost of the server. When a request is made to the load balancer, the load balancer assigns this start-up cost value to inactive servers. A new server is not started unless it is determined that its cost (the start-up cost) is less than the cost of the rest of the servers in the cluster. This field enables the administrator to control the order in which servers are started. After a server is started, the cost value is 0. When a client connects to the server, the server's cost value is increased.

5 Click **OK** in the Properties dialog box.

You can now start your new object spawner. You should also restart your original object spawner. At this point, the load balancing that you have set up will be in effect.

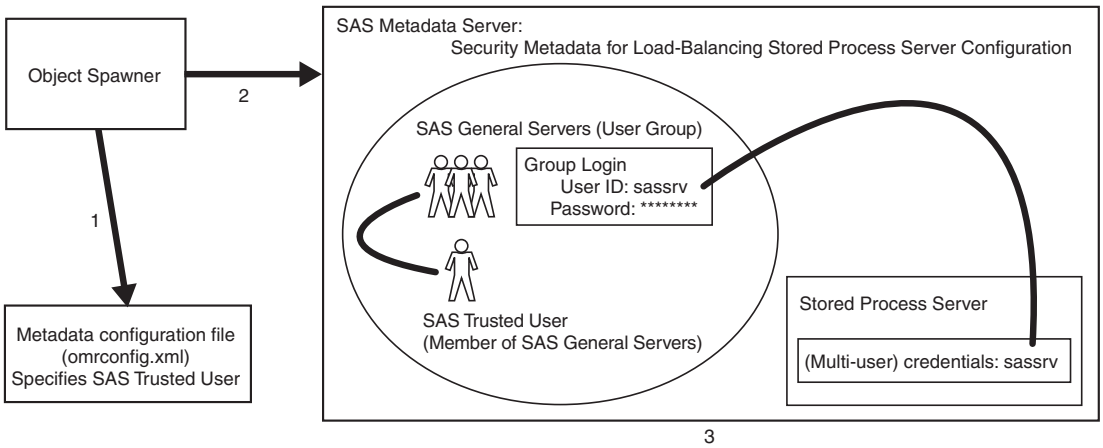
Note: For information about how to make sure that your workspace servers can access your data sources as you scale your system up, see "Data and Catalogs for Servers on Multiple Machines" on page 414. △

Overview of the Initial Load Balancing Setup for Stored Process Servers

In the initial load balancing SAS Stored Process Server configuration, three MultiBridge connections are set up for the stored process server so that the object spawner can start up to three stored process server processes. The object spawner balances the workload across these processes. The object spawner runs on the server host, listens for client requests, and connects clients to the appropriate server process.

The metadata server's foundation repository contains the spawner, server, and security metadata for the load balancing stored process server configuration. The object spawner must connect to the metadata server, and the metadata must be configured appropriately, in order for the spawner to start the load balancing stored process server processes. The following figure shows the initial security setup for the load balancing stored process server and spawner configuration.

Figure 18.1 Security Metadata for a Load Balanced Stored Process Server



Note: On Windows, all user IDs would be host or domain qualified, for example, *domain-name***sastrust**. △

In the preceding figure, the object spawner obtains the metadata it needs to start a load balancing stored process server as follows:

1. When the spawner is started, it reads a metadata configuration file named **omrconfig.xml** that contains information required to access the metadata server. This metadata configuration file specifies
   - the location of the metadata server
   - the user ID that the spawner will use to connect to the metadata server.

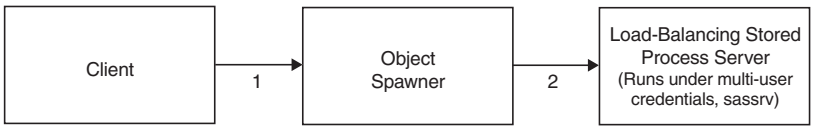
   By default, the **omrconfig.xml** file contains the user ID **sastrust**, which is owned by the SAS Trusted User (in the metadata).
2. The object spawner connects to the metadata server using the user ID specified in **omrconfig.xml**. This user's credentials are authenticated by the metadata server's authentication provider (usually the operating system).
3. On the metadata server, the connection from the object spawner is associated with the user that owns the **sastrust** user ID, SAS Trusted User. The spawner—as the SAS Trusted User—reads the metadata for the server and spawner configuration.

   Note: The SAS Trusted User can view the stored process server's multi-user login credentials (**sassrv**) because the SAS Trusted User is a member of the SAS

General Servers group, and the SAS General Servers group owns the server's multi-user login credentials. △

At this point, the object spawner has the necessary metadata to launch a stored process server. The following figure shows the flow for a client request and server launch.

Figure 18.2 Launching a Stored Process Server



1. When a client requests a server, the client is authenticated against the host authentication provider for the server.
2. If the object spawner needs to launch a new stored process server, the object spawner uses the server's multi-user login credentials (**sassrv**) to launch the load balancing stored process server.

Note: Because the stored process server runs under the credentials for the multi-user stored process server, each client can only access information that **sassrv** has permission to access. △

To summarize, in your initial load balancing stored process server configuration, you must ensure that the following security is set up properly:

- Ensure that the SAS Trusted User's credentials are specified in the metadata configuration file **omrconfig.xml**.
- Ensure that, in the foundation metadata repository, the SAS Trusted User is a member of the SAS General Servers group.
- Ensure that, in the foundation metadata repository, the group login owned by the SAS General Servers group is specified in the stored process server definition. (Using SAS Management Console, look at the Credentials tab in the properties dialog box for the server.)
- Ensure that the user ID and password of the group login for the SAS General Servers group match the credentials in a user account defined in the stored process server's host authentication provider.

To improve performance, you can distribute a workload across stored process server processes running on multiple hosts. For details, see "Load Balancing Stored Process Servers on Multiple Hosts" on page 407.

Load Balancing Stored Process Servers on Multiple Hosts

You are probably already using a load-balanced stored process server. If you performed the default configuration of your servers, requests for a stored process server might be channeled to any one of three stored process server processes. One way to scale up such a system is to define additional MultiBridge connections for an existing stored process server. However, as with workspace servers, you can also add a new host to your system, set up a stored process server there, and balance a load across hosts (as well as across processes on a host).

Installing the Software

Installing the software is easy. Just let your SAS representative know that you want to add a new host to your intelligence platform and that you want to run a second stored process server on that host. Your SAS representative will be able to get you the software that you need—SAS Foundation and SAS Management Console—and to prepare the planning file that you need for the installation

Configuring the Stored Process Server and Object Spawner

After you have installed SAS Foundation software and SAS Management Console software, you should run the SAS Configuration Wizard on the new host. The wizard will prompt you for information about a configuration directory and about the credentials for certain users. Because you are not installing a metadata server on this machine, the wizard will also prompt you for an application server as shown in the following display.

Display 18.2 Enter SAS Application Server Information Window

The assumption behind this prompt is that your stored process server will be part of a new application server. In the case we are considering, this assumption is incorrect. You want to add the new server to an existing logical stored process server, which is part of an existing application server (probably called **`SASMain`**). The best practice is to enter the name of your new host when you are prompted for an application server. This will not have any effect—except on the HTML instructions that the SAS Configuration Wizard generates.

After you have provided the SAS Configuration Wizard with all of the input that it needs, it will—as usual—create a configuration directory and generate a set of HTML instructions that you should follow to complete the configuration. However, the instructions will not be 100 percent correct because of the assumption mentioned previously. See the following sections for information on where you need to deviate from the generated instructions.

Note: The section titles that follow match section names in the **`instructions.html`** file. △

Start the SAS Management Console

SAS Management Console should start automatically. If it does not start automatically, start the application by following the instructions in this section. You will need this application to define your stored process server and object spawner in the metadata.

Defining Your Application Server and Workspace Server

Skip the instructions in the sections "Defining Your Application Server and Workspace Server" and "Edit the SAS Command for the Workspace Server." You have already defined an application server in your metadata—when you first installed your system—and you are not adding a workspace server.

Define Your Stored Process Server

Also, skip the instructions in the sections "Define Your Stored Process Server" and "Define the Stored Process Server as Supporting Load Balancing." We are assuming that you have already defined a logical stored process server called **`SASMain - Logical Stored Process Server`** and that you have configured that logical server for load balancing.

Note: Although you are not going to follow all of the directions in these sections of **`instructions.html`**, you can cut and paste some responses from these sections in order to prevent typographical errors. △

In the section "Edit the Properties of the Stored Process Server Component," replace steps 1 to 4 with the following instructions:

1. Expand the **`Server Manager`** node. Fully expand all three levels of **`SASMain`**.
2. Highlight **`SASMain - Logical Stored Process Server`**. Using the right mouse button, select **`Add Server`**. The New Server Wizard starts.
3. In the wizard's first screen, enter the name "*host-name* – **`Stored Process server`**," and click **`Next`**.
4. In the wizard's second screen, set the **`Command`** to **`sas -config "`***path-to-config-dir***`\Lev1\`***host-name***`\StoredProcessServer\sasv9_StorProcSrv.cfg"`** and click **`Next`**.
5. In the wizard's third screen, select the **`Bridge Connection`** radio button, and click **`Next`**.
6. In the wizard's fourth screen, enter the following information:
   - **`Authentication Domain`**: DefaultAuth
   - **`Host Name`**: *new-stored-process-server-host*
   - **`Port Number`**: 8601
7. In the wizard's fifth and final screen, review the information that you have supplied, and click **`Finish`**. A new stored process server icon new appears in SAS Management Console.

From this point, you can follow the directions generated by the SAS Configuration Wizard for adding MultiBridge connections to the stored process server.

Define Your Object Spawner

Follow the instructions in this section to create an object spawner on your new host—with one exception. Remove the bulleted item "*host-name*–**`Workspace Server`**" from step 6. You have not defined a workspace server on this host.

Load SAS Stored Process Samples

Skip this section because you should have loaded the metadata for these samples during your initial installation.

Start the Object Spawner

The directions in this section will be correct. However, you should not start your object spawner until after you have performed the configuration steps detailed in the sections "Setting Logical Stored Process Server Properties" on page 410 and "Setting the Load Balancing Properties for Each Stored Process Server" on page 412.

Setting Logical Stored Process Server Properties

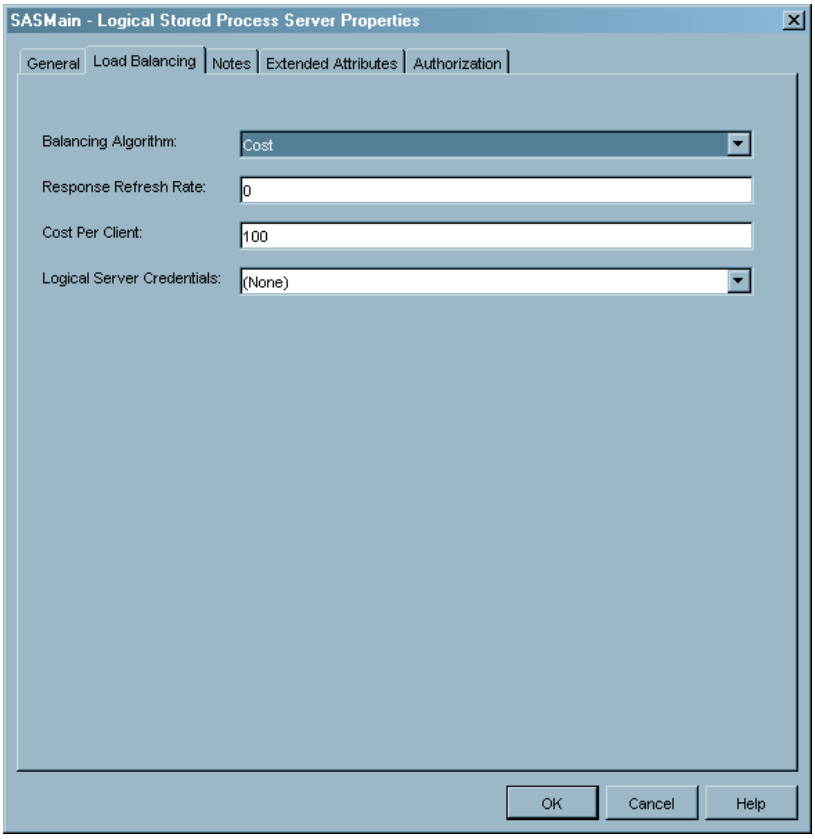
After you have defined your new stored process server and object spawner in the metadata, you can use SAS Management Console to set certain parameters for the logical stored process server that affect how the load balancing will work. To set these parameters:

1. If necessary, create a network user account that the object spawners on the two hosts will use to communicate. If you defined the SAS General Server User (**sassrv**) as a network account during pre-installation, you can skip this step.

   a. Create a network account for the SAS General Server User (**sassrv**). If you are creating a Windows domain account, be sure to grant the user the user right "Log on a batch job."
   b. In the metadata, add a new login to the group SAS General Servers. (If there is an existing login for a local **sassrv** account, remove it.) The object spawners will use the network account for **sassrv** to communicate with one another.

   Note: Because (1) the object spawners communicate with the metadata server using the **sastrust** account and (2) **sastrust** is a member of the SAS General Servers group, the object spawners are able to read the password for the sassrv account. △

2. Right-click the icon for your logical stored process server, and select **Properties** from the pop-up menu. A properties dialog box appears.
3. Select the **Load Balancing** tab. This tab contains the parameters that you can set.



4 Set these parameters using the explanations in the following table, and click **OK**.

Table 18.8 Load Balancing Parameters

Parameter	Explanation
Balancing Algorithm	Specifies the algorithm that the load balancer should use to control load balancing. Valid values are Cost and Response Time. Selecting the Cost algorithm specifies that client requests are processed based on the current server cost and the start-up cost of a new server. The clients' costs are added to or subtracted from the server cost as they connect and disconnect. Selecting Response Time specifies that client requests are allocated based on server response times. For more information about these load balancing algorithms, see the *SAS Integration Technologies: Server Administrator's Guide* at **support.sas.com/rnd/itech/doc9/admin_oma/**.
Response Refresh Rate	Specifies how often the server response times are checked. You only enter a value in this field if you selected Response Time in the **Balancing Algorithm** field, and the value should always be set to **-1**.

Parameter	Explanation
Cost Per Client	Specifies the default amount of weight (cost) that each client adds to (on connection) or subtracts from (on disconnection) the total cost of the server. (Cost algorithm only.)
Logical Server Credentials	Shows the credentials that the object spawners on the two hosts will use to communicate about load balancing. We recommend that you use the SAS General Servers group login (**sassrv**) for this purpose. This account will be used in both directions, so it must be a network account that will be valid on both spawner hosts.

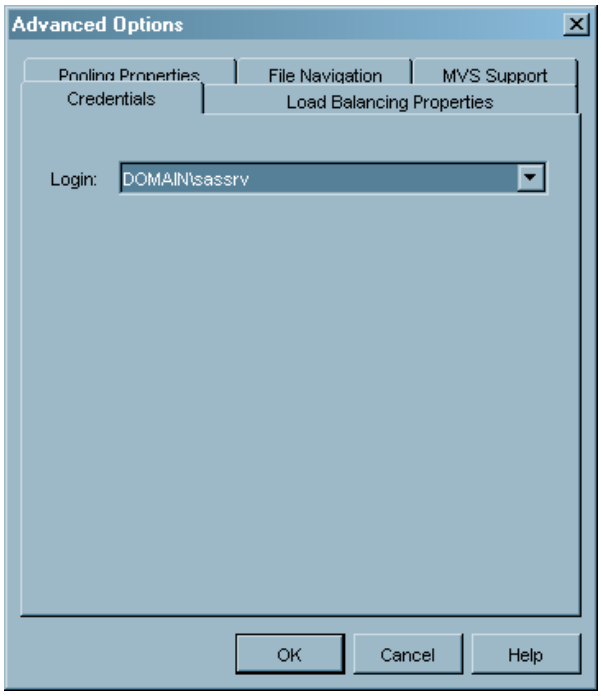
Setting the Load Balancing Properties for Each Stored Process Server

There are also some load balancing properties that you should set for each of your physical stored process servers. Follow these steps to set these properties:

1 In SAS Management Console, right-click the icon for the stored process server, and select **Properties** from the pop-up menu that appears. A Properties dialog box appears.

2 Select the **Options** tab.

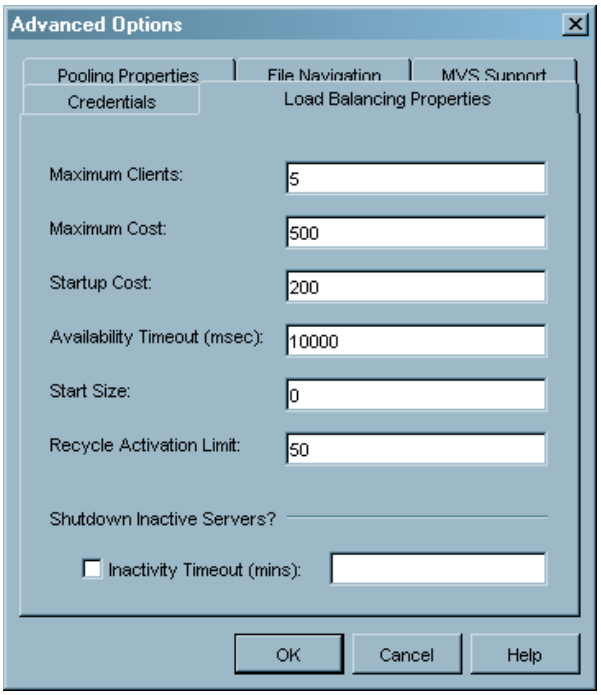
3 Click **Advanced Options**. An Advanced Options dialog box appears, and the **Credentials** tab displays.



4 From the **Login** list box, select the account that will be used to start stored process servers. We recommend that you use the SAS General Servers group login (**sassrv**) for this purpose.

Note: This should be a network account so that the stored process servers will be able to access data resources on file servers on the network. △

5 Select the **Load Balancing Properties** tab.



6 Set the load balancing properties using the information in the following table; then, click **OK**.

Table 18.9 Load Balancing Properties

Property	Explanation
Maximum Clients	The maximum number of simultaneous clients connected to this server. (Response Time algorithm only.)
Maximum Cost	The maximum cost allowed on each server before requests to the server are denied. (Cost algorithm only.)
Startup Cost	The cost of starting a server. (Cost algorithm only.)
Availability Timeout (msec)	The number of milliseconds to wait for a load balancing server to become available. This parameter is used (1) when all servers have allocated the maximum number of clients per server and (2) when the load balancer is waiting for a server to start and become available for its first client.
Start Size	The number of MultiBridge connections to start when the spawner starts.
Recycle Activation Limit	The number of times a connection to the server will be reused before it is disconnected ("recycled"). If the value is 0, then there will be no limit on the number of times a connection to the server can be reused. This property is optional. The default value is 0.

Property	Explanation
Shutdown Inactive Servers?	Indicates what you want a server to do when it is not currently serving a client. Select this check box to indicate that you want the process to terminate; otherwise, the server will remain active.
Inactivity Timeout (mins)	If you elected to shut down inactive servers, this field specifies how many minutes of inactivity must pass before the server terminates.

7 Click **OK** in the Properties dialog box.

You can start your object spawner now.

Note: For information about how to make sure that your stored process servers can access your data sources as you scale your system up, see "Data and Catalogs for Servers on Multiple Machines" on page 414. △

Data and Catalogs for Servers on Multiple Machines

In several sections of this chapter, we have discussed situations in which your original system had one server of a particular type and your scaled-up system has a server of that type on two or more machines. In particular, we have talked about

- creating a pool of workspace servers that spans machines
- placing load-balanced workspace servers on multiple machines
- placing load-balanced stored process servers on multiple machies

After scaling your system in any of these ways, you need to ensure that the new server can access the data—and possibly format catalogs—that the original server was working with. For example, suppose that your original server was using a library of SAS data sets and that the metadata object representing that library contains the path **C:\SAS***configuration-directory***\Lev1\SASMain\Data**. Your new server will not be able to access this library at its original location. One possible solution to this problem is to move a copy of the data to the new server and place it at the location stored in the library metadata object. However, this strategy may introduce data-synchronization problems. A generally preferable solution is to make sure that both servers can access a single copy of the data by providing a network path to the library.

Updating SAS Libraries

As mentioned above, when you scale your system, you need to update any SAS libraries that are being referenced using a local path. The procedure below explains how to update the definition of a library on a Windows machine that currently contains the path **C:\SAS***configuration-directory***\Lev1\SASMain\Data**:

1 In SAS Managment Console, expand the **Data Library Manager** and the **SAS Libraries** folder so that you see the icon reprenting your SAS library.

2 Right-click the library and select **Properties** from the pop-up menu that appears. A properties dialog appears.

3 In the properties dialog, select the **Options** tab.

4 Deselect the currently selected path by highlighting it in the **Selected items** list and clicking the left-arrow button.

5 Create a new path, and select it, by performing these steps:

   a Click the **New** button to bring up the New Path Specification dialog.

b In the **Name** text box, enter the UNC path to the library, for example, **\\D1234\SAS\EntBIServer\Lev1\SASMain\Data**. (Different machines on the LAN can use this same path to access the library.)

c Click **OK**.

6 Click **OK** in the properties dialog.

Updating References to User-Defined Formats

Like SAS data sets, existing user-defined format catalogs may only be available to servers running on your original SAS-server host. It is common for a server to look for such catalogs in the configuration directory on the original server host—in the directory *SAS-config-dir***\Lev1\SASMain\SASEnvironment\SASFormats**. It is also possible to specify the location of the catalog in a configuration file—as explained in "Create a User-Defined Formats Configuration File" on page 169. However, the path recorded in such a file is often a local path.

One solution to this problem is to replicate the catalog on all server hosts, but this can be less than ideal if the catalog is subject to change. A better solution is to specify the location of the catalog in *SAS-config-dir***\Lev1\SASMain\sasv9.cfg** and to make sure that the path to catalog is a network path. For example, you might change

```
-set fmtlib1 "C:\SAS\configuration-directory\Lev1\SASMain\Data\orformat"
-fmtsearch (fmtlib1.orionfmt)
```

to

```
-set fmtlib1 "\\D1234\SAS\configuration-directory\Lev1\SASMain\Data\orformat"
-fmtsearch (fmtlib1.orionfmt)
```

Accessing Data in Database Management Systems

If your new workspace or stored-process server needs to access data in a DBMS, you may again need to do some administrative work on the new server host before such access is possible. For example, you may need to:

- install a SAS/ACCESS product
- install database client software
- install a database driver
- configure a data source name

The simplest way to explain this is that you need to repeat whatever steps you took on the original SAS-server host on the newly added server.

CHAPTER

19

Best Practices for Configuring Your Middle Tier

Overview of Middle Tier Configuration

This chapter explains how you can configure middle-tier components of the SAS Intelligence Platform for better efficiency and performance.

The middle tier provides an environment for running the following SAS Web clients:

- SAS Web Report Studio
- SAS Web Report Viewer
- SAS Information Delivery Portal, or the portal Web application that is included in the SAS Web Infrastructure Kit
- SAS Web OLAP Viewer for Java

Note: In this chapter, all references to the SAS Information Delivery Portal can be assumed to include the portal Web application that the Web Infrastructure Kit includes. △

Refer to *SAS Intelligence Platform: Overview* for an introduction to the middle-tier components.

Tuning the Java Virtual Machine

Your J2EE application server or servlet container's JVM can be started with a number of options that affect its behavior. For a quick overview of where to set these options and a list of generally applicable settings, see the following subsections:

- "Which JVM Are You Using?" on page 418
- "Where to Specify JVM Options" on page 419
- "Quick Start Settings" on page 420

For more details about the various JVM options, see the following subsections:

- "Setting Just-in-Time Compiler and Memory Options" on page 421
- "Selecting a Garbage Collector" on page 423
- "Configuring the Garbage Collector" on page 423
- "Tuning the J2EE Application Server or Servlet Container" on page 424
- "Detecting Changes in JavaServer Pages and Servlets" on page 424
- "Setting the Number of Available Worker Threads (SAS Information Delivery Portal Only)" on page 424

Which JVM Are You Using?

The information in this section applies both to the JVM that is supplied by Sun Microsystems and to the JVM that is supplied by IBM. Note that these two JVMs use

different parameters. Most installations that use BEA WebLogic should use Sun's JVM and its corresponding parameters. Most installations that use IBM WebSphere should use IBM's JVM and its parameters. At the time this is being written, two exceptions are known:

1. On IBM's AIX operating system, only the JVM that is supplied by IBM should be used, even for BEA WebLogic.
2. On Sun's Solaris operating system, only the JVM that is supplied by Sun should be used, even for IBM WebSphere.

Where to Specify JVM Options

BEA WebLogic

For WebLogic version 8.1, you can set your JVM options in one of two ways: by editing the script **startManagedWeblogic.***extension* or by using the product's administrative console. Use whichever approach you have used to set other server options. The paragraphs below explain how to use each approach.

Note: If you are not running the Node Manager, you must specify your JVM options using the script. △

To set your JVM options in the **startManagedWebLogic.***extension* script, perform the following steps:

1. Change directories to *WebLogic-install-dir***\user_projects\domains***domain*.
2. Open the script in a text editor.
3. Uncomment the line reserved for setting the **JAVA_OPTIONS** environment variable, and set this variable as explained later in this section.
4. Save your changes, and close the file.

To set the JVM options from the administrative console, perform the following steps::

1. From the BEA WebLogic console, in the left panel, expand the **Servers** node in the tree.
2. Select the server that you want to configure.
3. In the right pane, select the **Remote Start** tab. That screen has an **Arguments** field where you can insert your JVM options. After inserting your options, click **Apply**.
4. Restart the server so that the new settings will be in effect.

IBM WebSphere

The IBM WebSphere administrative console is used to set Java startup parameters. The following procedure will enable you to set the parameters for IBM WebSphere version 5.1. For more information, see the IBM WebSphere documentation.

1. From the IBM WebSphere console, on the left panel, select **Servers**.
2. Select **Application Servers**.
3. On the right will be displayed a list of your servers. Select the appropriate server.
4. On the resulting screen, select **Process Definition**, then **Java Virtual Machine**.
5. Enter your Java parameters into the **Generic JVM Arguments** field.

   Note: Some parameters can be specified either in other boxes on this screen, or in the **Generic JVM Arguments** field as recommended here. Avoid placing the same information in both areas; this can have unpredictable results. △

Apache Tomcat

When the SAS Configuration Wizard runs, it creates a script that you can use to start your servlet container or J2EE application server. This script is called **startServletContainer.bat** or **startServletContainer.sh** and resides in the directory *path-to-config-dir***\Lev1\web**. This script provides one place in which you can specify JVM options. For example, if you are using the Apache Tomcat servlet container on a Windows system, the contents of the **startServletContainer.bat** script will look something like this:

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

call "%CATALINA_HOME%\bin\catalina.bat" run -security
```

In this case, you can change the options that are used to start the JVM by changing the value of the environment variable CATALINA_OPTS. You can add JVM options to other versions of this script in a similar manner.

Quick Start Settings

If you want to start with a group of settings that will provide you with a convenient starting place, find the description in the following examples that matches your situation. Then, use the options that follow that description. (You can later fine tune these settings as necessary.)

Few Users, Sun-Based JDK

If you will have 10 or fewer concurrent users and are using a Sun-based JDK whose revision level is 1.4.2 or later, use these settings:

```
-server -Xms512m -Xmx512m -XX:NewSize=64m -XX:MaxNewSize=64m
-XX:MaxPermSize=128m -Xss128k -XX:-UseTLAB -XX:+UseConcMarkSweepGC
-XX:+DisableExplicitGC -Dsun.rmi.dgc.client.gcInterval=3600000
-Dsun.rmi.dgc.server.gcInterval=3600000 -Djava.awt.headless=true
```

Notes:

- The value of **-Xss** shown above is appropriate for Windows systems. On UNIX systems, use the option **-Xss256k** instead.
- Do not use the option **-XX:-UseTLAB** on HP-UX Itanium systems. There is a known problem with its use.
- In version 1.4.2_04 of the JVM and earlier versions, setting **-XX:MaxNewSize** to a value greater than the value of **-XX:NewSize** will not enable the Young Generation to grow larger than the latter value.

More Users, Sun-Based JDK

If you will have more than 10 concurrent users and are using a Sun-based JDK whose revision level is 1.4.2 or later, use these settings:

```
-server -Xms1280m -Xmx1280m -XX:NewSize=160m -XX:MaxNewSize=160m
-XX:MaxPermSize=128m -Xss128k -XX:-UseTLAB -XX:+UseConcMarkSweepGC
-XX:+DisableExplicitGC -Dsun.rmi.dgc.client.gcInterval=3600000
-Dsun.rmi.dgc.server.gcInterval=3600000 -Djava.awt.headless=true
```

Notes:

- The value of **-Xss** shown above is appropriate for Windows systems. On UNIX systems, use the option **-Xss128k** instead.
- Do not use the option **-XX:-UseTLAB** on HP-UX Itanium systems. There is a known problem with its use.
- In version 1.4.2_04 of the JVM and earlier versions, setting **-XX:MaxNewSize** to a value greater than the value of **-XX:NewSize** will not enable the Young Generation to grow larger than the latter value.

Few Users, IBM-Based JDK

If you will have 10 or fewer concurrent users and are using an IBM-based JDK whose revision level is 1.4.1 or later, use these settings:

```
-Xms256m -Xmx512m -Xgcpolicy:optavgpause -Xss128k -Xoss128k
-Xpartialcompactgc -Xgcpolicy:optthruput
-Dsun.rmi.dgc.client.gcInterval=3600000
-Dsun.rmi.dgc.server.gcInterval=3600000 -Djava.awt.headless=true
```

If the JDK is installed on a UNIX system, change the value of the **-Xss** and **-Xoss**options to 256k.

More Users, IBM-Based JDK

If you will have more than 10 concurrent users and are using an IBM-based JDK whose revision level is 1.4.1 or later, use these settings:

```
-Xms640m -Xmx1280m -Xss128k -Xoss128k
-Xpartialcompactgc -Xgcpolicy:optthruput
-Dsun.rmi.dgc.client.gcInterval=3600000
-Dsun.rmi.dgc.server.gcInterval=3600000 -Djava.awt.headless=true
```

If the JDK is installed on a UNIX system, change the value of the **-Xss** and **-Xoss**options to 256k.

Setting Just-in-Time Compiler and Memory Options

A number of JVM options affect which compiler the JVM uses and the amount of memory that the JVM uses for such things as the object heap.

Sun-Based JVM

You should always use the Just-in-Time compiler if one is available. You enable the JIT compiler with the **-server** option. The following list includes the relevant memory settings and their respective options:

- Minimum heap size (**-Xms**)
- Maximum heap size (**-Xmx**)
- Thread stack size (**-Xss**)
- Minimum new generation size **(-XX:NewSize**)
- Maximum new generation size (**-XX:MaxNewSize**)
- Maximum permanent generation size (**-XX:MaxPermSize**)

Minimum and maximum heap size, **-Xms** and **-Xmx** respectively, define the total amount of memory that the JVM has at its disposal. To eliminate heap growth overhead, set both of these options to the same value. The recommended value for heap size

depends on which applications will be running and how much load will be supported. For 10 or fewer concurrent users, set the heap size to 512 MB (**-Xms512m -Xmx512m**). For more than 10 users, set the heap size to 1280 MB (**-Xms1280m -Xmx1280m**).

Although a 1280 MB heap is the maximum possible heap size on a 32-bit Windows system, most versions of UNIX allow the JVM to address up to 3 GB of memory. If your J2EE application server is running on UNIX and you anticipate a load of hundreds of users, you might want to scale the implementation. However, large heap sizes can cause longer garbage collection times, so it is important to expand the heap only as much as necessary. We recommend an upper limit of 1.5GB.

Note: If this maximum heap size is affecting the number of users that you can accommodate, you should consider creating a cluster of servers. △

Thread stack size defines the default amount of memory allocated to each native thread spawned by the JVM. Keeping this value as low as possible allows the JVM to dedicate more process memory to the heap, which in turn increases scalability. The recommended setting is 128 KB for Windows systems and 256 KB for Solaris systems. You might need to adjust this number for other platforms or for additional load. An additional optimization is possible for Sun JVMs running on SPARC processors. You can include the **-XX:-UseTLAB** option, which tells the JVM to minimize thread stack usage. For all other Sun JVMs, this is the default behavior.

Finally, you can size different memory regions, or generations, appropriately to ensure efficient garbage collector performance. You use the minimum new generation size (**-XX:NewSize**), maximum new generation size (**-XX:MaxNewSize**), and maximum permanent generation size (**-XX:MaxPermSize**) to control the sizes of the various memory regions. The new generation should receive about 12.5% of total memory, up to a maximum of about 256 MB. Both the minimum and maximum new generation settings should be set to the same value to avoid generation growth overhead. Assuming a 1280 MB heap, the calculation and associated options would look like this:

1280 x 12.5% = 160 MB (**-XX:NewSize=160m -XX:MaxNewSize=160m**)

The permanent generation is used to store loaded Java class definitions and extremely long-lived objects. Given the variety of components, services, and frameworks in use by SAS applications, the maximum permanent generation size should be set at 128 MB (**-XX:MaxPermSize=128m**).

IBM JVM

The following list includes the relevant memory settings and their respective options:

- Minimum heap size (**-Xms**)
- Maximum heap size (**-Xmx**)
- Native code thread stack size (**-Xss**)
- Java code thread stack size (**-Xoss**)

Minimum and maximum heap size, **-Xms** and **-Xmx** respectively, define the total amount of memory that the JVM has at its disposal. To allow the JVM to manage memory efficiently, set the minimum heap size below the maximum heap size. The recommended value for heap size depends on which applications will be running and how much load will be supported. A good rule of thumb is to set the minimum heap size to one half of the maximum heap size, but do not set it lower than 256 MB. For 10 or fewer concurrent users, set the heap sizes to 256 MB and 512 MB (**-Xms256m -Xmx512m**). For more than 10 users, set the heap sizes to 640 MB and 1280 MB (**-Xms640m -Xmx1280m**).

Although a 1280 MB heap is the maximum possible heap size on a 32-bit Windows system, most versions of UNIX allow the JVM to address up to 3 GB of memory. If your

J2EE application server is running on UNIX and you anticipate a load of hundreds of users, you might want to scale the implementation. However, large heap sizes can cause longer garbage collection times, so it is important to expand the heap only as much as necessary. We recommend an upper limit of 1.5 GB.

Note: If this maximum heap size is affecting the number of users that you can accommodate, you should consider creating a cluster of servers. △

Native code and Java code thread stack size defines the default amount of memory allocated to each native thread spawned by the JVM. Keeping the values as low as possible allows the JVM to dedicate more process memory to the heap, which in turn increases scalability. The recommended setting for **`-Xss`** and **`-Xoss`** is 128 KB for Windows systems and 256 KB for UNIX systems.

Selecting a Garbage Collector

The following subsections explain how to specify a garbage collector.

Sun-Based JVM

You should use the concurrent mark and sweep collector, which you enable by specifying the **`-XX:+UseConcMarkSweepGC`** option. The concurrent mark sweep collector is implemented in the old generation and, by default, its use automates the parallel copying collector in the young generation.

IBM JVM

You control the IBM JVM's garbage collection behavior using the **`-Xgcpolicy`** switch. The **`optthruput`** policy provides the default garbage collection behavior for the IBM JVM, which is optimized for response time. If you see long pauses in response time, you might want to investigate the use of the **`optavgpause`** policy.

Configuring the Garbage Collector

After you have chosen a garbage collector, configure the collector appropriately.

Sun-Based JVM

You can also influence how often the garbage collector runs by using the three options mentioned below. The **`-XX:+DisableExplicitGC`** option prevents Java code from invoking the garbage collector. In addition, there are two Java properties that you can set to control distributed garbage collection. (This is important because most SAS Java applications use Remote Method Invocation, which in turn uses distributed garbage collection.) Setting the **`sun.rmi.dgc.client.gcInterval`** and **`sun.rmi.dgc.server.gcInterval`** command line properties to 3600000 will reduce the frequency of full collections on the host machines..

IBM JVM

Select the **`-Xpartialcompactgc`** parameter, which spreads the time of compacting the memory across multiple collections, rather than waiting until memory is completely fragmented. If this parameter is not selected, garbage collection times can be excessive, especially under heavy load.

Tuning the J2EE Application Server or Servlet Container

In addition to specifying Java Virtual Machine options, you can improve the performance of SAS Web Report Studio, SAS Information Delivery Portal, and SAS Web OLAP Viewer for Java by configuring other aspects of your servlet container or J2EE application server's behavior. For example, two obvious ways to improve the performance of any Web application are

- to limit the frequency with which servers check for updated JavaServer Pages and servlets
- to make sure that the server can create sufficient threads to service incoming requests.

The following sections explain how to change these settings on the Java application servers that are supported by the SAS Intelligence Platform.

Detecting Changes in JavaServer Pages and Servlets

Most servlet containers and J2EE application servers perform a periodic check of compiled class files and source files to determine whether a servlet or JavaServer Page has been edited. This behavior is only appropriate while applications are under development. In your production environment, you should disable these features. The following subsections explain how to do this for the supported J2EE application servers and Apache Tomcat.

BEA WebLogic Server

BEA WebLogic does not require tuning for JSP changes for use with SAS Web Report Studio and SAS Information Delivery Portal. The **weblogic.xml** file that is shipped with both applications performs all necessary tuning functions.

IBM WebSphere

IBM WebSphere does not require tuning for JSP changes or the number of threads available.

Apache Tomcat

1. Open the file *Tomcat-install-dir***\conf\web.xml**.
2. Add the following XML to the <init-param> block for the servlet with the servlet name jsp:

```
<init-param>
   <param-name>reloading</param-name>
   <param-value>false</param-value>
</init-param>
<init-param>
   <param-name>development</param-name>
   <param-value>false</param-value>
</init-param>
```

3. Restart the Tomcat server.

Setting the Number of Available Worker Threads (SAS Information Delivery Portal Only)

Because each user request requires a server thread to service it, the server must be able to start a sufficient number of threads to service the expected load. The following

sections explain how to control the number of available of threads on the different supported servers.

WebLogic Server

1. Log on to the administration console.
2. Open the Servers node of the tree.
3. Right-click a server definition.
4. Select the **`View Execute Queues`** menu option.
5. Select the **`Configure a new Execute Queues`** menu option.
6. Enter **`sas.portal.default`** as the queue name.
7. Click **`Create`**.

Note: The default **`Thread Count`** value of 25 should be sufficient for most loads. The BEA Web site (**`http://e-docs.bea.com/platform/docs81/index.html`**) contains specific information about tuning execute queues. △

Apache Tomcat

1. Open the file *Tomcat-install-dir***`\conf\server.xml`**.
2. Locate the **`Connector`** element for the server's listening port. By default, this is port 8080.
3. Change the value of the **`maxProcessors`** attribute to a number greater than 75 (the default).
4. Restart the server.

Note: The default value of 75 provides good performance for most loads. You should need to change this setting only for very heavy loads. △

Improving the Performance of WebLogic on HP-UX

If you are running BEA WebLogic Server on a HP-UX system, use the HPjconfig utility to determine the optimal settings for your kernel parameters. You might need to change some of the values to improve performance and to prevent Out of Memory errors.

For more information about the HPjconfig utility, see **`http://www.hp.com/products1/unix/java/java2/hpjconfig/index.html`**.

Sample Middle-Tier Deployment Scenarios

This section describes sample deployment scenarios for the middle-tier components. These scenarios can help you design a middle-tier configuration that meets the needs of your organization with regard to performance, security, maintenance, and other factors that are described later in this section.

Overview of Middle-Tier Deployment Scenarios

As with all tiers in the SAS Intelligence Platform, deployment of the middle tier involves careful planning. When you design and plan the middle tier, you must balance performance requirements against a number of other criteria. To understand these criteria and to evaluate sample deployment scenarios, see the following subsections:

□ "Criteria for Choosing a Middle-Tier Configuration" on page 426

The scenarios that are presented here range from simple to complex. Scenario 1 represents the deployment that results from running the SAS Software Navigator and the SAS Configuration Wizard to install and configure middle-tier components. Scenarios 2 and 3 provide advanced features, such as greater security and efficiency, but require more effort to implement and to maintain.

All scenarios include the same SAS server tier, which consists of a SAS Metadata Server that resides on a dedicated machine, and additional systems that run various SAS application servers, including SAS Workspace Servers, SAS Stored Process Servers, and SAS OLAP Servers.

Criteria for Choosing a Middle-Tier Configuration

Before you can design a middle-tier configuration that is suitable for your organization, you must understand your organization's requirements. For example, the size of your organization and the level of security that your organization requires are two factors that you should consider. There are additional considerations that must be made as well in order to provide a robust, scalable, and secure middle-tier environment.

In addition to defining clear requirements, you should identify the priorities of those requirements and determine whether there are any conflicting requirements. For example, you might want a high level of security, but high security can be expensive to maintain and can restrict the scalability of the system. Establishing priorities for potentially competing objectives will facilitate your decision-making process. The information in the following tables can help you identify and clarify requirements.

Note: The implementation considerations in the following tables are all described in the remaining sections of this topic. △

Security

Here are some things to consider when you plan your security implementation:

Table 19.1 Security Criteria

Category	Questions to Ask	Implementation Considerations
Physical access	How will clients physically access your application? Will they need external access through the Internet—that is, from outside your corporate firewall? Will access come from workstations and other client devices that reside on your intranet? Will external access require or support access through a virtual private network (VPN) ?	Whether and how to use firewalls to protect internal components. Whether and where to use the Secure Sockets Layer (SSL) protocol.
Authentication and authorization	Is it acceptable for all members of your organization to look at your organization's data and reports? Do any resources require access control? Do you have more than one type of user? Do you need to limit access to certain types of users? Will people outside your organization have access to some of your data? What is the business cost that you will incur if your sensitive data is accessed by the wrong people?	Whether there is a need to authenticate users in order to support authorization.
Threats	What threats concern you the most? □ External unauthorized entry? □ Internal unauthorized access to sensitive data? □ Packet sniffing to gain unauthorized information? □ Denial of service attacks? □ Modification or falsification of sensitive data?	Whether and how to use firewalls to protect internal components. Whether and where to use SSL.

Availability

Here are some things to decide about the level of availability that you provide to Web application users:

Table 19.2 Availability Criteria

Category	Questions to Ask	Implementation Considerations
Planned downtime	Does the deployment require some sort of fault isolation so that one system failure won't cripple all usage? How much downtime can your application tolerate? Will you have planned periodic outages, or will the system be expected to be up continuously?	Whether to use a J2EE application server cluster and bring down only one server at a time. Whether to use clustered HTTP proxy servers.
Unplanned outages	How fast must the system be brought back up after an unplanned outage? What is the business cost associated with unplanned failure?	Whether to use a J2EE application server cluster, so that if a server goes down, the remaining servers in the cluster can continue with limited functionality. Whether to use clustered HTTP proxy servers.

Performance and Scalability

Here are some considerations related to how well your Web applications handle the load of expected users:

Table 19.3 Performance and Scalability Criteria

Category	Questions to Ask	Implementation Considerations
Responsiveness	What kind of response time is acceptable to your expected users? What is the business cost associated with delays?	Whether to use a J2EE application server cluster to maintain an even load across all servers. Whether to use clustered HTTP proxy servers. Whether to use load-balancing hardware or software.
Number of users	Does the deployment need to support large numbers (hundreds) of concurrent users? Will requests come in as a steady stream, or will you have a burst of traffic at certain points in the business cycle, such as a particular time of the day or a particular day of the month?	Whether to use a J2EE application server cluster. Whether to use clustered HTTP proxy servers. Whether to use load-balancing hardware or software.
Extensibility	Do you anticipate growth in usage over time? How quickly must your production application environment adapt to change?	Whether to use an HTTP server as a proxy to the J2EE application server. Whether to use a J2EE application server cluster. Whether to use clustered HTTP proxy servers. Whether to use load-balancing hardware or software.

Maintainability

These considerations help you quantify how difficult it will be to maintain the system:

Table 19.4 Maintainability Criteria

Category	Questions to Ask	Implementation Considerations
Maintenance	Who will maintain your system? Will the maintenance staff consist of one person or a team of people? What level of monitoring will take place? Will automated processes be required in order to deploy and configure components of the system?	Whether you can maintain a particular level of security, scalability, and availability.
Upgrades	How will new components and updates be introduced? Will there be a test environment for staging modifications and updates? How much manual work is involved with upgrades?	Whether upgrades will require more resources than you have.

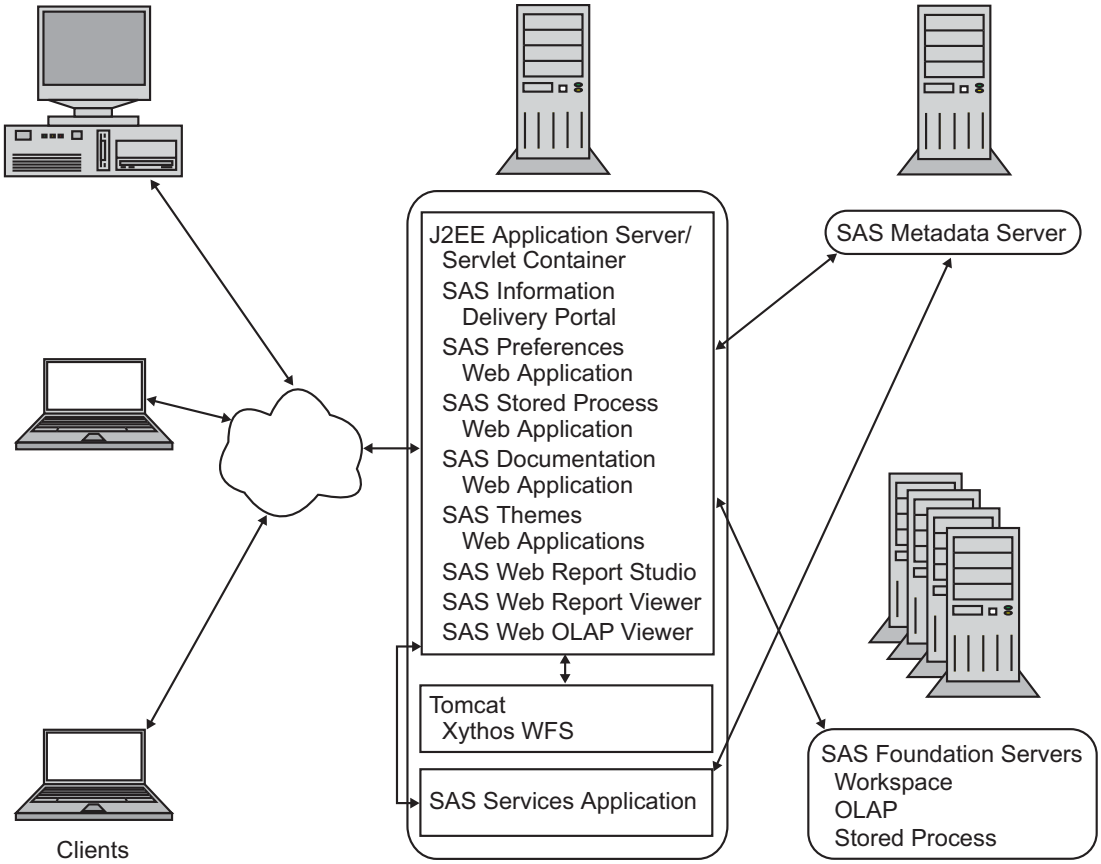
Scenario 1: Web Applications Deployed in a Single J2EE Application Server

Scenario 1 illustrates the most basic setup, one in which all of the SAS middle-tier components are installed on a single system.

Note: This scenario represents the deployment that results from running a planned Advanced installation of the middle tier using the SAS Software Navigator. If you want more sophisticated configurations, such as those presented in Scenarios 2 and 3, then you must manually modify your deployment after you run the SAS tools. △

The following figure illustrates the configuration for Scenario 1.

Figure 19.1 Scenario 1: Middle Tier on a Single System



After installation, the system contains the following software:

- J2EE application server or servlet container (BEA WebLogic, IBM WebSphere, or Apache Tomcat)
- The following SAS Web applications, which run in the J2EE application server or servlet container:
  - SAS Information Delivery Portal (or a portal Web application that you develop using the SAS Web Infrastructure Kit)
  - SAS Preferences Web application
  - SAS Themes Web applications
  - SAS Stored Process Web application
  - SAS Documentation Web application
  - SAS Web Report Viewer
  - SAS Web Report Studio
  - SAS Web OLAP Viewer
- WebDAV content services, which run in a separate Tomcat container

  The Xythos WebFile Server (WFS) is a WebDAV server that is configured by default to run in its own separate Tomcat servlet container. Xythos WFS requires a database, but this detail is not shown in the diagram. Xythos WFS can be configured with a PostgreSQL, IBM DB2, Oracle, or Microsoft SQL Server database.
- The SAS Services application, which runs in a separate Java Virtual Machine process

This type of deployment requires little additional configuration or tuning when you use an appropriately designed SAS planning file.

Following are the advantages and disadvantages of using this scenario:

- Advantages:
  - is easy to set up and configure
  - is least expensive of the three scenarios
  - is ideal for a development or test environment where frequent changes might be required
- Disadvantages:
  - does not adequately support large numbers (hundreds) of concurrent users
  - is less secure and scalable than the other scenarios
  - has no provision for improving performance
  - has no provision for serving users during planned or unplanned downtime

Scenario 2: Static Content Deployed in an HTTP Server Proxy

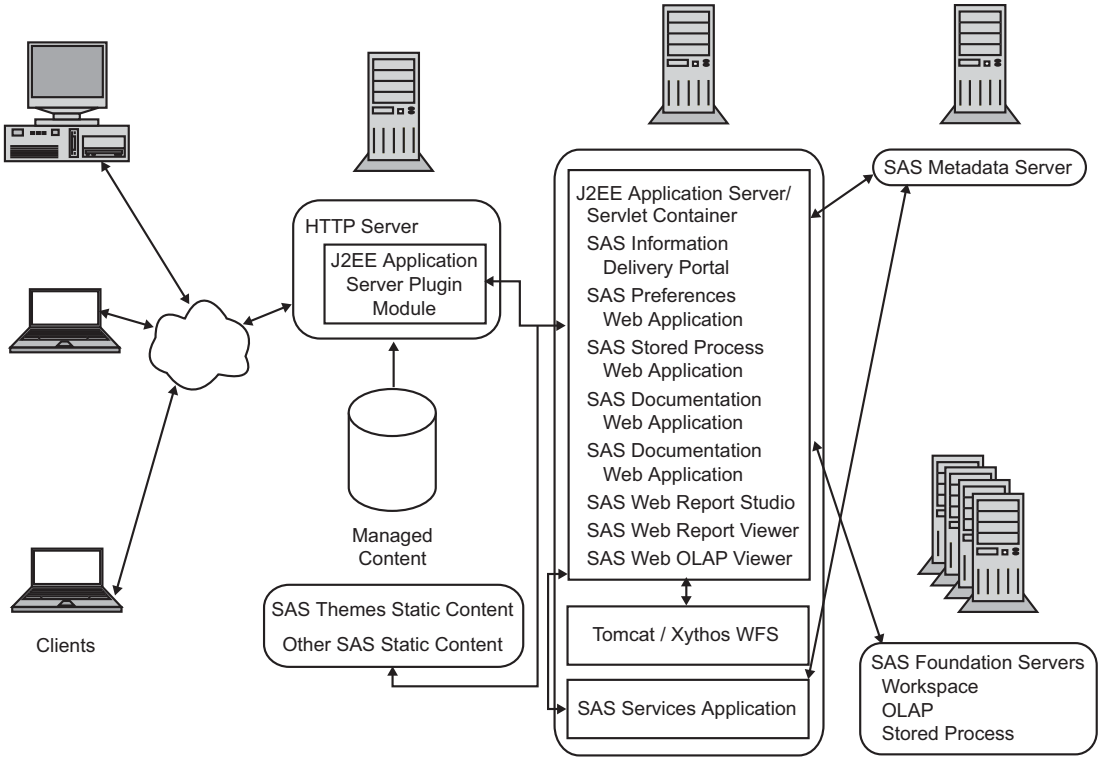
This configuration delivers static HTML content to clients from a separate HTTP server, such as Apache HTTP Server or Microsoft Internet Information Services (IIS), rather than from the J2EE application server or servlet container.

When a browser requests static content (for example, an HTML file) from a server, the server simply returns the requested document to the browser, and the browser displays the document. Dynamic content requests, however, involve some type of data manipulation that the server must perform before returning the requested page.

By design, an HTTP server is ideally suited for handling static content efficiently. Allowing an HTTP server to handle as much static content as possible enables the J2EE application server to concentrate on dynamic, more resource-intensive content.

The following figure illustrates the configuration for Scenario 2.

Figure 19.2 Scenario 2: Using a Proxy HTTP Server



In a typical configuration, the HTTP server acts as a proxy that handles all client requests for static content, and forwards requests for dynamic content to the J2EE application server or servlet container. The J2EE application server provides a module or plug-in that enables communication with the HTTP server. In this configuration, the J2EE application server is not directly exposed to clients.

Static Content to Deploy for SAS Web Applications

SAS Web Report Studio, SAS Web Report Viewer, and SAS Information Delivery Portal include static content in the form of HTML files, images, cascading style sheets, XML files, and scripts. The SAS Information Delivery Portal also contains two or more SAS Themes applications, which consist entirely of static files that control the portal's design and appearance (you must unpack the applications before you can deploy the static content). For information about deploying this static content, see "Configuring Apache HTTP Server to Serve Static Content for SAS Web Applications" on page 440.

Note: If you deploy SAS static content in an HTTP server, then you must be sure to redeploy this content if you later install a SAS software upgrade that includes new files for the static content. △

Advantages and Disadvantages of Using Scenario 2

Following are the advantages and disadvantages of using this scenario:

- Advantages:
  - provides faster response time for static content
  - stores all configuration details about Web applications in a central location (the HTTP server), so it is easier to move Web applications to different machines

- protects the J2EE application server better because it is not exposed directly to clients
- can be scaled to support a cluster of application servers and a demilitarized zone (described in the next scenario)

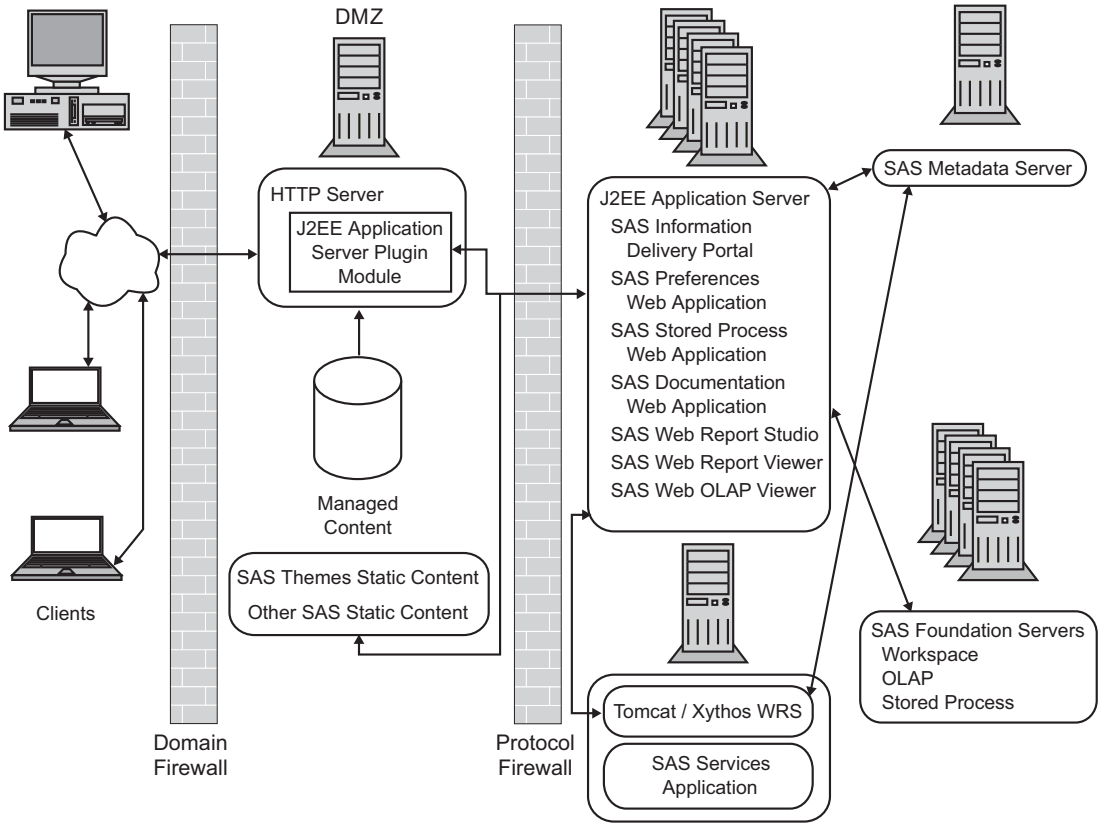
- Disadvantages:
  - requires multiple network hops and multiple servers for the proxy model to handle each dynamic request and each response
  - is not configured using the SAS Configuration Wizard; must be manually maintained, and must be manually upgraded in the event of a new SAS release

Scenario 3: Web Applications Deployed Across a J2EE Application Server Cluster

Scenario 3 includes a cluster of J2EE application servers in a deployment that implements a secure demilitarized zone (DMZ).

The following figure illustrates the configuration for Scenario 3. Note that the middle-tier software is now distributed across multiple machines. The entire column of nodes (J2EE application server and SAS Web applications, SAS Services application, and Xythos) represents the middle-tier.

Figure 19.3 Scenario 3: Clustered J2EE Application Servers and a Demilitarized Zone



Note: As indicated in the above figure, if you configure a cluster of J2EE application servers, then you must deploy all the SAS Web applications to all nodes of the cluster. △

In the previous figure, note that the SAS Services Application resides on a host that is separate from the cluster of J2EE application servers. This separation serves only to illustrate that the SAS Services Application is not replicated along with the other Web applications that are deployed across the cluster. The SAS Services Application could just as well reside on any of the hosts in the cluster. If you choose to deploy the SAS Services Application on a separate host, be aware that this deployment requires additional configuration after the initial installation. For instructions, see "Redistributing the SAS Services Application (and Java RMI Server)" in the *SAS Web Infrastructure Kit: Administrator's Guide* at **`http://support.sas.com/rnd/itech/doc9/portal_admin/deploy/ag_distssapp.html`**.

Understanding Clusters

In order to provide greater scalability, availability and robustness, WebLogic and WebSphere support some form of clustering. With clustering, multiple J2EE application server instances participate in a load-balancing scheme to handle client requests. Workload distribution is usually managed by the same application server plug-in module that enables the use of an HTTP proxy server for static content (see "Scenario 2: Static Content Deployed in an HTTP Server Proxy" on page 432).

Individual J2EE application server instances in a cluster can coexist on the same machine, or they can be distributed across a group of server machines. Each distribution mode has advantages and disadvantages, and you should evaluate these carefully when designing your deployment.

A different approach to load distribution involves merely deploying individual Web applications on separate, non-clustered J2EE application servers. Though this approach reduces the memory load for any given server, a clustering strategy is preferable. Deployment is easier to manage with a cluster because all machines and server instances are identically configured. Furthermore, J2EE application servers provide deployment management services that facilitate management of a cluster. It is relatively easy to add additional nodes and increase the size of the cluster.

Note: The Apache Tomcat version that is used in the current release of the SAS Intelligence Platform provides only limited support for clustering. For this reason, we recommend that you use WebLogic or WebSphere to implement clustering. △

Requirement for Session Affinity

For SAS Web Report Studio and SAS Information Delivery Portal to be deployed into a clustered environment, the J2EE application servers must implement session affinity. *Session affinity* is an association between a J2EE application server and a client that requests an HTTP session with that server. With session affinity, once a client has been assigned to a session with an application server, the client remains with that server for the duration of the session. This association is known in the industry by several terms, including session affinity, server affinity, and sticky sessions. By default, session affinity is enabled in WebSphere and WebLogic.

Why session affinity is required: Although WebSphere and WebLogic provide the ability to migrate HTTP sessions from one server to another, the SAS Web applications do not support this capability. Business intelligence sessions often contain large data elements, such as results sets from ad-hoc queries, reporting, and analytical tasks, that cannot be migrated easily among J2EE application servers.

Understanding Demilitarized Zones

Many organizations use a series of firewalls to create a demilitarized zone (DMZ) between their servers and the client applications. A DMZ buffers the servers from clients, whether those clients reside within the organization's computing infrastructure or reside outside the organization on the Internet.

In the previous figure, the outer firewall that connects to the public network is called the Domain Firewall. Only a few ports are typically allowed to be opened through this firewall. Often, only HTTP and HTTPS connections using the standard ports 80 and 443 respectively can be opened. Servers that reside directly behind this firewall are exposed to a wide range of clients through these limited ports, and as a result the servers are not fully secure.

An additional firewall named the Protocol Firewall sits between these non-secure machines and the secure server network. The Protocol Firewall might allow a few additional ports to be open, but the range of machines allowed to make connections is typically restricted to the servers that reside in the DMZ.

The DMZ usually contains HTTP servers, proxy servers, and load-balancing proxy software. The DMZ should not contain your application servers or any SAS servers that handle important business logic, data, or metadata.

If your applications will be accessed by clients through the Internet, then you should include a DMZ as part of your deployment in order to safeguard critical information. To be extra safe, you might want to implement a DMZ even when user workstations and client devices connect to secure servers within your organization's intranet.

Advantages and Disadvantages of Using Scenario 3

Following are the advantages and disadvantages of using this scenario:

- Advantages:
  - provides a level of fault isolation that is not possible with a single J2EE application server. If a node fails in a cluster, only those users on that node will be affected by the failure. For example, if you have an even distribution of users on a three-node cluster, only one third of the users would be affected by the loss of a single node.
  - provides a higher level of security for environments where external clients connect over the Internet, because the back-end servers are isolated from clients by the DMZ.
  - provides better overall performance because of clustering.
- Disadvantages:
  - is more expensive than the other scenarios.
  - is not configured using the SAS Software Navigator and SAS Configuration Wizard; must be manually maintained, and must be manually upgraded in the event of a new SAS release.
  - is unsuited to the version of Tomcat that is used by the current release of SAS Intelligence Platform because that version does not adequately support a clustered environment. You should use WebSphere or WebLogic.

    Note: While SAS does not support Tomcat in the clustered configuration that is described here, you can set up a DMZ and HTTP server proxy to work with a single Tomcat server. △

Additional Considerations for Planning a Deployment

This section presents a few more considerations that you might want to evaluate when you plan your middle-tier deployment.

Load-Balancing Software and Hardware for the HTTP Servers

In Scenario 3, the J2EE application servers are clustered to balance the load and to provide increased availability. While this scenario provides redundancy for the application servers, the HTTP servers remain as potential bottlenecks and single points of failure. You can also distribute HTTP traffic across multiple HTTP servers by placing load-balancing software or hardware in front of those servers. A single load-balancing component can accept client HTTP requests and distribute those requests across a cluster of HTTP servers.

A number of vendors sell load-balancing software and hardware products for HTTP servers, including IBM, Cisco, and Nortel Networks. If you are interested in this type of load-balancing, you can explore the product lines for these and other vendors.

Secure Sockets Layer

If you are moving sensitive information across the Internet, then you will probably want to use HTTPS and Secure Socket Layer (SSL) to encrypt your communication links. SSL uses Public Key Cryptography, which is based on the implementation of a public/private key pair. Each of your servers that handles encrypted communications will need to manage certificates that contain both the private key and the public key. A description of how Public Key Cryptography and SSL work is beyond the scope of this document. However, there are many good sources for that information.

Here are some factors to consider when determining whether and how to use SSL:

- Which links do you want to encrypt? In the figures shown for the various scenarios, each arrow represents a potential communications link that could be encrypted. You should consider encrypting the following:
  - Encrypt any data that is capable of moving across the public Internet. If connections to your site go through a virtual private network (VPN), then those connections are already encrypted. Otherwise, traffic to and from your site is open to packet-sniffing by Internet users.
  - Encrypt all traffic that moves between the client and your HTTP proxy server that resides in the DMZ.
  - Always encrypt credit card numbers, social security numbers, and any other sensitive information.

  To achieve strong security, encrypt links all the way to the J2EE application server. If you are concerned about internal packet sniffing, you could encrypt everything. However, total encryption comes with a cost, as explained in the remaining considerations below.
- Some load-balancing schemes might rely on packet content for routing. When that is the case, encryption can impede the work that is performed by load-balancing software or hardware because encryption renders the packet content undecipherable.
- Cryptography requires resource-intensive computation, and this resource requirement typically reduces the amount of traffic that your servers are able to handle.
- The certificates that are used with SSL expire at fixed intervals. When a user's certificate expires, the user must obtain a new certificate before logging on to your applications. If you want a highly available system, then you should prepare for certificate renewal in advance to avoid unexpected downtime.
- You must decide whether to use certificates that are generated by a Certification Authority (CA), or whether self-signed certificates are adequate for your application. Self-signed certificates can save you money, but you are responsible for managing their distribution to clients.

For instructions on setting up SAS Information Delivery Portal and other Web applications to use SSL, see "Configuring the Portal for Secure Sockets Layer " in the *SAS Web Infrastructure Kit: Administrator's Guide* at **http://support.sas.com/rnd/itech/doc9/portal_admin/security/ag_ssl.html**.

Middle-Tier (Trusted Web) Authentication

By default, the SAS Web applications authenticate users on the metadata server. You can configure the Web applications to authenticate on the middle tier instead. When users log on to a Web application that is configured to authenticate on the middle tier, a Web server or a servlet container handles the initial authentication. First, the Web server's authentication provider verifies the user's identity. Then, the Web application uses a trusted user connection to access the metadata server.

For a description of how this trusted authentication works, see "Initial Authentication on a Web Application Server" in the "Understanding Authentication" section in the *SAS Intelligence Platform: Security Administration Guide*. See also "Configuring Portal Authentication" in the *SAS Web Infrastructure Kit: Administrator's Guide* at **http://support.sas.com/rnd/itech/doc9/portal_admin/security/ag_authprov_intro.html**. The latter reference also contains information about achieving single sign-on.

There are several places where middle-tier authentication (sometimes called Web authentication) can occur. The servlet container, an HTTP server, or a proxy server can take responsibility for authentication. In general, it is more secure to perform authentication in the DMZ, as close to the client as possible. It's best to filter out all unauthorized requests before they reach your secure servers. If a proxy HTTP server is used, it can authenticate users and pass the user credentials on to the J2EE application server or servlet container as part of the HTTP request packet.

Performing authentication in the DMZ also facilitates single sign-on. Most likely, your organization will have several applications behind a common set of proxy and HTTP servers. By having a common server handle authentication, users will not need to re-authenticate for each back-end application.

You incur some limitations when you use middle-tier authentication. For example, the SAS Information Delivery Portal's Public Kiosk is not available when you authenticate users on the middle tier. In addition, you cannot customize authentication based on your own business logic. If either of these are important capabilities, then you might prefer to use the default host method of authentication.

For instructions on setting up SAS Information Delivery Portal and SAS Web Report Studio to use middle-tier authentication, see "Changing from Host to Trusted Web Authentication" in the *SAS Web Infrastructure Kit: Administrator's Guide* at **http://support.sas.com/rnd/itech/doc9/portal_admin/installmig/ag_settrust.html**.

SAS Services Application

SAS Information Delivery Portal uses the SAS Services Application to pass session and user context to content viewers, remote portlets, and Web applications launched from the portal (for example, SAS Web Report Viewer). This Java application enables the user to pass seamlessly through to the target without the requirement for a separate logon. By default, the SAS Services Application's JVM options are set to handle a moderate number of concurrent users, in the neighborhood of 750 to 1000. For the services application to support a larger number of concurrent users, you must increase its minimum and maximum heap size (**-Xms** and **-Xmx**, respectively). The application will need approximately 128 MB of additional memory for every additional 750 to 1000 users.

To increase the minimum and maximum heap size of the SAS Services Application, add the following properties to the end of the **install.properties** file located in the

PortalConfigure subdirectory of the portal's installation directory. For example, in a typical Windows installation, this directory would be **C:\Program Files\SAS\Web\Portal2.0.1\PortalConfigure**. The values shown below will increase the minimun and maximum heap size to 256 MB.

- **$SERVICES_REMOTE_JVM_INIT_HEAP$=256**
- **$SERVICES_REMOTE_JVM_MAX_HEAP$=256**

After updating the properties file, perform the following steps:

1. Stop the Java application server running the portal Web application and the SAS Services Application.
2. Run the **configure_wik** utility to update the SAS Services Application's memory configuration.
3. Restart the SAS Services Application.
4. Restart the Java application server.

Configuring a Cluster of J2EE Application Servers

Cluster configuration varies widely between J2EE application server vendors. You should consult your vendor's documentation for configuration instructions. Note, however, that you must deploy all the SAS Web applications to all nodes of the cluster. For a visual representation, see "Scenario 3: Web Applications Deployed Across a J2EE Application Server Cluster" on page 434.

It's possible to configure a cluster that consists of just one node. You might set up a single-node cluster when your sole objective is to route browser requests to an HTTP server instead of to the J2EE application server. For this configuration, you would set the address of the single-node cluster equal to the address of the HTTP server.

The following table can help you find vendor documentation about cluster configuration:

Table 19.5 Vendor Documentation for Cluster Configuration

Product	Location of the Documentation
BEA WebLogic Server	**http://e-docs.bea.com/wls/docs81/cluster/load_balancing.html#1044135**
IBM WebSphere Application Server	**http://publib.boulder.ibm.com/infocenter/ws51help/topic/com.ibm.websphere.nd.doc/info/ae/ae/trun_wlm.html**

Note: SAS currently does not support Apache Tomcat clustering . △

The next sections describe additional manual steps that you should perform.

Additional Manual Steps for WebLogic

If you are using BEA WebLogic, then after configuring the basic cluster, you must perform a few additional manual steps before the cluster is ready to use with the SAS Web applications.

Following are the steps to configure a WebLogic cluster for the SAS Information Delivery Portal. These steps assume that Apache has been configured as the cluster

front-end. If you are using some other HTTP server, then alter the last step below as applicable:

1 Add the following lines to the bottom of the WebLogic **`httpd.conf`** file (located in the Apache installation):

```
<IfModule mod_weblogic.c>
   <Include conf/weblogic.conf />
</IfModule>
```

2 Create a file named **`weblogic.conf`** in the same directory as **`httpd.conf`**. Add the following lines to **`weblogic.conf`**:

```
WebLogicCluster commaSeparatedListOfServers
WLLogFile pathToLogFile
Debug NONE
<Location /Portal>
SetHandler weblogic-handler
CookieName sas.portal.sessionid
</Location>
```

3 Save and close both files.

4 Make sure that each of the cluster nodes points to the Apache proxy server's host name and port. This is done by using the WebLogic Administrator Console (**Protocols ► HTTP ► Advanced Options**).

Configure a WebLogic cluster for SAS Web Report Studio, SAS Web Report Viewer, and SAS Web OLAP Viewer for Java in a similar way. Their respective location and cookie names are as follows (note that the capitalization is different for "sessionID"):

```
<Location /SASWebReportStudio>
SetHandler weblogic-handler
CookieName sas.wrs.sessionID
</Location>

<Location /SASWebReportViewer>
SetHandler weblogic-handler
CookieName sas.wrv.sessionID
</Location>

<Location /SASWebOLAPViewer>
SetHandler weblogic-handler
CookieName (?? sas.swov.sessionID
</Location>
```

Additional Manual Steps for WebSphere

No manual steps are necessary if you are using IBM WebSphere.

Configuring Apache HTTP Server to Serve Static Content for SAS Web Applications

Your middle-tier deployment can use an HTTP server to handle requests for static content and to forward requests for dynamic content to your J2EE application server or

servlet container. This configuration makes efficient use of the HTTP server, and enables your servlet container to devote its resources to dynamic content. The performance benefits are particularly notable for large-scale deployments that include a cluster of servlet containers. For an overview of this configuration, see "Sample Middle-Tier Deployment Scenarios" on page 425.

In order to offload static content to the HTTP server, all incoming traffic must be routed through the HTTP server, which then passes requests for dynamic content to the servlet container. This section provides an example of how to deploy static content on the Apache HTTP server for the following SAS Web applications:

- SAS Web Report Studio
- SAS Web Report Viewer
- SAS Information Delivery Portal
- SAS Web OLAP Viewer for Java

There is more than way one to configure Apache, and your configuration might differ from what is presented here.

The example also illustrates the use of a cluster of J2EE application servers. This configuration helps ensure that the application that runs in the container correctly sets the address of the HTTP server as the address for all of the application's pages. The cluster configuration also provides greater computation capacity, redundancy, and other benefits as described in "Scenario 3: Web Applications Deployed Across a J2EE Application Server Cluster" on page 434. As noted in scenario 3, Apache Tomcat currently does not support clustering. Therefore, this example would not work with Tomcat.

Note: Although these instructions apply to Apache HTTP Server, you can use any server that is compliant with HTTP 1.1 to deliver static content. For general information about supported third-party software, see **`http://support.sas.com/documentation/configuration/thirdpartysupport/index.html`** △

Main Steps for Setting Up Apache to Serve Static Content

Follow these steps in order to deploy static content and to configure Apache to handle that content. These steps assume that you have already installed and configured the SAS middle tier:

1. Verify that the SAS middle-tier components operate properly in your J2EE application server or your servlet container. One way to verify proper operation is to start SAS Web Report Studio or SAS Information Delivery Portal (preferably both, if both are installed) and log on. For instructions on starting either Web application, see the **`instructions.html`** document that you used during configuration of the Web application.
2. If you have not already done so, install Apache HTTP Server and verify that it runs properly. For more information, see the *SAS Intelligence Platform: Installation Guide*.
3. Install and configure the proxy plug-in that enables your J2EE application server to interact with Apache. See the documentation that is provided for BEA WebLogic Server, IBM WebSphere Application Server, or Apache Tomcat. For

general information and examples, see "Using a Proxy Plug-in Between the J2EE Application Server and the HTTP Server" on page 445.

4 Optionally, you can configure a cache timeout value for the static content in order to reduce the requests that the browser sends to the HTTP server to check for updated content. For more information, see "Configuring Apache Cache Control for Static Content" on page 451.

5 Configure a cluster of J2EE application servers or servlet containers (even if the cluster contains only one node). You must set up the cluster in order to route browser requests to Apache instead of to your J2EE application server. In effect, this configuration hides the J2EE application server from public access, since all public requests will target Apache. For instructions on configuring a cluster, see "Configuring a Cluster of J2EE Application Servers" on page 439.

6 Stop and restart Apache HTTP Server.

7 Before you deploy static content, verify proper interaction between Apache HTTP Server and the SAS Web applications (SAS Web Report Studio or SAS Information Delivery Portal) that reside in the J2EE application server. For example, use the URL address and port of Apache HTTP Server to access SAS Web Report Studio. (Conversely, you should *not* be able to access SAS Web Report Studio using the URL of the J2EE application server in which SAS Web Report Studio is deployed.)

8 Deploy the static content to Apache HTTP Server as follows:

- If you are deploying content for SAS Web Report Studio or SAS Web Report Viewer, then see "Deploy Static Content for SAS Web Report Studio and SAS Web Report Viewer" on page 442.
- If you are deploying content for SAS Information Delivery Portal, then see "Deploy Static Content for SAS Information Delivery Portal" on page 443.
- If you are deploying content for SAS Web OLAP Viewer for Java, then see "Deploy Static Content for SAS Web OLAP Viewer for Java" on page 445.

9 Stop and restart Apache HTTP Server.

10 If you deployed content for SAS Information Delivery Portal, then the configuration changes that you made in a previous step require that you stop and restart the SAS Services application. For instructions, see the **`instructions.html`** document that you used during configuration of the SAS Information Delivery Portal.

11 Stop and restart the J2EE application server or the servlet container.

Deploy Static Content for SAS Web Report Studio and SAS Web Report Viewer

SAS Web Report Studio and SAS Web Report Viewer use static content for images, styles, themes, and scripts.

Note: The deployment that is described here is only one of several steps that are required to set up Apache to serve static content. You must also configure the proxy plug-in that enables your J2EE server to interact with Apache. In addition, you must configure a cluster of J2EE application servers (even if the cluster contains only one node). To view all of the required steps, see "Main Steps for Setting Up Apache to Serve Static Content" on page 441. △

To deploy this static content to Apache HTTP Server, follow these instructions:

1 In the Apache **`htdocs`** directory, create a directory to hold SAS Web Report Studio contents. For example, create a directory named **`SASWebReportStudio`**.

2 Locate the following four directories that contain SAS Web Report Studio static content:

- **images**
- **scripts**
- **styles**
- **themes**

These directories reside in the **wrspackaging** directory of your SAS Web Report Studio installation. For example, on Windows the directory might be **C:\Program Files\SAS\SASWebReportStudio\3.1\wrspackaging**.

3 Copy the four directories and all their contents to the directory that you created in Apache.

4 Deploy the static content for SAS Web Report Viewer in a similar way. Create a **SASWebReportViewer** directory in Apache, and then copy the four same directories from the SAS Web Report Viewer installation.

Apache reads these files in the location that you created. For example, if the URL for SAS Web Report Studio is **http://www.yourdomain.com/SASWebReportStudio/**, then the static content for all images is served from **http://www.yourdomain.com/SASWebReportStudio/images/**.

Note: After you deploy static content to Apache, you must be sure to redeploy this content if you later install a SAS Web Report Studio or SAS Web Report Viewer upgrade that includes new files for the static content. △

Deploy Static Content for SAS Information Delivery Portal

SAS Information Delivery Portal uses static content for images, styles, themes, and scripts.

Note: The deployment that is described here is only one of several steps that are required to set up Apache to serve static content. You must also configure the proxy plug-in that enables your J2EE server to interact with Apache. In addition, you must configure a cluster of J2EE application servers (even if the cluster contains only one node). To view all of the required steps, see "Main Steps for Setting Up Apache to Serve Static Content" on page 441. △

To deploy static content to Apache HTTP Server, follow these instructions:

1 In the Apache **htdocs** directory, create the following directories (use the spelling and capitalization shown below):

- **SASTheme_default**
- **SASTheme_winter**
- **sasweb\graph**
- **Portal\images**
- **Portal\scripts**
- **Portal\styles**
- **Portal\themes**
- **SASStoredProcess\images**
- **SASStoredProcess\scripts**
- **SASStoredProcess\themes**

2 Locate the **Portal** directory in your portal installation directory. For example, on Windows the default portal installation directory is **C:\Program Files\SAS\Web\Portal2.0.1**.

Copy all subdirectory files as follows:

- Copy the contents of **Portal\images** to the **htdocs\Portal\images** directory that you created.
- Copy the contents of **Portal\scripts** to the **htdocs\Portal\scripts** directory that you created.
- Copy the contents of **Portal\styles** to the **htdocs\Portal\styles** directory that you created.
- Copy the contents of **Portal\themes** to the **htdocs\Portal\themes** directory that you created.

3 Locate the **SASStoredProcess** directory in your portal installation directory.

Copy all subdirectory files as follows:

- Copy the contents of **SASStoredProcess\images** to the **htdocs\SASStoredProcess\images** directory that you created.
- Copy the contents of **SASStoredProcess\scripts** to the **htdocs\SASStoredProcess\scripts** directory that you created.
- Copy the contents of **SASStoredProcess\themes** to the **htdocs\SASStoredProcess\themes** directory that you created.

4 Copy the contents of the **sasweb\graph** directory, located in the portal installation directory, into the **htdocs\sasweb\graph** directory that you created.

5 Modify and run **UpdatePreferencesConnection.sas** in order to update the URL for the SAS Preferences Web application. The URL should reference Apache HTTP Server. (Although the SAS Preferences Web application has no static content to copy to Apache, we recommend that you update this URL setting. This URL is surfaced to clients so that they can customize portal preferences. It is preferable for this URL to reference the Apache server rather than your back-end J2EE application server.)

For instructions, see "Redistributing the SAS Preferences Web Application" in the *SAS Web Infrastructure Kit: Administrator's Guide* at **http://support.sas.com/rnd/itech/doc9/portal_admin/deploy/ag_distsprfapp.html**.

6 Deploy the static theme contents as follows:

a Unpack the **SASTheme_default.war** file, located in the portal installation directory, into the **htdocs\SASTheme_default** directory that you created.

Note: It is recommended that you use the Java **jar** command to unpack the **WAR** file. △

b Modify and run the appropriate SAS program so that the URL for the theme points to Apache HTTP Server:

- If you have already deployed this theme to WebLogic, WebSphere, or Tomcat, then modify and run **UpdateThemeConnection.sas**. For instructions, see "Redistributing the SAS Themes Web Application" in the *SAS Web Infrastructure Kit: Administrator's Guide* at **http://support.sas.com/rnd/itech/doc9/portal_admin/deploy/ag_diststhapp.html**.
- If you are deploying this theme for the first time, then modify and run **LoadThemeConnection.sas**. For instructions, see "Theme Deployment" in the *SAS Web Infrastructure Kit: Administrator's Guide* at **http://support.sas.com/rnd/itech/doc9/portal_admin/deploy/ag_themedeploy.html**.

c Repeat the previous two steps for the **SASTheme_winter.war** file. Unpack the file into the **htdocs\SASTheme_winter** directory that you created; then modify and run the appropriate SAS program for this theme.

Note: If you have previously deployed any additional custom themes that your organization has developed, repeat the above steps for those themes as well. △

d Optionally, remove the themes from the servlet container. To remove the themes, manually delete the theme files (Tomcat) or use the administrator console (WebLogic and WebSphere) to remove the themes. One reason to remove the themes is to avoid possible confusion if you upgrade to a new release or service pack. If the themes reside on both the servlet container and the HTTP server, then it's possible that you might accidentally update the themes in the wrong location.

Note: After you deploy static content to Apache, you must be sure to redeploy this content if you later install a SAS portal upgrade that includes new files for the static content. △

Deploy Static Content for SAS Web OLAP Viewer for Java

SAS Web OLAP Viewer for Java uses static content for images, styles, scripts, and templates.

Note: The deployment that is described here is only one of several steps that are required to set up Apache to serve static content. You must also configure the proxy plug-in that enables your J2EE server to interact with Apache. In addition, you must configure a cluster of J2EE application servers (even if the cluster contains only one node). To view all of the required steps, see "Main Steps for Setting Up Apache to Serve Static Content" on page 441. △

To deploy this static content to Apache HTTP Server, follow these instructions:

1. In the Apache **`htdocs`** directory, create a directory to hold SAS Web OLAP Viewer for Java contents. For example, create a directory named **`SASWebOLAPViewer`**.
2. Locate the following directories that contain SAS Web OLAP Viewer for Java static content:
   - **`images`**
   - **`scripts`**
   - **`styles`**
   - **`templates`**

   These directories reside in the **`SASWebOLAPViewer`** directory of your SAS Web OLAP Viewer for Java installation.
3. Copy the directories and all their contents to the directory that you created in Apache.

After you deploy static content to Apache, you must be sure to redeploy this content if you later install a SAS Web OLAP Viewer for Java upgrade that includes new files for the static content.

Using a Proxy Plug-in Between the J2EE Application Server and the HTTP Server

BEA WebLogic Server, IBM WebSphere Application Server, and Apache Tomcat servlet container provide plug-in modules that enable integration with an HTTP server, such as Apache HTTP Server or Microsoft Internet Information Services (IIS).

The plug-ins are useful for either or both of the following:

- To forward requests for dynamic content to the J2EE application server or servlet container. In this scenario, the HTTP server handles all the static content and relies on the J2EE application server for dynamic content.
- To forward requests and distribute those requests among a cluster of J2EE application servers using a load-balancing algorithm. For this functionality, we recommend that you use WebLogic or WebSphere rather than Tomcat.

The plug-in enables the HTTP server to behave as a proxy server, which typically filters requests and passes requests that meet the filter requirements to the J2EE application server. Many proxy servers use a local cache of Web pages to respond to requests.

This chapter does not contain the instructions for configuring the plug-ins. The configuration instructions vary greatly depending on your particular architecture. For the best information, see the vendor's documentation. The following table can help you find that documentation:

Table 19.6 Vendor Documentation for the Proxy Plug-In

Product	Plug-In Configuration File	Location of the Documentation
BEA WebLogic Server	**mod_wl**	**http://e-docs.bea.com/wls/docs81/plugins/overview.html**
IBM WebSphere Application Server	**plugin-cfg**	**http://publib.boulder.ibm.com/infocenter/ws51help/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/twsv_plugin.html**
Apache Tomcat	**mod_jk**	**http://jakarta.apache.org/tomcat/tomcat-4.1-doc/config/jk.html**

CAUTION:
When using WebLogic with an Apache HTTP server, you must explicitly enable the pooling of connections between the WebLogic server and the Apache plug-in. You enable pooling by setting the **KeepAliveEnabled** parameter to *ON* in the Apache plug-in configuration. By default, that parameter is set to *OFF*. (The vendor documentation is misleading about this.) If **KeepAliveEnabled** is set to *OFF*, then you can experience session failures under heavy load conditions. △

The following sections provide two sample configurations for Apache HTTP Server. Both configurations assume that all static content has been copied to the appropriate subdirectory of the Apache **htdocs** directory. See "Configuring Apache HTTP Server to Serve Static Content for SAS Web Applications" on page 440.

Sample: Proxy Setup for SAS Web Report Studio

This sample shows the configuration for an Apache HTTP Server that proxies SAS Web Report Studio static content to a WebLogic cluster.

```
WebLogicCluster host1.yourdomain.com:7101,host2.yourdomain.com:7101
WLLogFile c:\apps\Apache2\logs\proxy_debug.log
```

```
Debug OFF

# Instruct Apache to forward WRS requests to BEA
<Location /SASWebReportStudio>
   CookieName sas.wrs.sessionID
   SetHandler weblogic-handler
</Location>

# Override static content (images)
<Location /SASWebReportStudio/images>
   SetHandler default-handler
</Location>

# Override static content (styles)
<Location /SASWebReportStudio/styles>
   SetHandler default-handler
</Location>

# Override static content (scripts)
<Location /SASWebReportStudio/scripts>
  SetHandler default-handler
</Location>

# Override static content (themes)
<Location /SASWebReportStudio/themes>
  SetHandler default-handler
</Location>
```

Sample: Proxy Setup for SAS Web OLAP Viewer for Java

This sample shows the configuration for an Apache HTTP Server that proxies SAS Web OLAP Viewer for Java static content to a WebLogic cluster.

```
#Configuremod_wl
# http://e-docs.bea.com/wls/docs81/plugins/apache.html
WebLogicCluster host1.yourdomain.com
WLLogicPort 7101
Debug OFF

# Instruct Apache to forward requests to BEA
<Location /SASWebOLAPViewer>
   CookieName sas.swov.sessionID
   SetHandler weblogic-handler
</Location>

# Override static content (images)
<Location /SASWebOLAPViewer/images>
   SetHandler default-handler
</Location>

# Override static content (styles)
<Location /SASWebOLAPViewer/styles>
```

```
   SetHandler default-handler
</Location>

# Override static content (scripts)
<Location /SASWebOLAPViewer/scripts>
   SetHandler default-handler
</Location>

# Override static content (templates)
<Location /SASWebOLAPViewer/templates>
   SetHandler default-handler
</Location>
```

Sample: Proxy Setup for the Portal and Related Web Applications

This sample shows the configuration for an Apache HTTP Server that proxies static content to a WebLogic cluster. This sample includes static content for SAS Web Report Studio, SAS Web Report Viewer, and SAS Information Delivery Portal.

```
# Configure mod_wl
#   http://e-docs.bea.com/wls/docs81/plugins/apache.html

LoadModule weblogic_module "C:/bea/weblogic81/server/bin/mod_wl_20.so"

WebLogicCluster host1.yourdomain.com:7101,host2.yourdomain.com:7101
WLLogFile c:\apps\Apache2\logs\proxy_debug.log
Debug OFF

# Instruct Apache to forward Portal requests to WebLogic
<Location /Portal>
  CookieName sas.portal.sessionid
  SetHandler weblogic-handler
</Location>

# The following entries override the location of static
#  content to Apache. Enable them only if you have manually
#  copied the static content to the appropriate
#  subdirectories in Apache.

<Location /Portal/images>
   SetHandler default-handler
</Location>

<Location /Portal/scripts>
  SetHandler default-handler
</Location>

<Location /Portal/styles>
   SetHandler default-handler
</Location>

<Location /Portal/themes>
```

```
  SetHandler default-handler
</Location>

# Instruct Apache to forward Stored Process requests to WebLogic
<Location /SASStoredProcess>
  CookieName sas.stp.sessionid
  SetHandler weblogic-handler
</Location>

# The following entries override the location of static
#  content to Apache. Enable them only if you have manually
#  copied the static content to the appropriate
#  subdirectories in Apache.

<Location /SASStoredProcess/images>
   SetHandler default-handler
</Location>

<Location /SASStoredProcess/scripts>
  SetHandler default-handler
</Location>

<Location /SASStoredProcess/themes>
  SetHandler default-handler
</Location>

# Instruct Apache to forward Preferences requests to WebLogic
<Location /SASPreferences>
  CookieName sas.preferences.webapp.sessionid
  SetHandler weblogic-handler
</Location>

# Instruct Apache to forward SASDoc requests to WebLogic
<Location /SASDoc>
  SetHandler weblogic-handler
</Location>

# By default, Apache will use the default-handler for
# content manually copied to the appropriate
# subdirectories of the Apache htdocs directory.
# The contents of sasweb, SASTheme_default, and SASTheme_winter
# should be unpacked and copied into these subdirectories.
# If instead, they are deployed as a WAR file into
# WebLogic, a Location entry must be added using the
# weblogic-handler.

# Instruct Apache to forward SAS Web Report Viewer
# requests to WebLogic
<Location /SASWebReportViewer>
  CookieName sas.wrv.sessionID
  SetHandler weblogic-handler
</Location>

# The following entries override the location of static
```

```
#  content to Apache. Enable them only if you have manually
#  copied the static content to the appropriate
#  subdirectories in Apache.

<Location /SASWebReportViewer/images>
    SetHandler default-handler
</Location>

<Location /SASWebReportViewer/styles>
   SetHandler default-handler
</Location>

<Location /SASWebReportViewer/scripts>
  SetHandler default-handler
</Location>

<Location /SASWebReportViewer/themes>
  SetHandler default-handler
</Location>

# Instruct Apache to forward SAS Web Report Studio
# requests to WebLogic
<Location /SASWebReportStudio>
  CookieName sas.wrs.sessionID
  SetHandler weblogic-handler
</Location>

# The following entries override the location of static
#  content to Apache. Enable them only if you have manually
#  copied the static content to the appropriate
#  subdirectories in Apache.

<Location /SASWebReportStudio/images>
   SetHandler default-handler
</Location>

<Location /SASWebReportStudio/styles>
   SetHandler default-handler
</Location>

<Location /SASWebReportStudio/scripts>
  SetHandler default-handler
</Location>

<Location /SASWebReportStudio/themes>
  SetHandler default-handler
</Location>

# Instruct Apache to forward SAS Web OLAP Viewer
# requests to WebLogic
<Location /SASWebOLAPViewer>
  CookieName sas.swovj.sessionID
  SetHandler weblogic-handler
</Location>
```

```
# The following entries override the location of static
#  content to Apache. Enable them only if you have manually
#  copied the static content to the appropriate
#  subdirectories in Apache.

<Location /SASWebOLAPViewer/images>
   SetHandler default-handler
</Location>

<Location /SASWebOLAPViewer/styles>
   SetHandler default-handler
</Location>

<Location /SASWebOLAPViewer/scripts>
  SetHandler default-handler
</Location>

<Location /SASWebOLAPViewer/templates>
  SetHandler default-handler
</Location>
```

Configuring Apache Cache Control for Static Content

To avoid sending unnecessary requests to the server each time a client requests a static content item, you can configure Apache HTTP Server to set cache timeout values for static content.

Typically, after a browser initially downloads a static resource from the HTTP server, the browser sends a conditional HTTP GET request each time the browser encounters that resource again. For example, when a browser first downloads a SAS Web Report Studio logo image, the browser stores a local copy of the image. For each subsequent page that references the logo, the browser requests that the image be resent if the image has been modified since the previous download. This sequence occurs for every static element and can result in large numbers of HTTP requests. Because the static content for SAS Web Report Studio, SAS Web Report Viewer, and SAS Information Delivery Portal is not modified often, most of these requests are unnecessary.

When you specify a cache timeout for each static element, clients (browser, proxy, or server cache) can avoid sending unnecessary requests to the HTTP server in order to check the validity of the content. When the browser first accesses a static element, the browser stores that element locally for the duration of the timeout value that is configured. During this time, subsequent queries to the HTTP server are suppressed for that element. The browser resumes queries as appropriate when the timeout period elapses within the session.

You can configure Apache HTTP Server to set cache timeout values for static content regardless of whether Apache HTTP Server is configured to serve that static content or is merely a front-end proxy to your J2EE application server.

Note: Although these instructions apply to Apache HTTP Server, you can use any server that is compliant with HTTP 1.1 and that supports timeout values for static content. △

The configuration instructions can vary greatly depending on your particular architecture. In general, you must complete these steps:

1. Create a configuration file for the Apache HTTP Server, or edit an existing configuration file. If you configured Apache HTTP Server to serve static content, then you can add the cache timeout settings to that configuration file.
2. In the configuration file, provide directives to load the cache timeout module:

   LoadModule expires_module modules/mod_expires.so

   LoadModule headers_module modules/mod_headers.so
3. Specify the timeout value for each static content location.

The following example shows the configuration for an Apache HTTP Server that serves SAS Web Report Studio static content with a cache timeout value and proxies traffic to a WebLogic cluster. This example is based on the example shown previously (see "Using a Proxy Plug-in Between the J2EE Application Server and the HTTP Server" on page 445). The new statements for timeout values are shown with a highlighted background.

(If Apache HTTP Server does not serve the static content, then your configuration file would be similar, but would exclude the **SetHandler** statements in the **# Override static content** blocks.)

```
# The following two entries must be uncommented to enable
# cache timeout headers to be sent by Apache
LoadModule expires_module modules/mod_expires.so
LoadModule headers_module modules/mod_headers.so


# The following section enables the expiration directives
<IfModule mod_expires.c>
ExpiresActive On
</IfModule>


WebLogicCluster host1.yourdomain.com:7101,host2.yourdomain.com:7101
WLLogFile c:\apps\Apache2\logs\proxy_debug.log
Debug OFF

# Instruct Apache to forward WRS requests to BEA
<Location /SASWebReportStudio>
   CookieName sas.wrs.sessionID
   SetHandler weblogic-handler
 </Location>

# Override static content (images)
<Location /SASWebReportStudio/images>
   SetHandler default-handler
   # set static timeout
   ExpiresDefault ''access plus 60 minutes''
</Location>

# Override static content (styles)
<Location /SASWebReportStudio/styles>
   SetHandler default-handler
   # set static timeout
   ExpiresDefault ''access plus 60 minutes''
```

```
</Location>

# Override static content (scripts)
<Location /SASWebReportStudio/scripts>
  SetHandler default-handler
  # set static timeout
  ExpiresDefault ''access plus 60 minutes''
</Location>

# Override static content (themes)
<Location /SASWebReportStudio/themes>
  SetHandler default-handler
  # set static timeout
  ExpiresDefault ''access plus 60 minutes''
</Location>
```

Redeploying Middle-Tier Applications

After initial installation and configuration, if you make changes to your middle-tier configuration, then you might be instructed to redeploy the middle-tier applications. For example, when you update passwords, you are instructed to redeploy all middle-tier applications that you have installed. All of the procedures in the documentation explicitly state when you must redeploy the middle-tier applications.

Each middle-tier application has its own deployment instructions. The following table indicates where to find complete instructions.

Note: Not all deployments include all the applications listed in the table. Redeploy only the applications that you have installed. △

Table 19.7 Deployment Instructions for Middle-Tier Applications

Application	Location of the Deployment Instructions
SAS Web Report Studio	**<SAS-install-dir>\SASWebReportStudio\3.1\deployment.html**
SAS Web Report Viewer	**<SAS-install-dir>\SASWebReportViewer\3.1\deployment.html**
SAS Information Delivery Portal	**<SAS-install-dir>\Web\Portal2.0.1\wik_readme.html** Some configuration changes require you to re-import the foundation services that the portal uses. For details, see "Redeploy the Portal Web Application" in the Web Infrastructure Kit: Administrator's Guide at **http://support.sas.com/rnd/itech/doc9/portal_admin/admintools/ag_portalconintro.html**
SAS Web OLAP Viewer for Java	**<SAS-install-dir>\SASWebOlapViewerforJava\3.1\config.pdf**
SAS BI Web Services for Java	**<SAS-install-dir>\Web\WebServicesforJava\1.0\xmla_readme.html**

CHAPTER

20

SAS Scheduling

Overview of SAS Scheduling

The Schedule Manager plug-in to SAS Management Console allows the user to schedule SAS jobs by using scheduling servers. In Schedule Manager, the user creates and schedules groups of jobs, called flows. Each job within a flow can be triggered to run based on a certain time, the state of a file on the file system, or the status of another job within the flow.

This chapter provides information on the following topics:

- the architecture used for scheduling jobs from SAS applications
- batch scheduling for DATA step and Java jobs
- setup tasks that must be performed by users of SAS Web Report Studio before they can use all of that product's report-scheduling capabilities

  Note: SAS Web Report Studio has a scheduling wizard that does not require users to work with jobs or flows. The SAS Web Report Studio built-in scheduling functionality is described in "Scheduling and Distributing Pre-generated Reports" on page 338. Regardless of the method that you use to schedule reports, you must perform the setup tasks that are described here. △
- specific scheduling-related issues for SAS Data Integration Studio, SAS Marketing Automation, and SAS Web Report Studio
- operating system issues specific to Windows and AIX
- an exit status issue with MVA SAS and scheduling servers from Platform Computing

Scheduling Architecture

The architecture used for scheduling jobs from SAS applications consists of a set of tools from SAS and Platform and a variety of servers. These scheduling components are used to process jobs deployed from SAS applications, such as SAS Data Integration Studio.

Tools

Schedule Manager

The Schedule Manager plug-in to SAS Management Console is used to schedule SAS jobs. Jobs created in applications that include SAS Data Integration Studio, SAS Marketing Automation, and SAS Web Report Studio can be scheduled with Schedule Manager. These jobs are processed as flows, which are sets of scheduled jobs. Although Schedule Manager is designed as a portable scheduling interface that provides SAS solutions with a scheduler-neutral interface, it functions as another Platform scheduling server client.

Platform Load Sharing Facility

Platform LSF (Load Sharing Facility) is a product from Platform Computing that handles the load balancing and execution of batch commands (which represent jobs). Platform LSF manages and accelerates batch workload processing for computer- and data-intensive applications. Platform scheduling servers use Platform LSF for the execution of the jobs within a flow.

Platform Scheduling Servers

Platform scheduling servers are used to schedule groups of jobs and any associated dependencies. Platform scheduling servers contain a Java API that allows for seamless integration with the Schedule Manager plug-in to SAS Management Console. Schedule Manager sends requests to Platform scheduling servers, which in turn send requests to Platform LSF. Platform scheduling servers allow a user to create flows and interdependencies between jobs.

Platform Calendar Editor

Platform Calendar Editor is a scheduling client for a Platform scheduling server. Platform Calendar Editor allows you to create new calendar entries to use for time dependencies in your jobs scheduled on Platform scheduling servers. You can use it to create custom versions of the calendars that are used to create time dependencies for jobs.

Platform Flow Manager

Platform Flow Manager provides a visual representation of scheduled flows and associated dependencies in Platform scheduling servers. It also allows you to view the status and information about the flows and jobs scheduled to Platform scheduling servers. You can run or rerun a single job directly from Platform Flow Manager, regardless of whether the job failed or completed successfully.

You can use Platform Flow Manager to view or update the status of flows that have been scheduled from Schedule Manager in SAS Management Console and submitted to a Platform scheduling server. You can also use Platform Flow Manager to view or update the status of jobs within a scheduled flow. To use Platform Flow Manager, the following conditions must exist:

- Platform Flow Manager must have been installed
- flows must have been scheduled

OS Services Scheduling Server

The OS Services Scheduling Server provides integration with operating system services such as the Windows AT command. These operating system scheduling servers use operating system commands to execute programs when a specified triggering condition occurs, such as a date and time. Before you add the metadata for an operating system scheduling server, identify the following:

- the network name of the server
- the port number for the server
- the directory on the server where scheduled flows should be stored (control directory)

- the command to start a SAS session on the server

Third-Party Schedulers

When using a SAS application that deploys a job for scheduling with Platform scheduling servers, the job is created in the SAS application, deployed to SAS Management Console, and scheduled using Schedule Manager. You can then use other schedulers to schedule the deployed jobs using a SAS batch command to run. Future releases will provide support for expressing flows in operating system shell scripts (such as PC batch, Korn Shell, and JCL) that can be more readily scheduled.

Servers

In the SAS scheduling architecture, you need to define a scheduler server and a batch server. You need to have a scheduler server and either at least one job made ready for scheduling or a SAS Data Step Batch server defined to use to deploy a SAS job for scheduling.

A batch server is really nothing more than a way to associate a batch command to a machine. Various applications that integrate with scheduling provide a batch command that represents their jobs when executed. For example, SAS Data Integration Studio uses the sasbatch command for its jobs.

Scheduler Server

A scheduler server is a server that executes programs when a specified trigger occurs, such as a date and time. The Platform scheduling server is an example. Typically, a server is installed and started before you specify the metadata for it. For details about installing a server, see the documentation that came with the server. See also the Administration and Deployment Guide for the SAS Intelligence Architecture. Before you add the metadata for a scheduler server, identify the following:

- the network name of the server
- the port number for the server

SAS Workspace Server

The SAS Workspace Server executes SAS code and accesses data. The SAS Workspace Server is a SAS integrated-OM server that is launched to fulfill client requests for IOM workspaces. A SAS Workspace Server is a server component that can be specified in the metadata for a SAS application server. For example, a SAS Workspace Server is one component in the metadata for the server that is used to deploy jobs for scheduling.

SAS Data Step Batch Server

The SAS Data Step Batch Server is designed to run traditional SAS programs (DATA step and procedure statements). A SAS batch server is a metadata object that includes the following:

- the network address of a SAS batch server
- a SAS start command that will run jobs in batch mode on the SAS Batch Server

You must identify a SAS start command that will run jobs in batch mode on the SAS Data Step batch server. Here is an example of an appropriate SAS command:

```
C:\Program Files\SAS\SAS 9.1\sas
```

SAS Java Batch Server

The SAS Java Batch Server is designed to run Java applications that are supplied by SAS and invoked and launched by the standard EXE shell.

SAS Generic Batch Server

The SAS Generic Batch Server is designed to run unknown stand-alone commands/ executables that are supplied by SAS. It is provided as a stop-gap measure for scheduling commands that were identified too late in the development cycle to be accommodated directly in a given release.

Architecture Summary

The table that follows summarizes the relationships between the components and servers used in SAS scheduling:

Table 20.1 Architecture Component and Server Summary

Component	Source	Works With	Purpose
Schedule Manager	SAS (a component of SAS Management Console)	Platform Computing scheduling servers, third- party schedulers, SAS applications that support scheduling	Schedules SAS jobs and manages job flows
LSF	Platform Computing	Scheduling servers	Handles load balancing and execution of batch commands
Scheduling Servers	Platform Computing	LSF	Schedule groups of jobs and dependencies
Calendar Editor	Platform Computing	Platform Computing scheduling servers	Creates new calendar entries for time dependencies in jobs
Flow Manager	Platform Computing	Platform Computing scheduling servers	Provides a view of scheduled flows and dependencies; also permits review of status and additional information about flows and scheduled jobs
OS Services Scheduling Server	SAS	Windows AT command and Unix cron command	Executes operating system commands to run schedules (via AT and cron commands)
Third-Party Schedulers	Various third parties	SAS batch command	Schedule deployed jobs using a SAS batch command
LSF Scheduler Server	SAS	Platform Computing scheduling servers	Executes programs when a specified trigger occurs, such as a date and time

Component	Source	Works With	Purpose
Workspace Server	SAS	SAS application server	Executes SAS code and accesses data to fulfill client requests for IOM workspaces
Data Step Batch Server	SAS	-	Schedules traditional SAS programs (DATA step and procedure statements)
Java Batch Server	SAS	-	Schedules Java applications supplied by SAS
Generic Batch Server	SAS	-	Schedules unknown stand-alone commands or executables supplied by SAS

Batch Scheduling

Schedule DATA Step Jobs

To use Schedule Manager to schedule SAS DATA step jobs, as used by SAS Data Integration Studio, the jobs must be deployed to the scheduling server. In order to deploy and schedule these jobs, two server definitions need to be added to your metadata server, and a deployment directory must be created, as follows:

1. In Server Manager in SAS Management Console, define a new scheduling server. Choose the host and port that you have installed the scheduling server on.
2. Also in Server Manager, add a Data Step Batch Server component to your existing application server. To do this, right-click your application server, and choose **`Add Server Component`**.
3. Define the full path of your MVA SAS (the location of the sasbatch command) installation on the machine your jobs will be scheduled on.

When you have completed these steps, you are ready to deploy SAS DATA step jobs from supporting application, such as SAS Data Integration Studio. To deploy jobs for scheduling in SAS Data Integration Studio, complete these steps:

1. Right-click a job in SAS Data Integration Studio and choose the **`Deploy for Scheduling`**. This brings up a dialog box that allows you to choose a SAS Server, where you want your jobs to be stored, and the filename of the job.
2. Click **`OK`** in this dialog box. Then, SAS Data Integration Studio generates SAS code from the job metadata, transfers the file to the scheduling server using the SAS Workpace Server defined in the SAS server that you have selected, and creates a piece of metadata for the Schedule Manager to use.

To schedule this job as part of a flow in the Schedule Manager, complete the following steps:

1. Right-click the schedule manager and create a new flow. Make sure that your flow name does not have spaces in it.

2 Choose the job that you have deployed to include in this flow and press **OK**. The flow displays in the navigation tree. If you expand the flow, you can see the jobs that are now a part of this flow.
3 Right-click the job and define dependencies for these jobs. Each job can have combinations of time, job, and file dependencies. When defining time dependencies, you will be prompted to log into the scheduling server in order to retrieve the default calendars that have been defined.
4 Specify the user ID and password for the machine on which the scheduling server is deployed. This actually checks your credentials with the scheduling server.
5 Schedule the flow to run after you have defined dependencies for each job. To do this, right-click the flow and choose **schedule flow**. You will be prompted to schedule this flow to run immediately, manually, or based on a dependency.
6 If you would like to just run this job once, choose to run the job immediately.
7 If you want to run this job on a recurring basis, then choose one of the dependencies that you have defined in the job as the trigger for the flow. For instance, if you have scheduled a job in your flow to run Mondays at 3:30, then you can schedule your entire flow to be kicked off at this time. After you have scheduled your flow, you can now view the flow in the Platform Flow Manager.

Schedule Batch Java Jobs

A batch Java job is the execution of a Java application that is deployed by SAS. These Java applications have an EXE shell and INI file that launch the Java application with the proper classpath and option settings.

To schedule batch Java jobs in Schedule Manager, the jobs must be deployed to the scheduling server. In order to deploy and schedule these jobs, two server definitions need to be added to your metadata server. Complete the following steps to add the definitions:

1 In Server Manager in SAS Management Console, define a new scheduling server. Choose the host and port that you have installed the scheduling server on.
2 Also in Server Manager, add a Java Batch Server component to your existing application server. To do this, right-click your application server, and choose **Add Server Component**.
3 Define the full path of your EXE shell for your Java application installation on the machine that your jobs will be scheduled on.
4 Select the subtype of Java Batch Server from the drop-down list. The subtype is usually named after the participating application which you will use to deploy the job.

When you have completed these steps, you are ready to deploy jobs from applications such as Marketing Automation. Each application provides a user interface for deploying jobs.

To schedule this job as part of a flow in Schedule Manager, follow these steps:

1 Right-click the Schedule Manager and create a new flow. Make sure that your flow name does not have spaces in it.
2 Choose the job that you have deployed to include in this flow and press **OK**. The flow shows up in the navigation tree. If you expand the flow, you can see the jobs that are now a part of this flow.

3 Right-click the job and define dependencies for these jobs. Each job can have combinations of time, job, and file dependencies. When defining time dependencies, you will be prompted to log into the scheduling server in order to retrieve the default calendars that have been defined.
4 Specify user ID and password for the machine where the scheduling server is deployed. This actually checks your credentials with the scheduling server.
5 Schedule the flow to run after you have defined dependencies for each job. To do this right-click the flow and choose **`Schedule Flow`**. You will be prompted to schedule this flow to run immediately, manually, or based off a dependency.
6 If you would like to just run this job once, choose to run the job immediately.
7 If you want to run this job on a recurring basis, then choose one of the dependencies that you have defined in the job as the trigger for the flow. For instance, if you have scheduled a job in your flow to run Mondays at 3:30, then you can schedule your entire flow to be kicked off at this time. After you have scheduled your flow, you can now view the flow in the Platform Flow Manager.

Enabling the Scheduling of Reports

Before the users of SAS Web Report Studio can use all of that product's report-scheduling capabilities, you must perform some setup tasks. This section covers those tasks in detail. These tasks are specific to Platform LSF.

Note: You must also have purchased Platform scheduling servers for SAS. The scheduling software is sold separately from the technology bundles that include SAS Web Report Studio. △

Before we look at these post-installation steps, however, let's review some related tasks that should already have been performed at this point. First, someone should have created four operating-system user IDs during pre-installation. These users and the roles that they play in report scheduling are shown in the following table:

Table 20.2 User IDs Needed for Report Scheduling

User ID	Purpose
lsfadmin (suggested name)	Serves as the primary administrator for Platform scheduling server. On Windows systems, this user owns the scheduling services. This user does not need to be added to the metadata repository.
lsfuser (suggested name)	Used to run an Output Generation Tool. This user does not need to be added to the metadata repository.
sastrust	Used by the Output Generation Tool to obtain user credentials from the metadata server.
saswbadm	Used by SAS Web Report Studio to perform the nightly cleanup of partially deleted jobs.

In addition, the person who installed Platform scheduling server should have specified that the Platform scheduling server and Platform LSF services should run as lsfadmin.

Finally, any target location for report output must be accessible to the scheduling software and to SAS Web Report Studio users. For example, suppose that you create a

channel archive for report output. Both SAS Web Report Studio and the scheduling software must be able to write to the channel archive. If the scheduling software and SAS Web Report Studio reside on different machines, then the archive should be created on a network location, and the channel archive file path should be defined for that network location.

Configure SAS Web Report Studio

In most regards, SAS Web Report Studio has been configured appropriately at this point. However, before you can use the product's report-scheduling feature, you must enter the credentials of lsfuser in a SAS Web Report Studio configuration file. You had the option to provide these credentials during installation. You can also provide the credentials after installation by following the instructions that are presented here.

To enter these credentials in the file, start SAS Management Console on the middle-tier server, connect to the metadata server using the credentials of the user who installed the software on this host, and perform the following steps:

1. In the SAS Management Console navigation tree, expand the **Application Management** node and, under that, the **Report Studio Configuration** node. You see a **Web Report Studio** node.
2. Right-click the **Web Report Studio** node, and select **Open Configuration File** from the pop-up menu. An Open Configuration File dialog box appears.
3. Use this dialog box to open the file **wrs.config**, which is located in the installation directory for SAS Web Report Studio. (On Windows systems, by default, this file resides in the directory **C:\Program Files\SAS\SASWebReportStudio\3.1**.) You see a set of values in the right-hand pane of the GUI.
4. Right-click the value in the column labeled **Host** (on the far left), and select **Modify Server Definition** from the pop-up menu. A Modify Server Definition dialog box appears.
5. Enter the user ID **lsfuser** in the **Scheduling Server User ID** text box. On Windows systems, this ID must be qualified with a domain or machine name.
6. Enter the password for lsfuser in the **Scheduling Server Password** text box. The password can be in clear text. However, for security reasons, you should enter a version of the password that you have encoded using SAS. To obtain this encoded password, use PROC PWENCODE.
7. Click **OK** in the Modify Server Definition dialog box.
8. Back in the main window of SAS Management Console, right-click the **Web Report Studio** node in the SAS Management Console navigation tree, and select **Save Configuration File** from the pop-up menu. A Save Configuration File dialog box appears.
9. Use this file browser to navigate to the file **wrs.config**. (This file name should appear in the **File name** box.) Then click **Save**.

After you edit this configuration file, you must reconfigure and redeploy SAS Web Report Studio. For instructions, see "Redeploy SAS Web Report Studio" on page 337. See also the document **deployment.html**, which resides in your SAS Web Report Studio installation directory.

Configure the Output Generation Tool

SAS Web Report Studio relies on an Output Generation Tool to create a set of reports in batch mode. To configure this tool, you must do the following:

- verify an output-generation configuration file
- edit the output-generation tool's initialization file

To verify the configuration file, open the file called **OutputManagementConfigTemplate.xml**, which is found in the installation directory for the SAS Query and Reporting Services (on the scheduling server). In this file, check parameters that are set during the SAS Query and Reporting Services installation. The parameters should provide the user ID and password for the SAS Trusted User (sastrust) using the format shown in this example:

```
<User>
   <Name>sastrust</Name>
   <Password>encodedPassword</Password>
   <Trusted>true</Trusted>
</User>
```

In the previous code, *encodedPassword* will be replaced with the actual password in encoded format.

Verify also that the mail server is correctly specified for your environment. The default value is the local host machine, as shown in this example:

```
<MailServer>
   <!-- Default: localhost -->
   <Name>mailhost</Name>
   <!-- Default: 25 -->
   <Port>25</Port>
</MailServer>
```

If the mail server is not the local host machine, then make sure the host name is specified.

In **OutputManagementConfigTemplate.xml**, edit the values of the following elements if necessary:

```
<Activated>${RENDERER_OPTIMIZER_SERVER_ENABLED}</Activated>
<Level>${RENDERER_OPTIMIZER_SERVER_LOGLEVEL}</Level>
<File>${OUTPUTGEN_DISTLIST_SERVER_LOGFILE}</File>
```

Depending on your installation, the variables might already be replaced with the correct values. If the variables haven't been replaced with the correct values, then you must replace the variables as follows:

- Replace **${RENDERER_OPTIMIZER_SERVER_ENABLED}** with **true**.
- Replace **${RENDERER_OPTIMIZER_SERVER_LOGLEVEL}** with **error**.
- Replace **${OUTPUTGEN_DISTLIST_SERVER_LOGFILE}** with the path where you would like your log files created.

  For example, on Windows:

  C:\SAS<*configuration_dir*>**\Lev1\SASMain\ReportingServices\logs\wrsdist.log**

  On UNIX, it might be:

  /users/<*user ID*>**/SAS/**<*configuration_dir*>**/Lev1/SASMain/ReportingServices/logs/wrsdist.log**

On Windows systems, the tool's initialization file, **outputgen.ini**, is also located in the installation directory for SAS Query and Reporting Services. In this file, add a logging directory at the bottom of the file. (This log is different from the one specified above.) To add this directory, complete the following steps:

1. Open the file in a text editor. Locate the line that starts with:

   ```
   ;LogDirectory=
   ```

2 Delete the semicolon at the beginning of the line (which marks the line as a comment).

3 After the equal sign, enter the path to the directory in which you want the tool to create its log file. We recommend that you specify the directory **`C:\Program Files\SAS\SASQueryandReportingServices\3.1\logs`**.

   Note: You must create this **`logs`** directory in the file system and give the lfsuser Write access to it. △

4 Verify that the **`CommandToExecute`** property is set to **`C:\j2sdk1.4.2_05\bin\java.exe`**. This is the path to the Java Runtime Environment that the Output Generation Tool will use. Be sure that the reference to the executable file includes the **`.exe`** extension. This version of the Java SDK must be installed on the system.

5 Save your changes.

On UNIX systems, the **`outputgen.ini`** is not available, so the standard out and standard error log statements are directed to the console.

Configure the Scheduling Server

On the host where the scheduling server is running, you might need to perform the following tasks:

- supply the scheduling server with the credentials for lsfuser
- give lsfuser Write access to the necessary log files

The first task is necessary only if the Platform scheduling server is running on a Windows machine and you are logging into the scheduler using a local account.

To perform the first of these tasks, complete the following steps:

1 Open a Command Prompt window.

2 Use the program **`lspasswd`** to register the credentials for the lsfuser.

   a Issue a command similar to this:

```
C:\LSF_51\bin\lspasswd.exe -u .\lsfuser
```

   The program prompts you for this user's password.

   b Enter the user's password and then confirm it. This completes the registration process. Again, this step is necessary only when you are running Platform scheduling server on Windows.

You must also give lsfuser Write access to the following directories:

- the logs directory that is specified in the **`LogDirectory`** parameter of the **`outputgen.ini`**.
- the logs directory that is specified in the**`outputgen`** log file.

Define a Scheduling Admins Group in the Metadata

This step and the next three steps involve creating required metadata objects.

First, follow these steps to create a metadata group named Scheduling Admins and to add sastrust and saswbadm to the group.

1 Start SAS Management Console, and connect to the metadata server as the SAS Administrator, sasadm. (You need to be logged into this application as this user when you perform the steps in the next two subsections as well.)

2 Right-click the **`User Manager`** plug-in icon, and select **New ▶ Group**. A New Group Properties dialog box appears.

3 Select the **General** tab, and enter **Scheduling Admins** in the **Name** box. Enter the text exactly as shown here (your text should use the same capitalization and have a single space between the words).
4 Select the **Members** tab, and move SAS Trusted User and SAS Web Administrator from the **Available Members** list to the **Current Members** list.
5 Click **OK**.

Update the Logins for Users Who Will Schedule Reports

Each user who will schedule reports must have his or her password stored in a metadata login. To do this, complete the following steps:

1 In the SAS Management Console navigation tree, select the **User Manager** plug-in. A list of users and groups appears in the right pane.
2 Right-click the user to whose login you want to add a password. From the pop-up menu that appears, select **Properties**. A Properties dialog box is displayed.
3 In the Properties dialog box, select the **Logins** tab.
4 On the **Logins** tab, select the login to which you want to add a password, and click **Modify**. An Edit Login Properties dialog box appears.
5 In the Edit Login Properties dialog box, enter the user's password in the **Password** and **Confirm Password** boxes. Then, click **OK**.
6 Click **OK** in the Properties dialog box.

Define a Java Batch Server

You also need to define a Java Batch Server as a component of your SAS application server. To do this, complete the following steps:

1 In the SAS Management Console navigation tree, expand the node for the **Server Manager** plug-in.
2 Right-click the SASMain application server, and select **Add Application Server Component** from the pop-up menu. The New SAS Application Server Component Wizard starts.
3 In the wizard's first screen, select **SAS Java Batch Server** and click **Next**.
4 In the wizard's second screen, enter the name **BRM Java Batch Server** in the **Name** box, and click **Next**.
5 In the wizard's third screen, select the name of the server on which the Platform scheduling server is installed from the **Associated Machine** drop-down list, and select **Business Report Manager** from the **SubType** list.

   In the **Command Line** box, enter a command that contains the following components:

   path-to-outputgen-executable
   The fully qualified path to the **outputgen** executable file, enclosed in quotation marks.

   –config-file *configuration-file-location*
   The fully qualified path to the output-generation configuration file, enclosed in quotation marks. (This is the file that is referenced in the topic "Configure the Output Generation Tool" on page 463.)

   –repository *metadata-repository*
   The name of the repository that contains the job definitions.

Here is an example command line on a Windows system:

```
"C:\Program Files\SAS\SASQueryandReportingServices\3.1\outputgen.exe"
 --config-file "C:\Program Files\SAS\SASQueryAndReportingServices\3.1\
OutputManagementConfigTemplate.xml" --repository Foundation
```

Here is an example command line on a UNIX system:

```
/usr/local/SAS/SASQueryandReportingServices/3.1/outputgen
--config-file OutputManagementConfigTemplate.xml --repository Foundation
```

Note: Enter the command with no line breaks, and be sure to put quotation marks around the arguments as shown in the example. △

6 In the wizard's final screen, review the information you have entered. If it is correct, click **Finish**.

Define a Java Batch Server to Run SAS Web Report Studio 2.1 and 3.1 Reports

If you are upgrading from SAS Web Report Studio 2.1 to version 3.1, but want to schedule reports for SAS Web Report Studio 2.1 and 3.1 on the same host machine, then define the Java Batch Server for version 3.1 as described in the previous section. Then, add the following lines to the SAS Web Report Studio **LocalProperties.xml** file. (For instructions on creating a **LocalProperties.xml** file, see "Create a LocalProperties.xml File" on page 325.)

```
<wrs.scheduling.batchServerComponentName>
Java Batch Server Name
</wrs.scheduling.batchServerComponentName>
<wrs.scheduling.schedulingServerComponentName>
Platform Job Scheduler
</wrs.scheduling.schedulingServerComponentName>
```

The name that you provide for the Java Batch Server should match the name that is shown for the 3.1 server in the Server Manager plug-in. By default, the name is **BRM Java Batch Server**. If the tags are not implemented, then 3.1 uses the first server definition.

To obtain a list of server names, add the following entry to the **DefaultLoggerProperties.xml** file.

```
<LoggingContext name = "com.sas.apps.citation.model.scheduler"
        priority = "INFO"
        chained  = "false">
   <OutputRef outputID = "WRS" />
  </LoggingContext>
```

The above logging context will generate numerous messages and is intended to be used temporarily only to obtain the names of Java Batch servers. After you have defined the 3.1 server, you should remove or comment out this logging context.

Channel-Specific Tasks

If anyone at your site will be publishing batch reports to a channel with an archive persistent store, you must use SAS Management Console to specify the path to a network share in the Publishing Framework plug-in by completing these steps:

1 In the SAS Management Console navigation tree, expand the **Publishing Framework** node.

2 Expand the desired metadata repository node.
3 Expand the **Channels** node.
4 Right-click the channel to which you will be publishing reports, and select **Properties** from the pop-up menu. A Properties dialog box appears.
5 In the Properties dialog box, select the **Persistent Store** tab.
6 Enter in the **Path** box the path to a network share to which all users who will be creating content have Write access. You must specify the path to a network share because a local path may not work.
7 Click **OK**.

In addition, you might need to edit the policy file to add or remove permissions for the folders that correspond to the channels. To learn how to set up a publication channel, see "Adding SAS Publication Channels" in *SAS Web Infrastructure Kit: Administrator's Guide* at **http://support.sas.com/rnd/itech/doc9/portal_admin/content/ag_saschannels.html**.

Scheduling Issues With Specific Applications

Issues With SAS Data Integration Studio

SAS Data Integration Studio has unique issues with schedule deployment and passwords and credentials, as discussed in the following sections.

Schedule Deployment

Deployment for scheduling creates the necessary metadata about the scheduled job and associates the job to a particular batch server. However, you need to create a deployment directory to set the location for placing the generated SAS DATA step program that will be executed by the batch server. To create the deployment directory, complete these steps:

1 Right-click Schedule Manager in SAS Management Console.
2 Select Deployment Directories.
3 Select your Application Server, and click **New**.
4 Enter directory properties for the name and path.

To redeploy jobs after promoting a repository, complete these steps:

1 Make sure that the command for the associated batch server is correct by checking Server Manager in SAS Management Console.
2 Redeploy all your jobs using the **Redeploy Jobs for Scheduling** option on the Tools menu in SAS Data Integration Studio.
3 Reschedule your flows in Schedule Manager for any flows containing any of the promoted jobs or use the **Reschedule All Flows** action in Schedule Manager.
4 Once the jobs have been redeployed, you need to go to Schedule Manager and either add the jobs to existing flows and reschedule the flows, or create new flows and schedule the flows.

Passwords and Credentials

Sometimes scheduled SAS Data Integration Studio jobs stop working after you have changed your password. When SAS Data Integration Studio deploys a job for scheduling, it generates a SAS file as the **-sysin** value for running MVA SAS in batch mode. Many times, the generated SAS program contains access to various data sources that specify the credentials within the LIBNAME statement. You will need to either redeploy the job to have the SAS file updated with the new credentials or edit the SAS file to update the credentials.

Other times, you can run a SAS Data Integration Studio job within SAS Data Integration Studio, but it fails because it cannot resolve the LIBNAME. One possible reason is that you have the wrong credentials in the LIBNAME statement. When SAS Data Integration Studio deploys a job for scheduling, it generates a SAS file as the **-sysin** value for running MVA SAS in batch mode. Many times the generated SAS program contains access to various data sources that specify the credentials within the LIBNAME statement. If the user that executes the job is not the same user who generates the job, then it is possible that you could have credentials for the LIBNAME statement that work in SAS Data Integration Studio but not in the scheduled environment.

Issues With SAS Marketing Automation

Launcher

To capture capture output from the SAS Marketing Automation launcher command, add the following option to the start of the INI file's command line:

```
-Dma.launcher.log
```

When you rerun the flow, you should see a log file in the home directory for the Marketing Automation launcher.

If the launcher command runs successfully on a command line but returns an exit status of 45 when scheduled under Platform scheduling servers, something must be wrong with the scheduling environment. Error code 45 generally means the launcher failed due to an unexpected error. This could be the result of permissions issues within the SAS Marketing Automation launcher install. There may be file permissions issues as well.

Return Codes

The possible return codes from a scheduled campaign include the following:

- 1: Unexpected error
- 100: Unknown argument
- 101: Missing argument
- 102: Invalid directive
- 200: Unable to log onto application server
- 300: Unable to acquire execution service
- 301: Execution failed
- 302: Not executable
- 400: Problem while logging off of application server

Issues With SAS Web Report Studio

SAS Web Report Studio requires particular users for scheduling. In addition, there are several possible return codes for scheduled reports.

Required Users

The following table summarizes the required SAS Web Report Studio users:

Table 20.3 User IDs Needed for Scheduling

Name	Purpose	Example User ID
ServerUser	An administrative OS /LSF user is required to run the Platform scheduling server daemons.	lsfadmin (omrstart on biddev09)
ClientUser	An LSF user is required to run command line commands such as outputgen.exe.	lsfuser
OMRUser	An OMR user is required to connect to OMR to perform tasks such as to read/ write metadata or create SAS Web Report Studio reports.	CARYNT\johndoe
BatchGenOMRUser	A Trusted OMR user is used by outputgen.exe to connect to OMR to get user credentials. Must also be a member of the Scheduling Admins group. Once the credentials are obtained, outputgen reconnects to OMR using the specified OMRUser to generate the report.	sastrust
WRSCleanupUser	A WRS OMR Cleanup user is used by WRS to connect cleanup-protected hidden arguments stored with jobs. This user is used to do nightly cleanup of partially deleted jobs. Must also be a member of the Scheduling Admins group.	saswbadm

Return Codes

The possible return codes from a scheduled report include the following:

- 1: Unable to read or validate configuration options
- 2: Unable to get Repository
- 3: Unable to extract an Entry
- 4: Unable to parse/read the input file listing the entries to be run/validated
- 5: Unable to find the input file listing the entries to be run/validated

- 6: Unable to write the extracted list to the output file
- 7: Caught a MetaDataException while getting a IntelligentQueryMetadataService
- 8: Caught a BatchGenerationException
- 9: Unable to get Session Context
- 11: Caught a RuntimeException
- 12: Caught a Throwable
- 13: Unable to get temporary user session
- 14: Unable to resolve metadatakey
- 15: Unable to deploy platform services
- 16: Unable to initialize logging service
- 17: Unable to retrieve User name from Job

Scheduling Issues Related to the Operating System

Windows

The Windows security policy has some requirements that are unique. In order to properly install the Platform Computing software, you must supply a valid user ID and password to run and administer the services. This requires that the user ID used to run the installation program have "Act as part of the operating system" privileges. The user ID that you specify to run the services under must have "Log on as a batch job" privileges. There can also be issues with the password expiring and the wrong domain name provided. Many times these user IDs are not your normal user ID. One simple way to test the domain, user, and password is to do the following:

1. Bring up a DOS command prompt.
2. Use the **`runas`** command to bring up a new DOS command prompt running as the other **`userid >runas /user:DOMAIN\userid cmd`**.
3. Enter the password. Then, a new DOS command prompt runs.

You can then use this new DOS command prompt to run the various scheduled commands that are failing to see if they work from the DOS prompt. If so, then you know there is some issue with the scheduler setup. If not, then chances are good that you will have additional information on the console of the DOS command window to help you determine the problem.

In a multi-user environment where you want more than one user submitting and running flows, there are permission settings on folders that need to be in place. The Platform scheduling server folders should already be set for scheduler service and administrator accounts and need no further changes. You will need to make sure that the following files installed by LSF have the following permissions:

Table 20.4 Permissions for Files Installed by LSF

Folder	User Group	Permissions
JSTOP\work	LSF service accounts	Full control (All)(All)
JSTOP\work	LSF administrators	Full control (All)(All)
JSTOP\work	Everyone	Special access (R)(R)
JSTOP\logs	LSF service accounts	Full control (All)(All)
JSTOP logs	LSF administrators	Full control (All)(All)
JSTOP logs	Everyone	Special access (R)(R)
JSTOP\conf\lsfuser.passwd	JS service accounts	Special access (R)(R)

You will need to make sure that the files installed by Platform scheduling servers have the following permissions:

Table 20.5 Permissions for Platform Scheduling Server-Installed Files

Folder	User Group	Permissions
JSTOP\work	LSF service accounts	Full control (All)(All)
JSTOP\work	JS administrators	Full control (All)(All)
JSTOP\work	Everyone	Special access (R)(R)
JSTOP\log	JS service accounts	Full control (All)(All)
JSTOP log	JS administrators	Full control (All)(All)
JSTOP log	Everyone	Special access (R)(R)

AIX

The AIX environment is unique in that you can configure your kernel to be in either 32-bit mode or 64-bit mode, and your 64-bit applications will run fine with either one. Platform Computing requires that the kernel be in 64-bit mode for their 64-bit applications. This means that you will need the kernel in 64-bit mode to use Platform scheduling servers and Platform LSF. To determine if your kernel is in 64-bit mode, use the **lsconf** command. Here is an example of 32-bit kernel running with 64-bit kernel installed:

```
$ lsconf -k
Kernel Type: 32-bit
$ lslpp -l | grep bos | grep 64
bos.64bit                5.1.0.50  COMMITTED  Base Operating System 64 bit
bos.mp64                 5.1.0.50  COMMITTED  Base Operating System 64-bit
bos.64bit                5.1.0.35  COMMITTED  Base Operating System 64 bit
bos.mp64                 5.1.0.35  COMMITTED  Base Operating System 64-bit
```

To switch to 64-bit mode you need to use the **rtfm** command.

MVA SAS Exit Behavior

When Platform scheduling servers are used, MVA SAS exits with warnings that cause it to have an Exit status instead of a Done status. Platform scheduling servers treat any exit code of non-zero as being Exit instead of Done.

There are two common ways to resolve this issue. One is to use a sasbatch script from a TUE installation. The other is to use a job starter. The TUE installation will create a script called "sasbatch" for running SAS in batch mode .

For Windows installations, you can modify the script as follows:

```
call sas.bat %*%
 set rc=%ERRORLEVEL%
 if %rc==1 goto makenormalexit

 exit %rc

:makenormalexit

exit 0
```

For UNIX installations, you can modify the script as follows:

```
sas $*
rc=$?
if [ $rc -eq 1 ]; then
   exit 0
else
   exit $rc
fi
```

The job starter is like an automatic job wrapper and is responsible for starting the task.

Further Resources

For additional information about managing job submissions, see the white paper *Implementing Site Policies for SAS® Scheduling with Platform JobScheduler* at **http://support.sas.com/documentation/whitepaper/technical/JobScheduler.pdf**.

CHAPTER

21

Supporting Grid Computing

Overview of Supporting Grid Computing

Grid computing has become an important technology for organizations that:

- have long running applications that can benefit from parallel execution
- want to leverage existing IT infrastructure to optimize compute resources and manage data and computing workloads

SAS applications that perform many replicate runs of the same fundamental task, or perform independent tasks against the same input data source, can often benefit from a grid environment. In addition, enterprise grids provide a virtualized pool of resources that enable multiple SAS applications to perform more efficiently through dynamic resource allocation and prioritization.

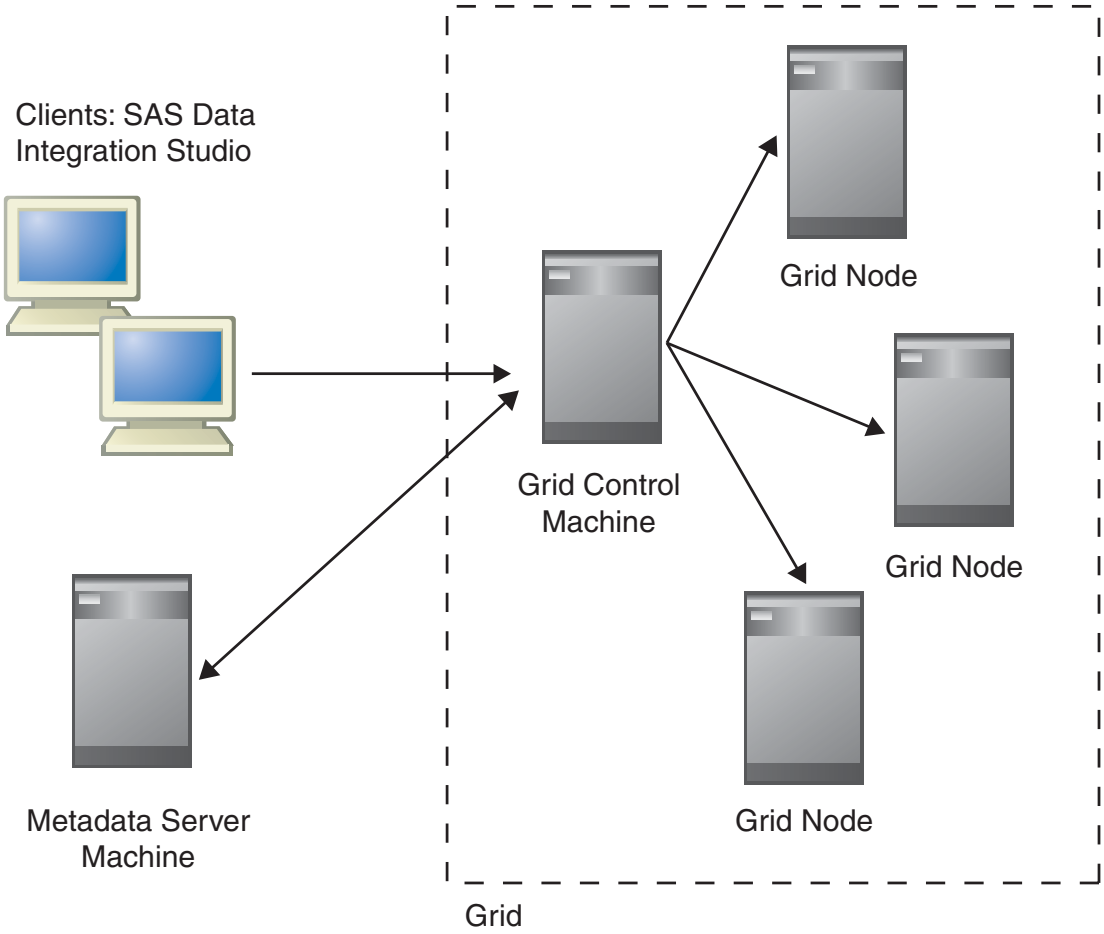
Grid computing is not a new concept to SAS. In fact, SAS has supported grid starting with Version 8.2 and the parallel processing capabilities of SAS/CONNECT. More recently, SAS has introduced enhanced support for grid computing by providing grid automation within several of the SAS products, as well as more dynamic resource based load-balancing capabilities and monitoring. SAS Data Integration Studio users, SAS Enterprise Miner users, and SAS programmers developing SAS applications can all benefit from these enhanced grid capabilities.

This chapter presents a typical architecture for a SAS ETL grid environment. Details are provided for the installation and configuration of the software and metadata components necessary to set up an ETL grid environment. While the focus of this chapter is an ETL grid, most of the information presented here applies to setting up a grid architecture to support any type of SAS application in a grid environment.

Architecture of an ETL Grid System

The figure below shows the typical architecture of an instance of the SAS Intelligence Platform that contains a computing grid. In this example, the grid is being used to run SAS Data Integration Studio jobs.

Figure 21.1 Architecture of an ETL Grid System



SAS Data Integration Studio is capable of generating code for grid-enabled jobs. It submits the code for such a job to the grid control machine, which hosts a workspace server and SAS/CONNECT so that it can distribute code to other computing resources in the grid. SAS/CONNECT interacts with Platform Suite for SAS to start a SAS session on the most appropriate grid node. This SAS session then runs the code generated by SAS Data Integration Studio. Jobs can also be scheduled to run on the grid.

For details about what software components must be installed and configured on the different machines shown in the figure, see "Installing and Configuring the Required Software" on page 478.

Topics Covered in This Chapter

The remainder of this chapter provides the following information:

- "Installing and Configuring the Required Software" on page 478 explains what software needs to be installed and configured on each machine in the system and goes over a few pre-installation tasks that you must perform before installing this software.
- "Basic Configuration of the Grid" on page 482 discusses the configuration of the Platform Computing software that you must perform manually—after running the SAS Configuration Wizard—to make the grid functional.
- "Verifying That the Grid Is Working" on page 483 explains how you can run a SAS-supplied job on the grid in order to make sure that the grid is working properly.
- "Advanced Configuration Tasks" on page 484 explains how to partition the grid into subgrids, how to optimize the number of jobs that can run on a processor, and how to set up a central deployment directory for scheduled jobs.
- "Troubleshooting" on page 488 explains how to solve problems that you might encounter while setting up your grid.
- "Improving Performance" on page 490 provides some tips for making jobs run as efficiently as possible.
- "Grid Manager Plug-In" on page 492 discusses a SAS Management Console plug-in called Grid Manager that you can use to monitor the runtime workload on your grid.

Installing and Configuring the Required Software

Installing and configuring an instance of the SAS Intelligence Platform that contains a grid of computing nodes is really no different from installing the platform without a grid. As usual, you should install any third-party products—including your Platform Suite for SAS software—before you install and configure your SAS software. You can then use the SAS Software Navigator and the SAS Configuration Wizard to install and configure the SAS software on each machine in your system. You can find information on both installing Platform Suite for SAS software and running the SAS installation and configuration tools in the *SAS Intelligence Platform: Installation Guide*.

However, because grid enablement requires the installation and configuration of a number of components, we have summarized this information in the following subsections:

- what components will be installed and configured on each machine in the system
- what requirements must be met before you begin the installation

Architecture of a Grid System

"Architecture of an ETL Grid System" on page 476 presents a high-level view of the makeup of a system that contains a grid to be used by SAS Data Integration Studio. This section takes a more detailed look at what software components are installed and configured on each machine in the system.

Metadata Server Machine

First, you install software on the metadata server machine.

Figure 21.2 Software Installed on the Metadata Server Machine

Metadata Server Machine

Installed Components:
- SAS Foundation
  - Base SAS
- SAS Management Console
- Grid Manager Plug-in for SAS Management Console

Configured Components:
- Metadata Server

When you configure the metadata server on this machine, you create your foundation metadata repository. As you create metadata objects in subsequent steps of the installation, those objects are stored in this repository.

Note: In the recommended configuration, the metadata server machine is not part of the grid. However, it is possible to put the metadata server on the grid control machine. You will need a custom deployment plan to do this. In addition, as we discuss later, this change will affect the way in which you configure Platform LSF. △

Grid Control Machine

Next, you install software on the grid control machine.

Figure 21.3 Software Installed on the Grid Control Machine

Grid Control Machine

Installed Components:
- Platform Process Manager and Platform LSF
- Platform Grid Management Service
- SAS Foundation
  - Base SAS
  - SAS Integration Technologies
  - SAS/CONNECT
- SAS Management Console
- Grid Manager Plug-in for SAS Management Console

Configured Components:
- Platform Process Manager Server
- Grid Monitoring Server
- SAS Application Server
  - SAS Data Step Batch Server
  - SAS Grid Server
  - Workspace Server
  - SAS/CONNECT Server
- Object Spawner
- SAS/CONNECT Spawner

As you can see, you create metadata objects representing a number of servers when you configure this machine:

- The Platform Process Manager Server enables you to schedule jobs in the grid.
- The Grid Monitoring Server enables you to use the Grid Manager in SAS Management Console to monitor the load on the nodes in the grid.
- The SAS Data Step Batch Server stores the command to execute for scheduled jobs.
- The SAS Grid Server stores the command that Platform LSF will use to start SAS/CONNECT sessions on the grid nodes.

- The workspace server executes jobs that are submitted by SAS Data Integration Studio clients.
- The object spawner starts Workspace Server processes as they are needed.

Other Grid Nodes

Once you have set up the grid control machine, you can set up the grid nodes.

Figure 21.4 Software Installed on the Grid Nodes

Grid Node

Installed Components:
- Platform LSF
- SAS Foundation
  - Base SAS
  - SAS/CONNECT
- SAS Management Console

Configured Components:
- SAS Data Step Batch Server*
- SAS Grid Server*

* A directory structure and a set of files are created in the node's configuration directory. No metadata objects are created.

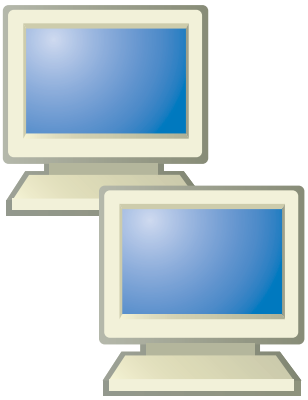
You install SAS/CONNECT on each of these nodes so that Platform LSF can start SAS/CONNECT sessions on these nodes as necessary.

You also configure a SAS Data Step Batch Server and a SAS Grid Server on these machines. When you configure these servers, no metadata objects are created; however, the SAS Configuration Wizard creates **`BatchServer`** and **`GridServer`** directories in the configuration directories on these machines. These directories contain files, such as log files, that you will use in the operation and management of the system.

Client Machines

Finally, you install software on your client machines.

Figure 21.5 Software Installed on Client Machines



Clients

Installed Components:
- SAS Data Integration Studio
- SAS Management Console
- Grid Manager Plug-in for SAS Management Console

Pre-Installation Tasks

For complete information about the tasks that you must perform before installing your SAS software, see the *SAS Intelligence Platform: Installation Guide*. This section discusses two special system requirements for installations that include a grid and the need for the SAS configuration directory to be located in the same place on each grid node. It also reiterates some other requirements that have a direct bearing on grid computing.

System Requirements

There are two main system requirements:

- First, if you are using a Windows machine as the grid control machine, that system must be running a server-class operating system like Windows 2003 Server. Because all of the nodes in the grid need to communicate with the control machine, it is easy to overrun the connection limit on a machine that is running an operating system that is not server-class.
- Second, you must make sure that machines that you have chosen to use for the grid control machine and grid nodes provide platforms that are supported by both Platform Computing and SAS. A machine that is running 64-bit Windows, OpenVMS for Integrity Servers, Tru64 UNIX, or z/OS cannot serve as the grid control machine or a grid node.

Configuration Directory

When specifying the configuration directory in the SAS Configuration Wizard, you must use the same value for all of the nodes of your grid. This is necessary in a grid environment because the same Grid Server definition is used for all grid nodes, and specifies how to launch a SAS session on any grid node.

For example, if your configuration directory were **`C:\SAS\GridETLServer`**, the SAS Configuration Wizard would configure your grid to use the same launching command on all grid nodes: **`C:\SAS\GridETLServer\Lev1\SASApp\GridServer\sasgrid.bat`**.

User Accounts

For a complete explanation of the operating-system user accounts that you must create before installing the SAS Intelligence Platform, see the *SAS Intelligence Platform: Installation Guide*. This section discusses the user accounts that relate specifically to grid computing:

- If you will be installing any part of the Platform Suite for SAS on a Windows system, you might want to create a Platform installation account. This can be a local account on each machine or a domain account. You must log on as this user when you install the Platform software on each Windows system. This user must be a local administrator on each of the Windows systems and must have the user right “Act as part of the operating system” on Windows NT and Windows 2000 systems.
- You must create an account for an LSF Administrator (lsfadmin). The LSF Administrator is the owner of the LSF configuration and log files. In addition, this user has permission to edit configuration files, reconfigure the cluster, and control jobs belonging to any user. Each machine on which Platform Process Manager or Platform LSF will run must be able to authenticate this user. In addition, on Windows systems, this user needs the user rights “Log on as a batch job” and “Act as part of the operating system.”

Note: It is also necessary to ensure that each user who will submit a job to the grid can be authenticated on each node in the grid. This means that local user accounts will only work if each user has an account (and the same password) on all of the nodes. Although this is not necessarily a pre-installation task, it must be performed before the grid is made available to your users. △

Platform Computing Software

Installation of all third-party software is a prerequisite to installing your SAS software. This means that before you begin using the SAS Software Navigator to install SAS software, you must install the appropriate components of the Platform Suite for SAS on the grid control machine and on all of the grid nodes. For information on how to install this software, see the *SAS Intelligence Platform: Installation Guide*.

Basic Configuration of the Grid

The SAS Configuration Wizard handles the great majority of the configuration of your grid for you. However, the LSF Administrator (lsfadmin) and the users who will run jobs on the grid must complete the basic configuration of the grid manually. See the instructions in the two following subsections.

Tasks Performed by the LSF Administrator

The LSF Administrator should log in to the grid control machine and perform the tasks described in this section.

1. Edit the file *LSF-install-dir*`\conf\lsf.shared` to register the grid as an LSF resource. Open this file, and in the Resource section, add a line similar to the third line below:

```
Begin Resource
RESOURCENAME  TYPE    INTERVAL INCREASING  DESCRIPTION        #Keywords
   SASApp     Boolean ()       ()          (Windows)
```

   By default, the resource name is the name of the SAS application server that contains your SAS Grid Server. (If your grid control machine also hosts your metadata server, this application server name will be SASMain.) This name is case sensitive. The description, of course, can vary.

2. Edit the file *LSF-install-dir*`\conf\lsf.cluster.`*cluster-name* (`lsf.cluster.sas_cluster`, by default) to indicate which machines, including the grid control machine, are part of the grid. You do this by editing a line in the Host section of the file for each machine in the grid, as shown below:

```
Begin Host
HOSTNAME  model    type        server rlm  mem  swp  RESOURCES    #Keywords
D1234     !        NTX86       1      -    -    -    (nt SASApp)
D1235     !        NTX86       1      -    -    -    (nt SASApp)
```

   Note that each grid node is being declared part of the resource SASApp (case sensitive).

   The following fields are required:

   HOSTNAME — the name of the grid node.

   model — a model name chosen from the HostModel section of `lsf.shared`, or an exclamation point (!). An exclamation point

indicates that the model is to be detected by the Load Information Manager (LIM) running on the host.

type — a host type chosen from the HostType section of **`lsf.shared`** or an exclamation point.

RESOURCES — a set of one or more strings defined in the Resource section of **`lsf.shared`**. One of the resources must be the name of the SAS application server. Normally, this is SASApp; however, if your grid control machine also hosts your metadata server, the name will be SASMain.

For more information about editing **`lsf.cluster.`***cluster-name*, see the *Platform LSF Reference*, which is available from your Platform Computing CDs.

3 On Windows systems, edit the file *LSF-install-dir***`\etc\lsf.conf`**, if necessary. Look for a line like the following one:

```
LSF_USER_DOMAIN=domain-name
```

If a domain name has been specified, remove it and save the file.

4 Issue the command **`lsadmin reconfig`** in order to restart LIMs on all of the hosts in the cluster. Note that restarting the LIMs can take a little time. Do not proceed to the next step until this step is complete.

5 Issue the command **`badmin reconfig`** to dynamically reconfigure LSF. If this command fails, the preceding step might not have completed. Wait a minute and try the command again.

Tasks Performed by the Users Who Will Run Jobs on the Grid

If the grid control machine is a Windows system, all users who will submit jobs must register their passwords with LSF. LSF uses these passwords to start jobs on behalf of the users. To register a password, the user should follow these steps:

1 At a Command Prompt on any machine in the grid, enter the command **`lspasswd`**.

2 Enter your password when you are prompted for it.

At this point, the basic configuration of your grid is complete, and you should be able to verify that the grid is working as explained in "Verifying That the Grid Is Working" on page 483.

Verifying That the Grid Is Working

At this point, it is a good idea to quickly verify that the grid is working properly. You can do this by running the short SAS program presented in this section.

On UNIX systems, make sure that you have set the environment variable **`LSF_SERVERDIR`** before you run this program. To set the variable, log on to the LSF master host as **`root`**, and issue the appropriate command:

- For csh or tcsh: **`% source`** ***`LSF-install-dir`***`/conf/cshrc.lsf`
- For sh, ksh, or bash: **`$ .`** ***`LSF-install-dir`***`/conf/profile.lsf`

1 Make sure that no one else is using the grid. Otherwise, the program might not finish, since it requires all job slots to be free.

2 Open the SAS Editor (by starting SAS).

3 Cut and paste into the editor the code at **`http://support.sas.com/rnd/scalability/grid/gridfunc.html#testprog`**.

4 Submit the program by selecting  . You will be prompted for the information needed to connect to the metadata server: a host name, a port number, a user ID, and a password. Supply this information so that the program can run.

   Follow the progress of the program in the Grid Manager. Make sure that SAS sessions are started on all of the grid nodes and that multiple sessions are started on any nodes that have multiple CPUs.

   Note: There may be some delay as jobs are submitted to the grid nodes. △

5 After the program finishes, check the log file. The value of the variable rc should be 0.

If this program does not write the expected information to the log, contact Technical Support for information about how to diagnose your problem.

Advanced Configuration Tasks

"Basic Configuration of the Grid" on page 482 discusses the configuration tasks that you must perform to make your system functional. This section discusses some further configuration that you might want to perform, depending on who is using the grid, what types of jobs are running on the grid, and whether jobs are being scheduled to run on the grid.

Partitioning the Grid

You can partition a grid so that different portions of the grid are allocated to specific types of workload. Administrators can partition the grid by performing the following tasks:

- The LSF Administrator defines a resource that includes a subset of the machines in the grid.
- A SAS administrator updates the metadata for the SAS Grid Server so that it contains the name of the LSF resource.

In SAS Data Integration Studio, users can specify the LSF resource in the properties for a Loop Transformation in a SAS Data Integration Studio job. When the job is submitted for execution, it is submitted to one or more machines that are specified in the resource.

Note: An LSF resource is called a *workload* in the SAS metadata. △

Add a Resource to the lsf.shared File

The LSF Administrator performs these steps to add a resource to the **`lsf.shared`** file. Later, in another configuration file, the LSF Administrator will associate this resource with a subset of the machines in the grid.

1 Log on to the grid control machine as the LSF Administrator.
2 Add a resource to the Resource section of the **`lsf.shared`** file. Specify at least a name and a brief description of the resource.
3 Save your changes and close the file.

In the example Resource section below, the name of the new resource is ETL.

```
Begin Resource
RESOURCENAME   TYPE    INTERVAL INCREASING  DESCRIPTION
ETL            Boolean ()       ()          (ETL_Workload)
End Resource
```

For details about the syntax for resources in the **`lsf.shared`** file, see *Platform LSF Reference*. For general information about adding resources to an LSF cluster, see *Administering Platform LSF*.

Configure the Resource in the lsf.cluster File

After a resource has been added to the **`lsf.shared`** file, the LSF Administrator performs these steps to associate the resource with a subset of machines in the grid.

1. Log on to the grid control machine as the LSF Administrator.
2. Locate the entries for the grid machines in the Host section of the **`lsf.cluster.sas_cluster`** file.
3. Identify the machines that will make up the resource.
4. In the Resources column for each machine in the partition, add the name of the new resource.
5. Save your changes and close the file.

In the example Host section below, the resource for the grid is SASApp, and the resource for the subset of the grid is ETL. The brackets indicate optional attributes that have been omitted for simplicity.

```
Begin Host
HOSTNAME [...] type    server [...] RESOURCES
D1234          NTx86   1            (nt SASApp ETL)
D1235          NTx86   1            (nt SASApp ETL)
D1236          NTx86   1            (nt SASApp ETL)
D1237          NTx86   1            (nt SASApp)
D1238          NTx86   1            (nt SASApp)
End Host
```

In the example, D1234, D1235, and D1236 are associated with the resource named ETL. Accordingly, jobs submitted to this resource will be executed by one or more of these machines. For more details about the syntax for hosts in an **`lsf.cluster.`***cluster-name* file, see *Platform LSF Reference*.

Update the Metadata for the SAS Grid Server

A SAS administrator performs these steps to specify an LSF resource/workload in the metadata for a SAS Grid Server. Typically, the resource would have already been defined and configured, as described in the previous sections.

1. In SAS Management Console, open the metadata repository that contains the metadata for the SAS Grid Server.
2. In the navigation tree, select **`Server Manager`**.
3. Expand the folders under **`Server Manager`** until you see the metadata objects for the SAS application server and its Logical SAS Grid Server component.
4. Expand the Logical SAS Grid Server component so that you see the metadata object for the SAS Grid Server.

5 Right-click the metadata object for the SAS Grid Server, and select **Properties**.
6 In the Properties window for the SAS Grid Server, click the **Options** tab.
7 On the **Options** tab, click **Advanced Options**.
8 In the Advanced Options window, specify a workload, such as the ETL workload discussed above.
9 Click **OK** twice to save your changes and close the Properties window.

Update the Metadata for the Loop Transformation

An ETL specialist performs these steps to specify an LSF resource in the properties for a Loop Transformation in a SAS Data Integration Studio job. When the job is submitted for execution, it is submitted to one or more machines that are associated with the resource.

It is assumed that the default SAS application server for SAS Data Integration Studio has a Logical SAS Grid Server component, and this component has been updated as described in the previous section.

1 In SAS Data Integration Studio, open the job that contains the Loop Transformation to be updated.
2 In the Process Designer window, right-click the metadata object for the Loop Transformation and select **Properties**.
3 In the Properties window, click the **Loop Options** tab.
4 On the **Loop Options** tab, in the **Grid workload specification** text box, enter the name of the desired workload (LSF resource), such as ETL. The name of the workload must match exactly the name of the resource that you defined in the LSF configuration files.
5 Click **OK** to save your changes and close the Properties window.

Changing the Number of LSF Job Slots

In LSF, a job slot is a basic unit of processor allocation. By default, each processor in a grid is a single job slot, which means that LSF will only execute one job at a time on that processor. However, it is possible to configure more than one job slot per processor on a host-by-host basis. If the jobs that you are submitting to the grid are I/O intensive, it might be a good idea to try configuring two job slots per processor—especially if the processor is a fast one.

Here is how to change the number of job slots on a machine:

1 Log into the grid control machine as the LSF Administrator (lsfadmin).
2 Open the file **lsb.hosts**, which is located in the directory *LSF-install-dir***conf****lsbatch***cluster-name***configdir**. In the Host section of the file, you will see, at a minimum, an entry for a **default** host.

```
Begin Host
HOST_NAME MXJ    r1m     pg    ls    tmp  DISPATCH_WINDOW  #Keywords
default    !     ()      ()    ()     ()     ()            #Example
End Host
```

3 Edit this file to specify either a single maximum number of jobs that applies to all hosts in the system, or separate maximums for each node in the grid.
   - To specify a maximum number of jobs that applies to all grid nodes, edit the entry for the **default** host. The value ! means one job *per processor*. You can

replace this value with a number representing the maximum number of jobs *on the host*.

- To specify a maximum number of jobs on a per-host basis, add a new line to the file for each host. See the example below.

```
Begin Host
HOST_NAME MXJ    r1m     pg    ls    tmp  DISPATCH_WINDOW  #Keywords
D1234      2     ()      ()    ()     ()     ()            #Example
default    !     ()      ()    ()     ()     ()            #Example
End Host
```

The line for D1234 specifies that a maximum of two jobs can run on this host concurrently.

4 Save your changes, and close the file.

5 Issue the command **`badmin reconfig`**.

Defining a Central Deployment Directory

In a non-grid system, where SAS code is often executed by a single machine, scheduled jobs are generally deployed to a special deployment directory in the configuration directory on that machine: *config-dir***`\Lev1\SASMain\SASEnvironment\SASCode\Jobs`**. In a grid system, you need to do things differently for a couple of reasons.

- First, there is no central deployment directory in this scenario. The directory structure mentioned above exists on every machine in the grid.
- Second, if you were to declare the deployment directory on one of the grid nodes to be *the* deployment directory, the other grid nodes would not be likely to have access to that directory.

Therefore, we recommend that you create a central deployment directory on a file server that is accessible from all of the grid nodes.

To create this central deployment directory and make it usable by SAS Data Integration Studio developers, you must perform the following tasks:

- Create the **`SASEnvironment`** directory structure on a file server, or copy it from the grid control machine.
- Use the Schedule Manager in SAS Management Console to define metadata for the deployment directory that references this new location.
- Tell your SAS Data Integration Studio developers to deploy jobs that will be run on the grid to this directory.

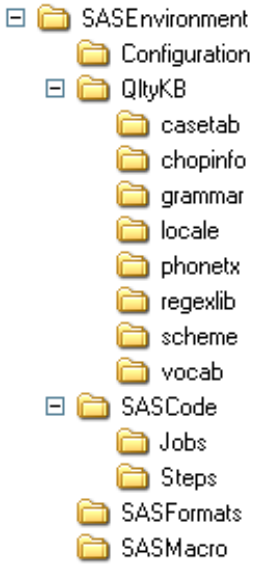
The following subsections explain briefly how to perform these tasks.

Create the SASEnvironment Directory Structure

To create the deployment directory, perform these steps:

1 Identify a file server that is accessible from all of the nodes in the grid.

2 On that file server, create the directory structure shown in the following display, or copy the structure from your existing grid control machine.



Note: Only the **SASEnvironment\SASCode\Jobs** directory is needed for deploying scheduled jobs. However, you may also want to have a central copy of other resources, such as format catalogs and macros. △

Create a New Deployment Directory in SAS Management Console

To define a new deployment directory, use SAS Management Console to perform the following steps:

1. Right-click **Schedule Manager**, and select **Deployment Directories** from the pop-up menu. A Deployment Directories dialog box appears.
2. In the Deployment Directories dialog box, choose the correct application server from the **Application Server** drop-down list. Normally, you will select SASApp.
3. Click **New** to bring up the New Directory dialog box.
4. In the New Directory dialog box, enter a name in the **Name** box and the path to the deployment directory in the **Directory** box. For example, on Windows, you might enter the directory *****server***\SASEnvironment\SASCode\Jobs**. Then click **OK**.

   Note: All of the nodes in the grid must be able to access the deployment directory using the path you specified above. △

5. Click **OK** in the Deployment Directories dialog box.

At this point, you have defined a deployment directory that SAS Data Integration Studio developers can use.

Use the New Deployment Directory from SAS Data Integration Studio

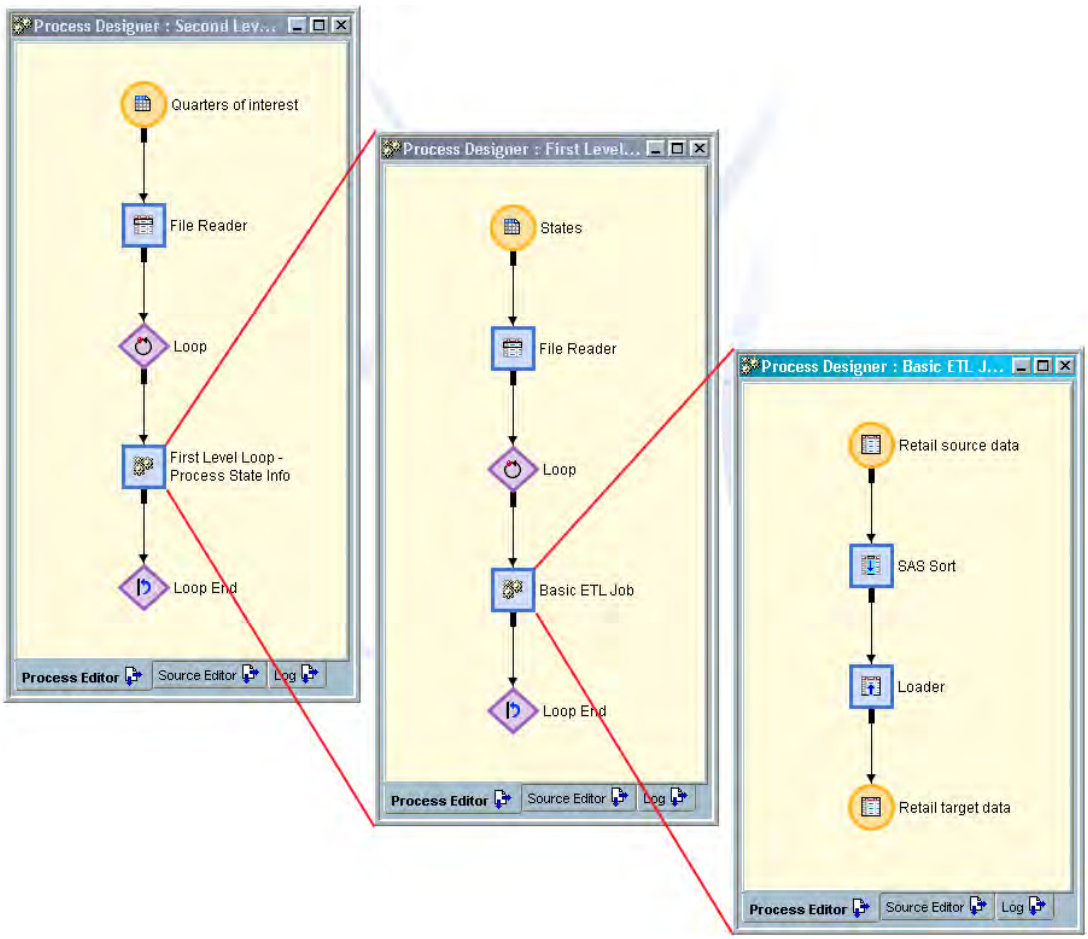
SAS Data Integration Studio developers can deploy jobs to the central deployment directory just as they deploy jobs to any other deployment directory. Just make sure that you supply these developers with the name of the application server that contains the SAS Grid Server (normally SASApp) and the path to the central deployment directory.

Troubleshooting

You should be alert to the following types of problems on the grid.

Deadlock

The display below shows a set of related SAS Data Integration Studio jobs. The first loop, running over the quarters of the year, needs to run a job for each state for that quarter.



If there are only four nodes in the grid, running four jobs to cover the four quarters in the year can consume all of the nodes in the grid. That would leave no remaining nodes to process data for each state. In this case, a deadlock would occur since no further jobs could be submitted, and no jobs could complete until more resources became available to process data for each state.

To avoid deadlock, you can add more nodes to the grid. Or in your SAS Data Integration Studio job, you can decrease the outer loop's setting for "Maximum number of concurrent processes." For more information about this setting, see the SAS Data Integration Studio online Help.

Resources on Remote Machines Cannot Be Accessed

All resources, such as libraries, files, and stored processes, must be available to all machines in the grid. Relative paths are not accessible by all nodes in the grid and cannot be used. If a non-domain account is used for the General Server User (such as mycomputer\sassrv) instead of a domain account, then the Stored Process Server will

not have access to the file system on all machines. Items such as the deployment directory and library paths will not be accessible and jobs will fail. Also, the user ID of the person submitting the job must be valid for all the machines in the grid.

LSF Cannot Start SAS/CONNECT Remote Sessions (Windows)

If LSF cannot start remote sessions, you might need to make a registry change. See the Microsoft Knowledge Base article at **`http://support.microsoft.com/default.aspx?scid=kb%3BEN-US%3B184802`** for an explanation. Be sure to back up the registry before attempting any modification.

Improving Performance

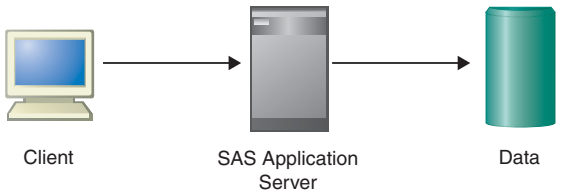
If you have set up a grid, you expect your ETL and other jobs to run faster than they would run on a single processor. However, not all jobs can benefit from running on a grid, and how much other jobs benefit from a grid can depend largely on the location of your data.

Short Running Jobs

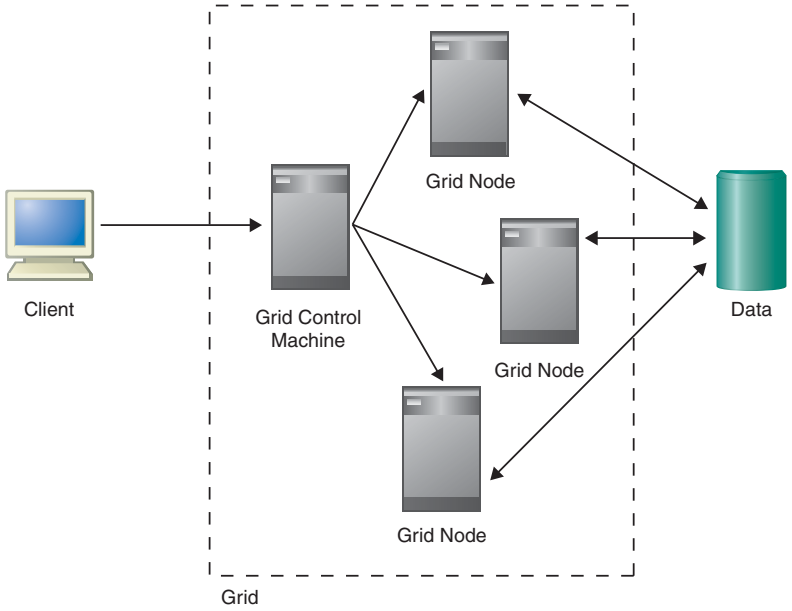
The first thing to bear in mind is that you may not see an increase in speed if you are executing short-running jobs—usually jobs that are not processing a great deal of data. See the following figure.

Figure 21.6 Using a Grid versus Using a Single Processor

Scenario A



Scenario B



It might seem that even a job that is processing a relatively small amount data would execute faster on three processors than on one processor. However, there is a certain amount of overhead taking place in Scenario B that is not necessary in Scenario A. Most obviously, in Scenario B, SAS sessions must be started on the grid nodes and code submitted to those sessions. This overhead can cause a short-running job to appear to execute more slowly on a grid than it would on a single processor.

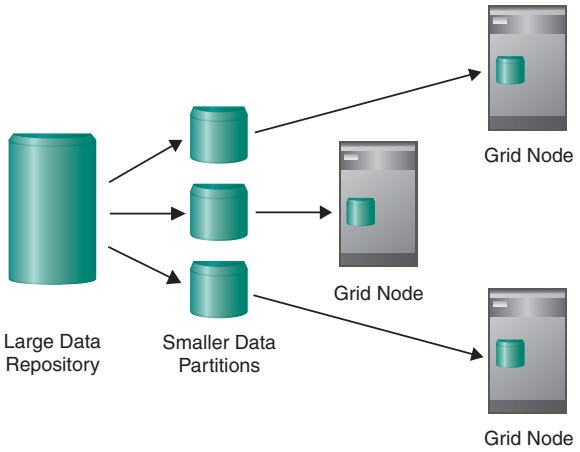
Contention for Data

Another factor that can affect processing speed is access to data. Let's assume that you are running a longer job, one that is processing a lot of data. Scenario B in the preceding figure may still be slower than Scenario A in cases in which all three SAS sessions in Scenario B are reading data from the same data repository. Contention among the processes for access to the data can render Scenario B slower than Scenario A. One way to address this would be to have sufficient network and storage infrastructure to support the I/O requirements of the application.

Partitioning Your Data and Moving It to Your Grid Nodes

As shown in the figure below, a way to make Scenario B much faster for long running jobs is to work with smaller pieces of data that are used by each grid node.

Figure 21.7 Partitioning Your Data and Relocating It



As the diagram above shows, it is important that pieces of the data be used locally on the grid node as much as possible. The large data repository is fine for the source or final target data, but all intermediate files should be stored and accessed on local disks for those nodes.

This is required if the I/O throughput supported by the large repository has only a few disk spindles (separate physical disk drives) or relatively slow network bandwidth to it. This is accomplished by using libraries (such as **WORK**) that are local to each grid node most of the time. This minimizes contention on the large repository and allows each grid node to run as efficiently as possible.

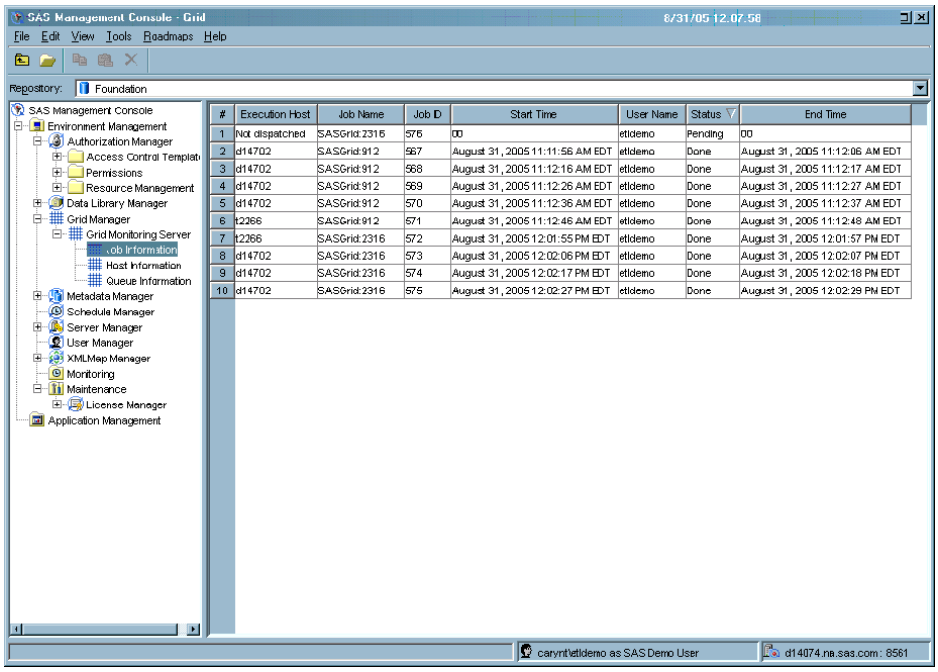
Grid Manager Plug-In

The Grid Manager plug-in for SAS Management Console enables you to monitor SAS execution in a grid environment. This plug-in enables you to manage workloads on the grid by providing dynamic information about the following:

- jobs that are running on the grid
- machines that are running under the grid
- job queues that are on the grid

All information is displayed in a table format. (See the Job View in the following display.) With Grid Manager, you have the ability to customize the view by selecting which columns of data to display and the order in which they should appear. In addition, you can filter, sort, and refresh jobs.

Each grid that you define must have one machine with the Platform Grid Management Service configured and running.



Monitoring Jobs

When you expand the Grid Manager node in the navigation tree, all of the grid monitoring servers that you have defined are listed under the name of the plug-in. To view information about the resources on a specific server, expand the server's node in the navigation tree. The resources for a server are grouped into three categories in the navigation tree:

- job information
- host information
- queue information

Select a category to display a table that contains resource information for the category. Right-click on a category in the navigation tree and select **`Properties`** from the pop-up menu to choose which columns are displayed in the table and to choose how to filter the resource information that is displayed.

Terminating Jobs

Note: When using Grid Manager to terminate a job in Windows, be careful to match the domain name, including the case, exactly.

If you logged on to SAS Management Console using a user ID that has been defined as an LSF Administrator ID, you can terminate jobs that have been submitted to the LSF servers. Any user can terminate his own job. The LSF Administrator can terminate any job. To terminate a job, follow these steps:

1. In the selection tree, select the **`Job Information`** node.
2. In the table, locate the job that you want to cancel.
3. Right-click on any column in the row for the job and select **`Terminate Task`** from the pop-up menu.

CHAPTER

22

Managing an Environment

Overview of Managing an Environment

A SAS environment identifies an entire set of related information such as levels, SAS application servers, scripts, utilities, and documentation. This chapter explains how to perform these tasks:

- customize the properties of a new environment
- manually change the properties of an existing environment
- add to an existing environment
- re-create an existing environment
- uninstall an environment.

Customizing the Properties of a New Environment

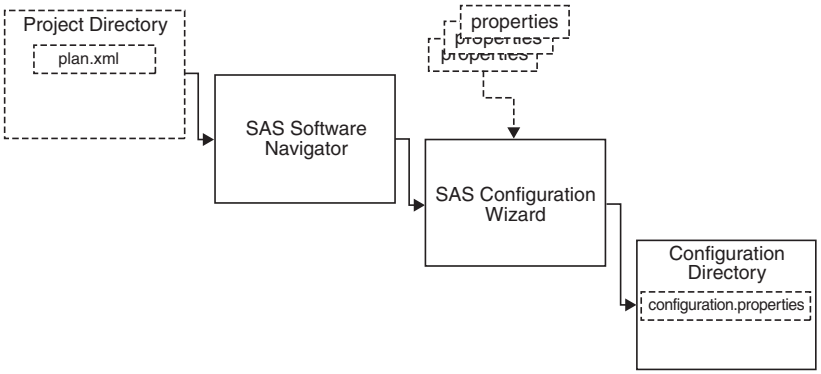
What Is a Configuration Property?

A configuration property is a simple name/value pair. The pairs are specified in a properties file, which contains one property pair (**`name=value`**) per line. Throughout the configuration process, property values are substituted for property names.

Note: The properties file is one of two data inputs to the SAS Configuration Wizard. The other input is the planning file or project directory. △

The following figure provides an overview of how properties are used during the configuration process. The properties that are shown as input to the SAS Configuration Wizard represent the default properties files that exist on the media as shipped from SAS. These default properties are distributed as a group of files in order to reduce duplication between generic properties and operating-specific properties.

Figure 22.1 An Overview of How Properties Are Used During the Configuration Process



The default properties are used in the following order:

- default English properties
- default locale properties (if they exist for the current locale)
- properties that override the default properties.

After the English properties are specified, the subsequent properties are specified as a subset of the English properties.

The SAS Configuration Wizard combines the properties into a **`configuration.properties`** file, which is saved in the environment's configuration directory. On Windows systems, the **`configuration.properties`** file is located in **`C:\SAS\`***configuration directory name*.

Ways to Modify the Configuration Properties File for a New Environment

There are two ways to manage the contents of the **`configuration.properties`** file. These methods can be used together.

Using a Command-Line Option to Specify a Custom Properties File

You can use a command-line option called **`OVERRIDE_PROPERTIES_FILE`** to specify the location of a custom properties file. The custom file can be used to override properties in a **`configuration.properties`** file that was created by a previous invocation of the SAS Configuration Wizard. The file needs to contain only the properties that you want to override.

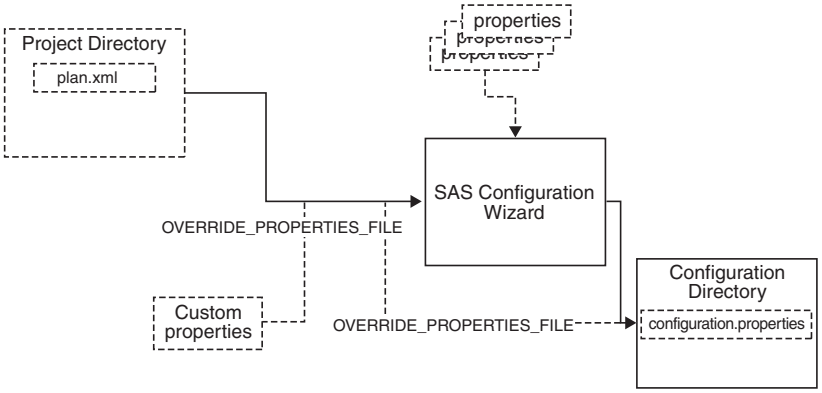
To use this option, you launch the SAS Configuration Wizard by using a command such as

```
setup -DOVERRIDE_PROPERTIES_FILE=''C:\myproperties\custom.properties''
```

Note: In the normal flow of initial deployment, the SAS Configuration Wizard is launched from the SAS Software Navigator. Therefore, you can only use the command-line option after initial deployment. △

The following figure shows at which point the custom properties are introduced into the configuration process.

Figure 22.2 Using the OVERRIDE_PROPERTIES_FILE Command to Specify a Custom Property File

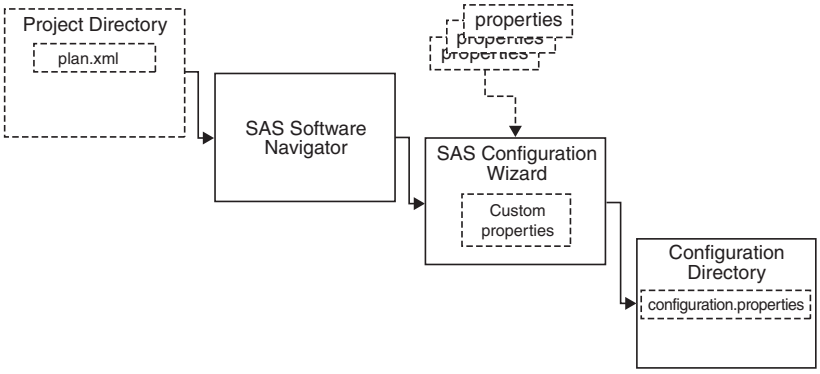


Editing the Configuration File from Within the SAS Configuration Wizard

You also can use the Advanced Properties Editor window in the SAS Configuration Wizard to modify the contents of the **`configuration.properties`**. On the Advanced Properties Editor window, click **`Edit Properties`** to open the **`configuration.properties`** file. Make your changes, save the file, and then continue completing the wizard.

The following figure shows at which point you modify the default properties by using the Advanced Properties Editor window in the SAS Configuration Wizard.

Figure 22.3 Using the Advanced Properties Editor Window from Within the SAS Configuration Wizard to Specify Custom Properties



Manually Changing the Properties of an Existing Environment

If you want to make simple changes to the environment without overwriting other, more extensive modifications, you should consider manually editing the affected files. (Rerunning the SAS Configuration Wizard overwrites your existing files without warning.)

These are the basic steps for making a manual change:

1 Use a text editor to open the **configuration.properties** file.
2 Examine the contents of the file to determine which property value that you want to change.
3 Search the directory structure for references to the value that you want to change. For example, to search for the property value **SASApp**, complete these steps:
   a Navigate to the correct level directory (e.g., **C:\SAS***configuration directory name***\Lev1**).
   b Enter a search command that is applicable to your operating system. For example:
      □ Windows:

```
findstr /S /I /M /P "SASApp" *
```

      □ UNIX:

```
find .-type f -exec grep -ls "SASApp" {}\;
```

4 In each located file, replace the original value with the new value. For example, you might replace **SASApp** with **FinancialServer**.

   Note: Although it is not technically required, it is a best practice to also update the **configuration.properties** file with the new value. △

Note: For information about how to manually reset passwords, see "How to Update Passwords for Required Accounts" in the "User ID and Password Management" section in the *SAS Intelligence Platform: Security Administration Guide*. △

Special Considerations for Client Applications

When you perform a search for a specific property value, the results of your search might include configuration files that are contained in the **clients** directory, which is located in the level directory (e.g., **C:\SAS***configuration directory name***\Lev1\clients**).

In addition to changing property values in the files in the **clients** directory, you must also modify any copied versions of the client configuration files. The SAS Configuration Wizard places the copied files in directories that are appropriate for the client applications. Often, the client application directories are located where SAS is installed, such as **C:\Program Files\SAS**.

Special Considerations for Web Applications

When you perform a search for a specific property value, the results of your search might include **.war** files that are contained in the **webapps** directory (e.g., **C:\SAS***configuration directory name***\Lev1\web\webapps**).

Instead of modifying the **.war** files in the **webapps** directory, you should modify the versions of those files that the SAS Configuration Wizard copied and expanded into your Web application server location.

Adding to an Environment

When you rerun the SAS Configuration Wizard in order to modify an environment, you might need to perform these tasks:

- Because the wizard will replace any existing files without notification, you should make copies of the files that you do not want to be overwritten.
- On Windows, the wizard will attempt to re-create services. To prevent this action, edit the **configuration.properties** file by setting the **_SERVICE** properties to **0**. For example, the service property name for the metadata server is **METADATA_SERVICE**.

Adding a SAS Application Server to an Existing Environment

The steps to add a SAS application server to an existing environment depend on whether you want to include the same servers that were used in the default SAS application server or you want to include different servers.

Including the Same Servers

If you want to use the same servers that exist in the default SAS application server, then you complete the SAS Configuration Wizard as you did for the original installation. When the Advanced Properties Editor window appears, click **Edit Properties**. In the **configuration.properties** file that opens, make these changes:

1. Change **AppName** to the name of the new SAS application server.
2. Change **MDAPDIR** to the name of the new SAS application server directory. This is the same value as **AppName**.

Including Different Servers

If you want the new SAS application server to contain different servers, then use one of these two methods:

- Run the SAS Configuration Wizard as you did for the initial deployment but specify a planning file that contains just the new servers.
- On the Select Install Set wizard window, select **Custom** and then choose to install just the servers that you want the new SAS application server to contain.

  If you use this method, then remember to also enter the name of the new application server and its directory. When the Advanced Properties Editor window appears, click **Edit Properties**. In the **configuration.properties** file that opens, make these changes:

  1. Change **AppName** to the name of the new SAS application server.
  2. Change **MDAPDIR** to the name of the new SAS application server directory. This is the same value as **AppName**.

Adding an Application Server Component to an Existing SAS Application Server

To add an application server component to an existing SAS application server, you complete the SAS Configuration Wizard by entering basically the same information that you entered to create the environment, including using same user IDs and passwords, the same configuration name, and the same paths. The modifications are listed in these steps:

1. When prompted, indicate that you are not using a planning file.
2. On the Select Install Set wizard window, select **Custom**.
3. When prompted, select one or more server components that you want to configure.

Note: Do not select any servers that already exist in the SAS application server. If you do, the original server information will be overwritten. △

Re-Creating an Existing Environment

You can use the command-line option **OVERRIDE_PROPERTIES_FILE** to specify a configuration file that was created through a previous invocation of the SAS Configuration Wizard. This effectively enables you to duplicate an environment.

Complete these steps to re-create an existing environment:

1. Copy the **configuration.properties** file from its directory in the existing environment to a new, temporary location.
2. Open the copied **configuration.properties** file and edit values that are specific to the original configuration. For example, change the **USER_MAGIC_FOLDER_1** property, which specifies the directory in which the environment will be created.
3. Directly launch the SAS Configuration Wizard by using a command such as

```
setup -DOVERRIDE_PROPERTIES_FILE=''C:\location of copied configuration file
\configuration.properties
```

When the SAS Configuration Wizard executes, the default values will be replaced by the values in the specified properties file.

To run the SAS Configuration Wizard in silent mode (no prompts), complete these steps:

1. Copy the **configuration.properties** file from the existing environment to a new, temporary location.
2. Open the copied **configuration.properties** file and then complete these steps:
   a. Uncomment the password properties and enter the passwords for which you would normally be prompted. Make sure that you enter the correct values because the wizard does not verify the accuracy of the information.
   b. Edit values that are specific to the original configuration. For example, change the **USER_MAGIC_FOLDER_1** property, which specifies the directory in which the environment will be created.
3. Directly launch the SAS Configuration Wizard by using a command such as

```
setup -DOVERRIDE_PROPERTIES_FILE=''C:\location of copied configuration file
\configuration.properties -i silent
```

Uninstalling an Environment

Complete these steps to remove a configuration directory that was created by the SAS Configuration Wizard:

1. Navigate to the **`UninstallerData`** directory. On Windows, the default location is **`C:\SAS\`***configuration directory name***`\UninstallerData`**.
2. From the **`UninstallerData`** directory, run the executable file for the uninstaller utility. On Windows, the file is named **`Uninstall SAS Configuration Wizard.exe`**. On UNIX, the file is named **`uninstall.sh`**.

   This application removes any files, scripts, services, etc. that were initially created by the SAS Configuration Wizard. It does not remove files that you have modified; those files will be listed in a window after the default files are uninstalled.
3. Review the remaining contents of the **`C:\SAS\`***configuration directory name* directory and decide which files that you want to keep.
4. Copy the files that you want to keep to another location.
5. Delete the entire directory structure. For example, you can use the following commands:
   - Windows:

     ```
     rmdir /S/Q configuration directory location
     ```
   - UNIX:

     ```
     rm -r -f configuration directory location
     ```

CHAPTER

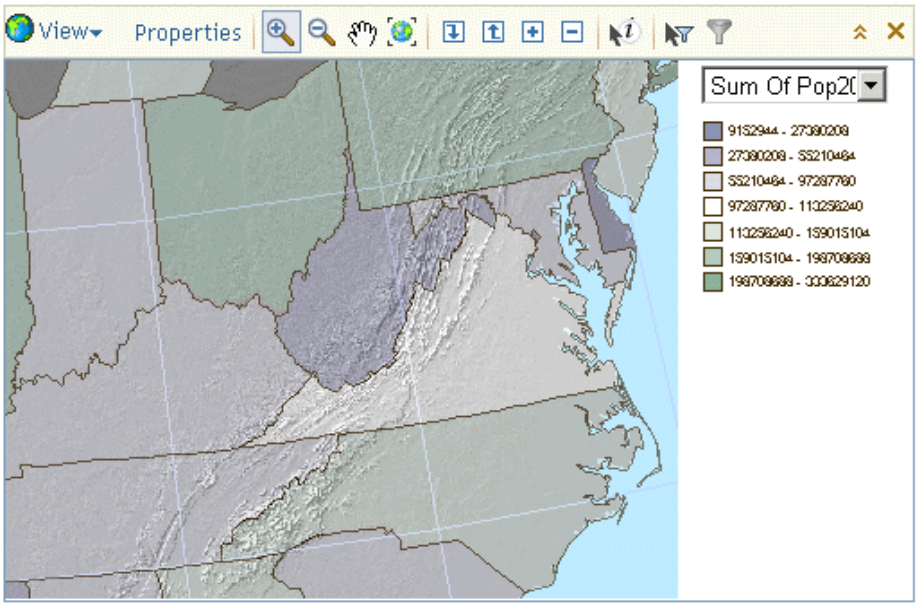
23

Configuring the ESRI Map Component

Overview of the ESRI Map Component

The ESRI map component is a SAS feature that enables you to plot your OLAP data onto an interactive geographic map.

The following image shows a geographic map within SAS Web OLAP Viewer for Java:



Depending on the SAS product that you use, you can interact with your geographic maps in several ways. For example, you can zoom, scroll the map, drill up or down, and expand or collapse regions.

The following SAS products can use the ESRI map component to display geographic maps:

- SAS Web Report Studio
- SAS Information Delivery Portal
- SAS Web OLAP Viewer for Java
- SAS Enterprise Guide

Software Requirements

The ESRI map component requires access to ESRI ArcGIS Server 9.0 with Service Pack 3, or later. Your SAS software and ESRI ArcGIS Server do not need to run on the same machine.

SAS Enterprise Guide requires ESRI ArcGIS Server 9.1 or later. In addition, you must install the ArcGIS engine runtime on the machine where SAS Enterprise Guide is running.

Define an ESRI Server

To create an ESRI server definition:

1. In SAS Management Console, select **Server Manager** and then select **Actions ► New Server** from the main menu bar.
2. In the New Server Wizard, select **ESRI Map Server**. Click **Next**.
3. Enter a name for the server in the **Name** field, and an optional description for the server definition in the **Description** field. Click **Next**.
4. Enter the appropriate information for the ESRI ArcGIS server in the **Software Version** and **Vendor** fields. Click **Next**.
5. Select a domain from the **Authentication Domain** drop-down list, or click **New** to create a new authentication domain. Enter the host name for the machine where the ESRI server is running in the **Host Name** field. Click **Next**.

   Note: It is recommended that you create a new authentication domain for the ESRI server. △
6. Confirm the information for your server definition, and then click **Finish**.

Configure Security for the ESRI Server

Each user who uses the ESRI map component must be able to access a login for the machine where ArcGIS Server is running. This login must be a member of the agsusers group, which is created automatically when you install ArcGIS Server.

For example, you might want to configure the security for the ESRI map component as follows:

1. On the machine where ArcGIS Server is running, create a new user named esriuser. Make this user a member of the agsusers group.
2. Define metadata for a special access group:
   a. In SAS Management Console, select **User Manager**. Select **New ► Group** to create a new group.

b On the **General** tab of the New Group Properties dialog box, enter the name **ESRI Users**.
c On the **Members** tab, move all of the groups that need to access the ESRI server to the **Current Members** pane.
d On the **Logins** tab, click **New** to create a new login.
e Specify the credentials for the user that you created in Step 1. Prefix your user name with your machine name (for example, **machine\esriuser**). From the **Authentication Domain** drop-down list, select the domain that you specified in the definition for the ESRI server.

In the preceding example, all of the members of the ESRI Users group can read the login metadata for esriuser and use that login to access the ESRI server.

Note: To grant ESRI access to every user who has a metadata identity, you can add your group login to the SASUSERS group. △

Define a Map Service

To create a map service definition:

1 In SAS Management Console, select **Map Service Manager**, and then select **Actions ▶ New Map Service**.
2 In the New Map Service wizard, enter the name for the map service in the **Name** field, and then select a map server from the **Map Server** drop-down list. Click **Next**.

   Note: When you click **Next**, SAS Management Console attempts to connect to the ESRI server. If the connection fails, a warning dialog box appears. Ensure that your user ID can access a login definition for the ESRI server, and that you are not logged on to SAS Management Console as an *unrestricted user*. △

3 From the **Configuration** drop-down list, select the map document that you want to use. Click **Next**.
4 In the **Layers** selection box, select the layers that you want to associate with OLAP data. Click **Next**.
5 For each layer that you selected, select the fields that you want to associate with OLAP data. Select one or more fields from the **Columns** selection box. Click **Next**.
6 Review the information that you specified, and then click **Finish**.

Configure Your OLAP Cubes for ESRI Integration

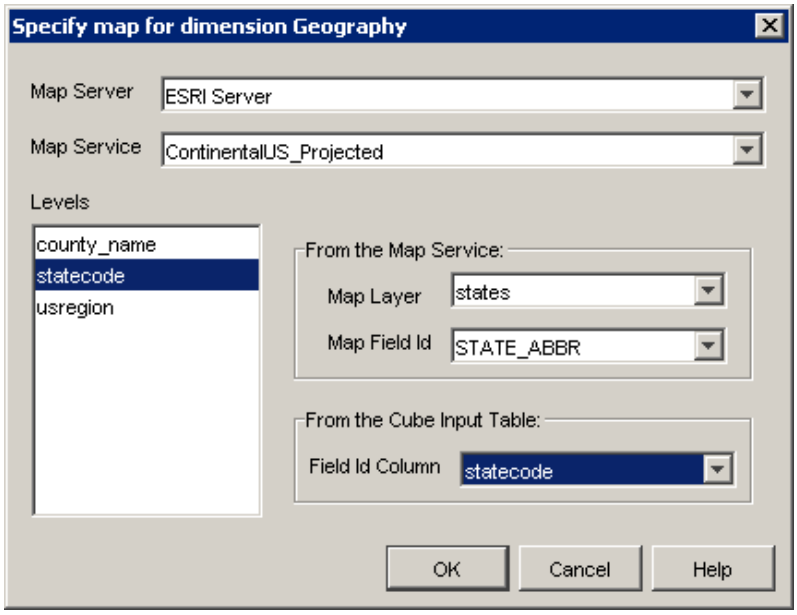
Overview of Integrating Your OLAP Data with ESRI

To use the ESRI map component with an OLAP cube, the cube must contain information about how its columns correspond to fields in the ESRI data. You can either add this information to an existing cube, or create a new cube.

Add ESRI Information to an Existing OLAP Cube

To add ESRI information to an existing OLAP cube:

1 In SAS OLAP Cube Studio, select the cube that you want to modify, and then select **Actions ▶ Edit Cube Structure**.
2 In the Cube Designer Wizard, click **Next** until you reach the Cube Designer – Dimensions page.
3 Select the dimension that contains geographic data, and then click **Modify**. In the Dimension Designer wizard, select **GEO** from the **Type** drop-down list.
4 Click **Next** until you reach the Dimension Designer – Hierarchy page, and then click **Finish**.
5 On the Cube Designer – Dimensions page, select the dimension that you edited in the preceding step, and then click **Specify Map**.
6 In the Specify Map for Dimension dialog box, select the resources that contain the ESRI data that you want to use from the **Map Server** and **Map Service** drop-down lists.



For each dimension level in the **Levels** selection box, select the level and fill in the following information:

Map Layer
specifies the ESRI map layer that corresponds to the selected OLAP dimension level.

Map Field ID
specifies an ESRI map field that identifies the regions of the selected map layer.

Field ID Column
specifies an OLAP data column that uniquely identifies the regions of the selected dimension level.

Note: For each level, the values of the OLAP data column must be identical (case sensitive) to the values of the ESRI map field. The names of the column and map field do not need to be identical. △

When you have filled out the information for each layer, click **OK** to return to the Cube Designer – Dimensions page.

7 Click **Next** until you reach the Cube Designer – Finish page. Select whether to re-create the cube immediately, and then click **Finish**.

Note: You must create your cube using SAS OLAP Cube Studio. The OLAP procedure does not support cubes with GEO dimensions. △

Specify ESRI Information in a New OLAP Cube

To specify ESRI information in a new OLAP cube:

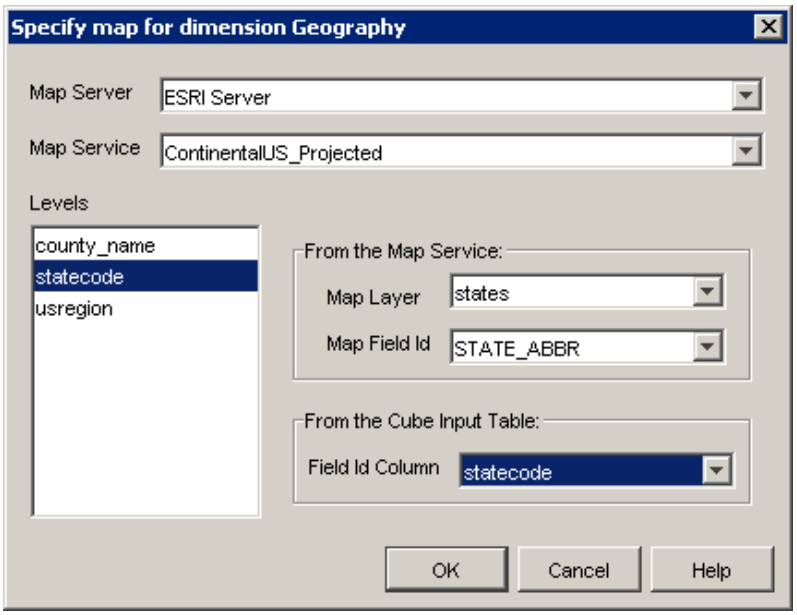
1 In SAS OLAP Cube Studio, select **File ▸ New Cube**.

2 In the Cube Designer wizard, fill out the General, Input, and Drill-Through pages as usual. See the SAS OLAP Cube Studio Help for more information about these pages.

3 On the Cube Designer – Dimensions page, create your dimensions and hierarchies as usual. For the dimension that contains geographic data, select **GEO** from the **Type** drop-down list.

When you have finished creating your dimensions, select the dimension that contains geographic data, and then click **Specify Map**.

4 In the Specify Map for Dimension dialog box, select the resources that contain the ESRI data that you want to use from the **Map Server** and **Map Service** drop-down lists.



For each dimension level in the **Levels** selection box, select the level and fill in the following information:

Map Layer
specifies the ESRI map layer that corresponds to the select OLAP dimension level.

Map Field ID
specifies an ESRI map field that identifies the regions of the selected map layer.

Field ID Column
specifies an OLAP data column that uniquely identifies the regions of the selected dimension level.

Note: For each level, the values of the OLAP data column must be identical (case sensitive) to the values of the ESRI map field. The names of the column and map field do not need to be identical. △

When you have filled out the information for each layer, click **OK** to return to the Cube Designer – Dimensions page.

5 Finish defining your cube as usual, until you reach the Cube Designer – Finish page. Select whether to create your cube immediately, and then click **Finish**.

Note: You must create your cube using SAS OLAP Cube Studio. The OLAP procedure does not support cubes with GEO dimensions. △

PART 6

Appendixes

APPENDIX

1

Users and Ports

Users

A number of operating system user accounts are needed to run SAS software. The user accounts that are created depend on the software that is installed and the platforms on which that software is installed. For information on these user accounts, see "Setting Up Required Users and Groups" in the *SAS Intelligence Platform: Installation Guide*.

Table A1.1 Special Platform Users

User	Role
SAS Administrator (sasadm)	The SAS Administrator account is used in a couple of ways. First, during setup, the SAS Configuration Wizard uses this account to connect to the metadata server from SAS Management Console. Later, you can use this account to administer the metadata and OLAP servers. In addition, because sasadm is an *unrestricted user*, you can use the account to access any metadata in the metadata repository (except for passwords, which an *unrestricted user* can overwrite, but cannot read). You should not use this account to run applications other than SAS Management Console. Also, you should use it to run SAS Management Console only when you are performing tasks that require special privileges. In particular, do not use this account in cases where passwords need to be acquired—for example, when you are defining a database library.
SAS Trusted User (sastrust)	The SAS Trusted User account is used by Web applications that use Web authentication and by the OLAP server to impersonate clients that are already authenticated on the metadata server. That is, the OLAP server and the Web applications authenticate clients; then, if a client needs to interact with the metadata server, these programs communicate with the metadata server on the client's behalf by using the sastrust account. This arrangement prevents clients from having to be authenticated multiple times and from having accounts on multiple back-end servers. The sastrust account is also used by the object spawner. When the spawner receives a request to start a workspace or stored process server, it uses this account to connect to the metadata server in order to read the appropriate server definition.
SAS Demo User (sasdemo)	The SAS Demo User account is an account that you can use to test any of the SAS clients.
SAS General Server User (sassrv)	This user is the process owner for stored process servers. In addition, both stored process servers and SAS/SHARE servers use this account when they communicate with the metadata server.
SAS Web Administrator (saswbadm)	The SAS Web Administrator account has permission to administer the portal Web application. The portal Web application shell uses the SAS Web Administrator to perform specific tasks, such as deploying portlets and creating SAS group permission trees. The SAS Web Administrator also has administrative privileges for all of the portal Web application content. The SAS Web Administrator can access a portal user's pages and share content with any SAS group.

User	Role
SAS Guest (sasguest)	The SAS Guest account is used to provide general access to your system's metadata. For example, if you have installed the SAS Information Delivery Portal, this user configures the Public Kiosk for the portal Web application.
SAS Installer (sas)	The installer uses the SAS Installer account when he or she installs and configures software on UNIX and z/OS systems. In addition, the SAS Installer is the owner of configuration directories and their contents and is the process owner for items such as the metadata server, the OLAP server, and the object spawner.
LSF Administrator (lsfadmin)	Necessary if you are installing the Platform Suite for SAS in support of either scheduling or grid computing, this user runs Platform LSF services. In addition, this user is the owner of the LSF configuration and log files and has permission to perform cluster-wide operations, edit configuration files, and reconfigure a cluster.
LSF User (lsfuser)	Necessary if you will be scheduling reports, this user runs command line programs such as the Batch Report Generation Tool.
PostgreSQL User (postgres)	Necessary if you will be using a PostgreSQL database with Xythos WebFile Server.
Pool Administrator (rpooladm)	Necessary only if you are implementing row-level security using workspace server pooling, this user handles requests for processes in the workspace server pool. The password for this account must be unique.
Puddle Log-in Account (rpoolsrv)	Necessary only if you are implementing row-level security using workspace server pooling, this user ID and its associated password must be present in the metadata group containing the SAS Web Report Studio users who will have access to the pool.
SAS Stored Process User (sasspusr)	This user is created on the stored process physical server and requires **`logon as batch job`** user rights. It should not have any access to data.
Solutions Installer (slninstl)	Necessary only if you are using SAS Solution Services, this user must exist as a user account on the data-tier machine and must belong to the machine's Administrators group and SAS Server Users group. This identity is used for loading data during the installation and can be removed (from the metadata and from the machine) once installation is complete. For more information, see the *SAS Solutions Services: System Administration Guide*.
Solutions Role Administrator (slnadm)	Necessary only if you are using SAS Solution Services, the Solutions Role Administrator is a system user that should always be a member of all solutions-created roles. It is used for cases in which a user must perform a query as a part of a larger process, but the query requires a role that the user does not generally need. For more information, see the *SAS Solutions Services: System Administration Guide*.

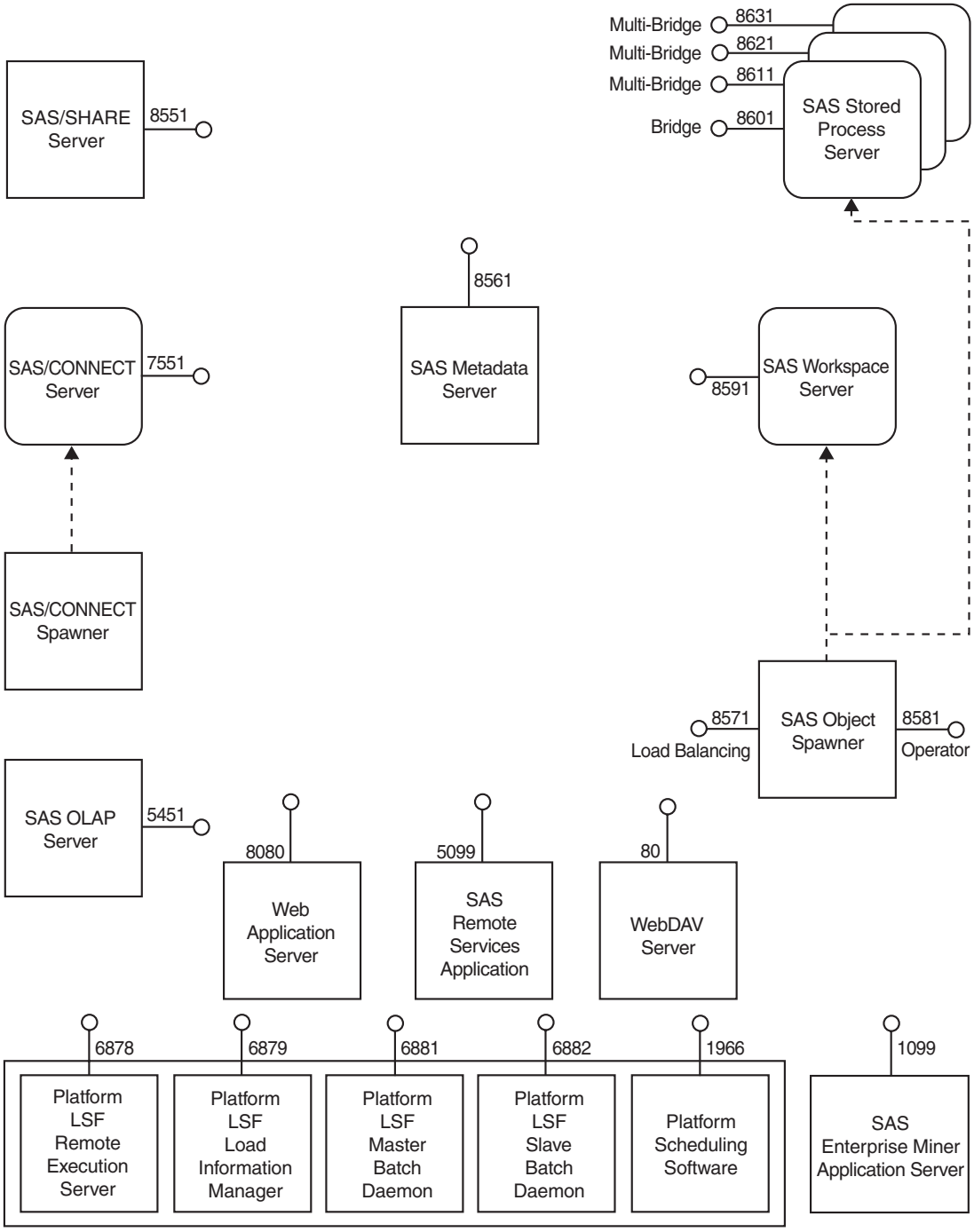
User	Role
Administrative user (srcadmin)	Necessary only if you will be promoting metadata from a source to a target machine, this user must set up on the source machine. It needs to be authenticated by the source machine's authentication provider and needs full access to the **`MetadataRepositories`** and **`ReplicationWorkArea`** directories.
SAS Replication Administrator (sasrpadm)	Necessary only if you will be promoting metadata from a source to a target machine, this user must be set up on the administration machine, for which it will be the administrator.

Default SAS Ports

The servers in the SAS Intelligence Platform communicate with clients and other servers using TCP/IP. Thus, each server listens on a particular port or ports for incoming requests. The SAS Configuration Wizard configures the servers to use a standard set of ports—unless you tell it to do otherwise.

The default port assignments are shown in the following figure.

Figure A1.1 Default Ports



If you will be using some nonstandard ports, you also need to know that the SAS Configuration Wizard reads a set of properties to get these port numbers. For example, the port number on which the metadata server will listen is stored in the property OMAPORT. The following table presents a complete list of port-number properties. (Although there are exceptions, you generally change the value of port-number properties using the SAS Configuration Wizard's Advanced Properties Editor.)

Table A1.2 Default Port Numbers, Property Names, and Descriptions

Property Name	Default Port Value	Description
OMAPORT	8561	SAS Metadata Server port. The SAS Configuration Wizard explicitly asks for this value.
CONNECT_PORT	7551	SAS/CONNECT Server port
SHAREPORT	8551	SAS/SHARE Server port
OLAP_PORT	5451	SAS OLAP Server port
SERVICES_RMI_PORT	5099	SAS Remote Services Application port
SPAWNER_OPERATOR_PORT	8581	SAS Object Spawner operator port
SPAWNER_LOADBALANCING_PORT	8571	SAS Object Spawner load balancing port
STP_PORT	8601	SAS Stored Process Server port
STP_PORT1	8611	SAS Stored Process Server port 1
STP_PORT2	8621	SAS Stored Process Server port 2
STP_PORT3	8631	SAS Stored Process Server port 3
IOM_PORT	8591	SAS Workspace Server port
DAV_PORT	80	WebDAV Server port (Apache HTTP Server)
DAV_PORT	8300	WebDAV Server port (Xythos WFS)
WEBSRV_PORT	8080	Servlet container port (Apache Tomcat)
WEBSRV_PORT	9080	J2EE Server port (IBM WebSphere Application Server port)
APP_SERVER_WEBLOGIC_ ADMIN_PORT	7501	WebLogic Administration Server port
APP_SERVER_WEBLOGIC_ MANAGED_PORT	7001	WebLogic Managed Server port
LSF_RES_PORT	6878	LSF Scheduler Remote Execution Service (RES) port*
LSF_LIM_PORT	6879	LSF Load Information Manager Service (LIM) port*
LSB_MDB_PORT	6881	LSF Master Batch Daemon (MBD) port*
LSB_SBD_PORT	6882	LSF Slave Batch Daemon (SBD) port*
JS_PORT	1966	Job Scheduler port**

Property Name	Default Port Value	Description
EM_APPSRV_PORT	1099	SAS Enterprise Miner application server port***
DBMS_PORT	3306	DBMS port
ANT_SERVER_LISTEN_PORT	17000	ANT server listen port
SERVICES_EVENT_BROKER_ ADMIN_PORT	5098	Services Event Broker administration port
SERVICES_EVENT_BROKER_ HTTP_PORT	8118	Services Event Broker HTTP port

* For information about how to change this value, see the LSF documentation.
** For information about how to change this value, see the Job Scheduler documentation.
***For information about how to change this value, see the SAS Enterprise Miner documentation.

Port Numbering in a Multiple-Level Environment

If you are setting up a multiple-level environment (for example, if your environment consists of development, test, and production levels), you will need to increment each of your port numbers by 1 for each level. For example, in setting up Lev2, use incremented port numbers as shown in Table A1.3 on page 517. In addition, you will need to set up an administration environment. Port numbers in the administration environment by convention end in 0 (zero). For example, the port number of the metadata server in the administration environment should be 8560.

Table A1.3 Incremented Port Numbers in Level 2 of Multiple-Level Environment

Server	Port Number
SAS Metadata Server	8562
SAS Object Spawner load balancing port	8572
SAS Object Spawner operator port	8582
SAS OLAP Server port	5452
SAS/CONNECT Server port	7552
SAS/SHARE Server port	8552
SAS Workspace Server port	8592
SAS Stored Process Server port	8602, 8612, 8622, 8632

Third-Party Ports

Table A1.4 Third-Party Ports Typically Used

Third-Party Product	Port Number
Tomcat	8080
Xythos	8300
Xythos or Apache DAV	80
WebLogic (Admin)	7501
WebLogic	7001
WebSphere	9080

APPENDIX

2

Understanding the SAS Configuration Environment

Overview of Understanding the SAS Configuration Environment

The SAS deployment process is based on a set of best practices for setting up a planned configuration environment. The environment has these characteristics:

- It provides a context under which all SAS servers can execute.
- It is easily extended to add development, test, and production levels (SAS Data Integration Studio only).
- It enables multiple users to work together and yet keep changes isolated.
- It eases integration and troubleshooting between multiple machines.

The Basic Concepts

Best practices for the default configuration were developed around these basic concepts:

Environment identifies an entire set of related information such as levels, SAS application servers, scripts, utilities, and documentation. The root of the environment is the configuration directory.

Level identifies the production status of the information contained within a specific area in the environment. Level 1 (Lev1) is the production level. Additional levels are supported only by SAS Data Integration Studio, which maintains an unchanged production environment. For SAS Data Integration Studio, you might have Lev1 for Production, Lev2 for Test, and Lev3 for Development.

Note: User-written scripts can take advantage of the consistent format of the level directory names. For example, a script that

searches for a particular piece of SAS code could search the lowest level (or the level where the request originates) and then continue to increment through the other levels, moving towards production. △

Change Management a general term for administering modifications to the information in the environment, such as moving information from one level to another (replication and promotion), as well as handling multiple users of a single level.

For more information, see "Setting Up Change Management" on page 248.

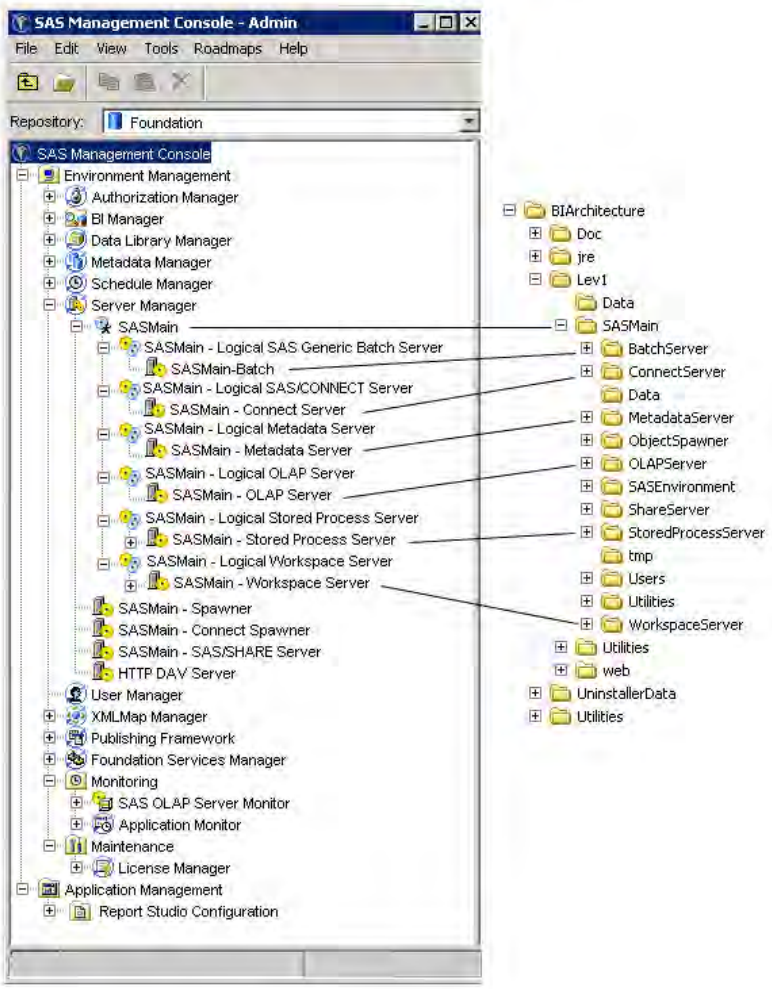
SAS Application Server a logical framework under which SAS applications execute. A SAS application server enables you to specify metadata that applies to all of the logical servers and servers that the SAS application server contains. A SAS application server provides a place to attach libraries, schemas, directories, and other resources that are available to SAS servers, regardless of the type of server. Providing this framework separate from the launching mechanism enables the administrator to deploy applications in several modes while ensuring that the applications will execute properly in that mode.

Logical Servers within SAS application server metadata, a logical server groups similar servers that run under the SAS application server. A logical server is referenced when a specific launch mode is requested, such as interactive or batch. Logical server definitions contain one or more server definitions.

Servers specific process instances that perform the requested work. A server definition contains the actual server metadata that is required to connect to a SAS server on a particular machine. Specified in the server definition are the details on where the process is executing, how a client should contact the server, and the options that describe how the server should behave.

Here is an illustration of how the metadata view of the environment matches up with the physical view of the environment on disk. In this example, **SASMain** is the name of the SAS application server. Beneath **SASMain** are the logical servers. Beneath the logical servers are the servers themselves.

Display A2.1 The Metadata View of the Environment in SAS Management Console and the Corresponding Physical View on Disk



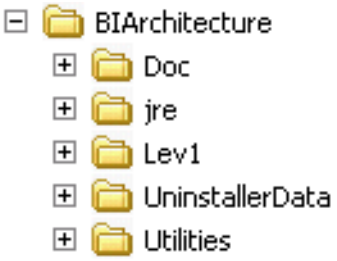
For an explanation of the **SASMain** directory structure, see "SASMain Contents" on page 524.

Note: **SASMain** is the name of the application server on the machine that hosts the SAS Metadata Server. Machines that are not hosting a SAS Metadata Server can have user-defined application server names; the default name is **SASApp**. This appendix only discusses **SASMain**. △

The Main Directory Structure

The first time that you run the SAS Configuration Wizard on a machine, it prompts you for the name of a configuration directory. The configuration directory is the root of the current environment. In this example, the configuration directory is named **BIArchitecture**.

Note: Depending on your operating system and your installed products, your site might also have subdirectories that are not shown in the following display. △



At the same level in the structure that is shown in the illustration are

Doc provides a location for you to save your own documents, which is related to the current environment.

jre contains the Java Runtime Environment, which interprets Java code.

Lev1 as explained in "The Lev1 Directory Structure" on page 522, Lev1 identifies information that is at production status. Beneath a level, you can create any number of directories that are unique to the level in which they are contained. As a best practice, you should name directories consistently across levels. For example, if you create a **`Lev2`** directory, it should also contain a **`SASMain`** directory with the same contents. When the content is consistent across levels, then promotion and replication are easier to manage.

Note: Additional levels are only supported by SAS Data Integration Studio, which maintains an unchanged production environment. For SAS Data Integration Studio, you might have Lev1for Production, Lev2 for Test, and Lev3 for Development. △

UninstallerData contains utilities to uninstall the current configuration (that is, uninstall the entire directory structure). On Window systems, this functionality is available from **Start ► Settings ► Control Panel ► Add or Remove Programs**.

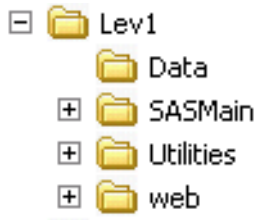
Utilities contains utilities that are used at the environment level. For example, this directory might contain scripts to replicate and promote between levels.

The Lev1 Directory Structure

For all configurations, the **`Lev1`** directory contains **`Data`** and **`SASMain`** subdirectories. If you are configuring a middle tier machine, then the **`Lev1`** directory also includes a **`web`** subdirectory.

Note: Depending on your operating system and your installed products, your site might also have subdirectories that are not shown in the following display. △

Display A2.2 Example of a Lev1 Directory Structure on a Windows System



Here are descriptions of the folders shown in the display:

Data contains data that is specific to the current level but that is shared across SAS application servers. When you use the utilities that are supplied in the **Utilities** directory, the **Data** directory is replicated and promoted to other levels by default.

SASMain on machines with an installed SAS Metadata Server (and perhaps other SAS servers), this directory represents a SAS application server that contains these items:

- a SAS configuration file named **sasv9.cfg**
- a SAS Metadata Server configuration file named **omaconfig.xml**
- a subdirectory for each logical server that is installed for the SAS application server in the current level
- a **Data** subdirectory that contains data that is specific to the current SAS application server but that is shared across all logical servers defined within the SAS application server
- a **Users** subdirectory for managing multiple users working in the SAS application server in the current level
- any other utility files that are used to manage the SAS application server

For more information about the contents of **SASMain** on a server-tier machine, see "SASMain Contents" on page 524.

Note: On Web-tier only machines, this directory is empty. △

utilities contains utilities used in deploying Web applications. Among them is a sample SAS application named **LoadPlannedStoredProcessSamples.sas** that can be used to load stored process samples into the metadata repository.

web for machines with installed Web components, this directory contains these items:

- a servlet container (or J2EE) start-up script named **startServletContainer**
- a **Deployments** subdirectory that contains information related to deploying Web applications, including policy files and service definition files
- a **Utilities** subdirectory for any utilities relevant to installed Web applications.
- a **webapps** subdirectory that contains the Web applications archive (WAR) files for your Web applications such as SAS Information Delivery Portal.

For more information about the contents of **web**, see "Web Contents" on page 528.

SASMain Contents

On machines with an installed SAS Metadata Server (and perhaps other SAS servers), this directory represents a SAS application server that contains configuration files, a variety of subdirectories, and utilities that support the SAS application server. On Web-tier only machines, this directory is empty.

About the SAS Configuration File

Regardless of the mechanism used to invoke SAS, all SAS sessions must begin with the same SAS configuration file. This practice has the following advantages:

- It enables the same SAS application to be executed via any of the SAS server technologies that support code submission, including SAS Workspace Servers, SAS/CONNECT servers, and batch processes.
- It ensures that the resources that are required to execute a SAS application are properly configured.

Depending on your operating system and your installed products, the **sasv9.cfg** file might contain the following information:

- the path to the directory that contains **sasv9.cfg** file
- the path to the root directory for the current SAS application server
- the path to the directory that will contain the SAS formats used by the current SAS application server
- the path to the directory that will contain the SAS macros used by the current SAS application server
- a list of the locales used during data cleansing and the location of the SAS Data Quality Server setup file
- SAS Metadata Server information such as port number, foundation repository name, server name, and connection protocol.

For example, the **sasv9.cfg** file for the **BIArchitecture** sample environment might have this content on a Windows system:

```
-config "C:\Program Files\SAS\SAS 9.1\sasv9.cfg"

-sasinitialfolder "c:\SAS\BIArchitecture\Lev1\SASMain"

-set library ("SASEnvironment/sasFormats")

-sasautos ("SASEnvironment/sasMacro" SASAUTOS)

-dqlocale (ENUSA)

-dqsetuploc "dqsetup.txt"

-metaserver "abcorp"
-metaport 8561
-metarepository "Foundation"
-metaprotocol BRIDGE
```

About the SAS Metadata Server Configuration File

On machines with installed SAS servers, the **SASMain** directory contains the **omaconfig.xml** file. Here are some of the SAS Metadata Server settings that are contained in the file:

- values for the libref and path of the metadata server's repository manager
- name or location of the adminUsers.txt and trustedUsers.txt files
- the engine used to create the repository manager
- the number of concurrent threads to run on the server.

Here is an example of an **omaconfig.xml** file:

```
<OMAconfig>
  <OMA
    ADMINUSERS="MetadataServer/adminUsers.txt"
    TRUSTEDUSERS="MetadataServer/trustedUsers.txt"
    MAXACTIVETHREADS="3"/>

  <RPOSMGR PATH="MetadataServer/rposmgr"/>

</OMAconfig>
```

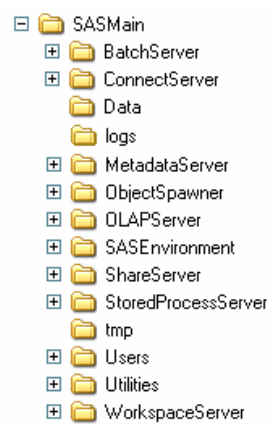
Note: For information about the **omaconfig.xml** file, see the *SAS Metadata Server: Setup Guide* available at **http://support.sas.com/rnd/eai/openmeta/v9/setup**. △

The SASMain Subdirectories

Depending on your operating system and your installed products, the **SASMain** application server directory might contain these subdirectories.

Note: Your site might also have subdirectories that are not shown in the following display. △

Display A2.3 Example of a SASMain Directory Structure on a Windows System



Here are descriptions of the folders that are shown in the display:

BatchServer contains log files and SAS code that are associated with the current SAS application server running in batch mode.

ConnectServer the physical directory that corresponds to a server definition in a SAS Metadata Repository. This server provides single-user server

functionality for Remote Library Services. (Remote Library Services provides transparent access to remote data libraries for moving data through the network as the local SAS session requests it.) This directory contains these items:

- one or more start-up scripts that are appropriate for the operating system.
- the **logs** subdirectory that stores any log files generated by this server invocation.
- any utility files that are used to manage the server.

Data
contains data that is specific to the current SAS application server but that is shared across all logical servers defined within the SAS application server.

logs
contains any log files that are specific to the current SAS application server.

MetadataServer
the physical directory that corresponds to a server definition in a SAS Metadata Repository. This multi-user server enables users to read metadata from or write metadata to one or more SAS metadata repositories. This directory contains these items:

- one or more start-up scripts that are appropriate for the operating system. Typically, the server is started as a service on Windows. Before the start command is executed, the script queries the metadata server for information, such as a port number, that is required for the launch.
- three security-related files: adminUsers.txt, trustedUsers.txt, and trustedPeers.xml (see "Security-Related Files" on page 14).
- the **MetadataRepositories** subdirectory that contains the production copy of the foundation repository. You can set operating level permissions on this directory.
- the **ReplicationWorkArea** subdirectory that is used to store work files and a backup copy of the repository during replication.
- The **rposmgr** subdirectory that contains the files that are used to manage the repositories in the current SAS application server.
- the **sasuser** subdirectory is a SAS library that contains SAS catalogs that enable you to tailor features of SAS for your needs. SAS assigns the SASUSER library at invocation.
- the **logs** subdirectory that stores any log files generated by this server invocation.
- any other utility files that are used to manage the server.

ObjectSpawner
the physical directory that corresponds to a server definition in a SAS Metadata Repository. This directory contains the start-up scripts needed to run a process-spawning service that instantiates SAS Workspace Servers and load-balanced SAS Stored Process Servers. Typically, the spawner is started as a service on Windows.

Note: During the deployment process, you are given the option to install and configure the spawner. However, if you are installing SAS Workspace Servers or SAS Stored Process Servers, then the spawner is automatically installed because those servers must be started by a spawner. △

OLAPServer the physical directory that corresponds to a server definition in a SAS Metadata Repository. This server provides access to cubes, which are logical sets of data that are organized and structured in a hierarchical, multidimensional arrangement. This directory contains these items:

- one or more start-up scripts that are appropriate for the operating system. Typically, the server is started as a service on Windows.
- the **sasuser** subdirectory is a SAS library that contains SAS catalogs that enable you to tailor features of SAS for your needs. SAS assigns the SASUSER library at invocation.
- the **logs** subdirectory that stores any log files generated by this server invocation.
- the **work** subdirectory that stores any other files generated by this server invocation.
- any other utility files that are used to manage the server.

SASEnvironment contains the elements that comprise the run-time environment for the SAS code when running on the current SAS application server. It contains the following subdirectories, which are specified in the **sasv9.cfg** file (see "About the SAS Configuration File" on page 524):

- the **QltyKb** subdirectory that contains one or more locales. Each locale contains definitions and other information that is referenced by the SAS Data Quality Server software during data analysis and data cleansing.
- the **SASCode** subdirectory that contains a **Jobs** directory that stores all the SAS code for each job in the environment, and a **Steps** directory that stores all the SAS code for each step inside a specific job. By sorting the SAS code in this way, developers on a project can share source code. You can set operating level permissions on these directories.
- the **SASFormats** subdirectory that contains the SAS format and informat catalogs that are necessary for the data and the code that is accessed through the current SAS application server. This information is available regardless of which SAS invocation technologies deploy the SAS code.
- the **SASMacro** subdirectory that contains the SAS Autocall macros that are invoked via SAS code that executes through the current SAS application server. Like SAS formats and informats, this information is available regardless of which SAS invocation technologies deploy the SAS code.

ShareServer the physical directory that corresponds to a server definition in a SAS Metadata Repository. This multi-user data server enables two or more clients to write to the same SAS file at the same time. This directory contains these items:

- one or more start-up scripts that are appropriate for the operating system. Typically, the server is started as a service on Windows.
- the **logs** subdirectory that stores any log files generated by this server invocation.
- any other utility files that are used to manage the server.

StoredProcess Server — the physical directory that corresponds to a server definition in a SAS Metadata Repository. This server executes stored processes. A stored process is a SAS program that is stored on a server and can be executed as required by requesting applications. This directory contains these items:

- the **logs** subdirectory that stores any log files generated by this server invocation.
- the **sasuser** subdirectory that stores any files relevant to the sasuser
- the **data** subdirectory that stores any data files generated by this server invocation.
- any other utility files that are used to manage the server.

tmp — contains temporary working files.

Users — contains a **TemplateUser** subdirectory that should be copied and renamed for each developer who will be working in the current SAS application server. Each user-specific directory contains these subdirectories:

- the **Data** subdirectory that contains temporary output as the result of testing changes in the user directory.
- the **SASEnvironment** subdirectory that contains a structure that matches the **SASEnvironment** subdirectory located under the **SASMain** directory, so that users can test changes to the SAS application server.

Utilities — contains utilities that are used at the SAS application server level. For example, this directory might contain scripts to backup the SAS application server and import users into the metadata.

WorkspaceServer — the physical directory that corresponds to a server definition in a SAS Metadata Repository. This server fulfills client requests for specific SAS sessions. This directory contains these items:

- the **logs** subdirectory that stores any log files generated by this server invocation.
- any other utility files that are used to manage the server.

Web Contents

Depending on your operating system and your installed products, the **web** directory might contain the following subdirectories:

Deployments — contains information related to deploying Web applications, including policy files, service configuration files, and service deployment files.

webapps — contains the Web applications archive (WAR) files for your Web applications such as SAS Information Delivery Portal. These WAR files are actually JAR files that contain all of the files that make up the Web application, including servlets, JavaServer Pages, and HTML documents.

Note: If you are using the Tomcat servlet container to execute your Web applications, the SAS Configuration wizard has already copied these WAR files to Tomcat's **webapps** directory. If you are using a J2EE application server for this purpose, you must manually deploy these files to your server's **webapps** directory.

Follow the instructions in your vendor's documentation for deploying an application. △

Utilities contains any utilities related to deploying Web applications.

Default Directory Permissions

For UNIX and z/OS systems, here are the default permissions for the directories, files, and scripts that are created in a planned installation.

Note: There are no default permissions set on Windows systems. All directories, files, and scripts are owned by the user who performs the installation. △

Table A2.1 Default Directory Permissions for UNIX and z/OS

Directories/Files/ Scripts	The `sas` user ID	The `sas` User Group	All Users
Server-specific directories, files, and scripts, except for the **`StoredProcessServer`** directory	Read, write, execute	No access	No access
`Lev1/SASMain/ StoredProcessServer`	Read, write, execute	Read, write, execute	No access
`Lev1/SASMain/Data`	Read, write, execute	Read, write, execute	Read, write, execute
All other directories and files	Read, write, execute	Read, execute	Read, execute
All other scripts	Read, write, execute	Read, execute	Read, execute

APPENDIX

3

Upgrading a SAS 9.1 or 9.1.2 System to a SAS 9.1.3 System

Overview of Upgrading a SAS 9.1 or 9.1.2 System to a SAS 9.1.3 System

This appendix explains how to upgrade from a 9.1 or 9.1.2 business intelligence system to a 9.1.3 system. For all platforms except z/OS, see "Your Two Options for Upgrading" on page 532 for pointers to the upgrade instructions. For instructions on upgrading a z/OS system, see "Upgrading a SAS 9.1 or 9.1.2 z/OS System to a SAS 9.1.3 System" on page 547.

The following subsections discuss

- the assumptions that we are making about your current installation and the software that you plan to install
- the tasks that you must perform prior to following the steps in this appendix
- your options for upgrading your non-z/OS machines.

Assumptions

We are making the following assumptions about your system:

- Your initial installation was a Project installation. (A Project installation is a planned installation, similar to an Advanced or a Personal installation of the 9.1.3 software.)
- You are not installing any new products. You are simply upgrading your existing environment.

Before You Begin the Upgrade

Before you actually begin the upgrade, perform the following tasks:

1. Plan a time to do the upgrade.
2. If you will be performing an Advanced installation of the 9.1.3 software, set up a new project directory. You will need a 9.1.3 Software Installation Data file(s) and a 9.1.3 planning file. The planning file must have the same name as your 9.1.2 planning file and must contain the same content.

   A recommended best practice is to generate your 9.1.3 plan by taking the following steps in the planning tool:

   a. Locate your 9.1.2 plan in the planning tool, and edit that plan in order to change its name.
   b. Make a copy of the 9.1.2 plan.
   c. Give the copy of the 9.1.2 plan the name that was originally assigned to your 9.1.2 plan, and set its **SAS Release** to 9.1.3.
   d. Save the 9.1.3 plan.

   Note: It is also a good practice to save a copy of old **plan.xml** files. Rename each old plan something similar to **plan_912_10APR04.xml**. △
3. Back up your current configuration.

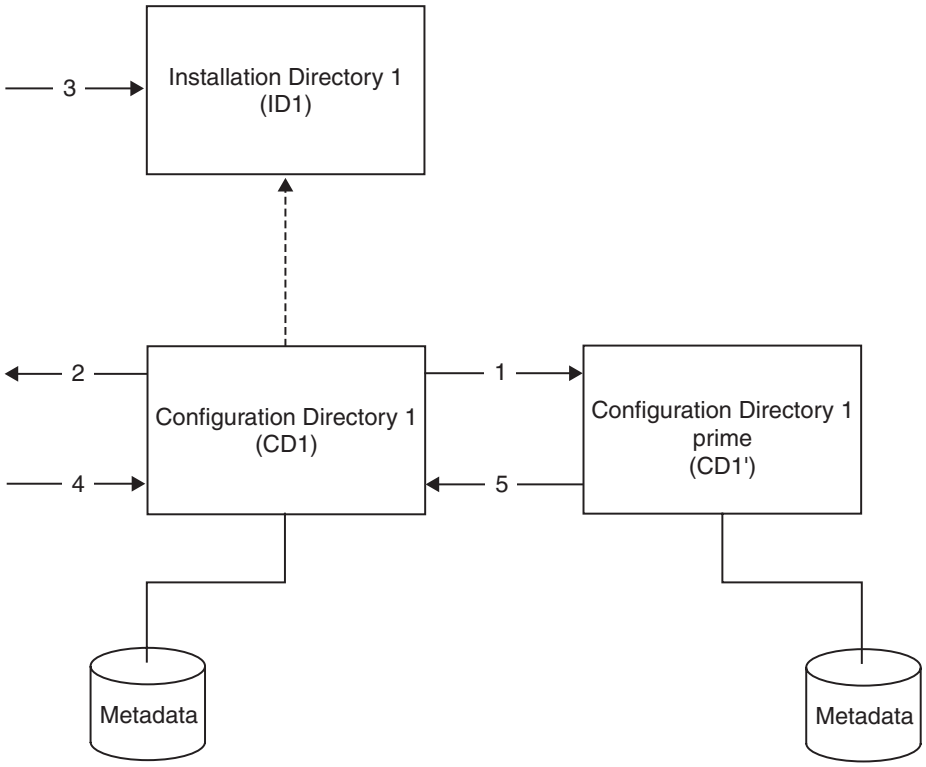
Your Two Options for Upgrading

You have two options for upgrading your system. You can simply replace your 9.1 or 9.1.2 software with your 9.1.3 software, or you can temporarily maintain separate 9.1[.2] and 9.1.3 systems so that you can test the 9.1.3 software before taking your 9.1 or 9.1.2 system out of production. For information about these options, see the following sections:

Upgrading in Place

The upgrade-in-place procedure is designed to allow you to minimize the amount of time involved in upgrading. The procedure is illustrated in the following figure.

Figure A3.1 Upgrading in Place



For each machine in your system, stop any servers that are running on that machine, and then perform the following tasks. (The following numbered steps correspond to the numbers in the previous figure.) Start with your metadata-server machine. Then proceed to other server-tier machines, middle-tier machines, and client machines, in that order.

1. Copy your 9.1[.2] configuration directory (CD1) to a backup location (CD1'). Also, back up any custom portlets that are located in the directory *SAS-install-dir*`\Web\Portal2.0.1\DeployedPortlets`.
2. Uninstall the 9.1[.2] configuration by running the appropriate uninstall program. On Windows systems, run the executable *path-to-config-dir*`\UninstallerData\Uninstall SAS Configuration Wizard.exe`. On UNIX systems, run *path-to-config-dir*`/UninstallerData/Uninstall_SAS_Configuration_Wizard`. If necessary, also delete the configuration directory from the file system.
3. Install the 9.1.3 software—including the 9.1.3 clients—in ID1. There is no need to uninstall the 9.1[.2] software first.

For third-party software, such as the BEA WebLogic Server, make sure that you have the supported version of the product installed. You can find information about the supported version of each such product at **`http://support.sas.com/thirdpartysupport`**.

4 Run the SAS Configuration Wizard, and specify the original configuration directory (CD1) as your configuration directory.

CAUTION:
Do not follow the HTML instructions that are generated by the SAS Configuration Wizard. These generated instructions are meant for a new configuration. Follow the instructions that are referred to in step 5 instead. (When the SAS Configuration Wizard finishes running, it starts SAS Management Console, and you are asked whether you want to create a new metadata repository. You do not need to create this repository; you will later copy your original metadata repository from CD1' to CD1.) △

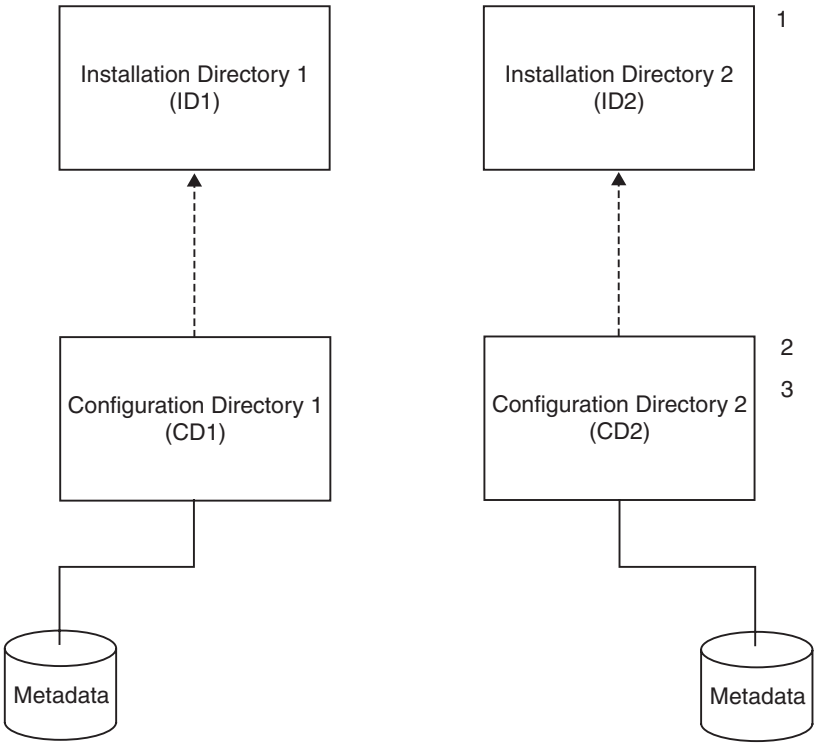
5 Execute the manual upgrade instructions that are documented in"Upgrading Each Machine" on page 536. Choose the correct set of instructions for the machine that you are currently working on.

Upgrading After Testing in a Test Environment

If you want to maintain both 9.1[.2] and 9.1.3 business intelligence systems for a testing period, before you make the 9.1.3 system your production system, see the following sections.

Building a Test System

Build a test system by following the steps that are shown in the following figure.



For each machine in your configuration, perform the following steps. (Start with your metadata-server machine. Then proceed to other server-tier machines, the middle-tier machine, and client machines, in that order.)

1 Install the 9.1.3 software on a different machine of the same type (or possibly on the same machine) in ID2.

For each Windows machine in your installation, you must install the SAS 9.1.3 software on a different Windows machine. For each UNIX machine, we recommend that you install the SAS 9.1.3 software on a different UNIX machine. However, it is possible to install the SAS 9.1.3 software on the same UNIX machine as SAS 9.1[.2] software.

CAUTION:
If you choose to install SAS 9.1[.2] and SAS 9.1.3 software on the same UNIX system, be aware that when you installed your SAS 9.1[.2] software, a preference file was created that will prevent you from installing the SAS 9.1.3 software in a new location. Therefore, before you can install SAS 9.1.3 software on a host where SAS 9.1[.2] software is installed, you must rename this preferences file. The **`vpd.properties`** file is located either in the **`sas`** user's home directory (Solaris and HP-UX IPF) or in **`/usr/lib/objrepos`** (AIX). Rename this file **`vpd.properties91`** or **`vpd.properties912`**.

Note that renaming this file will prevent you from uninstalling your SAS 9.1[.2] software later. To uninstall that software, perform the following steps:

a Rename the SAS 9.1.3 **`vpd.properties`** file to **`vpd.properties913`**.
b Change the name of the SAS 9.1[.2] properties file back to **`vpd.properties`**.
c Uninstall your SAS 9.1[.2] software.
d Delete the SAS 9.1[.2] properties file.
e Change the name of the SAS 9.1.3 properties file back to **`vpd.properties`**.

△

For third-party software, such as the BEA WebLogic Server, make sure that you install the supported version of the product. You can find information about the supported version of each such product at **`http://support.sas.com/thirdpartysupport`**.

2 Run the SAS Configuration Wizard to produce CD2. If you are installing a 9.1.3 system side by side with a 9.1.[.2] system, be sure to use different port numbers than the port numbers that you used to configure your 9.1[.2] system.

CAUTION:
Do not follow the HTML instructions that were generated by the SAS Configuration Wizard. These generated instructions are meant for a new configuration. Follow the instructions in step 3 instead. (When the SAS Configuration Wizard finishes running, it starts SAS Management Console, and you are asked whether you want to create a new metadata repository. You do not need to create this repository; you will later copy your original metadata repository from CD1 to CD2.) △

3 Execute the manual upgrade instructions that are documented in the section "Upgrading Each Machine" on page 536. Choose the correct set of instructions for the machine that you are currently working on.

Then test the entire system.

Updating Your Production System

After you are satisfied with your testing, you will need to update your production system. To do this, you follow the same steps that you follow to upgrade a system in place—see "Upgrading in Place" on page 533—with the following caveat. Depending on where you installed your new software in building your test system, you might not need to reinstall the 9.1.3 software. Instead, you might be able to use the existing installation from your test system. Your hardware deployment and standard operating procedures will dictate whether this is feasible. For example, you might be able to use UNIX NFS mount points to accomplish this.

CAUTION:
Do not copy your test system's configuration directory and metadata repositories. In the process of creating that system, you made several edits to that directory that might or might not be appropriate for your production system. △

Upgrading Each Machine

Perform the steps in each of the following sections whose title describes the machine on which you are working.

- □ If your metadata server is configured on the machine, follow the directions in "Metadata Server Host" on page 537.
- □ If the machine hosts SAS servers—but not the metadata server—follow the directions in "Server Machine, No Metadata Server" on page 538.
- □ If your middle-tier software, for example your J2EE application server, is installed on the machine, follow the directions in "Middle-Tier Machine" on page 539.

You might need to perform the steps in two sections. For instance, if you have installed the SAS Intelligence Platform on a single machine, you should consult the

first and third sections. In such a case, follow the directions in the first relevant section before performing those in the second relevant section.

Metadata Server Host

On the machine where the SAS Metadata Server is configured, perform the steps in the following sections:

1. "Stop the Metadata Server" on page 537.
2. "Copy Files from the Original Configuration Directory" on page 537.
3. "Start the Metadata Server" on page 537.
4. "Upgrade Your Metadata" on page 537.
5. "Set Up the SAS/CONNECT Spawner (Windows only)" on page 538.
6. "Reestablish Existing Schedule Flows" on page 538.

Stop the Metadata Server

The SAS Configuration Wizard automatically starts the metadata server. To make sure that there are no file contention issues, stop the metadata server before you copy any files into the new configuration directory.

Copy Files from the Original Configuration Directory

In the typical use of the configuration, you will have created and modified files in your original configuration directory. For example, you might have added SAS system options to the configuration file *path-to-config-dir*`\Lev1\SASMain\sasv9.cfg`. You need to determine which of these new or modified files you should copy to the new configuration. Also, you need to move your original metadata repository to the new configuration. Use the following directions:

- If you changed a host name, pathnames, or port numbers that are associated with the configuration, you need to ensure that the new configuration directory contains the correct information.
- If you added tuning options or other options to your original configuration, you should add those options to (or maintain those options in) the new configuration.
- You must always copy the SAS Metadata Server files to the new configuration. These files are contained in the `MetadataServer/reposmgr` and `MetadataServer/MetadataRepositories` directories. You should place these files in the same location in the new configuration directory.

Start the Metadata Server

You need to restart the metadata server before continuing this procedure.

Upgrade Your Metadata

Using SAS Management Console, upgrade your existing metadata repositories:

1. Log on to SAS Management Console as an unrestricted user.
2. Expand the Metadata Manager node in the SAS Management Console tree.
3. Right-click the Active Server icon, and select `Upgrade Metadata` from the pop-up menu.

All non-project repositories on the active server will be upgraded.

Note: If you have changed a host name, pathnames, or port numbers that are associated with the configuration, you will need to ensure that the metadata definitions contain the new information. To validate host names and port numbers, use the Server Manager plug-in to SAS Management Console: select the physical server definition; then, examine the properties of the connection shown in the right-hand view of the console. To verify pathnames, use the same plug-in to edit the properties of the physical server definition. △

Note: If you are working on a Windows machine, have registered local users (*host-name**user-ID*) in the metadata in your original configuration, and are running the new configuration on a different machine, you must modify the host-name qualifier on each user ID. You do this using the User Manager plug-in to SAS Management Console. Edit users and groups as appropriate, changing the user IDs on the **Logins** tab. △

Set Up the SAS/CONNECT Spawner (Windows only)

If you plan to run a SAS/CONNECT server on this machine, you should run the script that was supplied by SAS that sets up the SAS/CONNECT spawner to run as a service. You can find this script in the directory *path-to-config-dir*`\Lev1\SASMain\ConnectServer`. It is called `InstallConnectServer.bat`.

Reestablish Existing Schedule Flows

When you upgrade your metadata, existing scheduled flows are preserved. Although flows that you create in 9.1.3 are represented slightly differently in the metadata, your 9.1[.2] flows will continue to work. The 9.1.3 Schedule Manager supports both types of flows. All you need to do is to reschedule the previously scheduled jobs in the Schedule Manager.

Note: If you have not installed your 9.1.3 scheduling software yet, wait to perform this step until after you have installed that software. △

For each flow that you want to schedule, perform the following steps:

1. Right-click the flow, and select **Schedule Flow** from the pop-up menu.
2. Click **Yes** in the Reschedule confirmation dialog box.
3. Choose the trigger to start the flow, and click **OK**. This will schedule the flow with Platform Computing's JobScheduler.

Server Machine, No Metadata Server

Follow the steps in this section on all machines that host SAS servers, but do not host the metadata server.

Reapply Customizations from the Original Configuration Directory

In the typical use of the configuration, you will have created and modified files in your original configuration directory. For example, you might have added SAS system options to the configuration file *path-to-config-dir*`\Lev1\SASMain\sasv9.cfg`. You need to determine which of these new or modified files you should copy to the new configuration. Use the following rules to make this decision:

- If you changed a host name, pathnames, or port numbers that are associated with the configuration, you need to ensure that the new configuration directory contains the correct information.

- If you added any tuning or other options to your original configuration, you should add those options to (or maintain those options in) the new configuration.

Set Up the SAS/CONNECT Spawner (Windows only)

If you plan to run a SAS/CONNECT server on this machine, you should run the script that was supplied by SAS that sets up the SAS/CONNECT spawner to run as a service. You can find this script in the directory *path-to-config-dir*`\Lev1\SASMain\ConnectServer`. It is called `InstallConnectServer.bat`.

Start Your Servers

You can now start the servers in your configuration. For details on how to perform this step, see the `instructions.html` file from your 9.1[.2] installation.

Middle-Tier Machine

You should perform the steps that are listed in the following sections on a machine where a middle tier is configured (for example, where the SAS Web Infrastructure Kit is installed). Note that a couple of these steps apply only when you are upgrading from 9.1 to 9.1.3 (not from 9.1.2 to 9.1.3).

1. "Reapply Customizations from the Original Configuration Directory" on page 539.
2. "Prepare to Reapply Customizations to Web Applications" on page 540.
3. "Upgrade User and Group Definitions (9.1 to 9.1.3 only)" on page 540.
4. "Define the SAS Foundation Services in the Metadata" on page 540.
5. "Update the SAS Web Infrastructure Kit Metadata" on page 541.
6. "Copy Custom Portlets to the Installation Directory" on page 541.
7. "Migrate Your WebDAV Data to a 9.1.3 Test System (SAS Information Delivery Portal)" on page 541.
8. "Migrate Your WebDAV Data to a 9.1.3 Test System (SAS Web Report Studio)" on page 542.
9. "Migrate Your WebDAV Data in the 9.1.3 Production System (SAS Information Delivery Portal)" on page 543.
10. "Migrate Your WebDAV Data in the 9.1.3 Production System (SAS Web Report Studio)" on page 545.
11. "Start the SAS Services Application" on page 546.
12. "Deploy Your Web Applications" on page 546.

Note: If your metadata server is configured on this machine, you can skip step 1. You should have performed that step earlier. △

Reapply Customizations from the Original Configuration Directory

In the typical use of the configuration, you will have created and modified files in your original configuration directory. You need to determine which of these new or modified files you should copy to the new configuration. Use the following rules to make this decision:

- If you changed a host name, pathnames, or port numbers that are associated with the configuration, you need to ensure that the new configuration directory contains the correct information.

- If you added any tuning or other options to your original configuration, you should add those options to (or maintain those options in) the new configuration.

Prepare to Reapply Customizations to Web Applications

If you have customized any files that are part of a SAS Web application, such as SAS Web Report Studio or SAS Information Delivery Portal, you must back up the files that you have modified. You should restore these files after you redeploy your Web applications as described in "Deploy Your Web Applications" on page 546.

Upgrade User and Group Definitions (9.1 to 9.1.3 only)

In SAS Management Console, add the SAS Web Administrator (**saswbadm**) to the SAS System Services group.

1. Select the User Manager plug-in to SAS Management Console. You will see a list of groups and users on the right side of the GUI.
2. Right-click the SAS System Services group, and select **Properties** from the pop-up menu. A properties dialog box appears.
3. Select the **Members** tab.
4. Move the SAS Web Administrator from the **Available Members** list to the **Current Members** list.
5. Click **OK** to save this group information.

Define the SAS Foundation Services in the Metadata

You need to load the 9.1.3 SAS Foundation Services into the metadata before anyone uses the SAS 9.1.3 Web applications. To do this, you use the Foundation Services Manager plug-in to SAS Management Console.

1. Right-click the Foundation Services Manager, and select **Import Service Deployment** from the pop-up menu. An Import Service Deployment dialog box appears.
2. Remove any existing service-deployment import files by selecting them and clicking the **Remove** button.
3. Click **Add** in this dialog box. A file browser (an Open dialog box) appears.
4. Use the file browser to display the contents of the directory *path-to-config-dir***\Lev1\web\Deployments\Portal**.
5. Select the following files and click **Open**.
   - **sas_services_idp_remote_omr.xml**
   - **sas_services_idp_local_omr.xml**

   The names of these files will be displayed in the Import Services Deployment dialog box.
6. In the same way, add the file *path-to-config-dir***\Lev1\web\Deployments\WebReportStudio***server-name***_sas_pfs_queryandreporting.xml** to the list.
7. Click **OK** in the Import Service Deployment dialog box.

Note: If you have installed the SAS BI Web Services for Java, you should use also use the Foundation Services Manager to import the file *path-to-config-dir***\Lev1\web\Deployments\WebServices\sas_services_websvc_local_omr.xml**. △

Update the SAS Web Infrastructure Kit Metadata

You must also update the metadata for the SAS Web Infrastructure Kit before anyone uses the SAS 9.1.3 Web applications. In the directory *SAS-install-dir*`\Web\Portal2.0.1\OMR`, you will find a set of SAS programs. You must run a subset of these programs—in the order indicated in the following list—and check the log for any errors. There should be no errors.

If you are upgrading a 9.1.2 system to 9.1.3, run the following programs:

1. **`Remove912PreferenceDefs.sas`**
2. **`LoadDefaultPreferences.sas`**

If you are upgrading a 9.1 system to 9.1.3, run the following programs:

1. **`LoadDefaultPreferences.sas`**
2. **`LoadPreferencesConnection.sas`**
3. **`LoadThemeConnection.sas`**

On Windows systems, you can run these programs from the Windows Explorer. Right-click the SAS file, and select **`Batch Submit with SAS 9.1`** from the pop-up menu. After executing each SAS program, check the corresponding log file. (This file will have the same name as the program, but will end with a **`.log`** extension.) Open each log file and search for the string "ERROR:". You should only find program statements that contain this string, not log messages.

For more details about this step, see "Install Metadata" in the *SAS Web Infrastructure Kit Deployment Notes*.

Copy Custom Portlets to the Installation Directory

Portlets are mini-applications that run inside the framework that is provided by the SAS Web Infrastructure Kit. If you have created any custom portlets, you must perform one of the following tasks:

- If you are building a test system, copy the portlets from the directory *SAS-install-dir*`\Web\Portal2.0.1\DeployedPortlets` on your 9.1[.2] system to the same directory on your 9.1.3 system.
- If you are replacing your production system, copy the custom portlets from their back-up location or your test system to your 9.1.3 production system.

Note: You do not need to copy portlets that were supplied with your SAS software. You need to copy only portlets that you developed yourself. △

Migrate Your WebDAV Data to a 9.1.3 Test System (SAS Information Delivery Portal)

Important Note: Follow the instructions in this section only if both of the following are true:

- you are moving your 9.1[.2] WebDAV data to a 9.1.3 test system
- you are using SAS Information Delivery Portal with its WebDAV features enabled.

If you are also using SAS Web Report Studio, see "Migrate Your WebDAV Data to a 9.1.3 Test System (SAS Web Report Studio)" on page 542 as well.

To move your 9.1[.2] WebDAV data to a 9.1.3 test system, follow these instructions:

1. Install Xythos WebFile Server on the middle-tier host in your test system.
   a. Create new databases or database schemas to hold your current WebDAV data. Using your database vendor's documentation, create one database or schema to hold your current WFS Global Schema, and create a database or schema for each of your existing Document Stores.

   b Install Xythos WFS and specify the new database locations during installation.

2 Copy your existing WebDAV content to the new repository.

   a Export the data from your 9.1[.2] Global Schema database, and import that data into the Global Schema database on your test system. Then, export the data from each 9.1[.2] Document Store, and import the data into the Document Stores that you just created. (For detailed information about how to import and export data, refer to your database vendor's documentation.)
   b If external storage is enabled in your 9.1[.2] Xythos WFS setup, copy the existing external storage files into the test copy of Xythos WFS.

3 On the test system, update the JDBC connection URL that are used by Xythos WFS so that it points to the new Xythos Global Schema.

   a Edit the **`xythos.properties`** file in your new installation, and set the **`BaseJDBCConnectionURL`** property to the connection string for the copied Xythos Global Schema.
   b Start the new copy of WFS.
   c From the Xythos Administration Pages, for each document store, update the parameters **`Storage Location`**, **`JDBC Connection URL`**, **`DB User Name`**, and **`DB Password`**. (To get to the page for a document store, select **SERVER ADMINISTRATION ▶ Servers ▶ Document Stores**.)

4 Update the metadata definition of the connection for your HTTP DAV Server. Using the Server Manager plug-in to SAS Management Console, set the **`Host Name`** to the name of the middle-tier host in your test system.

Migrate Your WebDAV Data to a 9.1.3 Test System (SAS Web Report Studio)

Important Note: Follow the instructions in this section only if both of the following are true:

- you are moving your 9.1[.2] WebDAV data to a 9.1.3 test system
- you are using SAS Web Report Studio.

If you are also using SAS Information Delivery Portal and that application's WebDAV capabilities are enabled, you should already have followed the instructions in "Migrate Your WebDAV Data to a 9.1.3 Test System (SAS Information Delivery Portal)" on page 541.

If you are using Xythos WebFile Server as your WebDAV server, migrate your WebDAV data to your test system using the instructions in "Migrate Your WebDAV Data to a 9.1.3 Test System (SAS Information Delivery Portal)" on page 541—if you have not already performed those tasks. Then, use SAS Management Console to update the properties of your BIP Tree:

1 Expand the Business Report Manager node.

2 Right-click the BIP Tree icon and select **`Properties`** from the pop-menu that appears. A BIP Tree Properties dialog appears.

3 In this dialog box, update the value of the **`Content Server`** field. Also, update the SAS Web Administrator's password, if necessary. Then, click **`OK`**.

If you are using the Apache HTTP Server as your WebDAV server, use the following directions to move your WebDAV content to your test system:

1 On your test system, create the directory structure **`/sasdav/wrs`** inside the **`htdocs`** directory in your Apache installation.

2 Move the contents of **`/sasdav`** on your 9.1[.2] system to **`/sasdav/wrs`** on the test system. You can use any program that enables you to move files for this task: Windows Explorer, FTP, etc.

3 On the test system, update the metadata definitions of your HTTP Server and BIP Tree.
   - Use the Server Manager plug-in to SAS Management Console to add a new **`Base Path`** (**`/sasdav/wrs`**) to the HTTP Server and to update the **`Host Name`** on the HTTP Server's connection.
   - Use the Business Report Manager plug-in to edit the properties of the BIP Tree as necessary. You will have to at least change the **`Content Server`** and the **`Content Base Path`**; you might also need to change the **`Content Server Authentication`** information.

Migrate Your WebDAV Data in the 9.1.3 Production System (SAS Information Delivery Portal)

Important Note: Follow the instructions in this section only if both of the following are true:

- you are upgrading your 9.1[.2] WebDAV repository in place so that it can be part of a 9.1.3 production system
- you are using SAS Information Delivery Portal with its WebDAV features enabled.

If you are also using SAS Web Report Studio, see "Migrate Your WebDAV Data in the 9.1.3 Production System (SAS Web Report Studio)" on page 545 as well.

When you upgrade your 9.1[.2] production system to 9.1.3, you should move your SAS Information Delivery Portal WebDAV data to a new location on your Xythos WebFile Server. You need to change your base path root from **`/`** to **`/sasdav`** and then move your WebDAV data to the new location. This change will result in better performance and will make the system consistent with any newly installed 9.1.3 systems. To make the change, perform the following steps:

1 Make sure that you have the latest User Management code installed in your Xythos WFS installation area. You can obtain the latest JAR files and scripts from the 9.1.3 Xythos WFS installation CD. Unzip **`saswfs.zip`** into *Xythos-install-dir***`wfs-4.0.48`**.

2 Run the **`WFSInstaller.bat`** script located in *Xythos-install-dir***`wfs-4.0.48`**. In the installation screen presented, fill in the appropriate values for each field. You can look at your existing *Xythos-install-dir***`wfs-4.0.48\saswfs.properties`** file to see how the values were set during the previous installation. In the **`Enter the path to the user area`** field, enter **`/sasdav`**.

   Note: For new installations, we recommend defining a user area **`/sasdav/Users`**. We do not recommend **`/sasdav/Users`** for an upgrade, because you will lose access to content in your personal repository, such as alerts. △

   Note: The base path root specified in the Xythos WebFile Server installation must match the value specified during the SAS Web Infrastructure Kit installation.

   △

3 Run the **`WFSInstaller2.bat`** script that is located in the same directory. This script will create a **`/sasdav`** top-level directory on the Xythos WFS. This directory is referred to as the base path root. If you set the user area to **`/sasdav/Users`**, the script will also create a **`/sasdav/Users`** directory to store each user's personal content. If you did not set the user area this way, each user's personal content will be stored in **`/sasdav`**.

4 Stop and restart the Xythos WFS so that the new settings take effect.

5 When you unpacked the file **`saswfs.zip`** in step 1, the migration utilities **`MovePersons`** and **`MoveResources`** were placed in the directory **`wfs-4.0,48\utils`**. The usage information for the two utilities is as follows:

```
-----------------------------
MovePersons (Version 1.0.001)
-----------------------------
Usage:  MovePersons [/v] dir1 dir2
eg. MovePersons / /sasdav/Users
where
    dir1 is the source directory for person entries
    dir2 is the target directory
options
    /v verify operation, but do not actually do the move

------------------------------
MoveResources (Version 1.0.001)
------------------------------
Usage:  MoveResources [/v] dir1 regex dir2
eg. MoveResources /v / ".*" /sasdav
where
    dir1 is the source directory for person entries
    dir2 is the target directory
options
    /v verify operation, but do not actually do the move
```

6 In both of these scripts, make sure that the **XYTHOS_HOME** and **JDBC_DRIVER** environment variables are set correctly. For example, you might need to set their values like this:

```
set XYTHOS_HOME=c:\xythos
set JDBC_DRIVER=c:\cygwin\usr\share\postgresql\java\postgresql.jar
```

It is also a good idea to define the environment variable **JAVA_HOME** at the top of both scripts to make sure that you know which version of the JDK you are using:

```
set JAVA_HOME=c:\j2sdk1.4.2
```

7 Migrate user folders from **/** to **/sasdav** using the **MovePersons.cmd** script. For example, you might use the command:

```
C:\xythos\wfs-4.0.48\utils>MovePersons.cmd / /sasdav
```

If you set the user area to **/sasdav/Users** in step 2, then the second argument to this script should be **/sasdav/Users**.

8 Migrate other top level SAS content folders using the **MoveResources.cmd** script. For instance, the example below shows how you might move a **/Sales** folder containing Microsoft Office documents, a **/sas/publish** folder with publication/subscription content, and a **/ReportStudio** folder containing reports:

```
C:\xythos\wfs-4.0.48\utils>MoveResources / "Sales" /sasdav
C:\xythos\wfs-4.0.48\utils>MoveResources / "sas" /sasdav
C:\xythos\wfs-4.0.48\utils>MoveResources / "ReportStudio" /sasdav
```

9 If you have defined your WebDAV server in your metadata, bring up SAS Management Console, and go to the Server Manager plug-in. Right-click the icon representing your WebDAV server, and select **Properties** from the pop-up menu that appears. In the Properties dialog box, select the **Options** tab. In the **Base Path(s)** list, change **/** to **/sasdav**.

10 If you have defined channels with a WebDAV persistent store in the metadata, bring up SAS Management Console, and go to the Publishing Framework plug-in. Expand the tree to list the channels. Right-click each channel that uses WebDAV for a persistent store, and select **Properties** from the pop-up menu that appears.

In the Properties dialog box, select the **General** tab. In the **Content Base Path** drop-down list, change **/** to **/sasdav**.

11 Make sure that the DAV_BASE name/value pair located in the file *SAS-install-dir***\Web\Portal2.0.1\PortalConfigure\install.properties** is set to **/sasdav**. This value is the DAV base bath root used to configure the Web Infrastructure Kit (WIK). The base path root is used in the WIK's local services deployment definition: there is a repository definition entry for the DAV server in the information services configuration.

12 Run the script **configure_wik**. This script is located in the directory *SAS-install-dir***\Web\Portal2.0.1**.

13 If the services deployment definitions are loaded in your metadata repository, reload the service definitions to make sure that the DAV repository definition entry has the correct base path.

Migrate Your WebDAV Data in the 9.1.3 Production System (SAS Web Report Studio)

Important Note: Follow the instructions in this section only if both of the following are true:

- you are upgrading your 9.1[.2] WebDAV repository in place so that it can be part of a 9.1.3 production system
- you are using SAS Web Report Studio.

If you are also using SAS Information Delivery Portal and that application's WebDAV capabilities are enabled, you should already have followed the instructions in "Migrate Your WebDAV Data in the 9.1.3 Production System (SAS Information Delivery Portal)" on page 543.

If you are replacing your 9.1[.2] production system, you should follow the appropriate directions below—depending on whether you are using the Xythos WebFile Server or the Apache HTTP Server as a WebDAV server. In either case, you should make your existing WebDAV content available at the content base path **/sasdav/wrs** instead of (or in addition to) **/dav/sas**. This section explains how to migrate your content for each of these servers.

If you are using the Xythos WebFile Server as your WebDAV server, use the following procedure:

1 Use the Xythos WFS administration console to create the folder structure **/sasdav/wrs**.

2 Use the Xythos WFS administration console to grant the SAS Web Administrator (**saswbadm**) full access to the **wrs** folder and to deny access to this folder to other users.

3 Copy your existing WebDAV content from **/dav/sas** to **/sasdav/wrs**.

4 In SAS Management Console, use the Server Manager plug-in to change the properties for your WebDAV server (HTTP DAV Server). Right-click the icon that represents this server, and select **Properties** from the pop-up menu. A properties dialog box appears. On the **Options** tab, create a new base path: **/sasdav/wrs**.

5 In SAS Management Console, use the Business Report Manager plug-in to update the properties of the BIP Tree. Expand the Business Report Manager. Then right-click the BIP Tree icon, and select **Properties** in the pop-up menu that appears. A properties dialog box appears. Set the **Content Base Path** to **/sasdav/wrs**, and enter the user ID and password for the SAS Web Administrator in the **Content Server Authentication** area.

If you are using the Apache HTTP Server as your WebDAV server, use the following procedure:

1 Edit the **`httpd.conf`** configuration file to create an alias for the content base path. This way, you do not have to actually copy your WebDAV content to a new location.

   When you first configured your system, you should have added an element like this to this file:

```
<Directory "C:\Program Files\Apache Group\Apache2\htdocs\dav\sas">
 SetEnv redirect-carefully 1
 Dav On
 DavDepthInfinity On
 Options None
 AllowOverride None
</Directory>
```

   Add the following line just before this element:

```
Alias /sasdav/wrs "C:\Program Files\Apache Group\Apache2\htdocs\dav\sas"
```

   Then, save the file.

2 In SAS Management Console, use the Server Manager plug-in to change the properties for your WebDAV server (HTTP DAV Server). Right-click the icon that represents this server, and select **`Properties`** from the pop-up menu. A properties dialog box appears. On the **`Options`** tab, create a new base path: **`/sasdav/wrs`**.

3 In SAS Management Console, use the Business Report Manager plug-in to update the properties of the BIP Tree. Expand the Business Report Manager. Then right-click the BIP Tree icon, and select **`Properties`** in the pop-up menu that appears. A properties dialog box appears. Set the **`Content Base Path`** to **`/sasdav/wrs`**, and enter the user ID and password for the SAS Web Administrator in the **`Content Server Authentication`** area.

Start the SAS Services Application

You should now start the SAS Services application. For more information about how to perform this step, see the directions in your initial configuration instructions, **`instructions.html`**.

Deploy Your Web Applications

Your Web applications will have been upgraded. Redeploy these new versions to your Web server.

1 If necessary, uninstall the current version of each SAS Web application. If you are using Tomcat as your Web server, delete the directory structure for a Web application from the **`webapps`** directory. If you are using the BEA WebLogic server as your Web server:

   - Use the WebLogic Server administration console to delete an application. (To see the currently deployed Web applications, expand the Deployments tree node, and select the Web Applications node. In the resulting list of Web applications, select the Delete icon for the appropriate application.)
   - Remove the directory structure for the Web application from the **`webapps`** directory.

2 To deploy the new Web applications, follow the directions in the section "Deploy Your Web Applications" in your configuration instructions, **`instructions.html`**.

   Note: For more detailed information about deploying your Web applications, see the section "Getting More Information" in **`instructions.html`**. △

3 If you are using Tomcat as your servlet container, start Tomcat. For directions on how to do this, see "Start Your Tomcat Server" in **`instructions.html`**.

Upgrading a SAS 9.1 or 9.1.2 z/OS System to a SAS 9.1.3 System

The following two sections explain how to create a SAS z/OS business intelligence installation that you can swap into a functioning SAS 9.1 or SAS 9.1.2 business intelligence installation. Be certain to follow the steps in order.

Installing and Configuring the SAS 9.1.3 Software

Follow these instructions to install and configure a 9.1.3 system that you can test before you put it into production.

1. Install SAS 9.1.3 Foundation for z/OS using the "Action A" instructions for installing a completely new SAS Foundation for z/OS. You can use the SAS Installation Wizard for z/OS for this task. The high level prefix that you use for this installation is *not* critical.

   Note: The "Action A" instructions are located in your Installation Kit, behind the Installation Guide tab. △

2. Perform all of the necessary post-installation customization tasks for the new SAS 9.1.3 Foundation for z/OS. These customizations are discussed in appendixes in the Configuration Guide section of the Installation Kit, and should mirror the customizations that you performed when installing the SAS 9.1 or SAS 9.1.2 Foundation for z/OS.
3. In the control data set used for installing the new SAS 9.1.3 Foundation for z/OS (hereafter referred to as the **CNTLDSN**), open an editor on the member **UGTARGET**. Edit the following line:

   ```
   UGPFX=SAS.V91.PROD.PFX      <<==SUPPLY YOUR PRODUCTION SAS PREFIX
   ```

   Substitute the high-level prefix of your production or existing SAS 9.1 or SAS 9.1.2 Foundation for z/OS system for **SAS.V91.PROD.PFX**. Then save the member.
4. Open the editor on **CNTLDSN(MAKEUPG)**. Locate the following JCL near the beginning of the job:

   ```
   //INSTEDT1 PROC CNTLDSN='your_ctrl_data_set',
   //         SASPROC='SAS'                  <=== << SUPPLY >>
   //         SASEDTP='SASEDITP',            <=== << VERIFY >>
   //         SYSOUT='*',                    <=== << VERIFY >>
   //         DISKUNI='DISK'                 <=== << VERIFY >>
   ```

   Then, follow these steps:

   a. In the line with **SASPROC='SAS'**, substitute the procedure name of a working SAS 9.1, 9.1.2, or 9.1.3 PROC.
   b. Do not change **SASEDITP=***parm* unless directed to do so by SAS Technical Support.
   c. Verify the **SYSOUT=** and **DISKUNI=***parms* DD statements, which should have been set correctly from the main install.
   d. Save the edit.
5. Submit **CNTLDSN(MAKEUPG)** for execution. This will create four utilities to be used later. The names will contain the *actual* TS level identifiers of the production (old) and newly installed systems, obtained by analyzing the executables library of the production and new systems. These two TS level identifiers will also be used in the new data set names created by running the utilities.

   In the section "Replacing Your Production System" on page 549, "TSpMp" (your production or old system) and "TSnMn" (your newly installed system) may have

actual values of TS1M0 and TS1M3 as one of a number of possible TS level combinations. Substitute your actual TS level values for **TSpMp** and **TSnMn**.

6 After the new system has been deemed production quality, and during system down time, perform the following steps:

a Stop the version 9.1 or 9.1.2 servers.

b Copy the SAS 9.1 or 9.1.2 configuration directory to the SAS 9.1.3 configuration directory. You can do this by running the following command from the UNIX System Services shell:

```
cp -R 9.1_config_dir 9.1.3_config_dir
```

c Remove old log files from the 9.1.3 configuration directory. Use a command similar to this one:

```
rm 9.1.3_config_dir/Lev1/SASMain/*/logs/*
```

d Edit the COPYIA job with the location of the 9.1.3 configuration directory. (Supply the value for the CONFIG_DIR variable.) The COPYIA job is located in the *sas_v913_installed_prefix*.**W0.SRVCNTL** data set. Run the COPYIA job, and verify a successful completion.)

e Copy **configuration.properties** from your 9.1 or 9.1.2 configuration directory to your 9.1.3 configuration directory. From the USS shell, issue the command:

```
cp 9.1_config_dir/Utilities/zOS_config/configuration.properties
  9.1.3_config_dir/Utilities/zOS_config/configuration.properties
```

f Change directories so that you are in the **zOS_config** directory inside the 9.1.3 configuration directory. To get there, use the USS shell command:

```
cd 9.1.3_config_dir/Utilities/zOS_config
```

g Edit the 9.1.3 version of **configuration.properties**, supplying new values for the following properties:

- $CONFIG_DIR$ - Point to the 9.1.3 configuration directory.
- $APPSERVER_DIR$ - Point to the new application-server directory, for example, *9.1.3_config_dir*/**Lev1/SASMain**.

h You need to change only the following values if your site wants to run the 9.1.3 servers in parallel with the 9.1 or 9.1.2 servers during a testing period. This is the recommended approach. If your site wants to run the 9.1.3 servers right out of the box, just as they were configured in 9.1 or 9.1.2, then you do not need to change any of the following values:

- $OMAPORT$ - Enter a new port number.
- $OMA_STCNAME$ - Enter a new started task name (different from the 9.1 or 9.1.2 name).
- $OMA_CFGNAME$ - Enter a new configuration file name (same as the started task name).
- $OMA_TKENVNM$ - Enter the new tk environment file name (same as the started task name).
- $SPAWNER_OPERATOR_PORT$ - Enter a new port number.
- $SPAWNER_LOADBALANCING_PORT$ - Enter a new port number.
- IOM_PORT - Enter a new port number.
- $SPAWNER_STCNAMES$ - Enter a new started task name.
- $SPAWNER_PRMNAME$ - Enter a new parm name.
- STP_PORT - Enter a new port number.
- STP_PORT1 - Enter a new port number.

- STP_PORT2 - Enter a new port number.
- STP_PORT3 - Enter a new port number.
- OLA_PORT - Enter a new port number.
- $OLA_STCNAME$ - Enter a new started task name.
- $OLA_CFGNAME$ - Enter a new configuration file name.
- $OLA_TKENVNM$ - Enter a new tk environment file name.
- $CONNECT_PORT$ - Enter a new port number.
- $SHAREPORT$ - Enter a new port number.
- $SHR_STCNAME$ - Enter a new started task name.
- $SHR_CFGNAME$ - Enter a new configuration file name.
- $SHR_TKENVNM$ - Enter a new tk environment file name.
- $CNT_STCNAME$ - Enter a new started task name.
- $CNT_PRMNAME$ - Enter a new parm file name.

i Run the **deploy_IA.sh** script.

```
deploy_IA.sh -p configuration.properties
```

If you are asked whether you want to overwrite the environment, answer **Yes**.

j Copy the new started task names to your site's started task library.

k Start the metadata server.

l Using SAS Management Console, upgrade your existing metadata repositories.

i Expand the Metadata Manager plug-in to SAS Management Console.

ii Right-click the Active Server icon, and select **Upgrade Metadata** from the pop-up menu.

m Using SAS Management Console, update the metadata for the 9.1.3 system (using the 9.1.3 **instructions.html** as a guide). Change

- the Launch Command for the workspace server
- the Launch Command for the stored process server
- the Launch Command for the SAS/CONNECT server
- all the port numbers listed previously.

n Start the remaining servers.

o Test all of the servers.

Replacing Your Production System

In the following steps, you will rename your production and newly installed SAS system data sets in order to facilitate the changeover between your existing production SAS environment and the newly installed SAS system. You will accomplish this by running the **RNMTSpMp** and **RNMTSnMn** utilities, which will rename the existing SAS data sets using the following guidelines:

- **SAS.V91.PROD.PFX.*** becomes **SAS.V91.PROD.PFX.TSpMp.XXX**
- **SAS.V913.PFX.*** becomes **SAS.V91.PROD.PFX.TSnMn.XXX**

Alias names will be created for the originally installed data set names, and they will now point to the recently renamed data set names.

CAUTION:

Note that the alias names are alternate names for data sets that are defined in the catalog. RACF does not allow alias names because it uses the RACF database, not the catalog,

for its processing. If you have existing RACF profiles for your SAS installation, you might need to adjust these profiles based on the new data set names. △

CAUTION:
If your installation currently uses, or has future plans to run, a SAS/CONNECT spawner or object spawner, note that the SAS executable library and the SAS/C transient library must be put under RACF program control. Because RACF does not support alias names, you must use the renamed, fully qualified data set names in the program control table. △

CAUTION:
After the RNMTSpMp utility is executed, the SAS/CONNECT and object spawners will not function until the SAS.V91.PROD.PFX.TSpMp.DLR (the renamed SAS/C transient library) and SAS.V91.PROD.PFX.TSpMp.DLD (the renamed SAS executable library) have been added to the program control table. △

1. During system downtime, submit the utility **CNTLDSN(RNMTSpMp)**. This utility will rename the production system data sets and create aliases from the existing names to the new names. If you need to run this utility outside of system downtime, you must follow a locally appropriate procedure to ensure that data sets are not in use or enqueued, which will cause the process to fail.
2. Add the **SAS.V91.PROD.PFX.TSpMp.DLR** and **SAS.V91.PROD.PFX.TSpMp.DLD** libraries to the program control table, if your existing production SAS installation uses a SAS/CONNECT spawner or object spawner.
3. Submit the utility **CNTLDSN(RNMTSnMn)**. This utility will rename the new system data sets and create aliases from the existing names to the new names.
4. Add the **SAS.V91.PROD.PFX.TSnMn.DLR** and **SAS.V91.PROD.PFX.TSnMn.DLD** libraries to the program control table, if your existing production SAS installation uses a SAS/CONNECT spawner or object spawner.

   Note: The TS*xxx* portion of these names is taken from the TS level that is obtained from the installed systems and will vary depending on the version of the installed systems. △

   You are now ready to perform a swap.
5. During system downtime, or during a testing period where access to and enqueues upon the SAS System can be controlled, submit the utility **CNTLDSN(USETSnMn)**, which will cause the production-named aliases to point to the new SAS System of that TS level. Note that once you have submitted the **USETSnMn** utility, the 9.1.3 SAS System will be the production environment. Therefore, only the BI server started tasks that were created in step 6j are valid for use. The 9.1 or 9.1.2 BI server started tasks that were previously defined in your started tasks library will not be valid unless you change the system back to 9.1 or 9.1.2 by running the **CNTLDSN(USETSpMp)** utility.
6. Perform tests as needed or scheduled to verify that the new production system is functioning properly.
7. You can perform an emergency restoration of the previous system by submitting the utility **CNTLDSN(USETSpMp)**.

APPENDIX

4

Applying Service Pack Maintenance in a z/OS Environment

Applying Service Pack Maintenance in a z/OS Environment

The following sections explain how to apply service pack maintenance to a SAS 9.1.3 z/OS installation of a Business Intelligence configuration directory. Be certain to follow the steps in order. The process to follow depends on the installation action used to install the software:

Action A

This is a new installation and configuration. To configure the BI servers, follow the instructions found in the appendix titled "Installing and Configuring the SAS Servers on z/OS" in *SAS Intelligence Platform: Installation Guide*.

Action B

This action installs directly to an existing SAS installation. Similarly, a configuration directory should already exist for this installation, and the maintenance updates will be applied directly to it.

CAUTION:
Action B can write directly to any SAS library specified in the installation High Level Qualifier parameter. Action B is NOT recommended for installation of updates to production libraries. Action B is provided to perform a direct update to stand-alone testing copies of your production SAS system. Use Action C to install updates in other circumstances. △

1 Edit the sas_v913_installed_prefix.W0.SRVCNTL(COPYIA) job and edit the values for the following variables:

CONFIG_DIR	This represents the existing configuration directory to which maintenance should be applied.
LEVEL	This represents the level which contains the application server context. Use the same value that was chosen for the initial installation. (The default is Lev1.)

APPNAME — This represents the directory name for the SAS Application Server. Use the same name that was chosen for the initial installation. (The default is SASMain.)

2 Stop all servers that are running under this application server context (i.e. running out of this configuration directory).

3 Submit the COPYIA job and verify a successful completion.

4 Change directories so that you are in the zOS_config directory inside the existing 9.1.3 configuration directory. To get there, use the USS shell command:

```
cd existing_9.1.3_config_dir/Utilities/zOS_config
```

5 Edit the configuration.properties file, supplying new values for any new properties added at the bottom.

6 Run the deploy_IA.sh script.

```
deploy_IA.sh -properties configuration.properties
```

7 Review instructions.html for new configuration instructions, and use SAS Management Console to update the metadata accordingly.

8 Copy sas_v913_installed_prefix.W0.SRVPROC(*) into the system started task library.

9 Restart the servers, and check the logs for successful deployment.

Action C

This action installs to a staged SAS environment, so a staged configuration directory is going to be deployed which contains the service pack maintenance. The existing (or production) configuration will still exist and is intended to run in parallel with the staged set of servers. The SAS install has created a complete new set of partitioned data sets for the staged servers to use, and they are located under the names sas_v913_installed_prefix.SL.W0.SRV*. Complete the following steps:

1 Stop all 9.1.3 servers.

2 Copy the existing 9.1.3 configuration directory to a new staged configuration directory. Do this by running the following commands from the UNIX System Services shell:

```
cp -R existing_9.1.3_config_dir staged_9.1.3_config_dir
```

3 Remove old log files from the 9.1.3 staged configuration directory. Use a command similar to this one:

```
rm staged_9.1.3_config_dir/Lev1/SASMain/*/logs/*
```

4 Edit the sas_v913_installed_prefix.SL.W0.SRVCNTL(COPYIA) job and edit the values for the following variables:

CONFIG_DIR — This represents the staged configuration directory to which maintenance should be applied.

LEVEL — This represents the level which contains the application server context. (The default is Lev1.)

APPNAME — This represents the directory name for the SAS Application Server. (The default is SASMain.)

5 Submit the COPYIA job and verify a successful completion.

6 Copy configuration.properties from your existing 9.1.3 configuration directory to your staged 9.1.3 configuration directory. From the USS shell, issue the command:

```
cp existing_9.1.3_config_dir/Utilities/zOS_config/configuration.properties
staged_9.1.3_config_dir/Utilities/zOS_config/configuration.properties
```

7 Change directories so that you are in the zOS_config directory inside the staged 9.1.3 configuration directory. To get there, use the USS shell command:

```
cd staged_9.1.3_config_dir/Utilities/zOS_config
```

8 Add new properties to the existing configuration.properties file by issuing the following command:

```
cat configuration.properties.SAS_* >> configuration.properties
```

9 Edit the staged 9.1.3 version of configuration.properties, supplying new values for the following properties:

- $CONFIG_DIR$ - Point to the staged 9.1.3 configuration directory.
- $APPSERVER_DIR$ - Point to the new application-server directory, for example, staged_9.1.3_config_dir/Lev1/SASMain.

Also uncomment and provide values for the new properties that are found at the bottom of the file.

10 You need to change the following values if your site wants to run the staged 9.1.3 servers in parallel with the existing 9.1.3 servers during a testing period. This is the recommended approach.

- $OMAPORT$ - Enter a new port number.
- $OMA_STCNAME$ - Enter a new started task name (different from the 9.1 or 9.1.2 name).
- $OMA_CFGNAME$ - Enter a new configuration file name (same as the started task name).
- $OMA_TKENVNM$ - Enter the new tk environment file name (same as the started task name).
- $SPAWNER_OPERATOR_PORT$ - Enter a new port number.
- $SPAWNER_LOADBALANCING_PORT$ - Enter a new port number.
- IOM_PORT - Enter a new port number.
- $SPAWNER_STCNAMES$ - Enter a new started task name.
- $SPAWNER_PRMNAME$ - Enter a new parm name.
- STP_PORT - Enter a new port number.
- STP_PORT1 - Enter a new port number.
- STP_PORT2 - Enter a new port number.
- STP_PORT3 - Enter a new port number.
- $JSSPORT$ - Enter a new port number.
- OLA_PORT - Enter a new port number.
- $OLA_STCNAME$ - Enter a new started task name.
- $OLA_CFGNAME$ - Enter a new configuration file name.
- $OLA_TKENVNM$ - Enter a new tk environment file name.
- $CONNECT_PORT$ - Enter a new port number.
- $SHAREPORT$ - Enter a new port number.

- $SHR_STCNAME$ - Enter a new started task name.
- $SHR_CFGNAME$ - Enter a new configuration file name.
- $SHR_TKENVNM$ - Enter a new tk environment file name.
- $CNT_STCNAME$ - Enter a new started task name.
- $CNT_PRMNAME$ - Enter a new parm file name.

11 Run the deploy_IA.sh script.

```
deploy_IA.sh -properties configuration.properties
```

If you are asked whether you want to overwrite the environment, answer Yes.

12 Copy the new started task names to your site's started task library.

13 Start the metadata server.

14 Using SAS Management Console, update the metadata for the staged 9.1.3 system (using the 9.1.3 instructions.html as a guide for new configuration information). Instructions for upgrading the metadata can be found in the "Upgrading Your Metadata Repository" section. In addition, the following will need to be changed if they existed in the original configuration, or if being configured for the first time:

- the Launch Command for the workspace server
- the Launch Command for the stored process server
- the Launch Command for the SAS/CONNECT server
- all the port numbers listed previously
- the Command Line and the Logs Directory for the Data Step Batch Server
- the Batch Job Deployment Directory for the Job Scheduling Server
- the Control Directory and the Launch Command for the Job Scheduler Server

15 Start the remaining servers.

16 Test all of the servers.

Action D

Once you have verified that the servers are functioning properly, you can promote the service pack maintenance updates into the production environment by following these steps:

1 Run the SAS install action "D" to apply staged service pack changes to the production SAS system.

2 From the UNIX System Services shell, change your working directory to the zOS_config directory inside the staged configuration directory using the following command:

```
cd staged_9.1.3_config_dir/Utilities/zOS_config
```

3 Run the deploy_IA.sh script to promote the changes to the production configuration directory:

```
deploy_IA.sh -properties configuration.properties -promote
```

4 Review the output from the script for successful completion.

5 Change directories so that you are in the zOS_config directory inside the existing 9.1.3 configuration directory. To get there, use the USS shell command:

```
cd existing_9.1.3_config_dir/Utilities/zOS_config
```

6 Edit the configuration.properties file, supplying new values for any new properties added at the bottom

7 Run the deploy_IA.sh script.

```
deploy_IA.sh -properties configuration.properties
```

8 The metadata in the existing configuration directory still contains information about the staged deployment. Use instructions.html as a guide for resetting the metadata to use your existing configuration directory. You will need to change the same information found in step 14 under Action C.

9 Copy sas_v913_installed_prefix.W0.SRVPROC(*) into the system started task library.

10 Restart the production servers to pick up the service pack maintenance changes.

APPENDIX

5

Recommended Reading

Recommended Reading

Here is the recommended reading list for this title:

- *Base SAS Procedures Guide*
- *SAS Data Providers: ADO/OLE DB Cookbook*
- *SAS Data Integration Studio: User's Guide*
- *SAS Integration Technologies: Administrator's Guide*
- *SAS Integration Technologies: Developer's Guide*
- *SAS Integration Technologies: Server Administrator's Guide*
- *SAS Language Reference: Concepts*
- *SAS Language Reference: Dictionary*
- *SAS Management Console: User's Guide*
- *SAS Metadata LIBNAME Engine: User's Guide*
- *SAS Metadata Server: Setup and Administration Guide*
- *SAS OLAP Server: Administrator's Guide*
- *SAS OLAP Server: MDX Guide*
- *SAS Scalable Performance Data Engine: Reference*
- *SAS Web Infrastructure Kit: Administrator's Guide*
- *SAS Web Infrastructure Kit: Developer's Guide*
- *SAS/ACCESS for Relational Databases: Reference*

For a complete list of SAS publications, see the current *SAS Publishing Catalog*. To order the most current publications or to receive a free copy of the catalog, contact a SAS representative at

SAS Publishing Sales
SAS Campus Drive
Cary, NC 27513
Telephone: (800) 727-3228*
Fax: (919) 677-8166
E-mail: **sasbook@sas.com**
Web address: **support.sas.com/pubs**
* For other SAS Institute business, call (919) 677-8000.

Customers outside the United States should contact their local SAS office.

Glossary

access control entry (ACE)
a set of identities and permissions that are directly associated with a particular resource. Each access control entry is directly associated with only one resource. More than one ACE can be associated with each resource.

access control template (ACT)
a reusable named authorization pattern that you can apply to multiple resources. An access control template consists of a list of users and groups and indicates, for each user or group, whether permissions are granted or denied.

ACE
See access control entry (ACE).

ACT
See access control template (ACT).

administration metadata server
the metadata server from which a metadata repository is replicated or promoted. See also metadata replication, metadata promotion.

administrative user
a special user of a metadata server who can create and delete user definitions and logins. An administrative user can also perform administrative tasks such as starting, stopping, pausing, and refreshing the metadata server. Unlike an unrestricted user, an administrative user does not have unrestricted access to the metadata. You are an administrative user if your user ID is listed in the adminUsers.txt file or if you connect to the metadata server using the same user ID that was used to start the metadata server.

aggregation
a summary of detail data that is stored with or referred to by a cube. Aggregations support rapid and efficient answers to business questions.

application server
a server that is used for storing applications. Users can access and use these server applications instead of loading the applications on their client machines. The application that the client runs is stored on the client. Requests are sent to the server for processing, and the results are returned to the client. In this way, little information is processed by the client, and nearly everything is done by the server.

architecture
the way in which components of a system are designed to fit or work together. This term can pertain to many types of complex systems, as in 'software architecture' and 'network architecture.'

ARM (Application Response Measurement)
an application programming interface that was developed by an industry partnership and which is used to monitor the availability and performance of software applications. ARM monitors the application tasks that are important to a particular business.

attribute
a characteristic that is part of the standard metadata for an object. Examples of attributes include the object's name, creation date, and modification date.

authentication
the process of verifying the identity of a person or process within the guidelines of a specific authorization policy.

authentication domain
a set of computing resources that use the same authentication process. An individual uses the same user ID and password for all of the resources in a particular authentication domain. Authentication domains provide logical groupings for resources and login s in a metadata repository. For example, when an application needs to locate credentials that enable a particular user to access a particular server, the application searches the metadata for logins that are associated with the authentication domain in which the target server is registered.

authentication provider
a software component that is used for identifying and authenticating users. For example, Windows NT and LDAP both provide authentication.

authorization
the process of determining which users have which permissions for which resources. The outcome of the authorization process is an authorization decision that either permits or denies a specific action on a specific resource, based on the requesting user's identity and group memberships.

basic installation
a method of installing a SAS business intelligence system that requires you to specify what software should be installed on each host. The basic installation method uses an installation tool called the SAS Software Navigator. See also project installation.

batch mode
a method of running SAS programs in which you prepare a file that contains SAS statements plus any necessary operating system control statements and submit the file to the operating system. Execution is completely separate from other operations at your terminal. Batch mode is sometimes referred to as running in the background.

buffer
a portion of computer memory that is used for special holding purposes or processes. For example, a buffer might simply store information before sending that information to main memory for processing, or it might hold data after the data is read or before the data is written.

cache
a small, fast memory area that holds recently accessed data. The cache is designed to speed up subsequent access to the same data.

change management
in the SAS Open Metadata Architecture, a facility for metadata source control, metadata promotion, and metadata replication.

channel
a virtual communication path for distributing information. In SAS, a channel is identified with a particular topic (just as a television channel is identified with a particular radio frequency). Using the features of the Publishing Framework, authorized users or applications can publish digital content to the channel, and authorized users and applications can subscribe to the channel in order to receive the content. See also publish, subscribe.

cleanse
to improve the consistency and accuracy of data by standardizing it, reorganizing it, and eliminating redundancy.

client application
an application that runs on a client machine.

client tier
the portion of a distributed application that requests services from the server tier. The client tier typically uses a small amount of disk space, includes a graphical user interface, and is relatively easy to develop and maintain.

cluster
a group of machines that participate in load balancing. Each machine in the cluster runs an object spawner that handles client requests for connections.

COM (Component Object Model)
an object-oriented programming model that defines how software components interact within a single process or between processes. For example, COM includes standard rules of communication that enable a user-interface object to be dragged and dropped from one application window to another.

component
a self-contained, reusable programming object that provides some type of service to other components in an object-oriented programming environment.

controller
a computer component that manages the interaction between the computer and a peripheral device such as a disk or a RAID. For example, a controller manages data I/O between a CPU and a disk drive. A computer can contain many controllers. A single CPU can command more than one controller, and a single controller can command multiple disks.

CORBA (Common Object Request Broker Architecture)
a standard API for distributed object communication. CORBA was created by the Object Management Group. It is the most widely used distributed object standard for connecting operating system platforms from multiple vendors.

credentials
the user ID and password for a particular user account that has been established either in the operating system or with an alternative authentication provider such as Microsoft Active Directory or Lightweight Directory Access Protocol.

cube
a logical set of data that is organized and structured in a hierarchical, multidimensional arrangement. A cube is a directory structure, not a single file. A cube includes measures, and it can have numerous dimensions and levels of data.

cube loading
the process of building a logical set of data that is organized and structured in a hierarchical, multidimensional arrangement. See also cube.

custom repository
in the SAS Open Metadata Architecture, a metadata repository that must be dependent on a foundation repository or custom repository, thus allowing access to metadata definitions in the repository or repositories on which it depends. A custom repository is used to specify resources that are unique to a particular data collection. For example, a custom repository could define sources and targets that are unique to a particular data warehouse. The custom repository would access user definitions, group definitions, and most server metadata from the foundation repository. See also foundation repository, project repository.

daemon
a process that starts and waits either for a request to perform work or for an occurrence of a particular event. After the daemon receives the request or detects the occurrence, it performs the appropriate action. If nothing else is in its queue, the daemon then returns to its wait state.

data mart
a collection of data that is optimized for a specialized set of users who have a finite set of questions and reports.

data partition
a physical file that contains data and which is part of a collection of physical files that comprise the data component of a SAS Scalable Performance Data Engine data set. See also partition, partitioned data set.

data quality
the relative value of data, which is based on the accuracy of the knowledge that can be generated using that data. High-quality data is consistent, accurate, and unambiguous, and it can be processed efficiently.

data warehouse
a collection of data that is extracted from one or more sources for the purpose of query, reporting, and analysis. In contrast to a data mart, a data warehouse is better suited for storing large amounts of data that originates in other corporate applications or which is extracted from external data sources such as public databases.

database library
a collection of one or more database management system files that are recognized by SAS and that are referenced and stored as a unit. Each file is a member of the library.

database management system
See DBMS (database management system).

database server
a server that provides relational database services to a client. Oracle, DB/2 and Teradata are examples of relational databases.

DBMS (database management system)
a software application that enables you to create and manipulate data that is stored in the form of databases. See also relational database management system.

DCOM (Distributed Component Object Model)
an extension to the Component Object Model (COM) that enables components to request services from components that are on other computers in a network. See also component, COM (Component Object Model).

default access control template
the access control template (ACT) that controls access to a particular repository and to resources for which definitive access controls are not specified. You can designate one default ACT for each metadata repository. The default ACT is also called the repository ACT.

default ACT
See default access control template.

descriptor information
information about the contents and attributes of a SAS data set. For example, the descriptor information includes the data types and lengths of the variables, as well as which engine was used to create the data. SAS creates and maintains descriptor information within every SAS data set.

development environment
a computing environment in which application developers use software tools to write, compile, and debug programs. See also testing environment, production environment.

dimension
a group of closely related hierarchies. Hierarchies within a dimension typically represent different groupings of information that pertains to a single concept. For example, a Time dimension might consist of two hierarchies: (1) Year, Month, Date, and (2) Year, Week, Day. See also hierarchy.

dimension table
in a star schema, a table that contains the data for one of the dimensions. The dimension table is connected to the star schema's fact table by a primary key. The dimension table contains fields for each level of each hierarchy that is included in the dimension.

domain
a database of users that has been set up by an administrator by using a specific authentication provider such as LDAP or the host operating system. The domain name should be unique within your enterprise. For example, you should not have a Windows domain a nd a Unix domain that are both named "SALES". See also authentication domain.

encryption
the act or process of converting data to a form that only the intended recipient can read or use.

extended attribute
a custom attribute that is not part of the standard metadata for an object. Extended attributes can be used to automate tasks that require a custom attribute to be associated with one or more objects. Extended attributes can be added either programmatically or manually (through an application window).

fact
a single piece of factual information in a data table. For example, a fact can be an employee name, a customer's phone number, or a sales amount. It can also be a derived value such as the percentage by which total revenues increased or decreased from one year to the next.

fact table
the central table in a star schema. The fact table contains the individual facts that are being stored in the database as well as the keys that connect each particular fact to the appropriate value in each dimension.

foundation repository
in the SAS Open Metadata Architecture, a metadata repository that is used to specify metadata for global resources that can be shared by other repositories. For example, a foundation repository is used to store metadata that defines users and groups on the metadata server. Only one foundation repository should be defined on a metadata server. See also custom repository, project repository.

global resource
a widely used resource, such as a server that is used to access many tables in a data warehouse. See also resource.

hierarchy
an arrangement of members of a dimension into levels that are based on parent-child relationships. Members of a hierarchy are arranged from more general to more specific. For example, in a Time dimension, a hierarchy might consist of the members Year, Quarter, Month, and Day. In a Geography dimension, a hierarchy might consist of the members Country, State or Province, and City. More than one hierarchy can be defined for a dimension. Each hierarchy provides a navigational path that enables users to drill down to increasing levels of detail. See also member, level.

HTTP (HyperText Transfer Protocol)
a protocol for transferring data to the Internet. HTTP provides a way for servers and Web clients to communicate. It is based on the TCP/IP protocol.

HTTP server
a server that handles an HTTP request from a client such as a Web browser. Usually the client's HTTP request indicates that the client wants to retrieve information that is pointed to by a URL. An example of a popular HTTP server is the Apache HTTP Server from the Apache Software Foundation.

HTTPS (HyperText Transfer Protocol Secure)
an HTTP protocol that enables secure connections to be made between a Web browser and a server.

identity
See metadata identity.

inbound login
a login that is used to determine your metadata identity. The login is inbound to a SAS Metadata Server. A login that functions only as an inbound login does not need to include a password or to specify an authentication domain.

information map
a collection of data items and filters that describes and provides a business-relevant view of physical data. Users of query and reporting applications such as SAS Web Report Studio can easily build business reports by using information maps as the building blocks for their reports.

Integrated Object Model
See IOM (Integrated Object Model).

Integrated Object Model server
See IOM server.

IOM (Integrated Object Model)
the set of distributed object interfaces that make SAS software features available to client applications when SAS is executed as an object server.

IOM bridge
a software component of SAS Integration Technologies that enables Java clients and Windows clients to access an IOM server.

IOM server
a SAS object server that is launched in order to fulfill client requests for IOM services. See also IOM (Integrated Object Model).

JAR file
a Java Archive file. The JAR file format is used for aggregating many files into one file. JAR files have the file extension .jar.

Java
a set of technologies for creating software programs in both stand-alone environments and networked environments, and for running those programs safely. Java is a Sun Microsystems trademark.

Java application
a stand-alone program that is written in the Java programming language.

Java Database Connectivity
See JDBC (Java Database Connectivity).

Java Development Kit
See JDK (Java Development Kit).

Java Virtual Machine
See JVM (Java Virtual Machine).

JavaServer page
See JSP (JavaServer page).

JDBC (Java Database Connectivity)
a standard interface for accessing SQL databases. JDBC provides uniform access to a wide range of relational databases. It also provides a common base on which higher-level tools and interfaces can be built.

JDK (Java Development Kit)
a software development environment that is available from Sun Microsystems, Inc. The JDK includes a Java Runtime Environment (JRE), a compiler, a debugger, and other tools for developing Java applets and applications.

job
a metadata object that specifies processes that create output.

JSP (JavaServer page)
a type of servlet that enables users to create Java classes through HTML.

JVM (Java Virtual Machine)
a program that interprets Java programming code so that the code can be executed by the operating system on a computer. The JVM can run on either the client or the server. The JVM is the main software component that makes Java programs portable across plat forms. A JVM is included with JDKs and JREs from Sun Microsystems, as well as with most Web browsers.

key
a value that uniquely identifies a specific record in a database.

LDAP (Lightweight Directory Access Protocol)
a protocol that is used for accessing directories or folders. LDAP is based on the X.500 standard, but it is simpler and, unlike X.500, it supports TCP/IP.

LDAP directory
a repository that contains data about an enterprise's users and resources, as well as related security information, and that stores this data and information in a format that clients on a network can access by using the Lightweight Directory Access Protocol (LDAP).

LDAP server
a server that provides access to one or more LDAP directories.

level
in a multidimensional database (or cube), an element of a dimension hierarchy. Levels describe the dimension from the highest (most summarized) level to the lowest (most detailed) level. For example, possible levels for a Geography dimension are Country, Region, State or Province, and City.

Lightweight Directory Access Protocol
See LDAP (Lightweight Directory Access Protocol).

load balancing
for IOM bridge connections, a program that runs in the object spawner and that uses an algorithm to distribute work across object server processes on the same or separate machines in a cluster.

locale
a value that reflects the language, local conventions, and culture for a geographic region. Local conventions can include specific formatting rules for dates, times, and numbers, and a currency symbol for the country or region. Collating sequences, paper s izes, and conventions for postal addresses and telephone numbers are also typically specified for each locale. Some examples of locale values are French_Canada, Portuguese_Brazil, and Chinese_Singapore.

localhost
a keyword that is used to specify the machine on which a program is executing. If a client specifies localhost as the server address, the client connects to a server that runs on the same machine.

logical server
in the SAS Metadata Server, the second-level object in the metadata for SAS servers. A logical server specifies one or more of a particular type of server component, such as one or more SAS Workspace Servers.

login
a combination of a user ID, a password, and an authentication domain. Each login provides access to a particular set of computing resources. In a SAS metadata environment, each login can belong to only one individual or group. However, each individual or group can own multiple logins. A login can function as an inbound login, an outbound login, or as both an inbound login and an outbound login. See also inbound login, outbound login.

MDX (multidimensional expressions) language
a standardized, high-level language that is used for querying multidimensional data sources. The MDX language is the multidimensional equivalent of SQL (Structured Query Language). It is used by the OLE DB for OLAP API.

measure
a special dimension that contains summarized numeric data values that are analyzed. Total Sales and Average Revenue are examples of measures. For example, you might drill down within the Clothing hierarchy of the Product dimension to see the value of the Total Sales measure for the Shirts member.

member
in a multidimensional database (or cube), a name that represents a particular data element within a dimension. For example, September 1996 might be a member of the Time dimension. A member can be either unique or non-unique. For example, 1997 and 1998 represent unique members in the Year level of a Time dimension. January represents non-unique members in the Month level, because there can be more than

one January in the Time dimension if the Time dimension contains data for more than one year.

metadata identity
a metadata object that represents an individual user or a group of users in a SAS metadata environment. Each individual and group that accesses secured resources on a SAS Metadata Server should have a unique metadata identity within that server.

metadata LIBNAME engine
the SAS engine that processes and augments data that is identified by metadata. The metadata engine retrieves information about a target SAS data library from metadata objects in a specified metadata repository.

metadata model
a definition of the metadata for a set of objects. The model describes the attributes for each object, as well as the relationships between objects within the model. The SAS Metadata Model is an example. See also SAS Metadata Model.

metadata object
a set of attributes that describe a table, a server, a user, or another resource on a network. The specific attributes that a metadata object includes vary depending on which metadata model is being used.

metadata profile
a client-side definition of where a metadata server is located. The definition includes a host name, a port number, and a list of one or more metadata repositories. In addition, the metadata profile can contain a user's login information and instructions for connecting to the metadata server automatically.

metadata promotion
in the SAS Open Metadata Architecture, a feature that enables you to copy the contents of a metadata repository to another repository, and to specify changes in the metadata that will be stored in the target repository. For example, you can use this feature to move metadata from a development environment to a testing environment. In such a scenario, you would probably have to change some ports, hosts, and/or schema names as part of the process of moving metadata from one environment to another.

metadata replication
in the SAS Open Metadata Architecture, a feature that enables you to copy the contents of a metadata repository to another repository. In contrast to metadata promotion, metadata replication makes an exact copy of a metadata repository in a new location. For example, metadata replication is used for backing up a repository.

metadata repository
a collection of related metadata objects, such as the metadata for a set of tables and columns that are maintained by an application. A SAS Metadata Repository is an example.

metadata server
a server that provides metadata management services to one or more client applications. A SAS Metadata Server is an example.

metadata source control
in the SAS Open Metadata Architecture, a feature that enables multiple users to work with the same metadata repository at the same time without overwriting each other's changes. See also change management.

middle tier
in a SAS business intelligence system, the tier in which J2EE Web applications and J2EE enterprise applications execute.

multi-tier server environment
a computing environment that includes both a middle tier, in which a servlet container or J2EE platform runs, and a server tier, in which the SAS Metadata Server runs.

multidimensional database (MDDB)
another term for cube. See cube.

object
in object-oriented programming, an instantiation or specific representation of a class.

Object Linking and Embedding
See OLE (Object Linking and Embedding).

object server
another term for IOM server. See IOM server.

object spawner
a program that instantiates object servers that are using an IOM bridge connection. The object spawner listens for incoming client requests for IOM services. When the spawner receives a request from a new client, it launches an instance of an IOM server to fulfill the request. Depending on which incoming TCP/IP port the request was made on, the spawner either invokes the administrator interface or processes a request for a UUID (Universal Unique Identifier).

OLAP (online analytical processing)
a software technology that enables users to dynamically analyze data that is stored in cubes.

OLAP schema
a group of cubes. A cube is assigned to an OLAP schema when it is created, and an OLAP schema is assigned to a SAS OLAP Server when the server is defined in the metadata. A SAS OLAP Server can access only the cubes that are in its assigned OLAP schema.

OLE (Object Linking and Embedding)
a method of interprocess communication supported by Windows that involves a client/server architecture. OLE enables an object that was created by one application to be embedded in or linked to another application.

OLE DB
an open specification that has been developed by Microsoft for accessing both relational and nonrelational data. OLE DB interfaces can provide much of the same functionality that is provided by database management systems. OLE DB evolved from the Open Data base Connectivity (ODBC) application programming interface. See also OLE (Object Linking and Embedding).

OLE DB for OLAP (ODBO)
an extension to OLE DB that enables users to access multidimensional databases in addition to relational databases. See also OLE DB, OLAP (online analytical processing).

outbound login
a login that applications can retrieve from a SAS Metadata Server and send to other systems that need to verify a user's identity. The login is outbound from the SAS Metadata Server to the other systems. An outbound login must specify an authentication dom ain and must include credentials that are appropriate for the systems to which the login provides access.

package
a container for data that has been generated or collected for delivery to consumers by the SAS Publishing Framework. Packages can contain SAS files (SAS catalogs; SAS data sets; various types of SAS databases, including cubes; and SAS SQL views), binary fi les (such as Excel, GIF, JPG, PDF, PowerPoint and Word files), HTML files (including ODS output), reference strings (such as URLs), text files (such as SAS programs), and viewer files (HTML templates that format SAS file items for viewing). Packages also contain metadata such as a description, an abstract, and user-specified name/value pairs.

page size
the number of bytes of data that SAS moves between external storage and memory in one input/output operation. Page size is analogous to buffer size for SAS data sets.

parallel I/O
a method of input and output that takes advantage of multiple CPUs and multiple controllers, with multiple disks per controller to read or write data in independent threads.

parallel processing
a method of processing that uses multiple CPUs to process independent threads of an application's computations. See also threading.

partition
part or all of a logical file that spans devices or directories. In the SPD Engine, a partition is one physical file. Data files, index files, and metadata files can all be partitioned, resulting in data partitions, index partitions, and metadata partitions, respectively. Partitioning a file can improve performance for very large data sets. See also data partition, partitioned data set.

partitioned data set
in the SPD Engine, a data set whose data is stored in multiple physical files (partitions) so that it can span storage devices. One or more partitions can be read in parallel by using threads. This improves the speed of I/O and processing for very large data sets. See also parallel processing, partition, thread.

permanent package
a container for content that was produced by a SAS program or by a third-party application, and that is written to a specific location. Permanent packages remain in existence even after the stored process completes execution and the client disconnects from the server. See also transient package.

permanent result package
See permanent package.

planned installation
a method of installing a SAS business intelligence system. This type of installation requires a planning file that contains information about the different hosts that are included in the system, the software to be installed on each host, and the SAS servers that should be configured on each server-tier host. The planning file then serves as input to an installation tool called the SAS Software Navigator and to a configuration tool called the SAS Configuration Wizard. See also basic installation.

planning file
an XML file that contains a list of the products to be installed and the components to be configured at a site. This file serves as input to both the SAS Software Navigator and the SAS Configuration Wizard.

pool
a group of server connections that can be shared and reused by multiple client applications. A pool consists of one or more puddles. See also puddle.

pooling
the act or process of creating a pool. See pool.

portlet
a Web component that is managed by a Web application and which is aggregated with other portlets to form a page within the application. A portlet processes requests from the user and generates dynamic content.

pre-installation checklist
a checklist that enumerates the tasks a customer must perform before installing the business intelligence platform. The primary task is to create a set of operating system user accounts on the metadata server host. See also metadata server.

process flow
See process flow diagram.

process flow diagram
in SAS Data Integration Studio, a diagram in the Process Editor that specifies the sequence of each source, target, and process in a job. In the diagram, each source, target, and process has its own metadata object. Each process in the diagram is specified by a metadata object called a transformation.

production environment
a computing environment in which previously tested and validated software is used (typically on a daily basis) by its intended consumers. See also development environment, testing environment.

project repository
a repository that must be dependent on a foundation repository or custom repository that will be managed by the Change Management Facility. A project repository is used to isolate changes from a foundation repository or from a custom repository. The project repository enables metadata programmers to check out metadata from a foundation repository or custom repository so that the metadata can be modified and tested in a separate area. Project repositories provide a development/testing environment for customers who want to implement a formal change management scheme. See also custom repository, foundation repository.

promotion
See metadata promotion.

Public Kiosk
a public page that is displayed when a user starts the SAS Information Delivery Portal but has not yet logged on.

publish
to deliver electronic information, such as SAS files (including SAS data sets, SAS catalogs, and SAS data views), other digital content, and system-generated events to one or more destinations. These destinations can include e-mail addresses, message queues, publication channels and subscribers, WebDAV-compliant servers, and archive locations.

puddle
a group of servers that are started and run using the same login credentials. Each puddle can also allow a group of clients to access the servers. See also pool.

RAID (redundant array of independent disks)
a type of storage system that comprises many disks and which implements interleaved storage techniques that were developed at the University of California at Berkeley. RAIDs can have several levels. For example, a level-0 RAID combines two or more hard drives into one logical disk drive. Various RAID levels provide various levels of redundancy and storage capability. A RAID provides large amounts of data storage inexpensively. Also, because the same data is stored in different places, I/O operations can overlap, which can result in improved performance. See also redundancy.

redundancy
a characteristic of computing systems in which multiple interchangeable components are provided in order to minimize the effects of failures, errors, or both. For example, if data is stored redundantly (in a RAID, for example), then if one disk is lost, the data is still available on another disk. See also RAID (redundant array of independent disks).

relational database management system
a database management system that organizes and accesses data according to relationships between data items. The main characteristic of a relational database management system is the two-dimensional table. Examples of relational database management systems are DB2, Oracle, SYBASE, and Microsoft SQL Server.

Remote Library Services (RLS)
a feature of SAS/SHARE and SAS/CONNECT software that enables you to read, write, and update remote data as if it were stored on the client. RLS can be used to access SAS data sets on computers that have different architectures. RLS also provides read-only access to some types of SAS catalog entries on computers that have different architectures. See also architecture.

replication
See metadata replication.

repository access control template
the access control template (ACT) that controls access to a particular repository and to resources for which access controls are not specified. You can designate one repository ACT for each metadata repository. The repository ACT is also called the default ACT.

resource
any object that is registered in a metadata repository. For example, a resource can be an application, a data store, a dimension in an OLAP cube, a metadata item, an access control template, or a password.

resource template
an XML file that specifies the information that is needed for creating a metadata definition for a SAS resource.

result type
the kind of output that is produced by a stored process. Result types include none, streaming, permanent package, and transient package.

RMI (remote method invocation)
a Java programming feature that provides for remote communication between programs by enabling an object that is running in one Java Virtual Machine (JVM) to invoke methods on an object that is running in another JVM, possibly on a different host. See also JVM (Java Virtual Machine).

SAS application server
a server that provides SAS services to a client. In the SAS Open Metadata Architecture, the metadata for a SAS application server specifies one or more server components that provide SAS services to a client.

SAS batch server
in general, a SAS application server that is running in batch mode. In the SAS Open Metadata Architecture, the metadata for a SAS batch server specifies the network address of a SAS Workspace Server, as well as a SAS start command that will run jobs in batch mode on the SAS Workspace Server.

SAS data set
a file whose contents are in one of the native SAS file formats. There are two types of SAS data sets: SAS data files and SAS data views. SAS data files contain data values in addition to descriptor information that is associated with the data. SAS data views contain only the descriptor information plus other information that is required for retrieving data values from other SAS data sets or from files whose contents are in other software vendors' file formats. See also descriptor information.

SAS format
a pattern or set of instructions that SAS uses to determine how the values of a variable (or column) should be written or displayed. SAS provides a set of standard formats and also enables you to define your own formats.

SAS Foundation Services
a set of core infrastructure services that programmers can use in developing distributed applications that are integrated with the SAS platform. These services provide basic underlying functions that are common to many applications. These functions include making client connections to SAS application servers, dynamic service discovery, user authentication, profile management, session context management, metadata and content repository access, activity logging, event management, information publishing, and s tored process execution. See also service.

SAS informat
a pattern or set of instructions that SAS uses to determine how data values in an input file should be interpreted. SAS provides a set of standard informats and also enables you to define your own informats.

SAS Information Maps
See information map.

SAS log
a file that contains a record of the SAS statements that you enter as well as messages about the execution of your program.

SAS Management Console
a Java application that provides a single user interface for performing SAS administrative tasks.

SAS Metadata Model
a collection of metadata types that are used for saving information about application elements.

SAS Metadata Repository
one or more files that store metadata about application elements. Users connect to a SAS Metadata Server and use the SAS Open Metadata Interface to read metadata from or write metadata to one or more SAS Metadata Repositories. The metadata types in a SAS Metadata Repository are defined by the SAS Metadata Model.

SAS Metadata Server
a multi-user server that enables users to read metadata from or write metadata to one or more SAS Metadata Repositories. The SAS Metadata Server uses the Integrated Object Model (IOM), which is provided with SAS Integration Technologies, to communicate with clients and with other servers.

SAS OLAP Cube Studio
a Java interface for defining and building OLAP cubes in SAS System 9 or later. Its main feature is the Cube Designer wizard, which guides you through the process of registering and creating cubes.

SAS OLAP Server
a SAS server that provides access to multidimensional data. The data is queried using the multidimensional expressions (MDX) language.

SAS Open Metadata Architecture
a general-purpose metadata management facility that provides metadata services to SAS applications. The SAS Open Metadata Architecture enables applications to exchange metadata, which makes it easier for these applications to work together.

SAS procedure
a program that produces reports, manages files, or analyzes data and which is accessed with a PROC statement. Many procedures are included in SAS software.

SAS Report Model
an XML specification that defines a standard reporting format and provides common reporting functions for SAS applications.

SAS statement
a string of SAS keywords, SAS names, and special characters and operators that instructs SAS to perform an operation or that gives information to SAS. Each SAS statement ends with a semicolon.

SAS Stored Process
a SAS program that is stored in a central location and which can be executed from the SAS Information Delivery portal at the user's request. When a stored process is executed, it creates a report that includes the most current data that is available. Stored processes can display input forms that enable users to customize the contents of reports.

SAS Stored Process Server
a SAS IOM server that is launched in order to fulfill client requests for SAS Stored Processes. See also IOM server.

SAS system option
an option that affects the processing of an entire SAS program or interactive SAS session from the time the option is specified until it is changed. Examples of items that are controlled by SAS system options include the appearance of SAS output, the handling of some files that are used by SAS, the use of system variables, the processing of observations in SAS data sets, features of SAS initialization, and the way SAS interacts with your host operating environment.

SAS table
another term for SAS data set. See SAS data set.

SAS Workspace Server
a SAS IOM server that is launched in order to fulfill client requests for IOM workspaces. See also IOM server, workspace.

SAS/ACCESS software
a group of software interfaces, each of which makes data from a particular external database management system (DBMS) directly available to SAS, as well as making SAS data directly available to the DBMS.

SAS/CONNECT server
a server that provides SAS/CONNECT services to a client. When SAS Data Integration Studio generates code for a job, it uses SAS/CONNECT software to submit code to remote computers. SAS Data Integration Studio can also use SAS/CONNECT software for interactive access to remote libraries.

SAS/SHARE server
the result of an execution of the SERVER procedure, which is part of SAS/SHARE software. A server runs in a separate SAS session that services users' SAS sessions by controlling and executing input and output requests to one or more SAS data libraries.

scalability
the ability of a software application to function well with little degradation in performance despite changes in the volume of computations or operations that it performs and despite changes in the computing environment. Scalable software is able to take full advantage of increases in computing capability such as those that are provided by the use of SMP hardware and threaded processing. See also SMP (symmetric multiprocessing).

Scalable Performance Data Engine
See SPD (Scalable Performance Data) Engine.

server component
in SAS Management Console, a metadata object that specifies information about how to connect to a particular kind of SAS server on a particular computer.

server tier
in a SAS business intelligence system, the tier in which the SAS servers execute. Examples of such servers are the SAS Metadata Server, the SAS Workspace Server, the SAS Stored Process Server, and the SAS OLAP Server. These servers are typically accessed either by clients or by Web applications that are running in the middle tier.

service
one or more application components that an authorized user or application can call at any time to provide results that conform to a published specification. For example, network services transmit data or provide conversion of data in a network, database services provide for the storage and retrieval of data in a database, and Web services interact with each other on the World Wide Web. See also SAS Foundation Services.

servlet
a Java program that runs on a Web server. Servlets can be considered a complementary technology to applets, which run in Web browsers. Unlike applet code, servlet code does not have to be downloaded to a Web browser. Instead, servlets send HTML or other appropriate content back to a browser or to another type of Web-based client application.

servlet container
an execution environment for Java servlets that contains a Java Virtual Machine. The servlet container also provides other services for servlets and for the Web applications that those servlets are part of. For example, the servlet container converts HTTP requests that are sent by clients to Java objects that servlets can work with, and it converts the output of servlets to HTTP responses. An example of a popular servlet container is the Apache Tomcat server.

single sign-on
an authentication model that enables users to access a variety of computing resources without being repeatedly prompted for their user IDs and passwords. For example, single sign-on can enable a user to access SAS servers that run on different platforms without interactively providing the user's ID and password for each platform. Single sign-on can also enable someone who is using one application to launch other applications based on the authentication that was performed when the user initially logged on.

SMP (symmetric multiprocessing)
a hardware and software architecture that can improve the speed of I/O and processing. An SMP machine has multiple CPUs and a thread-enabled operating system. An SMP machine is usually configured with multiple controllers and with multiple disk drives per controller.

source metadata server
the metadata server from which metadata is promoted or replicated. See also metadata promotion, metadata replication, target metadata server.

spawner
See object spawner.

SPD (Scalable Performance Data) Engine
a SAS engine that is able to deliver data to applications rapidly because it organizes the data into a streamlined file format. The SPD Engine also reads and writes partitioned data sets, which enable it to use multiple CPUs to perform parallel I/O functions. See also parallel I/O.

spooling
the process of saving data that has been read to a temporary disk location so that computer resources are available to perform other tasks.

SQL (Structured Query Language)
a standardized, high-level query language that is used in relational database management systems to create and manipulate database management system objects.

star schema
tables in a database in which a single fact table is connected to multiple dimension tables. This is visually represented in a star pattern. SAS OLAP cubes can be created from a star schema.

stored process
See SAS Stored Process.

streaming result
a type of output that is generated by a stored process. In a streaming result, the content that the stored process generates is delivered to the client through an output stream. The output stream is generally accessible to the stored process as the _WEBOUT fileref. See also result type.

subscribe
to sign up to receive electronic content that is published to a SAS publication channel.

target metadata server
the metadata server to which the metadata is promoted or replicated. See also metadata promotion, metadata replication, source metadata server.

testing environment
a computing environment in which application developers typically use real-life data and scenarios to test software that has been migrated from a development environment. See also development environment, production environment.

thin client
an application that is deployed across a network, thereby reducing the need for disk space on client machines. Thin-client development tools reduce the cost of deploying and maintaining applications. Costs are lower because thin-client applications need to be updated only on the server. Otherwise, multiple user machines that perhaps run multiple operating systems would have to be updated.

thread
a single path of execution of a process in a single CPU, or a basic unit of program execution in a thread-enabled operating system. In an SMP environment, which uses multiple CPUs, multiple threads can be spawned and processed simultaneously. Regardless of whether there is one CPU or many, each thread is an independent flow of control that is scheduled by the operating system. See also SMP (symmetric multiprocessing), thread-enabled operating system, threading.

thread-enabled operating system
an operating system that can coordinate symmetric access by multiple CPUs to a shared main memory space. This coordinated access enables threads from the same process to share data very efficiently.

threading
a high-performance method of data I/O or data processing in which the I/O or processing is divided into multiple threads that are executed in parallel. In the boss-worker model of threading, the same code for the I/O or calculation process is executed simultaneously in separate threads on multiple CPUs. In the pipeline model, a process is divided into steps, which are then executed simultaneously in separate threads on multiple CPUs. See also parallel I/O, parallel processing, SMP (symmetric multiprocessing).

throughput
the rate at which requests for work are serviced by a computer system.

transformation
in SAS Data Integration Studio, a metadata object that specifies how to extract data, transform data, or load data into data stores. Each transformation that you specify in a process flow diagram generates or retrieves SAS code. You can specify user-written code in the metadata for any transformation in a process flow diagram.

transient package
a container for content that was produced by a SAS program or by a third-party application for immediate use, and that is not saved. After the client program disconnects from the server, the transient package disappears. See also permanent package.

transient result package
See transient package.

trusted user
a special user of a metadata server who can acquire credentials on behalf of other users in a multi-tier server environment.

tuple
a data object that contains two or more components. In OLAP, a tuple is a slice of data from a cube. It is a selection of members (or cells) across dimensions in a cube. It can also be viewed as a cross-section of member data in a cube. For example, ([time].[all time].[2003], [geography].[all geography].[u.s.a.], [measures].[actualsum]) is a tuple that contains data from the Time, Geography, and Measures dimensions.

unrestricted user
a special user of a metadata server who can access all metadata on the server (except for passwords, which an unrestricted user can overwrite but cannot read). An unrestricted user can also perform administrative tasks such as starting, stopping, pausing, and refreshing the metadata server. You are an unrestricted user if your user ID is listed in the adminUsers.txt file and is preceded by an asterisk.

URL (Uniform Resource Locator)
a character string that is used by a Web browser or other software application to access or identify a resource on the Internet or on an intranet. The resource could be a Web page, an electronic image file, an audio file, a JavaServer page, or any other ty pe of electronic object. The full form of a URL specifies which communications protocol to use for accessing the resource, as well as the directory path and filename of the resource.

Web application
a J2EE application that can execute in a servlet container. Such applications are distributed as Web application archive (WAR) files and can include servlets, JavaServer pages, JavaBeans, and HTML pages.

Web browser
a software application that is used to present Web content. To accomplish this task, the browser submits URL (Universal Resource Locator) requests to a Web server and handles any results that the request generates.

Web Distributed Authoring and Versioning
See WebDAV (Web Distributed Authoring and Versioning).

Web Infrastructure Kit
a set of infrastructure components that can be used to develop new portlets for the SAS Information Delivery Portal, to customize the SAS Information Delivery Portal, or to build new Web applications using portal technology. The kit includes common Java components as well as SAS Foundation Services. It is included with SAS Integration Technologies.

Web server
a server machine and software that enable organizations to share information through intranets and through the Internet.

WebDAV (Web Distributed Authoring and Versioning)
a set of extensions to the HTTP protocol that enables users to collaboratively edit and manage files on remote Web servers.

WebDAV repository
a collection of files that are stored on a Web server so that authorized users can read and edit them. See also WebDAV (Web Distributed Authoring and Versioning).

WebDAV server
an HTTP server that supports the collaborative authoring of documents that are located on the server. The server supports the locking of documents, so that multiple authors cannot make changes to a document at the same time. It also associates metadata with documents in order to facilitate searching. The SAS business intelligence applications use this type of server primarily as a report repository. Common WebDAV servers include the Apache HTTP Server (with its WebDAV modules enabled), Xythos Software's Web File Server, and Microsoft Corporation's Internet Information Server (IIS).

workspace
in the IOM object hierarchy for a SAS workspace server, an object that represents a single session in SAS. See also IOM (Integrated Object Model).

XML (Extensible Markup Language)
a markup language that structures information by tagging it for content, meaning, or use. Structured information contains both content (for example, words or numbers) and an indication of what role the content plays. For example, content in a section heading has a different meaning from content in a database table.

Index

A

B

C

D

E

F

G

H

I

J

L

M

N

O

P

R

S

T

U

V

W

X

Z

Your Turn

If you have comments or suggestions about *SAS® 9.1.3 Intelligence Platform: Administration Guide, Fifth Edition*, please send them to us on a photocopy of this page, or send us electronic mail.

For comments about this book, please return the photocopy to

SAS Publishing
SAS Campus Drive
Cary, NC 27513
E-mail: **yourturn@sas.com**

For suggestions about the software, please return the photocopy to

SAS Institute Inc.
Technical Support Division
SAS Campus Drive
Cary, NC 27513
E-mail: **suggest@sas.com**