SAS Publishing

SAS® 9.1 Intelligence Architecture: Planning and Administration Guide

The Power to Know.

The correct bibliographic citation for this manual is as follows: SAS Institute Inc. 2003. *SAS 9.1 Intelligence Architecture: Planning and Administration Guide*. Cary, NC: SAS Institute Inc.

SAS 9.1 Intelligence Architecture: Planning and Administration Guide

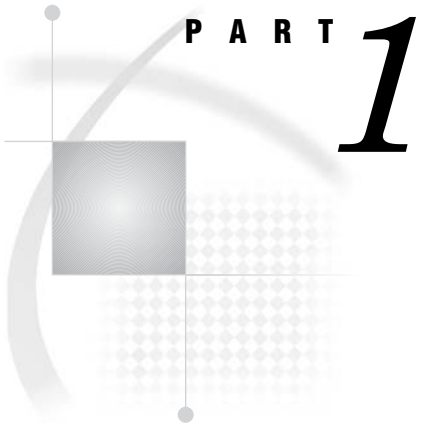
SAS Institute Inc., SAS Campus Drive, Cary, North Carolina 27513.

1st printing, November 2003
2nd printing, January 2004

3rd printing, February 2004

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

SAS Intelligence Architecture

Overview of the SAS Intelligence Architecture

Your job is to design an enterprise intelligence infrastructure, perhaps one that your company can use to build data warehouses and data marts and to query those warehouses and marts. This means that you must put in place a series of applications and the components and services on which those applications depend. Because you have chosen to use SAS software, this task will be considerably easier than it would have been if you had purchased applications from a number of vendors.

Why?

First, the products that you purchased from SAS have been designed to work together and to provide a complete solution. That is, SAS provides an application for each step in an intelligence solution, from extracting data from operational data sources and integrating it in a data warehouse to enabling a user to view data from the warehouse in a Web browser. You simply install and configure the appropriate set of applications, taking advantage of the SAS Intelligence Value Chain. For more information, see "SAS Intelligence Value Chain" on page 4.

Second, all of these applications are built using what SAS calls the SAS Intelligence Architecture. This architecture specifies a set of software components and services that are used by all applications. Because applications share this infrastructure, you do not have to learn about and manage application-specific resources. For more information, see "SAS Intelligence Architecture" on page 7.

SAS Intelligence Value Chain

The SAS Intelligence Value Chain represents the links required to build an entire intelligence solution. Each link in the chain, except the planning link, corresponds to a set of SAS products. This means that you deal with a single vendor and, most importantly, you can run all of your applications within a single framework.

There are five key components in the SAS Intelligence Value Chain.

Figure 1.1 SAS Intelligence Value Chain

Note: Not all solutions require products from each link. △

The *Plan* link of the SAS Intelligence Value Chain delivers proven, best-practice roadmaps that are supported by integrated, industry-specific analytic models, project methodologies, and consulting expertise. Therefore, you can create and deploy custom solutions reliably because these processes and services embody SAS experience from hundreds of business cases in areas such as finance, telecommunications, retail, pharmaceuticals, and manufacturing. The Plan link's customized models and services enable you to structure project deployment for maximum efficiency and consistency, to reduce implementation time and risk, and to quickly deliver solutions that show bottom-line results.

As part of the Plan link, you spend time with a SAS representative to decide on the platforms and SAS software products that you will need to build your intelligence solution. Based on the information needs of your users and your IT infrastructure, you should consider such things as how you will store the data that makes up your data warehouse and data marts, how you will query these data stores, and how information consumers will access data. Your planning also typically includes building data models for a data warehouse and data marts. If you do build such models, you can later import information about those models into SAS's data warehousing solution.

The *ETL*[Q] link is the stage in which you create a data warehouse by integrating data in existing operational data sources, such as SAS data sets, DBMS tables, and Enterprise Resource Planning (ERP) systems. The software components in this link enable you to perform the following tasks:

- *Extract* data from the data sources mentioned previously, regardless of the platform on which the data sources reside or the format of the data.
- *Transform* the data before writing it to target data sources. For example, you might change the structure of your data by joining the contents of several tables into one table.
- *Load* the transformed data into the data warehouse.
- Ensure the *Quality* of the data to be loaded into the warehouse by providing a product that can review and "cleanse" the data so that it is accurate, up-to-date, and consistently represented.

The *Intelligent Storage* link is the stage in which you determine how to store your data to achieve the best performance. Your storage options include SAS or third-party relational databases, parallel storage, and multidimensional databases. Or you can combine any of these storage structures to satisfy your company's business requirements.

The *Business Intelligence* link consists of a set of enterprise-wide query and reporting tools and interfaces that enable different types of users to surface meaningful intelligence from consistent, company-wide data. Multiple clients surface interfaces targeted at various user skill levels and needs, enabling users to generate their own answers while IT retains control over the quality and consistency of the data they are using. The products in this area also enable you to build Web portals that guide users to the information they need.

Finally, the *Analytic Intelligence* link involves such capabilities as predictive and descriptive modeling, forecasting, optimization, simulation, experimental design, and more. In this document, we focus on data mining as part of an intelligence solution.

Note: In addition to the products that enable you to build an intelligence system, you will make extensive use of a management application called SAS Management Console. You use this application to define metadata for such entities as users, data, and servers and to control the operation of a metadata server. △

As you build your intelligence system, you will be installing and configuring a number of SAS products. The following sections describe the products that you will be working with at each stage in the SAS Intelligence Value Chain.

Plan Link

During the planning stage, you and your SAS representative will use a planning application that is available from the SAS Web site; you will not be installing any SAS products yet. This application enables you to capture all of the decisions you and your SAS representative make about the platforms on which you will host your solution and about the SAS software technology packages and products that your solution will require. The output of this program is a planning file called **`plan.xml`**. This file later becomes the input to the installation software that you use to build your system. Having this planning file available makes the installation software much smarter—and your installation experience much simpler.

Note: It is also possible to perform unplanned installations, called *basic installs*. △

If your planning activities include data modeling, you will probably use a third-party product such as AllFusion ERwin Data Modeler to build the model. You can then export the metadata for your model to a Common Warehouse Metamodel-compliant XML file.

You can subsequently import that metadata into a SAS metadata repository where it can be used by products like SAS ETL Studio.

ETLQ Link

The main component in this area is SAS ETL Studio. SAS ETL Studio is a Java application that enables you to perform the tasks in the ETL^{Q} link of the SAS Intelligence Value Chain: the extraction of data from operational data stores, the transformation of this data, and the loading of the extracted data into your data warehouse. This application actually extends into the Intelligent Storage link, because it enables you to design the flow of data into SAS data sets, Online Analytical Processing (OLAP) cubes, or third-party relational database tables.

Note: SAS OLAP Cube Studio also enables you to build and maintain OLAP cubes. △

A number of products augment the capabilities of SAS ETL Studio. For example, the SAS/ACCESS interfaces to relational databases enable you to read, write, and update data regardless of its native database and platform. And the SAS Data Surveyors enable you to build SAS ETL Studio jobs that help you read data from SAP, Siebel, Oracle, and other Enterprise ERP vendors.

There are also several components that enable you to improve the quality of your data. For instance, the SAS Data Quality Server allows you to analyze, cleanse, and standardize your data. This product is often used in conjunction with products like dfPower Studio from DataFlux Corporation, which enables you to customize the Quality Knowledge Base that the SAS Data Quality Server uses to store its data-cleansing guidelines.

Finally, the products Platform LSF and Platform JobScheduler enable you to schedule the execution of a set of SAS ETL Studio jobs.

Intelligent Storage Link

Providing intelligent storage means providing storage solutions to meet a variety of needs. SAS provides intelligent storage by supporting the following:

- SAS data sets, which are analogous to relational database tables, and third-party relational database management systems
- OLAP cubes
- Scalable Performance Data Engine (SPD Engine) tables.

SAS data sets and OLAP cubes (multidimensional databases) are managed by Base SAS and the SAS OLAP Server, respectively. SPD Engine tables are managed by the SPD Engine. This engine enables you to store and retrieve large quantities of data at very high rates of speed. This bulk loading and reading of data is made possible through parallel processing. Large SAS data sets can be partitioned, and separate I/O streams can be initiated on multiple threads, which might execute on separate processors.

Note: SPD Engine is part of Base SAS. A higher-performance, multiuser version of this product is available in the Intelligent Storage technology package. It is called the Scalable Performance Data Server (SPDS). △

Business Intelligence Link

The products in the Business Intelligence link in the SAS Intelligence Value Chain enable you to explore the data in a data warehouse or data mart and to control the

presentation of the results in business reports. SAS has several tools that provide data access, data visualization, and information access. Two such products are SAS Enterprise Guide and the SAS Information Delivery Portal:

- SAS Enterprise Guide is a Microsoft Windows application for analyzing data and generating analytical reports. This application also enables users to create SAS Stored Processes and to store that code in a repository that is available to a SAS Stored Process Server. Stored processes are similar to SAS programs run in batch mode, with the added benefit that they can accept input parameters. Stored processes are used for Web reporting and analytics, among other things.
- SAS Information Delivery Portal is a Java 2 Enterprise Edition (J2EE) Web application that enables you to aggregate data from a variety of sources and present them to the user in a Web browser. The content can include the output of SAS Stored Processes, links to Web addresses, documents, and syndicated content from information providers.

Analytic Intelligence Link

SAS has multiple solutions for analytic intelligence for areas such as the following:

- Enterprise Intelligence
- Supplier Intelligence
- Organizational Intelligence
- Customer Intelligence
- Supply Chain Intelligence.

In this document, we deal primarily with SAS Enterprise Miner. This product provides a complete set of data mining tools for data preparation and visualization, predictive modeling, clustering, association discovery, model management, model assessment, and reporting.

SAS Management Console

SAS also provides a tool that you use to manage your entire intelligence solution, SAS Management Console. Before you can envision the role of SAS Management Console, it is important to understand that most of the objects involved in your SAS system are described by *metadata*. These objects include everything from SAS servers to users to libraries and tables. This metadata is stored in one or more metadata repositories, and access to these repositories is controlled by a metadata server.

SAS Management Console enables you to manage the metadata server and to create metadata. Among other tasks, SAS Management Console enables you to

- create metadata repositories and operate the metadata server
- create users and groups, and specify their access rights to metadata and other resources
- define SAS application servers
- define data libraries
- control the scheduling and execution of SAS ETL Studio jobs.

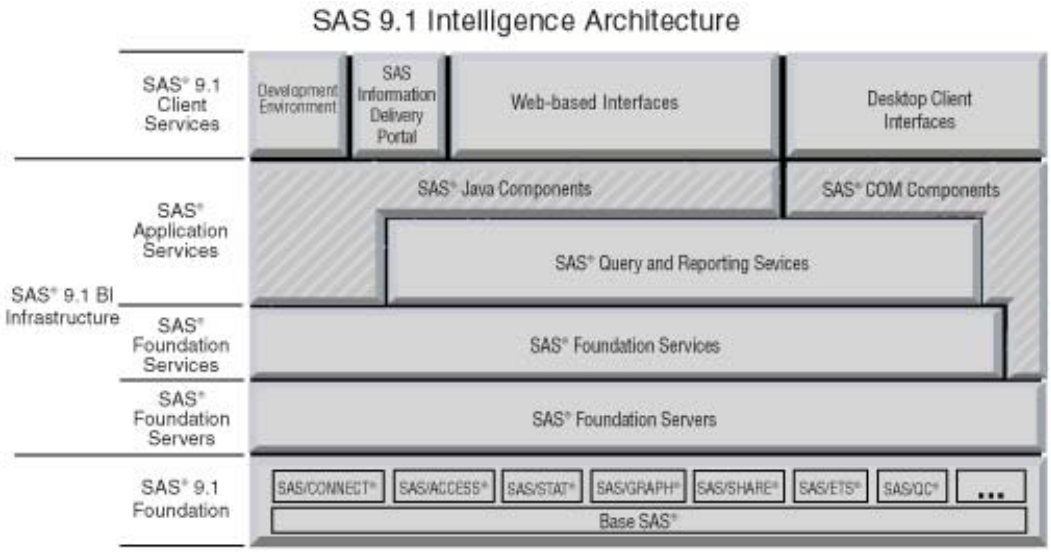
SAS Intelligence Architecture

SAS creates and delivers enterprise intelligence through the SAS Intelligence Architecture. This architecture fully integrates world-class SAS technology in data

extraction, transformation, and loading; data storage; business intelligence; and analytic intelligence into a single cohesive platform. These capabilities provide the end-to-end infrastructure necessary to ensure consistent and reliable enterprise-wide intelligence that is needed for exploring, analyzing, optimizing, and understanding your organization.

The SAS Intelligence Architecture is built to provide enterprise-class performance. By taking advantage of the multiple functional layers of the new n-tier architecture, SAS can efficiently process large amounts of data while simultaneously delivering relevant content to users throughout the enterprise. See Figure 1.2.

Figure 1.2 SAS 9.1 Intelligence Architecture



SAS 9.1 Foundation

The SAS 9.1 Foundation layer (also referred to as SAS 9.1) consists of SAS products like Base SAS, SAS/CONNECT, SAS/GRAPH, SAS/ACCESS, SAS/STAT, SAS/ETS, SAS/OR, SAS/QC, and others in the SAS product line that deliver the data processing and statistical and analytical power of SAS. These products provide a broad range of core data manipulation functions, such as distributed data management, data access across multiple database sources, data visualization, data mining, and advanced analytical modeling applications.

SAS 9.1 Business Intelligence Infrastructure

The SAS 9.1 Business Intelligence Infrastructure (BI Infrastructure) layer provides a suite of servers and services that deliver *The Power to Know* throughout the enterprise. With the BI Infrastructure, SAS can be deployed in multi-tier environments where Web servers and application servers operate.

SAS Foundation Servers

The servers in the BI Infrastructure include the following:

- *SAS Metadata Server*
  The SAS Metadata Server enables centralized, enterprise-wide metadata delivery and management: one metadata server provides metadata to SAS applications across the enterprise.
- *SAS OLAP Server*
  The SAS OLAP Server delivers pre-summarized "cubes" of data to OLAP clients like SAS Enterprise Guide using OLE DB for OLAP. The SAS OLAP Server is a multidimensional database server that is designed to reduce the load on traditional back-end storage systems by delivering different summarized views of data to business intelligence applications, irrespective of the amount of data underlying these summaries.
- *SAS Stored Process Server*
  The SAS Stored Process Server executes and delivers results from SAS Stored Processes in a multi-client environment. A SAS Stored Process is a SAS program that can be called through the SAS Stored Process Server. Using the SAS Stored Process Server, clients can execute parameterized SAS programs without having to know the SAS language.
- *SAS Workspace Server*
  The SAS Workspace Server surfaces the SAS programming environment through an API to calling clients.

SAS Foundation Services

This suite of Java-based APIs provides core middleware infrastructure services including user authentication, profile management, session management, activity logging, metadata and content repository access, and connection management. Extension services for information publishing, event management, and SAS Stored Process execution are also provided.

SAS Application Services

SAS Application Services provide business-oriented query and reporting services to calling clients. By using a business metadata layer and a universal report definition, SAS Query and Reporting Services provide a solid foundation for enterprise reporting and application development. Java and COM-based interfaces to SAS Application Services surface to clients the functionality provided by SAS Query and Reporting Services. SAS Application Services can also be used by application developers to provide custom business intelligence capabilities within their solutions.

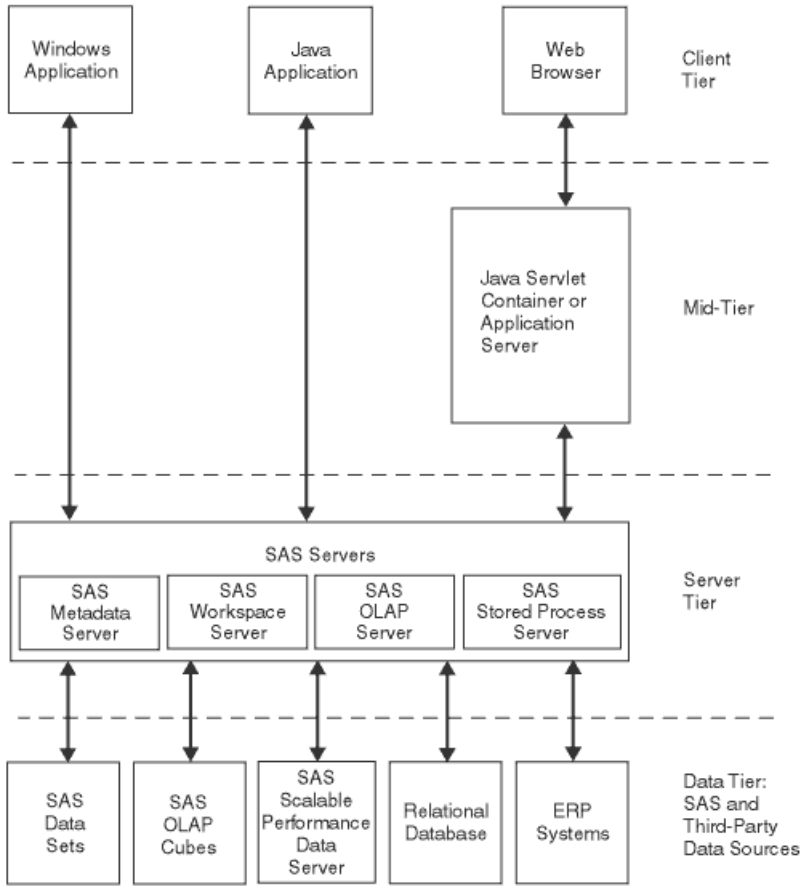
SAS 9.1 Client Services

The SAS 9.1 Client Services layer provides a suite of Web-based and desktop front-end interfaces to the content and applications generated from the SAS BI Infrastructure and the SAS Foundation. In today's business environment, organizations need to allow all levels of decision makers direct access to information to improve decision making and enhance operational effectiveness. SAS Client Services can provide centralized access to content, appropriate query and reporting interfaces, and business intelligence functionality to all decision makers within an enterprise—from the CEO to business analysts to customer service agents.

Architecture from an Administrator's Point of View

From your perspective as an administrator, it might be helpful to think about the SAS applications as n-tiered applications and to envision the tiers at which different application components must be installed. See Figure 1.3.

Figure 1.3 Tiers in the Architecture



Client Tier

In the client tier, you install the portion of the application that presents a user interface to the user. This could involve installing a Windows application, a Java application, or a Web browser (if one is not already installed).

The following client runs only on Microsoft Windows systems:

- SAS Enterprise Guide.

The Java applications include the following:

- SAS Management Console
- SAS ETL Studio
- SAS OLAP Cube Studio
- SAS Enterprise Miner and other SAS Analytic Intelligence solutions.

You can run SAS Management Console on Windows systems and several UNIX platforms. The remaining applications are supported on Windows systems. All of these applications require the Java Runtime Environment (JRE), which includes a Java Virtual Machine (JVM) that executes the application and a set of standard Java class libraries. If you have installed the SAS 9.1 Foundation on a host, the JRE will already be present on that machine. Otherwise, you can install the JRE from a CD of third-party products before you install the first Java client.

For the following applications, you only install a Web browser on each client machine:

- SAS Information Delivery Portal
- SAS Stored Process Web Application
- Webdoc.

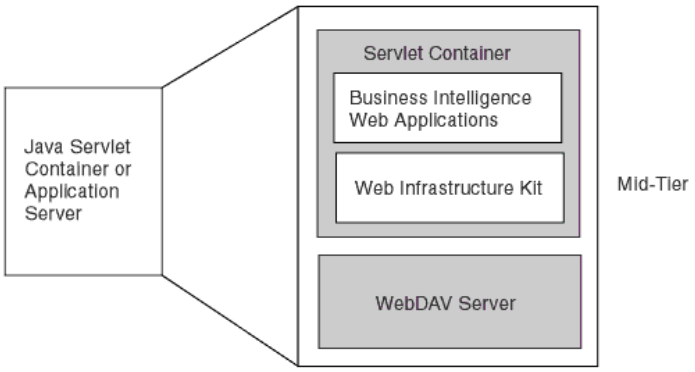
These products run in a J2EE servlet container on the mid-tier. They communicate with the user by sending data to and receiving data from the user's Web browser. For example, an application of this type displays its user interface by sending an HTML document to the user's browser. The user can submit input to the application by sending it an HTTP response—usually by clicking a link or submitting an HTML form.

Mid-Tier

The mid-tier provides an environment where the business intelligence Web applications, such as the SAS Information Delivery Portal, can execute. This environment includes the following elements:

- a J2EE servlet container or application server
- the Java 2 Software Development Kit, Standard Edition (J2SE SDK)
- a WebDAV server
- the SAS Web Infrastructure Kit.

Figure 1.4 Mid-Tier Components



The servlet container and the SDK are required because the products previously mentioned are written using JavaServer Pages and servlets. At runtime, the JavaServer Pages are translated to servlets, which must be compiled before execution. You need the SDK, which includes a Java compiler, to compile the servlets. The servlet container provides the execution environment for the compiled servlets.

The WebDAV server stores documents that users might want to access through the SAS Information Delivery Portal. The server acts like a network accessible file system

and can serve not only HTML and XML documents, but also items such as Microsoft Word documents.

The SAS Web Infrastructure Kit is a SAS product that provides certain services to Web applications. These services handle such tasks as the following:

- user logon and logoff
- page navigation
- searching
- integration with SAS via stored processes
- interacting with basic content types.

Server Tier

The server tier consists of a set of SAS servers. Each server exposes a different set of Integrated Object Model (IOM) interfaces and has a different purpose. The servers are listed as follows:

- The SAS Metadata Server controls access to a central repository of metadata, which is shared by all of the applications in the system. This repository contains metadata that represents items such as SAS servers, users, libraries, and tables.
- The SAS Workspace Server executes any type of SAS program.
- The SAS OLAP Server handles MDX (multidimensional expression language) queries.
- The SAS Stored Process Server executes stored processes, which support input parameters.

Data Tier

SAS provides all of the products that you need to build your data tier. As mentioned earlier, SAS provides for intelligent storage by supporting the following:

- SAS data sets, which are analogous to relational database tables
- SAS SPD Engine tables, which can be read or written by multiple threads
- SAS OLAP cubes.

In addition, SAS provides products that enable you to access data in existing third-party DBMSs and ERP systems. The SAS/ACCESS interfaces provide direct access to DBMSs such as the following:

- DB2
- Informix
- MS SQL Server
- Oracle
- Sybase.

The SAS/ACCESS interfaces also provide direct access to ERP systems such as the following:

- SAP
- PeopleSoft
- Baan.

Advantages of Using a SAS Intelligence Solution

If you have built intelligence solutions using earlier versions of SAS software, you will notice a number of improvements in SAS 9.1 systems. These include improvements in the following areas:

- manageability
- scalability
- interoperability
- usability.

Manageability

The main improvement in the area of manageability is that SAS 9.1 includes SAS Management Console, which serves as the central application for managing an intelligence system. SAS Management Console enables you to

- create metadata repositories and operate the SAS Metadata Server
- create users and groups, and specify their access rights to metadata and other resources
- define SAS application servers
- define data libraries
- control the scheduling and execution of SAS ETL Studio jobs.

Scalability

The main improvement in the area of scalability is the use of threading. SAS 9.1 makes better use of multiple machines on a network (load balancing) and of the capabilities of symmetric-multiprocessing machines. Some places in which this threading occurs include the following:

- the SAS Metadata Server and the SAS OLAP Server
- SAS procedures
- SAS/ACCESS engines
- the Scalable Performance Data Engine, which is used to read very large SAS data sets.

There have also been improvements in the area of server configuration. You can create pools of connections to workspace servers so that clients do not need to open a connection to such a server each time they need to execute SAS code. In addition, an object spawner can balance a workload across multiple workspace or stored process servers.

Interoperability

The main improvement in the area of interoperability is that SAS 9.1 supports the SAS Open Metadata Architecture and the Common Warehouse Metamodel (CWM). The SAS Open Metadata Architecture defines a general purpose metadata management facility that provides common metadata services to SAS and other applications. Third parties can access metadata in the SAS Metadata Server using an API that is supplied by SAS. SAS supports CWM as a standard for metadata interchange.

Usability

A good example of the usability improvements in SAS 9.1 is the way that the ETL and business-intelligence applications have been targeted at specific audiences so that a single person (such as an ETL specialist or a web portal designer) needs to use a single interface to do his or her work. For instance, the ETL specialist may only need to learn the SAS ETL Studio interface. Using this client, the ETL specialist can create data warehouses and data marts. Likewise, portal designers will only need the SAS Information Delivery Portal to perform their work.

Deployment Process

We assume that you and a SAS representative have decided what hardware and software you need to build your SAS intelligence system. This section *summarizes* how you will proceed from here to build your system.

As you will see below, there are three general types of activities you must perform:

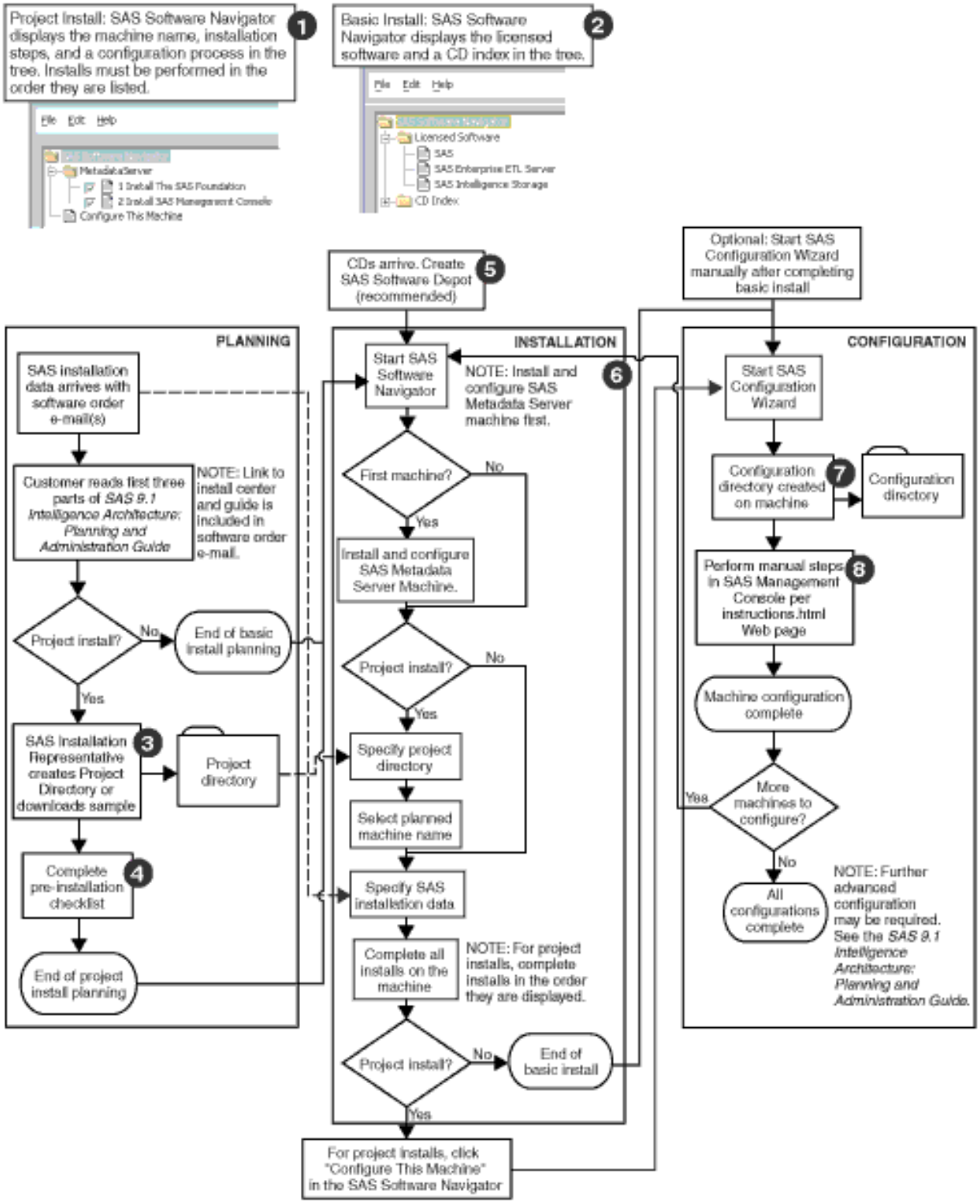
- Planning your installation. We recommend that you and your SAS representative create a planning file, **`plan.xml`**, that contains information about:
  - the different machines in your system
  - the software to be installed on each machine
  - which machine you should install software on first, which you should install software on second, and so forth
  - the order in which you should install products on a particular machine

  This plan serves as input to an installation tool, the SAS Software Navigator, and a configuration tool, the SAS Configuration Wizard.
- Installing software. On each machine in your system, you run a software installation program that, for each product to be installed, provides you with a set of installation instructions and enables you to start the product's installation program. (You should install all of the software on a machine before configuring that software.)
- Configuring the software on a machine. SAS also provides a tool that will assist you in configuration, once you have installed all of the software on a machine. There will be some manual configuration you need to perform as well.

The diagram below shows in flowchart form the procedure you will follow to build your system (and the two types of installs that are possible). The sections that follow the diagram provide annotations for the diagram and direct you to additional information about important topics.

Note: You follow a different procedure to install the SAS 9.1 Foundation for z/OS and to configure the SAS servers on z/OS systems. For installation instructions for this platform, see the Installation Instructions for SAS 9.1 Foundation for z/OS, and for configuration information see Chapter 6, "Pre-Installation Tasks," on page 93 and Chapter 7, "Installing and Configuring Your Software," on page 123. △

Figure 1.5 Planning, Installation, and Configuration Workflow



Types of Installs

There are two types of installs: project installs and basic installs.

❶ In a project install, you have available an XML planning file called **`plan.xml`**. You and your SAS representative might have developed this file using a planning tool available on the SAS Web site, or you might have selected a standard SAS plan. In either case, both the SAS installation tool, the SAS Software Navigator, and the configuration tool, the SAS Configuration Wizard, use this plan as input. When you run the SAS Software Navigator, the plan file tells the installation tool which products need to be installed on the machine and in what order. When you run the SAS Configuration Wizard, the plan file tells the configuration tool which products need to be configured.

We recommend that the initial installation of your system be a project install. You will find detailed instructions for performing a project install in Chapter 6, "Pre-Installation Tasks," on page 93 and Chapter 7, "Installing and Configuring Your Software," on page 123.

❷ A basic install does not require a planning file. Without a planning file, the SAS Software Navigator does not give you a customized list of products to install on a machine. Instead, it gives you two alternatives:

- A Licensed Software folder enables you to select a technology package that you have licensed for the machine on which you are installing software. The SAS Software Navigator will then enable you to install the required products in that package in the correct order. This option is for use by expert installers, because the installer will have to install any non-required components using the method described in the next bullet.
- A CD Index folder enables you to look at the contents of all the CDs from which you can install software on the current machine. You can select any licensed product from any CD and install it. This capability is useful primarily for installing individual products after the initial installation of your system.

For more information about basic installs, see Appendix 2, "Basic Installs," on page 321.

Planning

For a project install, you need to perform the following tasks:

- Place your planning file and other files in a shared directory, called a *project directory*.
- Perform a set of tasks described on a pre-installation checklist.

Note: You also need to perform tasks from a pre-installation checklist if you are performing a basic install. See Appendix 2, "Basic Installs," on page 321 for more information. △

❸ Before you begin a project install, you must create a project directory that contains your planning file. Both the SAS Software Navigator and the SAS Configuration Wizard read files in this directory. For more information about setting up a project directory, see "Setting Up Your Project Directory" on page 94.

❹ You must also fill out a pre-installation checklist (or checklists). This checklist explains the set of tasks you must perform before you begin installing your SAS software. Typical tasks include setting up operating system user accounts where your metadata server will run, and installing the Java 2 SDK, a servlet container, and a WebDAV server. For more information on pre-installation tasks, see the following sections:

- "Pre-Installation Checklists" on page 95

- “Setting Up Required User Accounts” on page 113
- “Preparing to Run Web Applications” on page 116.

Installation

❺ We recommend that before you begin installing your software, you copy the contents of your SAS CDs to the network, creating a SAS Software Depot. This approach is particularly helpful if you will be installing software on a large number of machines. For more information about building a software depot, see “Creating a SAS Software Depot” on page 126.

❻ You then install a set of products on a machine using the SAS Software Navigator. *You begin by installing software on the machine that will host the metadata server* and then proceed to install software on your other computers. For each product that you want to install, the SAS Software Navigator enables you to read a set of installation instructions and to start an installation program. For further information about installing software, see “Installing Software on a Machine” on page 130.

Configuration

Once you have installed all of the software on a machine, you run the SAS Configuration Wizard and use SAS Management Console to configure the software on that machine.

❼ On machines where you install a SAS server or a servlet container, the SAS Configuration Wizard automatically creates a *configuration directory*, which contains a set of directories and files that make it easy for you to manage your system. For more information on this subject, see the following topics:

- “Running the Configuration Wizard on Windows and UNIX Systems” on page 135
- “Configuring SAS Servers on z/OS Systems” on page 141
- “Configuration Directory: Server Tier Machines” on page 148
- “Configuration Directory: Mid-Tier Machines” on page 150.

❽ The SAS Configuration Wizard also produces an HTML document that explains what configuration you need to perform manually after the wizard exits. Often, this configuration involves adding objects to your metadata repository using SAS Management Console. For further information on this subject, see “Performing Manual Configuration Steps” on page 143.

CHAPTER

2

Understanding the SAS Application Servers

Overview of Understanding the SAS Application Servers

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. Four of the SAS application server components are Integrated Object Model (IOM) servers. The IOM provides distributed object interfaces that are based on industry-standard technologies, including Microsoft's Distributed Component Object Model (DCOM) and the Object Management Group's Common Object Request Broker Architecture (CORBA).

Note: DCOM is an extension to the Component Object Model (COM) that enables components to request services from components that are on other computers in a network. CORBA is a standard, widely distributed API for connecting operating system platforms from multiple vendors. △

- The SAS Metadata Server is a multi-user server that enables users to read metadata from or write metadata to one or more SAS Metadata Repositories. SAS Metadata Repositories contain metadata that represents items such as SAS application servers, users in the metadata environment, libraries, tables, stored processes, and cubes.
- The SAS Workspace Server fulfills client requests for a specific SAS session. For example, SAS ETL Studio submits requests to the SAS Workspace Server to initiate SAS sessions for building OLAP cubes that are based on metadata in a SAS Metadata Repository.
- The SAS OLAP Server provides access to cubes, which are logical sets of data that are organized and structured in a hierarchical, multidimensional arrangement. Cubes are queried by using the multidimensional expression (MDX) language.

- The SAS Stored Process Server executes stored processes, which are SAS programs that are stored on a server and can be executed as required by requesting applications.

The other two SAS application server components follow:

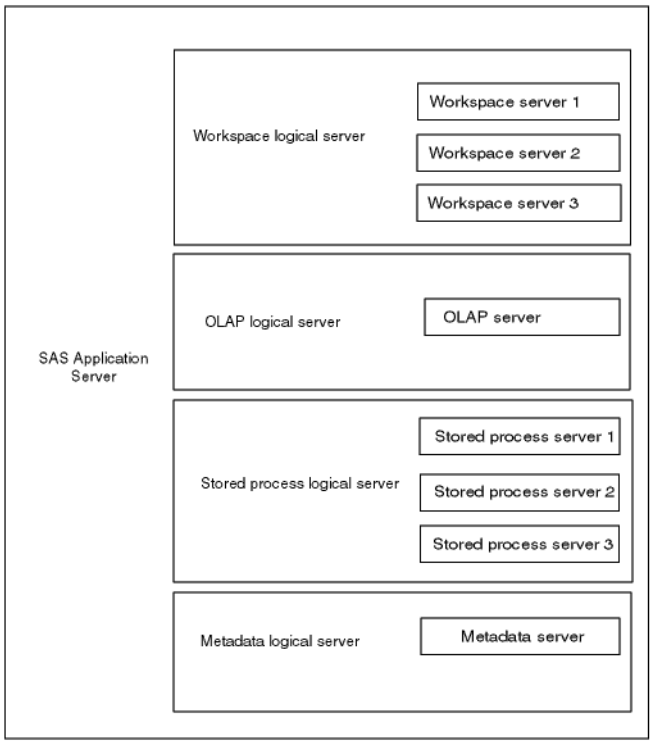
- The SAS batch server, which invokes SAS in batch mode. In the SAS Open Metadata Architecture, the metadata for a SAS batch server specifies
  - the network address of a SAS Workspace Server
  - a SAS start command that will run jobs in batch mode on the SAS Workspace Server.

  For example, in SAS ETL Studio, a SAS batch server is used to deploy jobs for scheduling.
- The SAS/CONNECT server, which runs a SAS session on a computer that receives requests for services from a SAS/CONNECT client. The server can run on a remote, single-processor computer or on a local or remote symmetric multiprocessing (SMP) computer. For example, when SAS ETL Studio generates code for a job, it uses the SAS/CONNECT server to submit code to remote machines.

How SAS Application Server Metadata Is Organized

In order for client applications to have access to the SAS application servers and the resources that they manage, the servers must be defined in a SAS Metadata Repository. The metadata for a SAS application server consists of at least three definitions: a SAS application server, a logical server (or application server component), and a server. The following illustration shows how one SAS application server definition contains metadata for four logical servers and eight servers.

Figure 2.1 How the SAS Metadata Model Organizes SAS Application Server Metadata



SAS Application Server

A SAS application server enables you to specify metadata that applies to all of the logical servers and servers that it 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. For example, when you define a SAS library by using the Data Library Manager, you can assign the library to the SAS application server and all the servers within that SAS application server will have access to the library.

The SAS application server definition and logical servers are not actual server definitions. In SAS Management Console, when you define the first server of a SAS application server, three definitions are created:

- the SAS application server definition
- a logical server definition
- a server definition.

The SAS application server definition contains the logical server definition and its corresponding server definition.

A SAS application server can contain one or more logical server definitions. However, each SAS application server can contain only one of each type of logical server. For example, in one SAS application server definition, you can have up to six logical servers: one workspace logical server, one OLAP logical server, one stored process logical server, one metadata logical server, one batch logical server, and one SAS/CONNECT logical server.

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 *mode of execution* is requested, such as interactive or batch. Logical server definitions contain one or more server definitions.

Note: Metadata logical servers only support one metadata server, and OLAP logical servers only support one OLAP server. △

Servers A server definition contains the actual server metadata 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.

How Workload Is Managed for SAS Workspace and SAS Stored Process Servers

Object Spawners

An Object Spawner is a process-spawning service that instantiates object servers that are using the IOM bridge protocol engine, such as a SAS Workspace Server and a SAS Stored Process Server. In effect, the spawner is a daemon on the server that listens for incoming client requests for IOM services. When the daemon receives a request from a new client, it launches an instance of either a SAS Workspace Server or a SAS Stored Process Server to fulfill the request. After the request is fulfilled, if nothing else is in the spawner's queue, then the spawner returns to its wait state.

During a planned SAS installation, an Object Spawner is defined and configured. For more information, see Chapter 7, "Installing and Configuring Your Software," on page 123.

For more information about using spawners, spawner security, and configuring spawners, see the *SAS Integration Technologies Administrator's Guide* at **`support.sas.com/rnd/itech/library/library9.html`**.

Load Balancing

For IOM bridge connections, load balancing is a program that runs in the Object Spawner and uses an algorithm to distribute work across object server processes on the same or separate machines in a cluster. A cluster is a set of machines that participates in load balancing. Each machine in the cluster runs an Object Spawner that handles and load balances client requests for connections. Load balancing is optional for use with SAS Workspace Servers and is required for SAS Stored Process Servers.

Load balancing is most useful when you need to perform these tasks:

- handle many clients
- distribute work across multiple machines by spreading the workload for different clients.

Load balancing provides the most value when all of the clients actively use the connection to SAS.

For information about load balancing, see the *SAS Integration Technologies Administrator's Guide* at **`support.sas.com/rnd/itech/library/library9.html`**. Also see Chapter 14, "Configuring SAS Servers for Better Performance," on page 259.

Pooling

Pooling is a mechanism that enables you to create a pool of connections to SAS Workspace Servers. These connections are then shared and reused among multiple client applications.

Without pooling, a connection must be created for each client connection request and must remain available for that client, regardless of the client's level of activity. These dedicated connections consume resources that are then unavailable for future client connections.

Because pooling is accomplished in the client process, only those applications running in the same client process can share a pool. A typical Web server can run hundreds of clients in the same process.

For information about pooling, see the *SAS Integration Technologies Administrator's Guide* at **`support.sas.com/rnd/itech/library/library9.html`**. Also see Chapter 14, "Configuring SAS Servers for Better Performance," on page 259.

How SAS Clients Interact with SAS Application Servers

Each SAS Intelligence product interacts with one or more SAS application servers, as explained in these individual sections:

- "SAS ETL Studio" on page 23
- "SAS OLAP Cube Studio" on page 24
- "SAS Enterprise Guide" on page 24
- "SAS Information Delivery Portal" on page 25
- "SAS Enterprise Miner" on page 26.

SAS ETL Studio

SAS ETL Studio is a Java application that enables you to create and manage extract, transform, load (ETL) process flows, which are sequences of steps for the extraction, transformation, and loading of data. SAS ETL Studio enables you to create metadata objects that define sources, targets, and the transformations that connect them. It uses this metadata to generate or retrieve code that reads sources and creates targets on a file system. The following table describes how SAS ETL Studio interacts with each application server.

Table 2.1 How SAS ETL Studio Interacts with SAS Application Servers

Server	Interaction
SAS Metadata Server	Read and write metadata for repository objects such as cubes, documents, jobs, notes, libraries, tables. Read metadata for objects such as SAS Workspace and SAS/CONNECT servers.
SAS Workspace Server	Generate and submit SAS code for jobs. Access resources such as tables.
SAS/CONNECT Server	Submit generated SAS code to machines that are remote from the default SAS application server. Gain interactive access to remote libraries.

For more information about SAS ETL Studio, see the SAS ETL Studio Help, which is available from within the product.

SAS OLAP Cube Studio

SAS OLAP Cube Studio is a Java application that enables you to register cube metadata in a SAS Metadata Repository and save physical cube data in your specified location. The following table describes how SAS OLAP Cube Studio interacts with each application server.

Table 2.2 How SAS OLAP Cube Studio Interacts with SAS Application Servers

Server	Interaction
SAS Metadata Server	Read and write metadata for repository objects such as cubes, OLAP schemas, libraries, and tables. Read metadata for objects such as SAS Workspace Servers.
SAS Workspace Server	Generate and submit SAS code to create and edit cube information. Access resources such as tables.

For more information about SAS OLAP Cube Studio, see the SAS OLAP Cube Studio Help, which is available from within the product, and *SAS OLAP Server: Administrator's Guide*.

SAS Enterprise Guide

SAS Enterprise Guide is a Microsoft Windows application that enables you to create SAS Stored Processes and to store that code in a repository that is available to a SAS Stored Process Server. SAS Enterprise Guide also enables you to access data locally or on SAS servers, perform basic reporting and data analyses, and export or publish results to SAS servers and other Windows or server-based applications. The following table describes how SAS Enterprise Guide interacts with each application server.

Table 2.3 How SAS Enterprise Guide Interacts with SAS Application Servers

Server	Interaction
SAS Metadata Server	Read and write metadata for repository objects such as stored processes. Read metadata for objects such as libraries, tables, IT channels, and SAS Workspace and SAS Stored Process Servers.
SAS Workspace Server	Execute SAS code and queries. Access resources such as tables. Move and copy library members and files between different SAS Workspace Servers. Perform a variety of maintenance tasks such as renaming and deleting files and folders, and copying server and PC SAS data files to libraries.
SAS OLAP Server*	Access cube data and process MDX queries.
SAS Stored Process Server	Execute stored processes and collect resulting output.

* Access to OLAP data on the SAS OLAP Server requires a separate license for the server.

For more information about SAS Enterprise Guide, see the SAS Enterprise Guide Help, which is available from within the product.

SAS Information Delivery Portal

SAS Information Delivery Portal is a Web application that enables you to aggregate data from a variety of sources and present them to the user in a Web browser. The content might include items such as the output of Web applications or SAS Stored Processes, documents, and links. The following table describes how SAS Information Delivery Portal interacts with each application server.

Table 2.4 How SAS Information Delivery Portal Interacts with SAS Application Servers

Server	Interaction
SAS Metadata Server	Read metadata for repository objects such as publication channels, stored processes, libraries, tables, and SAS Workspace and SAS Stored Process Servers. Read and write metadata for repository objects such as portal content (pages, portlets, links, bookmarks, alerts).
SAS Workspace Server	Publish packages and retrieve published packages. Execute stored processes with package results.
SAS Stored Process Server	Execute stored processes with *streaming results*.

For more information about SAS Information Delivery Portal, see the SAS Information Delivery Portal Help, which is available from within the product.

Additional information is in the *SAS Web Infrastructure Kit Administrator's Guide and the SAS Web Infrastructure Kit User's Guide*, which are part of the SAS Integration Technologies documentation available at **`support.sas.com/rnd/itech/library/library9.html`**.

SAS Enterprise Miner

SAS Enterprise Miner is a Java application that enables you to create and manage data mining process flows, which are sequences of steps for the examination, transformation, and processing of data to create models to predict complex behaviors of economic interest. The following table describes how SAS Enterprise Miner interacts with each application server.

Table 2.5 How SAS Enterprise Miner Interacts with SAS Application Servers

Server	Interaction
SAS Metadata Server	Read metadata for repository objects such as libraries, tables, and SAS Workspace Servers. Write metadata for repository objects such as user profile preferences, project definitions and locations, and Result Set Packages.
SAS Workspace Server	Spawn processes to create and persist project information such as data source definitions, process flow diagrams and flow node properties, and Result Set Packages.

For more information about SAS Enterprise Miner, see the SAS Enterprise Miner Help, which is available from within the product.

PART 2

Security Administration

CHAPTER

3

Understanding the Security Concepts in the SAS Intelligence Architecture

Introduction to Security Concepts

This chapter explains the security concepts in the SAS Intelligence Architecture, focusing on how the authentication and authorization processes work. Systems architects and administrators who perform security-related tasks will benefit from understanding the information that is presented in this chapter.

Authentication is the process of verifying the identity of a person or process within the guidelines of a specific policy. Authentication is a prerequisite for authorization. Understanding how authentication works in the SAS Intelligence Architecture will help you perform these tasks:

- make preliminary decisions about your security architecture
- determine which user accounts you must create
- plan your authentication domains
- identify which user credentials you must store in the metadata

Authorization is the process of evaluating rules to determine which users have which permissions for which resources. The outcome of the authorization process is an authorization decision that permits or denies a specific action on a specific resource based on the requesting user's identity and group memberships. Understanding how authorization works in the SAS Intelligence Architecture will help you do these things:

- manage access to resources across multiple authorization layers, as described in "Authorization Layers" on page 41
- define an effective, manageable set of access controls in the metadata authorization layer.

Authentication

By default, servers in the SAS Intelligence Architecture rely on the host environment to authenticate users. SAS Metadata Servers and SAS OLAP Servers can use alternative authentication providers such as Lightweight Directory Access Protocol (LDAP) server and Microsoft Active Directory server.

In the SAS Intelligence Architecture, not all servers have trust relationships with one another, so authentication occurs in two distinct phases:

1. In the *initial authentication* phase, you log in with a SAS Intelligence client and your identity is verified by the authentication provider for a SAS Metadata Server, a SAS OLAP Server, or a Web application. In most cases, the authentication provider is the host operating system. After your credentials (your user ID and password) are verified, the metadata server uses your user ID to determine your metadata identity.
2. In the *additional authentication* phase, you make a request that requires access to a server that must verify your identity (such as a SAS Workspace Server, a SAS Stored Process Server, or a database server). The application that you are using obtains your credentials from the metadata server and provides those credentials to the target server. This enables the target server's authentication provider to verify your identity.

The following sections describe the two phases of the authentication process in detail.

Initial Authentication

Initial authentication is the verification of your identity based on credentials that you submit when you log in with a SAS Intelligence client. Initial authentication requires that you have an account with the authentication provider that verifies the user ID and password that you submit. Storing credentials in the metadata does not eliminate the need for this account. The account can be any of the following:

- a local user account in the operating system of the computer on which the authenticating server is running
- a network user account that provides access to the operating system of the computer on which the authenticating server is running
- an LDAP or Active Directory user account (if the authenticating server is using one of these alternative authentication providers).

The initial authentication process can vary depending on the software component with which you log in. The following table describes how each software component verifies identities.

Table 3.1 Initial Authentication

Software Component	Identity Verification
A desktop application (such as SAS ETL Studio or SAS Management Console).	The SAS Metadata Server's authentication provider verifies the user ID and password that you submit.
A Web application (such as SAS Information Delivery Portal) that is using metadata server authentication.	The SAS Metadata Server's authentication provider verifies the user ID and password that you submit.
A Web application (such as SAS Information Delivery Portal) that is using Web server authentication.	The Web application's authentication provider verifies the user ID and password that you submit. As a *trusted user** of the metadata server, the Web application establishes the connection on your behalf. You do not need to have your own account with the SAS Metadata Server's authentication provider.
A component that connects directly to a SAS OLAP Server (such as SAS Enterprise Guide).	The SAS OLAP Server's authentication provider verifies the user ID and password that you submit. As a *trusted user** of the SAS Metadata server, the SAS OLAP Server establishes the connection on your behalf. You do not need to have your own account with the SAS Metadata Server's authentication provider.

* The trusted user account supports a multi-tier server environment where user identities are authenticated by a server other than the metadata server.

How the Metadata Server Determines Your Metadata Identity

After your credentials are verified, the metadata server determines your metadata identity. A *metadata identity* represents an individual user or a group of users in the metadata environment. In a metadata repository, each metadata identity

- can own one or more logins that contain credentials for various systems
- can be assigned permissions to computing resources
- can be a member of one or more user groups.

Your metadata identity can be used by

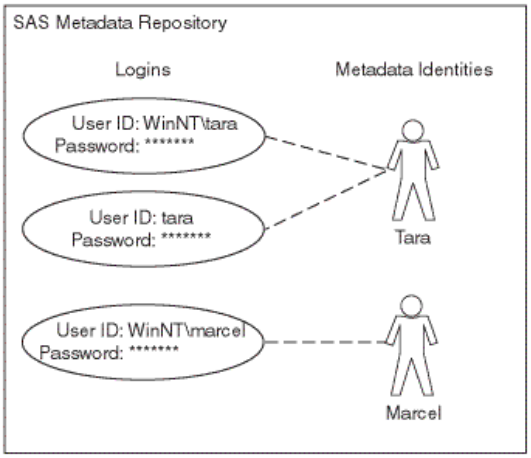
- the metadata server to respond to requests that other applications make for your credentials
- the metadata server's authorization facility to respond to requests for authorization decisions.

In order to determine your metadata identity, the metadata server searches the metadata repository for a user ID that matches the user ID that you submitted during initial authentication. In this process, the metadata server attempts to match your fully qualified user ID. For example, when you store a user ID for a Windows host in the metadata, you must specify the user ID in the form *domain-name\userID* (or *machine-name\userID* for a local account).*

* For additional information about how to specify fully qualified user IDs, see "Defining SAS Users, Groups, and Logins" in the *SAS Integration Technologies Administrator's Guide*, which is available at **`support.sas.com/rnd/itech/doc9/admin_oma/security/auth`**.

In the metadata repository, each user ID is stored as part of a login. Each login is owned by only one metadata identity. Each metadata identity can own multiple logins. These relationships between user IDs, logins, and metadata identities are depicted in the following figure.

Figure 3.1 User IDs, Logins, and Metadata Identities



If the metadata server finds a matching user ID in the repository, the metadata server determines which metadata identity owns the login that includes that user ID. Your access to resources is defined by the logins and access controls that have been defined for your metadata identity (or for a user group to which your metadata identity belongs). If the metadata server does not find a matching user ID, your access is limited to the logins and access controls that have been defined for the PUBLIC implicit group, which includes all users who can access the metadata server.

Note: If you are an *unrestricted user* or an *administrative user*, you can perform certain tasks even if you do not have a metadata identity. For details, see "Special Users of the Metadata Server" on page 49. △

When a login is used to determine your metadata identity, the login is functioning as an inbound login. A login that functions only as an inbound login does not have to include a password because the metadata server does not examine passwords while attempting to determine your identity.

Note: A login can be used for inbound purposes, outbound purposes, or both inbound and outbound purposes. Outbound use of logins is described in "Additional Authentication" on page 35. △

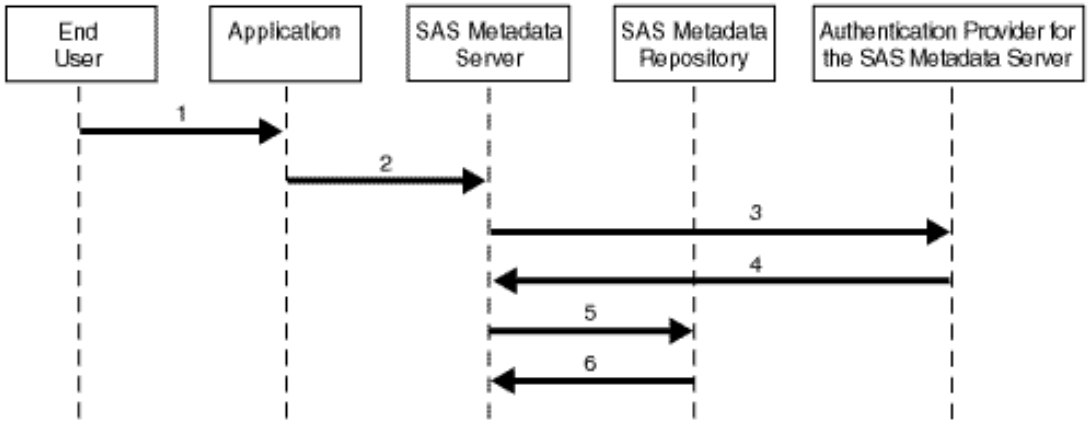
Initial Authentication on a Metadata Server

The metadata server handles initial authentication when you log in with the following applications:

- a desktop application such as SAS Management Console, SAS ETL Studio, or SAS OLAP Cube Studio
- a Web application (such as SAS Information Delivery Portal) that is configured to authenticate users on the metadata server.

The following figure depicts a successful initial authentication for a user who logs in to an application that authenticates users on a metadata server.

Figure 3.2 Initial Authentication on a Metadata Server



In this figure, the numbered arrows correspond to the following activities:

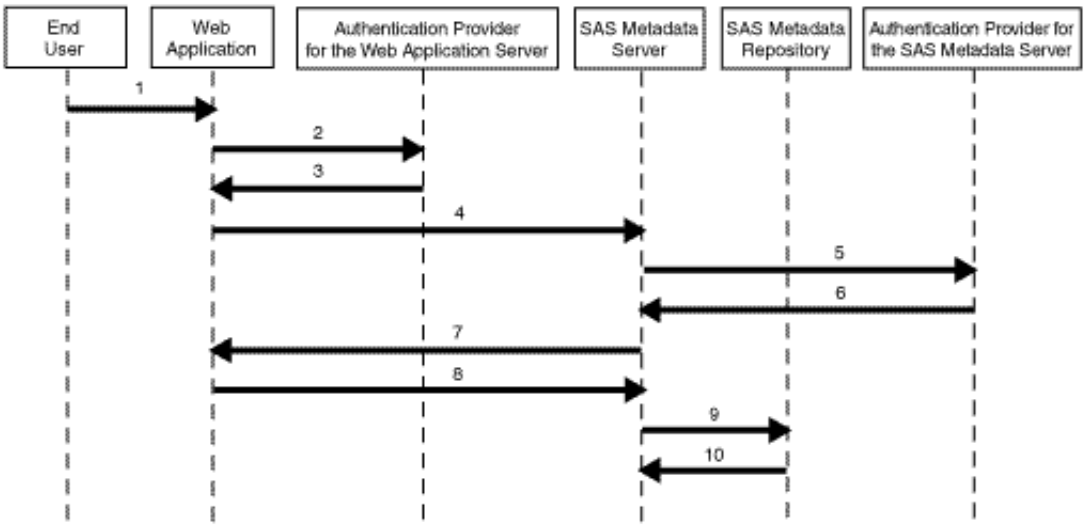
1. The user submits a user ID and password to a SAS application.
2. The application sends the user's credentials to the metadata server.
3. The metadata server passes the credentials to its authentication provider for authentication. By default, the metadata server uses host authentication.
4. The metadata server's authentication provider returns the authenticated user ID to the metadata server.
5. The metadata server looks up the user ID in the logins that are stored in the metadata repository.
6. The metadata server determines which metadata identity owns the login that contains the matching user ID.

Initial Authentication on a Mid-Tier Server

If you log in with a Web application that is configured to authenticate on a Web server, the Web server handles the initial authentication and then uses a trusted user connection to communicate with the metadata server. A similar process occurs when you log in to a SAS OLAP Server.

The following figure depicts a successful initial authentication for a user who logs in to a Web application that is configured to authenticate users on a mid-tier server.

Figure 3.3 Initial Authentication on a Mid-Tier Server



In this figure, the numbered arrows correspond to the following activities:

1 The user submits a user ID and password to a Web application.

2 The Web application passes the user's credentials to its authentication provider for verification.

3 The Web application's authentication provider passes the authenticated user ID back to the application after verification.

4 The application requests a connection to the metadata server using a trusted user ID.

5 The metadata server passes the credentials of the trusted user to its authentication provider for verification.

6 The metadata server's authentication provider verifies the credentials of the trusted user.

7 The metadata server tells the Web application that the trusted user connection has been accepted.

8 The application sends the user's ID to the metadata server over the trusted user connection. The metadata server trusts that the application has verified the user's ID.

9 The metadata server looks up the user ID in the logins that are stored in the metadata repository.

10 The metadata server determines which metadata identity owns the login that contains the matching user ID.

Note: Application-specific implementation details have been omitted from the last three steps in this process description. △

Trusted Peer Servers

In addition to the trusted *user* connections that are described in the previous section, the SAS Intelligence Architecture also supports trusted peer *server* connections. You can configure your deployment so that the SAS Workspace Servers and SAS Stored Process Servers can connect to the metadata server without explicitly providing credentials. The connection is made with a proprietary protocol that causes the metadata server to trust the authentication that the connecting server has already

performed. Applications can use this trusted peer server mechanism to generate code or run batch jobs without providing credentials to the metadata server.

Note: Use of this proprietary protocol implies that the SAS Metadata Server trusts the authentication mechanism of the connecting server. You must implement the appropriate security for your network to prevent untrusted machines and untrusted authentication that could compromise the SAS Metadata Server. For additional information about trusted server connections, see "Implementing Trusted Authentication Mechanisms" in the *SAS Integration Technologies Administrator's Guide* at **`support.sas.com/rnd/itech/doc9/admin_oma/security/auth`**. △

Additional Authentication

Additional authentication is the use of your credentials by other systems after initial authentication. For example, when you use an application such as SAS Information Delivery Portal to run a stored process, the application must provide your credentials to a SAS Stored Process Server (or a SAS Workspace Server) so that server can verify your identity.

Additional authentication requires that you have an account with the authentication provider for each system that will verify your identity. Storing credentials in the metadata does not eliminate the need for these accounts. These accounts can be any of the following:

- local user accounts in the operating system of the computer on which the authenticating server is running
- network user accounts that provide access to the operating system of the computer on which the authenticating server is running
- LDAP or Active Directory user accounts (if the authenticating server is using one of these alternative authentication providers).

The SAS Intelligence Architecture uses a single sign-on model for authentication. This model streamlines the user experience by enabling users to access computing resources across multiple logical servers, physical servers, operating systems, and network domains without being repeatedly prompted for their user IDs and passwords.

The SAS Intelligence Architecture supports single sign-on by enabling you to store multiple sets of credentials in the metadata for each user and user group. Each set of credentials is stored in a login that is associated with a distinct set of computing resources. When an application needs to provide your credentials to another server, the application searches the metadata repository for a login that contains credentials that can be used to access the target server. The login must be owned by your metadata identity or by a user group to which your metadata identity belongs.

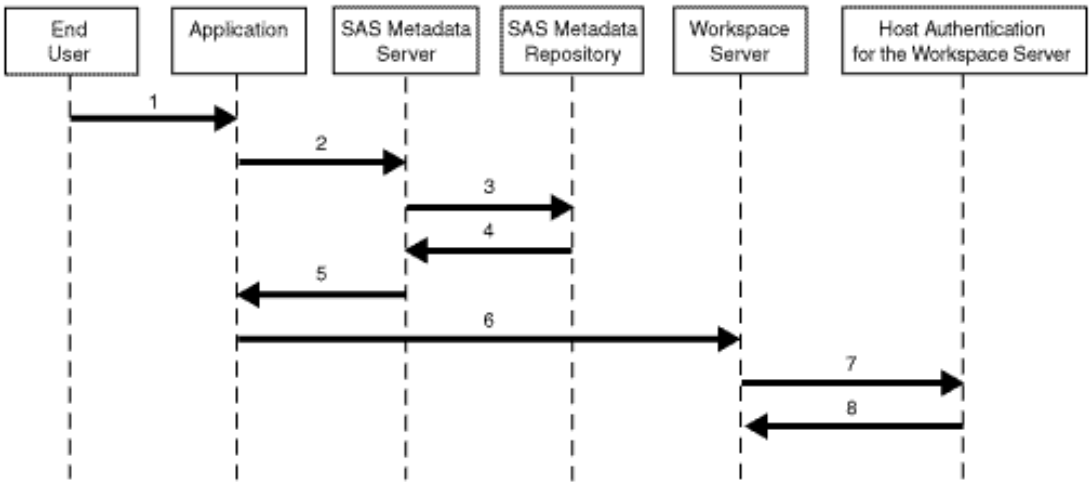
If the application finds an appropriate login, the application submits that user ID and password for authentication by the server that you need to access. Some applications will prompt you for your credentials in these circumstances:

- a login that provides access to the target server has not been defined in the metadata for you or for a group to which you belong.
- the login information is incorrect or incomplete
- more than one login meets the criteria that is specified in the application's request and the application is unable to determine which login to use.

When a login is retrieved from the metadata server and sent to another system, the login is functioning as an *outbound login*. An outbound login must specify an authentication domain and include credentials that are appropriate for the server and host to which the login provides access.

The following figure depicts a successful additional authentication for a user who has already logged in with SAS Information Delivery Portal and completed initial authentication. In this example, the user is represented in the repository by a metadata identity that owns a login that contains credentials for accessing the SAS Workspace Server.

Figure 3.4 Additional Authentication



In this figure, the numbered arrows correspond to the following activities:

1 The user makes a request in SAS Information Delivery Portal that requires access to the SAS Workspace Server.

2 The application (SAS Information Delivery Portal) recognizes that the request requires access to the workspace server, so the application goes to the metadata server to get credentials that will give the user access to the workspace server. The application communicates with the metadata server using the connection that was established during initial authentication.

3 The metadata server looks for the requested credentials in the metadata repository. The credentials must meet both of these criteria:

- The credentials are stored in a login that is owned by the requesting user's metadata identity (or by a group to which that identity belongs).
- The credentials are stored in a login that is associated with the authentication domain in which the workspace server is registered. For more information about this requirement, see "Authentication Domains" on page 37.

4 The metadata server locates the appropriate credentials in the metadata repository and retrieves those credentials from the metadata repository.

5 The metadata server sends the credentials to the requesting application.

6 The application sends the credentials to the workspace server.

Note: Typically an object spawner is used to start the workspace server, so the application actually sends the credentials to the object spawner rather than directly to the workspace server. This implementation detail is not depicted in the figure. △

7 The workspace server passes the credentials to its host environment for authentication.

8 The host authentication mechanism tells the workspace server that the credentials are valid, and accepts the connection.

Cached Credentials

SAS Web applications (such as SAS Information Delivery Portal) will cache the credentials that you provide when you log in. The cached credentials can be used during additional authentication. This enables you to access servers that use the same authentication process with which you were initially authenticated— even if those credentials are not stored in the metadata.

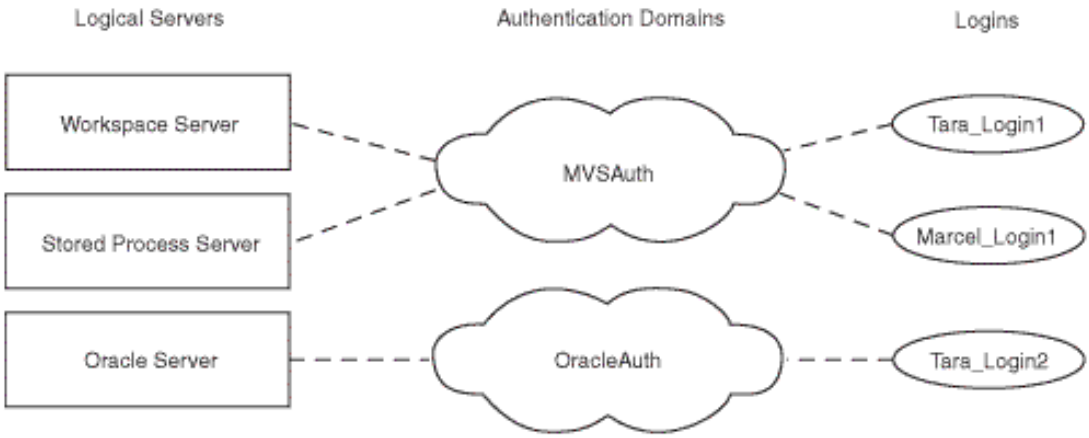
The cached credentials are used to provide access to servers within one specified authentication domain. For that authentication domain, your cached credentials are used even if those credentials are also stored in the metadata. When an administrator installs and configures the application, the administrator specifies which authentication domain will use the cached credentials. By default, the specified authentication domain is named DefaultAuth. For additional details, see the installation instructions for your application.

Authentication Domains

Authentication domains enable you to define logical groupings of computing resources and logins within a metadata repository. All of the computing resources within an authentication domain use the same authentication process. You can choose to use the same groupings and names for your authentication domains as you do for your host domains or network domains, but you are not required to do so.*

You must associate an authentication domain with each logical server that you register in the metadata. You must associate an authentication domain with each login that you define in the metadata to support additional authentication. The following figure depicts the relationships between servers, authentication domains, and logins.

Figure 3.5 Servers, Authentication Domains, and Logins



* For information about the role of host and network domains in the SAS Intelligence Architecture, see "Understanding the Server Authentication Process" in the *SAS Integration Technologies Administrator's Guide* at **support.sas.com/rnd/itech/doc9/admin_oma/security/auth**.

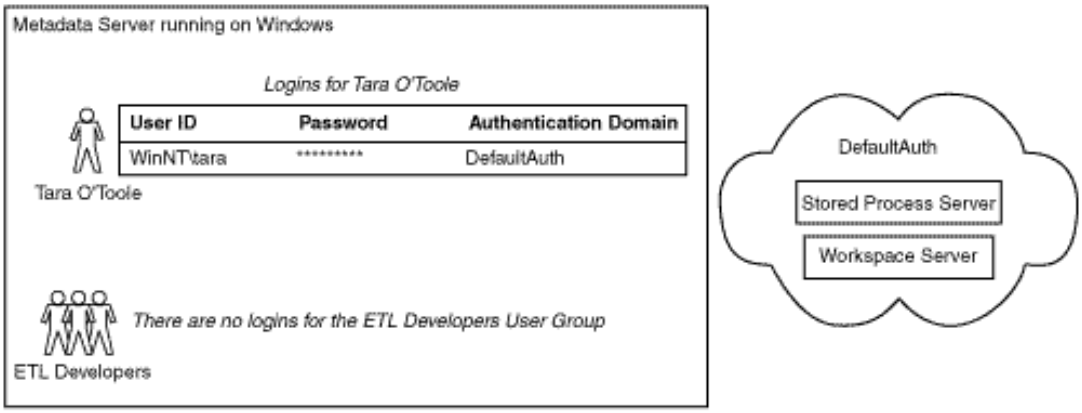
During additional authentication, an application searches the metadata for a login that is associated with the authentication domain in which the target server is defined. If Tara makes a request that requires access to the Oracle server, the application that Tara is using will first determine that the Oracle server is associated with the OracleAuth authentication domain. The application will then ask the metadata server for a login that is associated with the OracleAuth authentication domain and owned by Tara's metadata identity (or by a group to which Tara's identity belongs). In this figure, the login Tara_Login2 meets these criteria. If this login includes Tara's password for the Oracle server, then Tara will be able to access that server. If Marcel makes a similar request, he will be denied access to the Oracle server because Marcel does not have a login for the OracleAuth authentication domain.

The following topics illustrate the relationships between servers, authentication domains, and logins in a variety of deployment models. In each model, the logins that are stored in the metadata for an individual user (Tara O'Toole) and a particular user group (ETL Developers) are identified.

Authentication Domains in a Single Platform Environment

In a homogeneous environment, you might need only one authentication domain. The following figure depicts a deployment in which all of the logical servers and all of the logins for all metadata identities are associated with an authentication domain that is named DefaultAuth.

Figure 3.6 Homogenous Environment, One Authentication Domain



In this figure, the metadata identity that represents Tara owns only one login, which functions as both an inbound and an outbound login. Because the servers are running under Windows, the user ID in the login is fully qualified with the name of the Windows domain (WinNT). Because Tara's password is stored in the login, Tara will be able to access the workspace server and stored process server without being prompted for her credentials.

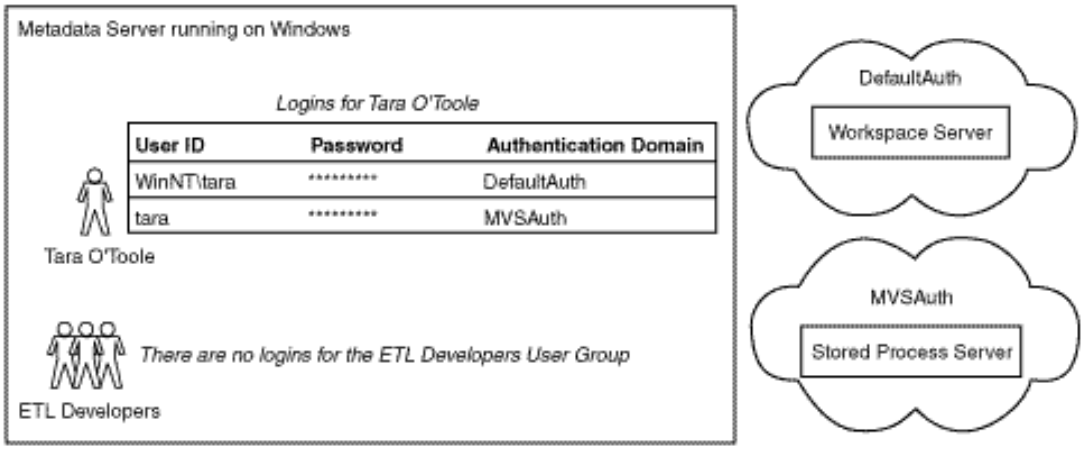
Note: If Tara uses a Web application, she can access the workspace and stored process servers using credentials that are cached from initial authentication. In this scenario, Tara's login would not have to include a password or specify an authentication domain. △

In this deployment, no logins have been defined for the ETL Developers user group. This user group exists to simplify administration of access controls.

Authentication Domains in a Mixed Platform Environment

In a multi-host environment, you will usually need more than one authentication domain. For example, if you modify the previous deployment by moving the stored process server to z/OS, then you will need an additional authentication domain, because your users access servers on z/OS using different credentials than they use on Windows. In the metadata, you need to link the stored process server to the logins that contain credentials for accessing that server. You create this link by associating both the server and the logins with a new authentication domain. The following figure depicts this modification to the previous deployment.

Figure 3.7 Mixed Environment, Two Authentication Domains



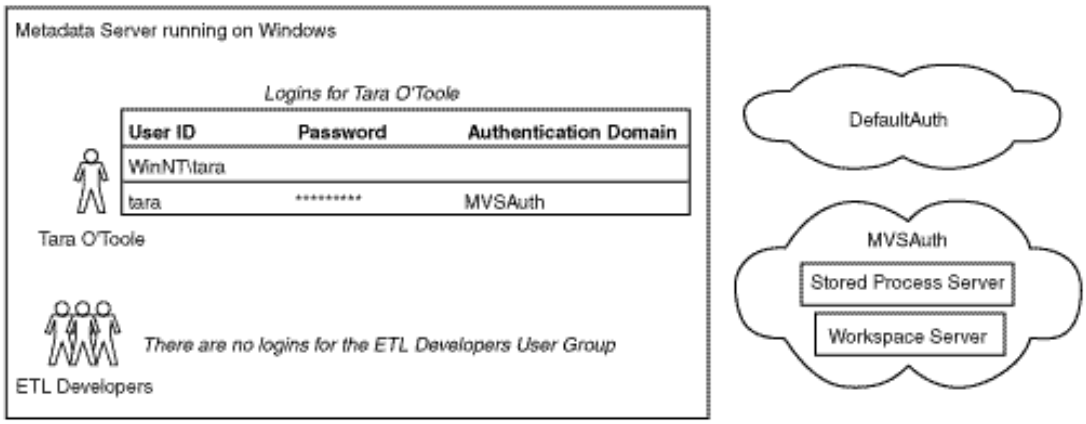
In this figure, a new authentication domain named MVSAuth has been defined, and the stored process server has been registered in that authentication domain. Two logins have been defined for Tara:

- The first login is for the DefaultAuth authentication domain. This login is used by the metadata server to determine Tara's identity and by the workspace server during additional authentication.
- The second login is for the MVSAuth authentication domain. This login enables Tara to access the stored process server during additional authentication.

Note: If Tara uses a Web application, she can access the workspace server using credentials that are cached from initial authentication. In this scenario, Tara's first login would not have to include a password or specify an authentication domain. △

The next figure depicts the deployment after you move the workspace server to z/OS. Now only the metadata server is running under Windows. All of the other servers are running under z/OS and are registered in the MVSAuth authentication domain.

Figure 3.8 Mixed Environment, One Authentication Domain

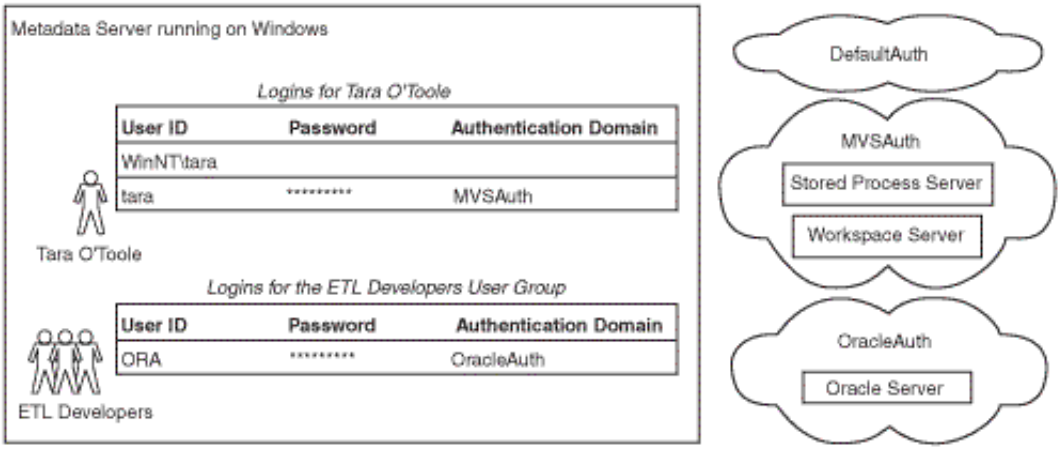


In this figure, the DefaultAuth authentication domain still exists, but it is not associated with any servers or logins. Tara still owns two logins, but it is no longer necessary to include a password or an authentication domain in the first login. Tara's first login is now only used to determine her metadata identity; it is not used to provide access to any other servers.

Authentication Domains in a Diverse Environment

In a diverse environment, you might need more authentication domains. For example, if you add an Oracle server to the previous deployment, then you will need to add another authentication domain, because your users access the Oracle server with different credentials than they use to access the other servers. In the metadata, you need to link the Oracle server to the logins that contain credentials for accessing that server. You create this link by associating both the Oracle server and the logins with a new authentication domain. The following figure depicts this deployment.

Figure 3.9 Diverse Environment, Multiple Authentication Domains



In the figure, a new authentication domain named OracleAuth has been defined, and an Oracle server has been registered in that authentication domain. The DefaultAuth

authentication domain still exists, but it is not associated with any servers or logins. The metadata identity that represents Tara O'Toole owns two logins:

- The first login is used by the metadata server to determine Tara's identity. It is not necessary to include a password in this login because this login is used only as an inbound login (it is inbound to the metadata server).
- The second login provides access to the stored process and workspace servers that are registered in the MVSAuth authentication domain. This login functions as an outbound login (it is outbound from the metadata server), so this login includes a password to support a single sign-on approach to additional authentication.

Note: A different set of logins might be required if your metadata server uses an alternative authentication provider or your deployment includes pooled servers. △

Tara does not directly own a login that provides access to the server in the OracleAuth authentication domain, so she can access that server only if she is a member of a user group that owns an appropriate login. In this example, Tara is a member of the ETL Developers user group, so she can use that group's shared login to get to the Oracle server in the OracleAuth authentication domain. If you give the ETL Developers group a login for the OracleAuth authentication domain, you should not also give Tara a login for the OracleAuth authentication domain. If more than one login for a particular authentication domain is available to Tara, then a requesting application might not be able to determine which set of credentials to use.

Note: In order to access the Oracle server from SAS ETL Studio, Tara must be able to access both the workspace server *and* the Oracle server. △

Authorization Layers

In a SAS Intelligence Architecture deployment, there are multiple authorization layers that can affect your access to resources. An *authorization layer* is a set of access controls that exists within a particular security framework, such as an operating system or a database management system. Your *effective permissions* to resources are limited to access that is permitted by all applicable authorization layers. For example, regardless of the access controls that have been defined for you in the metadata, you cannot access a particular file if the operating system permissions do not permit the action.

The SAS Intelligence Architecture provides a *metadata authorization layer* that enables you to set access controls on metadata objects that represent computing resources. This authorization layer is described in detail in the next topic.

Other authorization layers that can affect your access to resources in a SAS Intelligence deployment include the following:

- The operating system authorization layer consists of the file, directory, and system permissions that you specify on a particular machine.
- The data source authorization layer consists of permissions to relational database objects, passwords for SAS data sets, and other data source-specific access controls.
- The WebDAV authorization layer consists of the third-party Web server access controls on report content objects such as files and directories.
- The physical authorization layer consists of tangible protective measures such as locking a server room or cabinet.

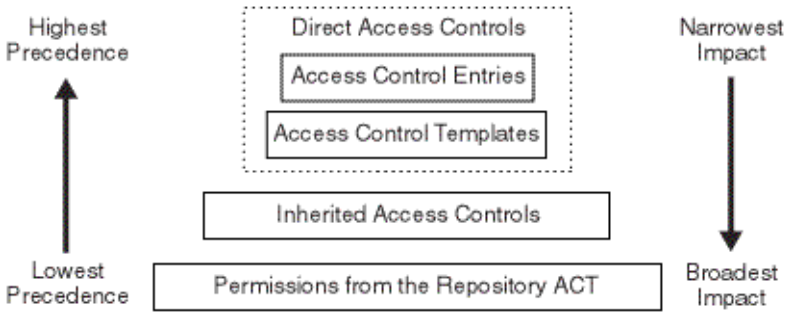
The Metadata Authorization Layer

In the metadata authorization layer of the SAS Intelligence Architecture, you specify access controls on metadata objects that represent your computing resources. The metadata authorization layer has the following characteristics:

- The management of access controls is centralized. All metadata access controls are defined, stored, and evaluated by the authorization facility of a SAS Metadata Server. SAS Management Console enables you to create access controls in the metadata authorization layer that define which users can perform which actions on which resources.
- The evaluation of access controls incorporates multiple inheritance, access control templates, and identity precedence rules. When it receives a request for access to a particular resource, the authorization facility evaluates all of the pertinent access controls in order to determine whether the request should be granted or denied.
- Requests to create, view, update, or delete metadata are initiated and enforced by the metadata server. For a list of metadata layer permissions and a description of the actions that are controlled by each permission, see "Metadata Layer Permissions" on page 64.
- Requests to create, view, update, or delete data are initiated and enforced by other applications. In the current release, not all applications enforce all of these permissions. See "Considerations for Defining Effective, Efficient Access Controls" on page 68 for best practice information for using these permissions.

In order to plan an efficient, maintainable framework of access controls in the metadata authorization layer, you must understand how permissions are assigned and evaluated in the metadata environment. The following figure provides an overview of the different ways that permissions can be assigned in the metadata authorization layer.

Figure 3.10 Access Controls in the Metadata Authorization Layer



From top to bottom in the figure, the types of permission assignment in the figure are ordered as follows:

- from highest precedence (hardest to override) to lowest precedence (easiest to override). For example, a directly assigned ACE can override a directly assigned ACT.
- from narrowest impact (most specific) to broadest impact (least specific). For example, a permission that is specified on the repository access control template (ACT) can affect all of the resources in a repository, while a directly assigned access control entry (ACE) can be directly associated with only one resource.

These permission assignment types are described in the following sections.

Direct Access Controls

Direct access controls are permissions that you set on the Authorization tab of the target resource.* A directly assigned permission can be an access control entry (ACE) or an access control template (ACT).

ACE — An access control entry (ACE) is an access control object that represents a set of identities that have the same set of permission values (such as a grant of ReadMetadata permission and a denial of WriteMetadata permission) for a particular resource. Each ACE is directly associated with a single resource. Each resource can be associated with multiple ACEs.

ACT — An ACT is an access control object that represents a set of identities with specific permission settings for each participating identity. For example, an ACT can consist of a denial of ReadMetadata permission for the PUBLIC group and a grant of ReadMetadata for a particular user. Each ACT can be directly associated with multiple resources. Each resource can be associated with multiple ACTs. An ACT can be updated independently of the resources that reference it.

A permission is directly assigned to you if an ACE or an ACT on the target resource grants or denies the permission to your metadata identity *or* to a user group to which your metadata identity belongs. Directly assigned permissions override permissions that are not directly assigned.

Identity precedence is used to resolve conflicts between directly assigned permissions. For example, you could have a scenario where

- a directly assigned ACT denies PersonA ReadMetadata access to DataSet1
- PersonA belongs to GroupB
- a directly assigned ACE grants GroupB ReadMetadata access to DataSet1.

In this scenario, PersonA does not have ReadMetadata access to DataSet1. PersonA's direct ACT denial overrides GroupB's direct ACE grant.**

Inherited Access Controls

An inherited access control is an ACT or an ACE that is specified on an object that is a parent to the target resource. In a SAS metadata environment, inheritance rules determine which objects can be parents for a particular resource. For example, the inheritance rules specify that the metadata object that describes a SAS library can be a parent to the metadata object that describes a SAS data set. For this reason, the metadata layer access controls that you assign to a particular SAS library are inherited by all of the SAS data sets that belong to that library. The inheritance rules for SAS data, relational database data, OLAP data, and custom folders are described in the following sections.

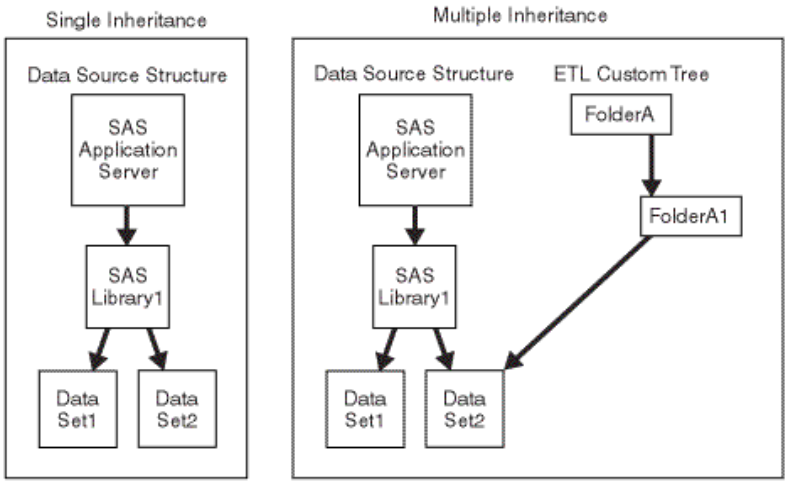
Note: A target resource can inherit access controls from only those parent objects that are registered in the same metadata repository as the target resource. Even when there is a dependency relationship between two repositories, objects in one repository cannot inherit access controls from objects in another repository. △

* For information about using the Authorization tab, see "Controlling Access to Resources" on page 86.

** The identity precedence rules are described in detail in "Understanding the Authorization Facility" in the *SAS Metadata Server: Setup Guide*, which is available from **`support.sas.com/rnd/eai/openmeta/`**.

The SAS metadata environment supports multiple inheritance, such that one object can inherit access controls from several parent objects. The following figure depicts examples of single inheritance and multiple inheritance in the SAS metadata environment.

Figure 3.11 Single and Multiple Inheritance



In the single inheritance figure, Data Set1 and Data Set2 share the same single chain of parent objects. The two data sets inherit access controls from SAS Library1, which in turn inherits access controls from the SAS Application Server. In the single inheritance case, if GroupA has ReadMetadata permission to SAS Library1, then Data Set1 and Data Set2 inherit that grant of ReadMetadata permission for GroupA.

Note: You can override an inherited access control. For example, you can set a permission on Data Set1 that denies GroupA the ReadMetadata permission. The denial that you specify on Data Set1 overrides the grant that Data Set1 inherited from SAS Library1. △

In the multiple inheritance figure, Data Set2 is assigned to both a parent library and a folder in the ETL custom tree. Data Set2 inherits access controls from two sets of parent objects:

- SAS Library1 and its parent SAS application server
- FolderA1 and its parents in the ETL custom tree.*

In the multiple inheritance case, if GroupA is granted ReadMetadata permission to SAS Library1 but is denied ReadMetadata permission to FolderA1 in the ETL custom tree, then GroupA can still access Data Set2. A grant on *any* parent object is sufficient to enable GroupA to read the metadata that describes Data Set2.

If GroupA is denied ReadMetadata permission on SAS Library1 and is neither granted nor denied ReadMetadata permission to FolderA1 in the ETL custom tree, then the inherited access controls on Data Set2 are not definitive, so the repository ACT is consulted.

If GroupA is denied ReadMetadata permission by all parents, then those denials are determinative.

* The ETL custom tree is available in deployments that include SAS ETL Studio. For information about how to use the ETL custom tree, see "Using Custom-Tree Folders for Security" on page 215.

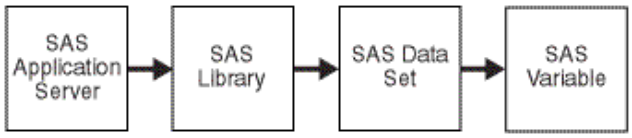
Inheritance in SAS Data

In a SAS metadata environment, inheritance of access controls for SAS data flows as follows:

- from a SAS Application Server to the SAS libraries that are defined on the logical workspace server component of the SAS Application Server
- from a SAS library to the data sets within that library
- from a data set to the variables within that data set.

The following figure depicts the flow of access controls for SAS data in the metadata environment.

Figure 3.12 Inheritance of Access Controls in SAS Data



Not all permissions are supported at all levels for SAS data.

- The metadata server enforces the ReadMetadata, WriteMetadata, and CheckInMetadata permissions at all levels—server, library, data set, and variable.
- The Metadata LIBNAME Engine enforces the Read, Write, Create, and Delete permissions at the library and data set levels only. Setting these permissions on a variable has no effect.

Note: SAS data objects can also inherit permissions from custom trees, which are described in "Inheritance in Custom Folders" on page 47. △

Inheritance in Relational Database Data

In a SAS metadata environment, inheritance of access controls for relational database data flows as follows:

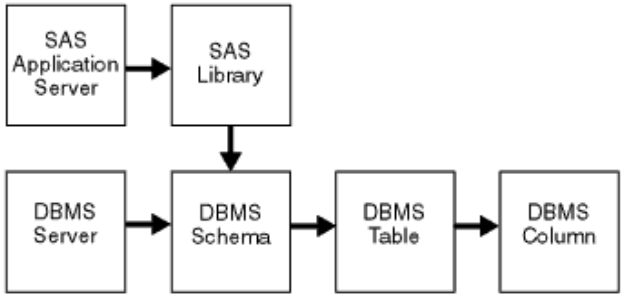
- from a database management system (DBMS) server definition to the DBMS schemas that are defined on that DBMS server.

  Note: A DBMS schema that is associated with a SAS library will also inherit access controls from that library and, in turn, from the SAS Application Server that includes the workspace server component on which the library is defined. △

- from a DBMS schema to the tables within that schema
- from a DBMS table to the columns within that table.

The following figure depicts the flow of access controls for database data in the metadata environment.

Figure 3.13 Inheritance of Access Controls in Relational Database Data



Not all permissions are supported at all levels for relational database data.

- The metadata server enforces the ReadMetadata, WriteMetadata, and CheckInMetadata permissions at all levels — server, schema, table, and column.
- The Metadata LIBNAME Engine enforces the Read, Write, Create, and Delete permissions at the library and table levels only. Setting these permissions on a column has no effect.

Note: Relational database objects can also inherit permissions from custom trees, which are described in "Inheritance in Custom Folders" on page 47. △

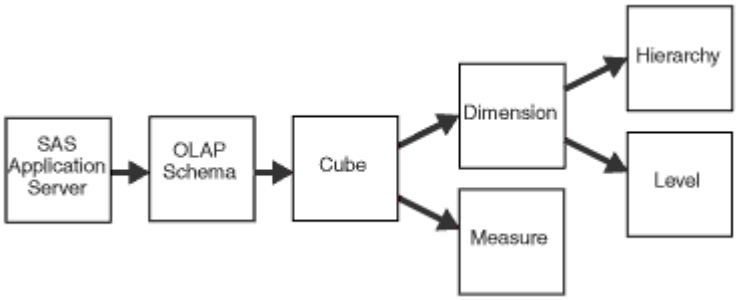
Inheritance in OLAP Data

In a SAS metadata environment, inheritance of access controls for OLAP data flows as follows:

- from a SAS Application Server to the OLAP schemas that are defined on the OLAP server component within the SAS Application Server
- from a schema to the cubes within that schema
- from a cube to the dimensions and measures within the cube

  Note: You cannot set access controls on a calculated measure. △
- from a dimension to the hierarchies and levels within the dimension.

Figure 3.14 Inheritance of Access Controls in OLAP Data



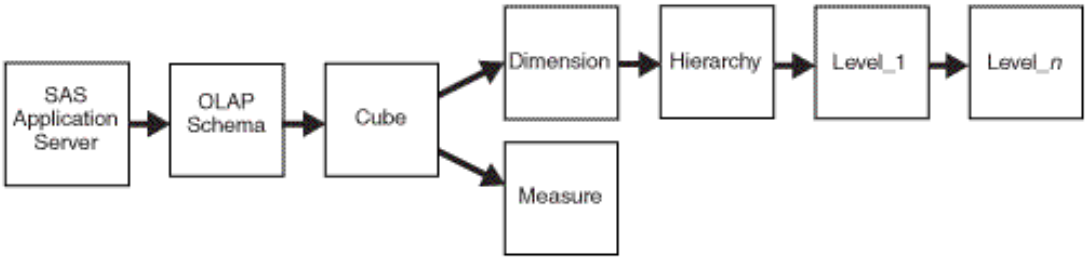
Not all permissions are relevant for all OLAP objects. In order to access a cube, you must have both ReadMetadata and Read permission for the cube. However, in order to access a dimension, measure, hierarchy, or level, only Read permission is required because the SAS OLAP Server requests and enforces decisions for these objects using the Read permission.

Your ability to access OLAP data is also affected by the requirements for drilling through a cube in order to access the data. If you do not have Read access to a particular object (such as an OLAP cube), then you cannot access other objects (such as dimensions and measures) within that object. For example, if a direct access control on an OLAP cube denies the Read permission to a particular user, then that user cannot access any data within the cube. Even if you give the user Read permission to a dimension within the cube, the user will be unable to access that dimension. The problem is not that the user does not have Read access to the dimension. Rather, the problem is that the user does not have the clear path of grants of Read access that is necessary to navigate through the cube to the dimension.

The following list and figure document these navigational access requirements for OLAP data:

- If you do not have Read permission to a cube, you cannot navigate to the dimensions and measures within the cube.
- If you do not have Read permission to a dimension, you cannot navigate to the hierarchies within the dimension.
- If you do not have Read permission to a hierarchy, you cannot navigate to the top levels within the hierarchy.
- If you do not have Read permission to a particular level in a hierarchy, you cannot navigate to the next level in that structure.

Figure 3.15 Access Requirements for Navigating through OLAP Data

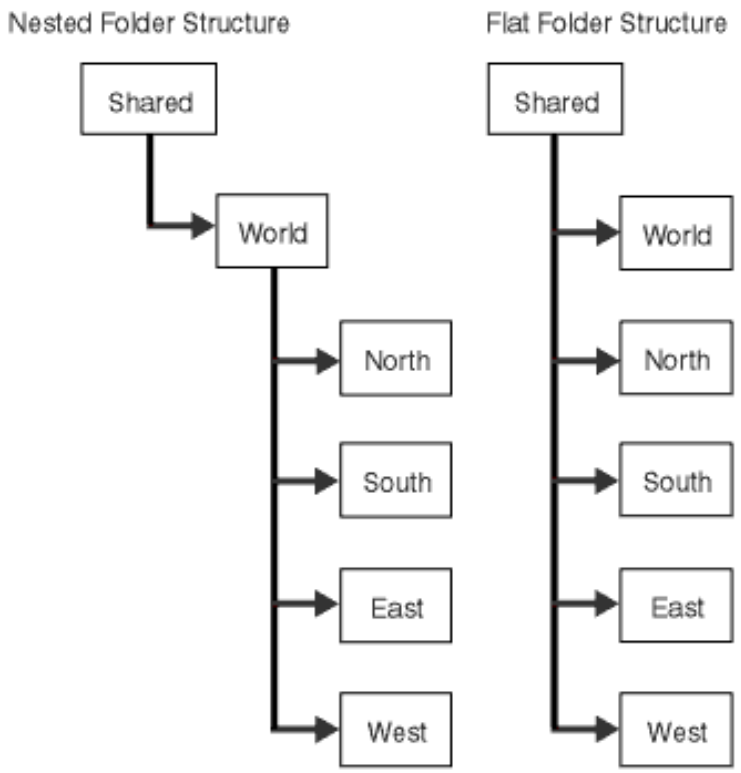


Inheritance in Custom Folders

SAS ETL Studio enables you to add folders and items to an ETL custom tree. Within a custom tree, each folder inherits access controls from its parent folder. Most items in the ETL custom tree inherit access controls from the folder in which the items are located. To learn how to create and manage folders in the ETL custom tree, see "Using Custom-Tree Folders for Security" on page 215.

Because inheritance of access controls flows in the ETL custom tree, selecting the optimal folder structure for your needs can help you minimize the number of access controls that you have to set and maintain. Within each tree, you can use a flat list structure, a nested tree structure, or a blend of the two structures. For example, the following figures illustrate two of the ways you could structure folders in your ETL custom tree if you are organizing metadata that describes sales data for a worldwide sales team and for four regional sales teams.

Figure 3.16 Custom Folder Structures



The arrows show how the inheritance of access controls flows in each custom tree.

- In the nested structure, the regions each inherit access controls from the World folder, which in turn inherits access controls from the Shared folder.
- In the flat structure, the World folder and all of the regional folders inherit from the Shared folder.

Permissions from the Repository ACT

A repository ACT permission comes from the ACT that has been designated as the default ACT for the current repository. In a SAS metadata environment, a repository ACT can be designated for each metadata repository.

The settings on the Users and Permissions tab of the repository ACT determine which users and groups can perform the following actions:

- accessing resources for which no other access controls have been established. For example, these settings control which users can perform which actions on a newly created metadata object for which no access controls have been specified.
- creating new resources. For example, these settings determine who can create new metadata objects in the repository.

The repository ACT is represented in SAS Management Console by a blue cylinder icon and is named "Default ACT" by default. Permissions for the repository are listed on the Users and Permissions tab of the repository ACT's properties dialog box. You can locate the repository ACT under

Environment Management ► Authorization Manager ► Access Control Templates

How Permissions Are Evaluated

In the metadata authorization layer, a user or group can have multiple permissions for a resource. When a user requests access to a resource, all of the relevant access controls are evaluated to determine whether the required permissions have been granted to the user. Each authorization decision is made by examining the access controls that pertain to the requesting user, the requested action, and the target resource.

For example, a user can have a directly assigned ACE grant of WriteMetadata permission for a particular resource and a directly assigned ACT denial of the same permission for the same resource. In this case, the authorization decision would be to grant the user access to the resource. The user's ACE grant overrides the conflicting ACT denial.

The following list summarizes the evaluation process:

1. ACEs and ACTs that are directly assigned to the target resource are examined.
   - Conflicting permissions that arise from group membership are resolved by identity precedence rules. For example, an ACT that is assigned to you overrides a conflicting ACE that is assigned to a group to which you belong.
   - If there is a conflict between an ACE and an ACT at the same level in the group precedence hierarchy, then the ACE takes precedence. For example, an ACE that is assigned to you overrides a conflicting ACT that is also assigned to you.
   - If no relevant ACEs or ACTs are found, then the evaluation process continues.
2. The inheritance rules are applied to identify all of the target resource's parent objects. The access controls that have been specified for the parent objects are examined by applying the entire evaluation process to each of the parent objects.
   - If *any* of the parent objects grants the requested permission to you (or to a group to which you belong), then that inherited grant is final.
   - If *all* of the parent objects deny the requested permission to you (or to a group to which you belong), then that inherited denial is final.
   - Otherwise, the evaluation process continues.
3. The Users and Permission tab of the repository ACT is examined to determine whether it grants or denies the requested permission to you (or to a group to which you belong). If there are conflicting permissions within the repository ACT, those conflicts are resolved by the identity precedence rules.
   - If the requested permission is granted or denied to you (or to a group to which you belong), then that grant or denial is determinative.
   - If the requested permission is neither granted nor denied to you (or to any group to which you belong), then the permission is denied.
   - If there is no repository ACT, then the permission is granted. (If there is no repository ACT, and no definitive access controls are found in steps one and two of the evaluation process, then any user who can access the metadata server can read and write metadata in the repository.)

Note: There are additional constraints on access to security-related objects such as logins and permissions. △

Special Users of the Metadata Server

Special users of the metadata server are highly privileged accounts and should be used only for tasks that require access that cannot be assigned by access controls in the

metadata. For instructions on how to create special users, see "Security-Related Files" on page 149.

The metadata server supports the following types of special users:

unrestricted users — can read or write any metadata object (except for passwords, which an unrestricted user can overwrite but cannot read) regardless of any access controls that are specified in the metadata. An *unrestricted user* can also perform administrative tasks such as starting, stopping, pausing, and refreshing the metadata server.

administrative users — can perform the following tasks regardless of any access controls that are specified in the metadata:

- create a user definition and logins to establish a metadata identity for another user
- modify the user definition of another user
- delete the user definition of another user.

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.

trusted users — A *trusted user* is a user ID that acquires credentials on behalf of other users in a multi-tier server environment. The trusted user functionality is used by mid-tier applications and servers to impersonate authenticated clients.

CHAPTER

4

Developing Your Security Plan

Overview of Security Planning

This chapter describes how to plan the users, user groups, and access controls for your deployment of the SAS Intelligence Architecture. This chapter assumes that you have an understanding of the security concepts that are explained in the previous chapter.

Most of the tasks that are described in this chapter are pre-deployment activities, such as making decisions about your security architecture, creating accounts in the operating system (or with other authentication providers), and organizing your users into groups. Some of the tasks that are mentioned in this chapter, such as creating user definitions and setting access controls in the metadata, are post-installation activities.

These are the phases in the security planning process:

1 Define your security goals.

2 Make some preliminary decisions about your security architecture.

3. Determine which user accounts you must create with your authentication providers and which user identities and logins you must establish in the metadata.
4. Determine how you will organize your users into groups.
5. Determine which users need which permissions to which resources, and develop a strategy for establishing those access controls.

Defining Your Security Goals

It is important to customize your security design for your site, so you should begin your security planning by analyzing your environment to determine your security needs. Consider the following guidelines:

- Different types of data require different levels of protection. It is important that your security policies reflect an understanding of your data and of the needs of the users who interact with that data.
- You should usually be more conservative in granting write access to metadata objects and computing resources than you are in granting read access.
- Even for data that is not highly sensitive, it might be desirable to constrain read access so that your users do not see information that is not relevant to their needs.
- You should evaluate how likely it is that someone will accidentally or intentionally compromise the security of your deployment, and how severe the consequences of a compromise would be.
- You should consider the nature of your user community and their expectations regarding security and privacy.
- Your security policy should meet any applicable organization or legal requirements for security and auditability.

In this process, remember that there is no absolute security. You should choose a security design that strikes the right balance between deployability, usability, maintainability, and security for your environment. After you have identified your security goals, you can develop a strategy for achieving those goals.

Making Preliminary Decisions about Your Security Architecture

As you would for any other application suite, you will consider a wide range of aspects of security when you plan your deployment of the SAS Intelligence Architecture. In the preliminary decisions phase, you address those security design and technology choices that have a direct impact on your strategy for defining users, user groups, and access controls. The site-specific security goals that you identified in the previous section should help guide your decisions.

In order to prepare for your deployment, you must make these decisions:

- Select an encryption method and decide whether you will encrypt only credentials or all transmitted data. See “Setting an Encryption Method” on page 157 for more information.
- Select the technology that the metadata server will use to verify the identity of users. This choice determines whether the accounts that you define for users who are authenticated by the metadata server must be in the operating system or with an alternative authentication provider such as a Lightweight Directory Access Protocol (LDAP) server or a Microsoft Active Directory server.

- Decide whether users who log in with Web applications will be initially authenticated on a mid-tier server or on the metadata server.
- Plan which authentication domain you will associate with each logical server. All servers within an authentication domain must use the same authentication provider. Each set of credentials that you store in the metadata provides access to all of the servers in a particular authentication domain.
- Decide whether you will allow users to access computing resources using shared accounts. You can enable multiple users to access a server with a shared user account by storing the credentials for the shared account in a login that you assign to a user group. The credentials for the shared account are available to every user who is a member of the user group.

  Shared accounts provide these advantages:

  - minimize the number of individual user accounts that you must create and maintain in the operating system (or other authentication provider)
  - minimize the number of user credentials that you must store in the metadata repository to support additional authentication
  - facilitate server pooling, which can enhance performance.

  Shared accounts provide these disadvantages:

  - reduce individual accountability
  - reduce your ability to make access distinctions in the operating system or database authorization layers
  - require you to carefully coordinate the logins so that no user has access to more than one login for a particular authentication domain.

After you make these preliminary decisions, you can begin more detailed planning for users, user groups, and access controls.

Planning Your Users

This section helps you plan the user accounts that you will need in the operating system (or with other authentication providers) and the user identities that you will need in the metadata. The initial set of role-based accounts that are required for installation are described in "Overview of Pre-Installation Tasks" on page 93.

In this phase, you begin by making a list of individuals in your organization who need to access resources in the SAS Intelligence environment. Then you analyze this information to determine these things:

- which accounts you must create in the operating system (or alternative authentication provider) for each user
- which user definitions you must create in the metadata repository
- which user credentials you must store in the metadata.

Planning the User Accounts

Make a list of the users who will access resources in your SAS Intelligence environment. Include everyone from consumers of unsecured published reports to information architects to system administrators. Plan to establish individual or shared accounts that enable each user to access every system that will verify that user's identity. These accounts can be any of the following:

- local accounts in the operating system of the computer on which the authenticating server is running
- network accounts that provide access to the operating system of the computer on which the authenticating server is running
- LDAP or Active Directory accounts (if the authenticating server is using one of these alternative authentication providers)
- user accounts for database authentication.

Note: On Windows platforms, the accounts must have certain user rights, as documented in "Pre-Installation Checklist for Windows" on page 96. △

User Accounts for Initial Authentication

The accounts that you create to support initial authentication enable a metadata server, a SAS OLAP Server, or a Web application to verify the credentials that users submit when they log in. To identify the user accounts that are needed to support initial authentication, complete the following analysis:

1. For each application
   - determine whether initial authentication will be handled by a SAS Metadata Server, a Web application, or a SAS OLAP Server.
   - identify the authentication provider. In most cases, the authentication provider will be the host operating system. In some cases, the authentication provider will be an alternative provider such as LDAP or Active Directory.
2. Identify groups of applications that will use the same authentication provider.
3. For each group of applications, create a list of users. If multiple users share a single account with the authentication provider, include the shared account in the list rather than the individual users who will use the shared account.
4. For each group of applications, determine whether any of the required accounts already exist in the authentication provider. Make a list of the additional accounts that you will need to create. You can create these accounts before you begin installation.

For example, in a deployment that includes SAS Information Delivery Portal, SAS Management Console, and SAS ETL Studio, you might have the following initial authentication processes:

- For SAS Information Delivery Portal, you choose to have the Web application handle authentication using LDAP as the authentication provider.
- For SAS Management Console and SAS ETL Studio, the metadata server handles authentication and you choose to use the operating system as the authentication provider.

In this example, you need the following user accounts to support initial authentication:

- an LDAP account for every user who logs in to SAS Information Delivery Portal.
- an operating system account on the computer on which the metadata server is running for each user who logs in to SAS ETL Studio or SAS Management Console.
- one operating system account on the computer on which the metadata server is running so the Web application (SAS Information Delivery Portal) can access the metadata server as a *trusted user*.

  Note: For information about setting up *trusted users*, see "Configuring Special Users" in the *SAS Metadata Server: Setup Guide* (**`support.sas.com/rnd/eai/openmeta/v9/meta`**). △

User Accounts for Additional Authentication

The accounts that you create to support additional authentication enable workspace servers, stored process servers, and other data servers to verify the identity of users who make requests that require access to resources on those servers.

To determine which user accounts are needed to support additional authentication, complete the following analysis:

1. Review your plan for which servers you will associate with which authentication domains. All servers that are associated with the same authentication domain must share the same authentication provider.
2. For each authentication domain, make a list of users who will access resources on servers in that authentication domain. If multiple users will use a shared account to access resources in a particular authentication domain, include the shared account in the list rather than the individual users who will use the shared account.
3. For each authentication domain, determine whether any of the required accounts already exist in the authentication provider. Make a list of the additional accounts that you will need to create. You can create these accounts before you begin installation.

For example, if the deployment described in the previous section includes stored process and workspace servers running on z/OS and a database server running on UNIX, you would have the following additional authentication processes:

- z/OS host authentication for the stored process and workspace servers
- database authentication for the database server.

In this example, to enable all of your users to authenticate to all servers, you would need these accounts:

- a z/OS operating system account for every user (or for each group of users who will share an account)
- an account on the database server for every user (or for each group of users who will share an account).

Planning the User Metadata Identities

In addition to their user accounts, many users must also have a unique metadata identity. You can define specific access controls, group memberships, or logins for only those users who have a unique metadata identity. Some sites create a unique metadata identity for every user. Other sites have a set of users who do not have their own metadata identities. These users access resources as members of the PUBLIC group, which includes everyone who can access the metadata server. You can define access controls for the PUBLIC group.

Plan to establish a unique metadata identity for any user who meets any of the following criteria:

- requires access to resources that differs from the access that you will give to the PUBLIC group
- logs in with an application that requires a metadata identity. For example,
  - SAS Enterprise Miner requires that every user has a unique metadata identity
  - SAS Information Delivery Portal requires that users have a metadata identity in order to access resources other than those in the Public Kiosk.

A user's metadata identity consists of a user definition that includes a login that the metadata server can use to resolve the user's identity, as described in "How the Metadata Server Determines Your Metadata Identity" on page 31.

You should create all of your user definitions in a single foundation repository. You can create user definitions in either of the following ways:

- import user and group data from an external system into a metadata repository by using the MDUIMPC.SAS and MDUIMPL.SAS autocall macros that SAS provides in the autocall libraries. For instructions, see "Registering Users Using SAS AutoCall Macros" in the *SAS Metadata Server: Setup Guide* at **`support.sas.com/rnd/eai/openmeta/`**.
- use the User Manager in SAS Management Console to create user and group definitions one at a time. For instructions, see the *SAS Management Console: User's Guide*.

Planning the Logins

In addition to the login that the metadata server uses to determine a user's metadata identity, determine which credentials you must store in the metadata for additional authentication.

Plan how you will store the necessary credentials in the metadata. You must coordinate the logins so that no more than one login is available to each user for each authentication domain.

- The user ID and password for an individual account are stored in a login that is owned by a user's metadata identity.
- The user ID and password for a shared account are stored in a login that is owned by a user group's metadata identity.

Plan who will add the necessary logins to your user and group definitions.

- Each user can add logins to his or her user definition by using SAS Management Console.

  Note: As an alternative, users can use the SAS Personal Login Manager utility to add or update their own login information. For information about this utility, contact your SAS representative. △

- *Administrative users* and *unrestricted users* can add logins to any user definition.

Note: Only an *unrestricted user* can access a login that is owned by another user in order to reset the user's password. An *unrestricted user* can overwrite the existing password but cannot view the password. △

User Planning Summary

The requirements for user accounts, metadata identities, and logins vary depending on factors such as the number of different authentication processes and the particular applications that are being used. One important factor is whether shared accounts or individual accounts are used to provide access to workspace, stored process, and data servers. Deciding whether to use shared accounts in a particular deployment requires careful consideration of the trade-offs between security, ease of deployment, maintainability, and performance.

The following table summarizes how the various factors affect user requirements. The table describes requirements for two environments:

- The Low Security Environment column lists the requirements for a deployment where shared accounts are used to access servers such as stored process servers, workspace servers, OLAP servers, and database servers during additional authentication. You can store the credentials for a shared account in the metadata as a login that is owned by any user group.

  Note: In the table, the shared logins are owned by the PUBLIC group to illustrate the lowest security configuration. Everyone who can access the metadata server is a member of PUBLIC, and PUBLIC is the only group that does not require its members to have their own metadata identities. △

- The High Security Environment column lists the requirements for a deployment where every user has an individual account for each authentication domain that contains resources that the user will access.

Table 4.1 User Planning Summary

Requirement	Low Security Environment	High Security Environment
Accounts for initial authentication	In most cases, each user must have an account with the metadata server's authentication provider. If a Web application (such as SAS Information Delivery Portal) is configured to authenticate users in the mid-tier, then each user of that application must instead have an account with the Web server's authentication provider. For details, see "Initial Authentication on a Mid-Tier Server" on page 33.	
Accounts for additional authentication	All users share a single account with the authentication provider for each authentication domain. The credentials for these shared accounts are stored in logins that are owned by the PUBLIC group.	Each user must have an account for each authentication domain that contains resources the user will access. The credentials for these individual accounts are stored in logins that are owned by the user's metadata identity.
User definitions	Only those users who log in with an application that requires a metadata identity* must have a user definition.	Every user must have a user definition.
Logins for inbound use	Each user definition must include a login that can be used to establish the user's metadata identity. This login must contain the user's fully qualified user ID. For details, see "How the Metadata Server Determines Your Metadata Identity" on page 31.	
Logins for outbound use	The PUBLIC group definition must include one login for each authentication domain. Each login must contain the user ID and password for a shared account and must specify the authentication domain to which it provides access.	In most cases, each user definition must include one login for each authentication domain that contains resources that the user will access. Each login must contain the user ID and password for a user account and must specify the authentication domain to which it provides access. In some cases, users of Web applications can access resources in one authentication domain using credentials that are cached from initial authentication. For details, see "Cached Credentials" on page 37.

* SAS Enterprise Miner requires each user to have a unique metadata identity. SAS Information Delivery Portal will not allow users who do not have a metadata identity to access resources beyond the Public Kiosk.

The preceding table assumes that you have a diverse environment where not all of the workspace, stored process, and data servers use the same authentication process. A diverse environment can include workspace and stored process servers running on different platforms and database servers that have their own authentication process. In

a diverse environment, you must have a separate authentication domain for each distinct authentication process that is used during additional authentication.

Rather than being strictly high security or low security, some sites will make limited use of shared accounts. For example a deployment might use a shared account to provide access to a database server without storing individual user credentials for the database server in the metadata.

Another intermediate approach is to define multiple shared accounts for a particular server and store the credentials for each shared account in a login that you assign to a particular user group. This enables you to differentiate users in the operating system or database authorization layers. For example, to use shared accounts that support two levels of access to a database server, you would perform the following tasks:

1. On the database server, define two accounts. Assign different privileges to each of the two accounts in the database authorization layer.
2. In the metadata, create two user groups. Give each of the group definitions a login that contains the credentials for one of the database server accounts.
3. In the metadata, add each user who will access the database server to one of the group definitions, in accordance with the privileges each user should have in the database. Do not assign a user to both groups because this causes more than one login for the database server's authentication domain to be available to the user.

User Requirements in Environments That Have Homogeneous Authentication

In a deployment that uses the same authentication provider for both initial authentication and for a particular authentication domain, the credentials that a user submits during initial authentication are also appropriate for accessing servers in that authentication domain. For example, if you log in with SAS Information Delivery Portal and your initial authentication is handled by a metadata server that is using Windows host authentication, the credentials that you submit can be used to access a stored process server that is also using Windows host authentication.

In a high security environment, using the same authentication provider for both initial and additional authentication reduces the user requirements as follows:

- In a deployment that has only one authentication domain, users who log in with an application that caches credentials do not need any outbound logins.
- In a deployment that has more than one authentication domain, users who log in with an application that caches credentials do not need an outbound login for the one authentication domain that is using the same authentication provider as is used for initial authentication.

In a low security environment, using the same authentication provider for both initial and additional authentication does not affect the user requirements. However, in such an environment, users who log in with a Web application access some resources with their personal, cached credentials rather than with a login that is assigned to the PUBLIC group.

Planning Your User Groups

This section helps you plan the user groups that you will define in the metadata. Your security goals might require that you also create user groups in other layers.

The main purpose of organizing your users into groups is to simplify the process of establishing and managing access controls for authorization. Granting access to resources on an individual basis can be cumbersome. After you define user groups, you can assign permissions to groups rather than to individual users. In some deployments,

you also use user groups to support server pooling or manage credentials for shared accounts.

In this phase, you begin by gathering information about your users' business tasks. You then analyze this information to develop a list of user groups, a diagram of your group hierarchy, and a list of members for each group.

Identifying Related Tasks

Make a list of the business tasks that each user performs and the content domain (such as the business unit, job title, or geographic region) in which each user operates. For example, your list might include these activities:

- viewing reports in a particular content domain (such as human resources)
- creating reports in a particular content domain
- scheduling jobs
- defining objects that represent computing resources in a metadata repository
- creating and maintaining user and group definitions.

Organize your list of user tasks into logical groups. In this process, look for variations in

- the content domain
- the computing resources that are involved
- the type of access that is required
- the level of sensitivity of the underlying data.

Analyze your task lists to identify which user groups you need. In this process, keep in mind your security goals. If you do not intend to define different levels of access for two sets of users, then you usually do not need to create separate user groups to represent each of those sets of users.

For example, if you want everyone to have read access to all of your data and you want to limit write access by job function, your list of tasks and groups might look like the tasks in the following table.

Table 4.2 A Simple Tasks-to-Groups Mapping

Tasks	Group
Create users and groups. Administer servers and repositories. Set repository level security.	Administrators
Define metadata for data resources. Define ETL processes. Schedule jobs.	ETL Developers
Create stored processes. Register stored process in the metadata.	Business Analysts
View all reports (run all stored processes).	Report Consumers
View unsecured reports (run PUBLIC stored processes).	Everyone (PUBLIC)

Note: Membership in a user group that is named Administrators does not enable you to perform tasks that require status as an *administrative user* of the metadata server. For details, see "Special Users of the Metadata Server" on page 49. △

Defining the Group Structure

Decide how the groups that you have identified should relate to each other. In the metadata layer, one group can be a member of other groups. For example, a regional sales group can be a member of a worldwide sales group.

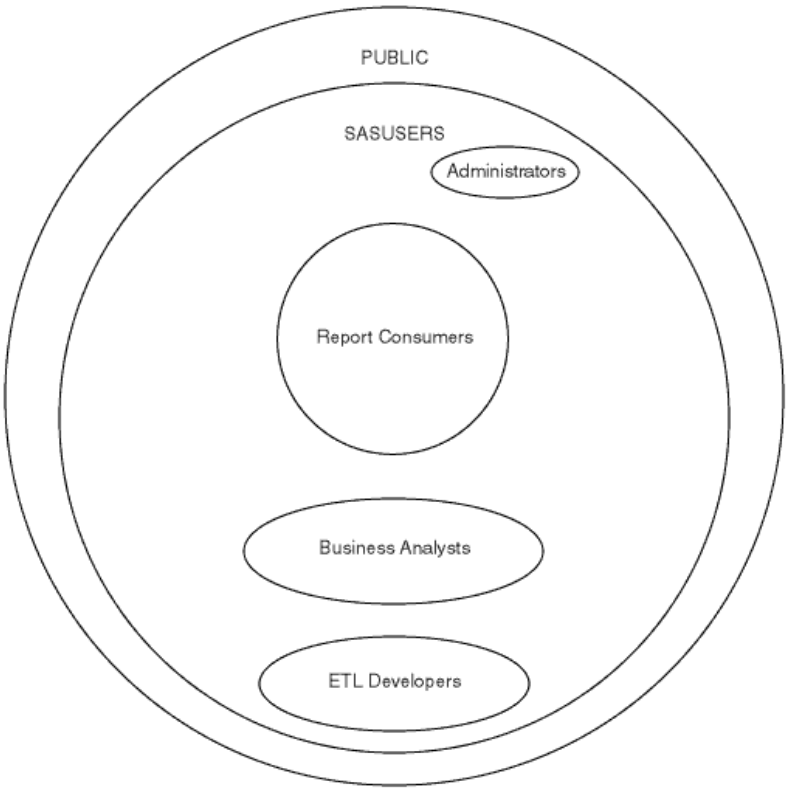
In the metadata layer, your group structure will consist of the user-defined groups that you have identified and the two standard groups that have implicit membership:

PUBLIC — If you can access the metadata server, then you are a member of the PUBLIC group. Members of the PUBLIC group do not have to have their own unique metadata identities. You can define logins and access controls for the PUBLIC group.

SASUSERS — If you are a member of the PUBLIC group and you have a metadata identity that owns a login with a user ID that matches the fully qualified form of the user ID with which you are initially authenticated, then you are also a member of the SASUSERS group.

The following figure depicts one way you could structure the user groups that were identified in the simple tasks-to-groups mapping in the previous section.

Figure 4.1 A Simple Group Structure



This group structure reduces the number of access controls that you have to define and maintain. For example, if you give the PUBLIC group ReadMetadata permission for the repository, you do not also have to give ReadMetadata permission to the ETL Developers group. Members of the ETL Developers group will have ReadMetadata permission for the repository because they are also members of the PUBLIC implicit group.

Note: The figure does not illustrate an individual user's group memberships. The figure shows only which entire groups are members of other groups. For example, the Report Creators group is a member of the Power Report Consumers group. △

Implementation Strategy for Defining Groups

Determine which individuals in your organization should be assigned to each of the user-defined groups that you identified. You will usually assign each user definition to one or more group definitions in the metadata. Remember that you do not always have to directly add individual users to every user group; groups can also be members of other groups.

You should create all of your user group definitions in a single foundation metadata repository. You can create group definitions in either of these ways:

- importing user and group data from an external system into a metadata repository by using the MDUIMPC.SAS and MDUIMPL.SAS autocall macros that SAS provides in the autocall libraries. For more information, see "Importing User Information" on page 86.
- creating the user and group definitions one at a time by using the User Manager in SAS Management Console. For instructions, see the *SAS Management Console: User's Guide*.

You can define logins in the metadata for your user groups, but you are not required to do so. A login that is associated with a group definition is available to all of the members of that user group.

Planning Your Access Controls

Immediately after you complete the standard installation, you should secure the foundation metadata repository. A recommended approach is described in "Protecting the Foundation Repository" on page 72. This section describes a planning process for defining the rest of your access controls in the metadata authorization layer. Your security goals might require that you also define access controls in other authorization layers.

For each resource that you define or register in a metadata repository, the initial access controls consist of

- permissions that are specified in the repository ACT
- permissions that are inherited from the resource's parent objects.

As you add resources to a metadata repository, your security goals might require that you modify the default access controls for those resources. It is recommended that you use a planned, organized approach to setting these controls. In any environment, access controls that are defined for individual users in an ad-hoc fashion quickly become difficult, if not impossible, to manage. Starting with an appropriate user group structure can greatly simplify the process of assigning and maintaining permissions.

Access control planning requires a thorough understanding of the way that metadata layer permissions work, of your computing resources, and of the needs of the users who interact with those resources. The following sections contain detailed information that will help you complete a three step access control planning process:

1. Familiarize yourself with the metadata authorization layer permissions that you can use to control access to resources. In order to set effective access controls in the metadata layer, you must understand which permissions control which actions

and recognize that in the current release, the application permissions (Read, Write, Create, Delete, Administer) are not always enforced.

2. For each task that your users perform, identify which resources are involved and which permissions to each resource are required. In this step, you identify the minimum set of permissions to resources that you must grant to each user group so that your users can accomplish their tasks.
3. Develop a strategy for establishing the necessary access controls. In this step, you decide *how* you will grant and deny permissions to resources.

Metadata Layer Permissions

Metadata layer permissions are permissions that are defined and evaluated by the metadata server's authorization facility. Some metadata layer permissions are enforced by the metadata server. Other metadata layer permissions are not enforced by the metadata server but can be enforced by other applications.

CAUTION:
In the current release, the only metadata layer permissions that are always enforced are ReadMetadata, WriteMetadata, and CheckInMetadata (which is applicable only to SAS ETL Studio users who are working in a change managed environment). See "Considerations for Defining Effective, Efficient Access Controls" on page 68 for best practice recommendations for using metadata layer permissions. △

ReadMetadata, WriteMetadata, and CheckInMetadata Permissions

The metadata server requests and enforces authorization decisions for the ReadMetadata, WriteMetadata, and CheckInMetadata permissions. Because all applications in the SAS Intelligence Architecture use the metadata server when accessing resources, the permissions that are enforced by the metadata server provide the strongest protections that are available in the metadata layer.

The following table summarizes the actions that are controlled by each of these permissions.

Table 4.3 Permissions That the Metadata Server Enforces

Permission (Abbreviation)	Actions Controlled
ReadMetadata (RM)	Reading a metadata object.
WriteMetadata (WM)	Creating, updating, or deleting a metadata object.
CheckInMetadata (CheckInM)	Checking in metadata from a project repository, and checking out metadata to a project repository.

Read, Write, Create, Delete, and Administer Permissions

Some applications request and enforce authorization decisions for the Read, Write, Create, Delete, and Administer permissions. The following table summarizes the actions that are controlled by each of these permissions. For details about how a SAS server or application enforces permissions, see the documentation for the server or application in SAS OnlineDoc (available from **`support.sas.com/documentation/onlinedoc/`**).

Table 4.4 Permissions That Other Applications Can Enforce

Permission (Abbreviation)	Actions Controlled
Read (R)	Reading data from the resource that is described by a metadata object. For example, on an OLAP cube the Read permission controls viewing of the data within the cube.*
Write (W)	Updating data in the resource that is described by a metadata object. For example, on a table the Write permission controls updating the rows in the table.**
Create (C)	Adding data to the resource that is described by a metadata object. For example, on a table the Create permission controls adding rows to the table.**
Delete (D)	Deleting data from the resource that is described by a metadata object. For example, on a library the Delete permission controls deletion of tables from the library.**
Administer (A)	Accessing the administrative interfaces of SAS servers, such as the SAS OLAP Server, the SAS Stored Processes Server, and IOM spawners.

* In the current release, the Read permission is enforced only if the SAS OLAP Server or the SAS metadata LIBNAME engine is used to access the data.

** In the current release, the Create, Write, and Delete permissions are enforced only if the SAS metadata LIBNAME engine is used to access the data.

Because not all applications enforce all permissions, the Read, Write, Create, Delete, and Administer permissions are not always sufficient to control access in the current release. For example, even if PersonA is denied Read access to DataSet1, PersonA *can* view the data in DataSet1 if PersonA is using SAS ETL Studio. SAS ETL Studio does not enforce the Read permission when accessing data sets. You can prevent PersonA from seeing DataSet1 by denying PersonA the ReadMetadata permission for the data set. You should also use another authorization layer (such as the data source authorization layer or the operating system authorization layer) to protect the data.

Access Requirements for Common Tasks

Before you assign permissions to resources, it is important to understand which resources are involved in each task and what type of access to each resource is required for each task. In order to perform a single task, you often must have access to multiple resources. For example, in order to run a stored process you must be able to access the stored process and the underlying data sources.

The following topics document the *metadata layer* permissions that are required in order to perform common tasks. For many of the tasks, there are also access requirements in other authorization layers.

Access Requirements for Working with Data

Users who use SAS ETL Studio, the metadata LIBNAME engine, or SAS Management Console to define and manage metadata that describes data sources must have permissions that enable them to view, create, modify, and delete that metadata. This topic describes the required metadata layer permissions for working with data.

Note: Permissions in other authorization layers are also required for most of these tasks. For example, although no metadata layer permissions to a data source are required in order to register the data source, some permissions in the data source and operating system authorization layers are usually required (because this task involves reading some information from the source). △

The column headings in the following table are the metadata objects that are most frequently involved when you define and manage metadata that describes data sources. Each table cell contains the permissions to a particular metadata object that are required for a particular task.

Table 4.5 Metadata Authorization Layer Access Requirements for Common Data Tasks

Task	Foundation Repository	Metadata Object That Describes the Data Source
Create metadata that describes a data source	RM, CheckInM (or WM)*	not applicable
View the metadata that describes a data source	RM	RM
Modify or delete metadata that describes a data source	RM, CheckInM (or WM)*	RM, CheckInM (or WM)*
View the data within a registered data source	RM	RM, R**

* If you are using SAS ETL Studio in a change-managed environment, then you must have CheckInMetadata permission. Otherwise, you must have WriteMetadata permission.

** In the current release, the Read permission is not always required because not all applications enforce this permission.

In a change-managed environment, the owner of each project repository should also have ReadMetadata and WriteMetadata permissions for the entire project repository. For more information about change management, see "Setting Up Change Management" on page 209.

Note: In order to navigate to a metadata object, you must also have ReadMetadata permission to the folder that contains the metadata object, and to all of that folder's parent folders. If you cannot see a folder, you cannot browse the objects that the folder contains. △

Access Requirements for Working with Stored Processes

Users who define and manage metadata that describes stored processes must have permissions that enable them to view, create, modify, and delete that metadata. A

stored process is a SAS program that generates output, such as a data set, a table or a graph. You can register a stored process in a SAS Metadata Repository, use metadata authorization layer access controls to secure the stored process, and run the stored process in the SAS Information Delivery Portal.

The column headings in the following table are the metadata objects that are involved when you register and manage stored processes. Each table cell contains the permissions to a particular metadata object that are required for a particular task.

Table 4.6 Metadata Authorization Layer Access Requirements for Common Stored Process Tasks

Task	Repository	Folder That Contains the Stored Process*	Server that Hosts the Stored Process	Stored Process	Data Sources
Create metadata that describes a stored process	RM, WM	RM, WM	RM, WM	not applicable	none
View metadata that describes a stored process	RM	RM	RM	RM	none
Modify metadata that describes a stored process	RM	RM	RM	RM, WM	none
Delete metadata that describes a stored process	RM	RM, WM	RM, WM	RM, WM	none
Run a stored process	RM	RM	RM	RM	RM, R**

* In order to navigate to a stored process, you must have ReadMetadata permission to the folder that contains the stored process, and to all of that folder's parent folders. If you cannot see a folder, you cannot browse the objects that the folder contains. If you access a stored process by searching, it is not necessary to have ReadMetadata permission for the parent folders.

** In the current release, the Read permission is not always required because not all applications enforce this permission.

CAUTION:

A stored process that runs on a stored process server (or a pooled workspace server) accesses data using the account under which the server is running. Because your account is not being used to access the data, your permissions to the data are not relevant. In these circumstances, it is particularly important to set appropriate access controls to secure the stored process. △

Note: For more information about these security considerations, see "Planning Security on Workspace and Stored Process Servers" in the *SAS Integration Technologies Administrator's Guide* at **`support.sas.com/rnd/itech/doc9/admin_oma/security/auth`**. △

Access Requirements for Working with Security-Related Objects

The metadata authorization layer provides additional protections for security-related objects such as user definitions, logins, and Permission objects. For example, only an *unrestricted user* or an *administrative user* can add user definitions to a metadata repository.

The access requirements for performing common security-related tasks are documented in the online help for the Authorization Manager plug-in to SAS Management Console and in the *SAS Metadata Server: Setup Guide*, which is available at **support.sas.com/rnd/eai/openmeta**.

Implementation Strategy for Assigning Permissions

This section contains best practice recommendations for setting access controls. Before you begin setting permissions, you should understand the different ways you can set permissions in the metadata authorization layer and be familiar with the way permissions are evaluated.

Considerations for Defining Effective, Efficient Access Controls

The following measures can enhance the effectiveness of the protections that you establish in the metadata layer:

- To control access to the computing resources that are represented by metadata objects, grant and deny metadata permissions in pairs. In the current release, the strongest protections come from the ReadMetadata, WriteMetadata, and CheckInMetadata permissions. The following table documents an approach that provides the best protections in the current release and the best compatibility for future releases.

Table 4.7 Recommended Use of Permissions

To Control This Action	Grant or Deny These Permissions
Reading a metadata object	ReadMetadata
Reading the data that is described by a metadata object	Read and ReadMetadata
Modifying a metadata object	WriteMetadata
Modifying data that is described by a metadata object	Write and WriteMetadata
Creating a new metadata object	WriteMetadata
Creating new data	Create and WriteMetadata
Deleting a metadata object	WriteMetadata
Deleting data that is described by a metadata object	Delete and WriteMetadata

- Use other authorization layers, such as operating system permissions and relational database controls, to secure data.
- Understand how permissions are evaluated in the metadata authorization layer. Remember that this layer supports multiple inheritance, and that the inheritance rules make it much easier to establish an effective grant of a permission than to establish an effective denial.

- Use caution when moving objects that inherit permissions from their folders. For example, moving a library from one folder to another in the ETL custom tree might change the metadata layer access controls for that library.
- Remember that your effective permissions are limited to access that is allowed in all applicable authorization layers.
- Remember that security is an ongoing process; you will need to define more access controls as you register additional resources in the metadata environment.

An important efficiency goal is to minimize the number of access controls that you have to set and maintain. Tactics that will help you achieve this goal include the following:

- Assign permissions to the highest appropriate user group in the group hierarchy.
- Use access control templates (ACTs) to centralize management of identity/permission patterns that you will apply to multiple resources.
- Assign permissions at the highest appropriate level in the resource inheritance structure.
- Use dedicated folders to manage access to stored processes. This enables you to set security for a stored process by adding the stored process to an appropriately secured folder, rather than setting controls on each stored process individually.
- If you have SAS ETL Studio, consider using dedicated folders in the ETL custom tree to manage access to the metadata that describes data.
- Consider whether it will be more efficient to assign permissions by inclusion or by exclusion.
  - When you assign permissions by inclusion, you begin by denying all access to resources and then selectively grant permissions where they are needed. This approach is typically used when you are following the rule of least privilege so that you grant only as much access as is required to do the job.
  - When you assign permissions by exclusion, you begin by granting broad access to resources and then selectively deny permissions where there is a need to protect resources or information. This approach is typically used when you are following the rule of least protection.
- Keep in mind the balance between deployability, usability, maintainability, and security that you selected for your environment.

CHAPTER

5

Implementing Security

Overview of Implementing Security

This chapter contains instructions for setting up users, user groups, and access controls in the metadata repository *after you complete the installation process that is documented in Part 3 of this guide*. Part 3 of this guide includes information about protecting your configuration directories with file system access controls, selecting an encryption method, and modifying your security configuration in "Configuring Security" on page 156.

In a medium or high security environment, these are the primary tasks in the security implementation process:

1. Protect the foundation metadata repository.

   CAUTION:
   At the end of the installation process, the foundation metadata repository is unprotected. Until you set some initial controls, anyone who can access the metadata server can create, view, modify, and delete most metadata objects in the repository. △

2. Set up security for users who will administer the metadata repository.
3. Secure the repository access control template (ACT) and the group definitions that you created during installation.

CAUTION:
By default, ACTs and user-defined groups are secured only by the repository ACT. You should set direct access controls on each user-defined group and ACT in order to control who can modify or delete these objects. △

4 Set up security for regular (non-administrative) users.

5 Perform ongoing security maintenance activities such as these tasks:
   - removing users
   - resetting passwords
   - adding users to a deployment
   - setting up security for resources that are added to an existing deployment
   - setting up security for servers that are added to an existing deployment

The following sections provide instructions for each of these tasks. The last section in this chapter describes a minimal security implementation that creates an extremely low security environment.

Protecting the Foundation Repository

In its initial state, the repository ACT for a foundation repository grants ReadMetadata and WriteMetadata permissions to the PUBLIC group, which includes everyone who can access the metadata server. This means that anyone who can access the metadata server can create, view, modify, and delete most metadata objects. This section describes how to set some initial access controls to protect a new foundation repository. Unless you intend to have a very low security environment, you should set some initial controls on the Users and Permissions tab of the repository ACT immediately after completing installation.

It is strongly recommended that you begin your security implementation by limiting the PUBLIC group's access to the repository. As your security implementation progresses, you can return to the repository ACT to selectively expand access to the repository. Typically, you will have the following access controls in place at the end of the security implementation process:

- All users who will access resources in the metadata environment should have ReadMetadata permission to the repository. Typically, you give either the PUBLIC group or the SASUSERS group ReadMetadata access to the repository, depending on your security goals.
- All users who will create new objects in the repository should have WriteMetadata permission to the repository. For example, WriteMetadata permission to the repository is required to register a stored process, add a login to your own user definition, or create metadata that describes data. Typically, you give either the SASUSERS group or selected user-defined groups WriteMetadata access to the repository, depending on your security goals.

One approach is to begin by denying the PUBLIC group all permissions to the repository. After you set these initial controls, you will use the *unrestricted user* account (SAS Administrator) to access the repository until you have set up security for your administrators.

Note: Depending on your security goals, you might choose to set more liberal initial controls. For example, you might leave the PUBLIC group's grant of ReadMetadata permission in place. The instructions in this chapter assume that you are following the more restrictive approach. △

To set initial protections for a foundation repository, complete the following steps:

1 Log in to SAS Management Console by opening a metadata profile with the *unrestricted user* account (SAS Administrator).
2 In the navigation panel, select

   Environment Management ► Authorization Manager

   ► Access Control Templates ► Default ACT

   Note: In SAS Management Console, the repository ACT is represented by a blue cylinder icon and is named "Default ACT" by default. △
3 Right-click and select Properties to open the Default ACT properties dialog box.
4 On the Users and Permissions tab, select PUBLIC in the **Names** list box.

   Note: If other users or groups are listed in the **Names** list box, do not modify or remove those entries. △
5 In the permissions list for the PUBLIC group, select the Deny check box for every permission.
6 Click OK to save the settings and close the Default ACT.

   Note: Do not set any permissions on the Authorization tab at this point. After you create an Administrators group in the next section, you will set direct access controls on the Authorization tab of the Default ACT so that only members of the Administrators group can modify or delete the Default ACT. △

At this point, only the *unrestricted user* account (SAS Administrator) can access the repository. With the repository well-protected, you can begin setting up security for administrators first and then for regular users.

Setting Up Security for Administrators

This section describes how to set up one or more users as administrators who can perform tasks such as adding other users, user groups, resources, and access controls to the metadata repository. A user who has status as an *administrative user* of the metadata server and who is granted ReadMetadata, WriteMetadata, and Administer permissions for the repository can perform all common metadata administrative tasks except viewing, updating, and removing logins for other users.

The recommended sequence is to create a user group for administrators, give the group broad access to the repository, and then populate the group by adding users to the deployment. This sequence is depicted in the following figure.

Figure 5.1 Sequence for Setting Up Security for Administrators

The process consists of the following steps:

1 Log in to SAS Management Console by opening a metadata profile with the *unrestricted user* account (SAS Administrator).
2 In the navigation panel, select User Manager.
3 Open the New Group properties dialog box by selecting the following path from the menu bar:

Actions ► New ► Group

Note: You can also right-click on User Manager and select these items from the pop-up menu. △

4 Create a group definition for your administrators.

a On the General tab of the New Group properties dialog box, enter "Administrators" as the group name.
b Click OK to save and close the group definition.

Note: Do not add any users or groups to the Members tab at this point. You will add users to the Administrators group in step 8. △

Note: Do not add any logins on the Logins tab. A login that you add to a group definition enables group members to access servers with a shared account. If your administrators need to access servers other than the metadata server, they will use individual accounts. (In the metadata, credentials for an individual account are stored as a login that you add to a user definition.) △

Note: Do not set any permissions on the Authorization tab at this point. You will secure the Administrator's group definition in step 6. △

5 Define the group's default access to the repository.

a In the navigation panel, select

Environment Management ► Authorization Manager

► Access Control Templates ► Default ACT

Note: In SAS Management Console, the repository ACT is represented by a blue cylinder icon and is named "Default ACT" by default. △

b Right-click and select Properties to open the Default ACT properties dialog box.
c On the Users and Permissions tab, click Add.
d In the Add Users and/or Groups dialog box, move the Administrators group to the **Selected Identities** list box and then click OK.
e On the Users and Permissions tab, select the Administrators group in the **Names** list and grant ReadMetadata, WriteMetadata, and Administer permissions to the Administrators group.*

Note: Within the Users and Permissions tab, permissions that are assigned to a user-defined group (such as Administrators) have precedence over permissions that are assigned to an implicit group (such as PUBLIC). △

6 Secure the ACT with directly assigned permissions. The goal is to prevent anyone other than a member of the Administrators group from modifying or deleting the Default ACT. You will take WriteMetadata permission away from PUBLIC and then give WriteMetadata permission back to the Administrators group. For detailed instructions on setting these permissions, see "Securing ACTs and User-Defined Groups" on page 75.

7 Secure the Administrators group definition with directly assigned permissions. The goal is to enable only members of the Administrators group to modify or

* The Administer permission enables you to access the administrative interfaces of SAS servers such as the SAS OLAP Server, the SAS Stored Processes Server, and IOM spawners. Depending on your security goals, you might choose to set this permission on specific server definitions rather than as a default permission for the entire repository.

delete the group definition. You will take WriteMetadata permission away from PUBLIC and then give WriteMetadata permission back to the Administrators group. For detailed instructions on setting these permissions, see "Securing ACTs and User-Defined Groups" on page 75.

8 Add individual administrators to the deployment by following the instructions in "Adding Users to a Deployment" on page 85.

9 Log out of SAS Management Console by closing the metadata profile that uses the *unrestricted user* account.

From this point forward, each administrator can use his or her own account to perform common administrative tasks, rather than continuing to share the highly privileged *unrestricted user* account.

Securing ACTs and User-Defined Groups

You should set direct access controls to protect every ACT in the repository and every group definition that you create. Until you set these access controls, any user who has WriteMetadata access to the repository can modify or delete your group definitions and ACTs.

Note: It is not necessary to set access controls to secure the PUBLIC and SASUSERS groups. Because they are standard groups with implicit membership, these groups have special protections. △

If you have not already done so, you should secure the repository ACT and any group definitions that you created during installation. To secure these objects, you set permissions on the Authorization tab of each group definition and each ACT. The permission settings on a group's Authorization tab determine who can make changes to the group definition; these settings do *not* define the actions that group members can take on other objects. Similarly, the permission settings on the Authorization tab for an ACT determine who can make changes to the ACT; these settings do not define the actions that users can perform on the repository.

If your goal is to enable only members of the Administrators group to modify or delete your group definitions and ACTs, you should take WriteMetadata permission away from PUBLIC and then give WriteMetadata permission back to the Administrators group.

Note: The data permissions (Read, Write, Create, Delete) are not relevant when you are protecting a group definition or an ACT because these objects exist only as metadata objects. △

You can set these controls by completing the following steps:

1 In SAS Management Console, select the group definition or ACT that you want to secure.

2 Right-click on the group definition or ACT and select Properties from the pop-up menu.

3 On the Authorization tab:

   a In the **Names** list, select PUBLIC. In the permissions list for the PUBLIC group, the Deny WriteMetadata check box should already be selected and have a gray background color. This denial comes from the pattern that you defined on the Users and Permissions tab of the repository ACT.

   Note: These instructions assume that you denied all permissions to the PUBLIC group on the Users and Permissions tab of the repository ACT, as instructed in "Protecting the Foundation Repository" on page 72. △

b Select the (already selected) Deny WriteMetadata check box to add a directly assigned denial of WriteMetadata permission for the PUBLIC group. The directly assigned denial is indicated by the absence of a background color.

Note: The directly assigned denial ensures that the group definition or ACT will remain protected as you expand default WriteMetadata access to the repository. The PUBLIC group's directly assigned denial of WriteMetadata will override any grants of WriteMetadata that come from the repository ACT. △

c In the **Names** list, select the Administrators group. In the permissions list for the Administrators group, the Grant WriteMetadata check box should already be selected and have a gray background color. This grant comes from the pattern that you defined on the Users and Permissions tab of the repository ACT.

d Select the (already selected) Grant WriteMetadata check box to add a directly assigned grant of WriteMetadata permission for the Administrators group. The directly assigned grant is indicated by the absence of a background color.

Note: Members of the Administrators group are also members of the PUBLIC group. In the identity precedence rules, permissions that are directly assigned to a user-defined group (such as Administrators) have precedence over permissions that are directly assigned to an implicit group (such as PUBLIC). Members of the Administrators group will be able to modify or delete the current group definition or ACT because the directly assigned grant to the Administrators group overrides the directly assigned denial to the PUBLIC group. △

4 In the properties dialog box, click **OK** to save the settings and close the group definition or ACT.

Using an ACT to Secure ACTs and Group Definitions

As an alternative to setting these access control entries (ACEs) on the Authorization tab every group definition and ACT that you create, you can use the following approach:

1 Create one ACT that has the desired identity/permission pattern on its Users and Permissions tab.

2 Apply that ACT on the Authorization tab of every group definition and ACT that you create.

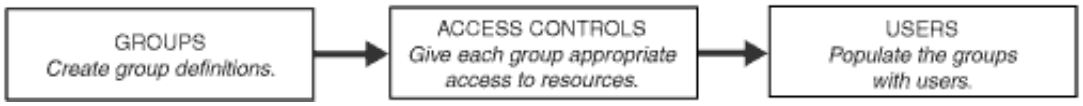
This gives you a centralized way to manage access to your ACTs and group definitions. If you change your mind about which groups should be able to modify or delete your ACTs and group definitions, you can make the change in one place (on the ACT) rather than revisiting every ACT and group definition to change the individual ACEs that you set on each object. For detailed instructions on creating an ACT and applying it to resources, see the *SAS Management Console: User's Guide*.

Note: If you want different groups to be able to make changes to particular ACTs or group definitions, then you cannot use a single ACT to manage access to all of these objects. You use an ACT to manage access when you want to apply the same identity/permission pattern to multiple resources. △

Setting Up Security for Regular Users

When you set up security for regular users, you use the same general sequence (groups, then access controls, then users) that you used to set up security for administrators. The following figure illustrates the sequence.

Figure 5.2 Sequence for Setting Up Security for Regular Users



In this sequence, you establish the user groups and access controls before you add individual users to the deployment.

- This reflects the best practice of centralizing management of access controls by assigning permissions to user groups rather than to individual users
- This enables you to separate tasks that are typically performed by a security architect (such as designing the user group structure and the access control strategy) from the more administrative tasks (such as adding individual users to a deployment).

A medium or high security environment will have several user-defined groups in addition to the two standard user groups (PUBLIC and SASUSERS) that have implicit membership. A process to help you identify the user-defined groups that you need is described in "Planning Your User Groups" on page 59. When you set up security for regular users, you can use either of the following approaches:

- Perform the entire sequence separately for each group (or set of groups) in the deployment. This phased approach is appropriate for a gradual roll-out by job function.
- Perform the entire sequence one time for the entire deployment. In this approach, you set up all of the groups, then create all of the access controls, then add users to the deployment. This enables you to do a small scale (but comprehensive) pilot by adding a few users to each group and then verifying that you get the security behaviors that you expect.

The following list provides step-by-step instructions for the latter approach.

1. Log in to SAS Management Console by opening a metadata profile with one of the *administrative user* accounts that you created in the previous section.

   Note: You do not have to be an *administrative user* in order to manage group definitions. Any user who has WriteMetadata permission to a repository can create user-defined groups in that repository. Any user who has WriteMetadata permission to a group definition can modify that group definition. △

2. In the navigation panel, select User Manager.
3. Open the New Group properties dialog box by selecting the following path from the menu bar:

   Actions ► New ► Group

   Note: You can also right-click on User Manager and select these items from the pop-up menu. △

4. Create the group definitions. For each group definition, complete the following steps:

a On the General tab, enter a name for the group.

b On the Logins tab, add a login to the group definition only if you are using the group to provide access to servers using a shared account. A login on a group definition should contain credentials for shared account that provides access to servers in a particular authentication domain. The login should be associated with the authentication domain to which it provides access. You can add multiple logins to a group definition so that each login contains credentials for a shared account in a separate authentication domain.

For each login that you add to a group definition, a corresponding shared user account must be established in the operating system (or other authentication provider). Storing the credentials for a shared account in the metadata does not eliminate the need to create the account in the authentication provider.

Note: In a very high security environment, you would not use any shared accounts. In a very low security environment, you might store credentials for shared accounts as logins that you add to the PUBLIC or SASUSERS group definitions. For a discussion of the security, convenience, and performance trade-offs of using shared accounts, see "Making Preliminary Decisions about Your Security Architecture" on page 52. △

c On the Members tab, add any other groups that are members of the current group, in accordance with your planned group structure. For information about planning a nested group hierarchy, see "Defining the Group Structure" on page 61.

Unless you define groups in a particular order, you might have to return to a group's Members tab to add other groups as members. For example, if you want GroupA to be a member of GroupB, you must do either of the following:

- Create the GroupA before you create GroupB. This enables you to add GroupA as a member when you define GroupB.
- Create GroupA after you create GroupB. This requires you to return to the Members tab of GroupB after you create GroupA to add GroupA as a member.

d Click OK to save and close the group definition.

5 Secure each group definition with directly assigned permissions. If your goal is to enable only members of the Administrators group to make changes to your group definition, you will take WriteMetadata permission away from PUBLIC and then give WriteMetadata permission back to the Administrators group. For detailed instructions on setting these permissions, see "Securing ACTs and User-Defined Groups" on page 75.

6 Define default access to the repository for each user group, beginning with the two standard groups—PUBLIC and SASUSERS. Membership in these groups is implicit, so default access to the repository for most users can be established through permissions that you set for PUBLIC or SASUSERS. Next, determine whether any of your user-defined groups need additional default access to the repository (beyond the access that you have already granted to PUBLIC and SASUSERS).

For example, in a high security environment you might set the following permissions on the Users and Permissions tab of the repository ACT:

- Leave the denials of all permissions to PUBLIC in place. (You denied these permissions when you set initial access controls to protect the metadata repository.)
- Grant ReadMetadata to SASUSERS.

- Grant WriteMetadata to selected user-defined groups in accordance with your security plan.

  Note: It is not necessary to also grant ReadMetadata to the selected user-defined groups because members of those groups are also members of SASUSERS, and SASUSERS has ReadMetadata access to the repository. △

Note: In a medium security environment, you might choose a less restricted configuration, such as granting ReadMetadata access to PUBLIC and WriteMetadata access to SASUSERS. In this scenario, it is not also necessary to grant ReadMetadata to SASUSERS because all members of SASUSERS are also members of PUBLIC, and PUBLIC has ReadMetadata access to the repository. △

7 Modify the default access to selected resources by setting additional ACEs and ACTs in accordance with your security plan. For more information, see "Controlling Access to Resources" on page 86.

8 Add individual users to the deployment by following the instructions in "Adding Users to a Deployment" on page 85.

Example: Set Up Security for Regular Users

This example demonstrates how to set up regular users in a diverse high security environment. The example assumes that you have already performed both of the following tasks:

- Set initial access controls on the Users and Permissions tab of the repository ACT as described in "Protecting the Foundation Repository" on page 72.
- Set up security for your administrators as described in "Setting Up Security for Administrators" on page 73.

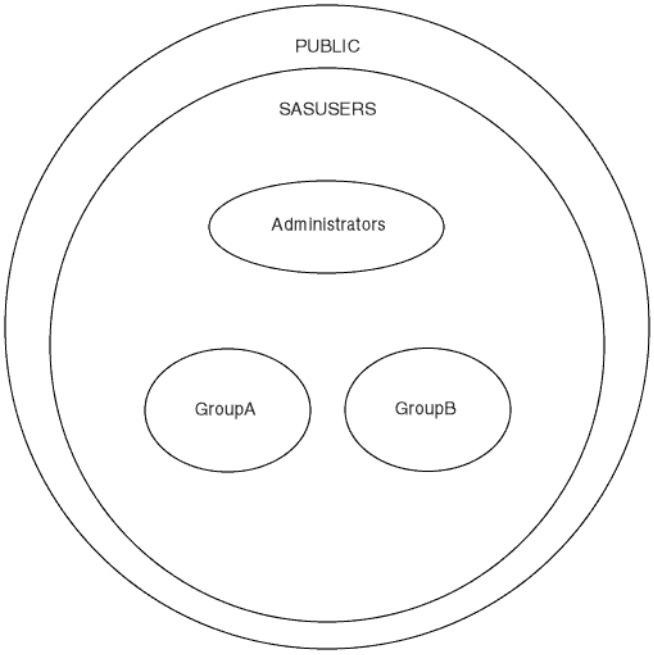
Security Goals and Configuration

These are the security goals for this example:

- To establish a relatively high security environment.
  - You will not give any access to the repository to users who do not have a metadata identity.
  - By default, any user who has a metadata identity has read access to all resources in the repository.
  - Users will access the workspace and stored process servers using individual accounts.
  - You will allow some users to access a third-party database server using a shared account. Because the data on this server is not sensitive, you are willing to trade-off individual accountability for the convenience of not having to maintain individual credentials in the metadata for the database server.
- To establish mutually exclusive access controls for the data in two SAS libraries so that
  - one set of users has exclusive access to the SAS data in LibraryA
  - another set of users has exclusive access to the SAS data in LibraryB
  - the Administrators group has ReadMetadata access to both libraries.
- To enable only the users who access data in LibraryA to access the third-party database server.
- To give only members of GroupA, GroupB, and Administrators default write access to the repository.

The following figure depicts the group structure for this example:

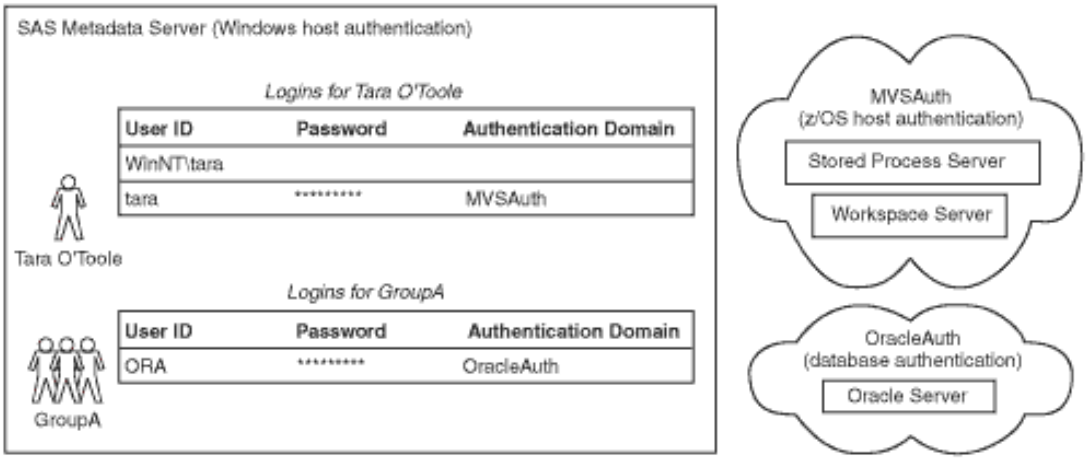
Figure 5.3 Group Structure



In this example, you have a diverse environment where more than one authentication process is being used. The following figure depicts these things:

- the servers and authentication domains for this example
- the logins for Tara O'Toole, which is an example of the metadata identities that you will create in this example
- the shared login that you will define to enable GroupA to access the Oracle Server.

Figure 5.4 Servers, Authentication Domains, and Logins



Implementation Process

To set up security for regular users in this example, complete the following steps:

1. Log in to SAS Management Console by opening a metadata profile with your *administrative user* account.
2. Use User Manager to create a group definition for GroupA.
   a. On the General tab, enter "GroupA" as the group name.
   b. On the Logins tab, add a login for the database server. This login should specify the user ID (ORA) and the password for a shared account that you have created on the database server. From the **Authentication Domain** drop-down list, select OracleAuth to associate the login with the authentication domain in which the Oracle server is defined.
   c. Click **OK** to save and close the group definition.

   Note: No other groups are members of GroupA, so you do not need to add any members on the Members tab. △
3. Use User Manager to create a group definition for GroupB.
   a. On the General tab, enter "GroupB" as the group name.
   b. Do not add any logins on the Logins tab. Members of GroupB do not access the database server.
   c. Click **OK** to save and close the group definition.

   Note: No other groups are members of GroupB, so you do not need to add any members on the Members tab. △
4. On the Authorization tab for each group, secure the group definition by directly assigning access controls that deny WriteMetadata to PUBLIC and grant WriteMetadata to the Administrators group. For detailed instructions, see "Securing ACTs and User-Defined Groups" on page 75.

   Note: Directly assigned permissions do not have a gray background color. If the background color for a permission on the Authorization tab for a group definition is gray, that permission comes from the repository ACT. A permission from the repository ACT is not sufficient to protect a group definition because you will expand WriteMetadata access to the repository as your security implementation progresses. △
5. Define each group's default access to the entire repository. On the Users and Permissions tab of the Default ACT, set these access controls:
   - Leave the permissions for PUBLIC as they are (all permissions are denied).
   - Grant Read and ReadMetadata to the SASUSERS group. By default all users who have a metadata identity will have read access to resources.
   - Leave the permissions for the Administrators group as they are (ReadMetadata, WriteMetadata, and Administer are granted).
   - Grant Write, Create, Delete, and WriteMetadata to GroupA and GroupB. By default, members of these groups will be able to make changes to most metadata objects and to the data that those objects represent.

   Note: It is not necessary to give GroupA and GroupB the Read and ReadMetadata permissions. Members of GroupA and GroupB are automatically members of SASUSERS, and SASUSERS has Read and ReadMetadata permissions. △
6. Modify the default access controls for selected resources in accordance with the security goals. The permissions that you set in the previous step are the default permissions for all resources. In this example, you will set additional controls for

LibraryA and LibraryB so that only GroupA can access LibraryA and only GroupB can access LibraryB.

- On the Authorization tab For LibraryA, set the following directly assigned permissions:
  - Deny Read, ReadMetadata, Create, Write, Delete, and WriteMetadata to PUBLIC.

    Note: The directly assigned (no background color) denial to PUBLIC will prevent the Administrators group and GroupB from accessing the data in LibraryA. However, this denial is not displayed in the permissions lists for the Administrators group and GroupB. △

  - Grant Read, ReadMetadata, Write, Create, Delete, WriteMetadata to GroupA.

    Note: For members of GroupA, these directly assigned grants to the user-defined group (GroupA) will override the directly assigned denials to the implicit group (PUBLIC). △

  - Grant ReadMetadata and WriteMetadata to Administrators.

    Note: For members of Administrators, these directly assigned grants to the user-defined group (Administrators) will override the directly assigned denials to the implicit group (PUBLIC). △

- On the Authorization tab For LibraryB, set the following permissions:
  - Deny Read, ReadMetadata, Write, Create, Delete, WriteMetadata to PUBLIC.
  - Grant Read, ReadMetadata, Write, Create, Delete, WriteMetadata to GroupB.
  - Grant ReadMetadata and WriteMetadata to Administrators.

Note: The permissions that you set on LibraryA and LibraryB will be inherited by the tables within each library. △

7 Create the necessary user accounts. Each user will need the following accounts:

- a Windows account that enables the user to get to the metadata server as described in initial authentication

  Note: On Windows platforms, assign user rights to these accounts as described in "Setting System Access Permissions" in the *SAS 9.1 Metadata Server: Setup Guide* at **`support.sas.com/rnd/eai/openmeta/`**. △

- a z/OS account that enables the user to get to the stored process and workspace servers in the MVSAuth authentication domain.

8 Create a user definition for every user whose access needs cannot be met through membership in the PUBLIC group. In this example, you did not give the PUBLIC group any access to the repository, so all users must have a metadata identity in order to access any resources. For each user definition, complete these tasks:

- On the General tab, enter the user's name in the **`Name`** field. The other fields on the General tab are optional.
- On the Groups tab, define the user's group memberships. In this example, all users are automatically members of both PUBLIC and SASUSERS. Add selected users to either GroupA or GroupB.
- On the Logins tab for each user definition, complete these tasks:
  - Add a login so the metadata server can determine the user's metadata identity. As depicted in the previous figure, this login does not have to

include a password or specify an authentication domain. This login functions only as an inbound login. For example, the login that is used to determine Tara's metadata identity consists of only her user ID (WinNT\tara).

- Add another login so the user can access the workspace server and stored process server in the MVSAuth authentication domain. As depicted in the previous figure, this login should include a password and be associated with the MVSAuth authentication domain. This login functions only as an outbound login. For example, the login that enables Tara to access the workspace server and the stored process server consists of her z/OS user ID (tara) and a password. This login is associated with the MVSAuth authentication domain.

Note: Do not give any individual users who are members of GroupA a login for the OracleAuth authentication domain. Members of GroupA will access the third-party database server using the login that you added on the Logins tab for GroupA. △

- Click **OK** to save and close the user definition.

Note: It is not necessary to set any permissions on the Authorization tab of the user definition. By default, only *administrative users*, *unrestricted users*, and the user who is represented by a particular user definition can make changes to that user definition. △

If you want to protect multiple resources (rather than protecting just one library for each user group), you should use an access control template. This enables you to define each identity/permission pattern only once and then apply each pattern to multiple resources. To establish mutually-exclusive security, you would create two ACTs and apply one of the ACTs to each resource that is accessed exclusively by either GroupA or GroupB.

Security Maintenance Activities

This section describes the security maintenance activities that you will perform as your deployment evolves.

Removing Users

Administrative users and *unrestricted users* can use User Manager to delete user definitions from a metadata repository. It is recommended that you log in with the *unrestricted user* account (SAS Administrator) in order to delete a user definition. Detailed instructions for deleting user definitions are provided in the online Help for User Manager.

In addition to removing the user definition from the metadata, you might need to remove any individual accounts that you established for the user in the operating system or with an alternative authentication provider.

Resetting Passwords

You must keep the credentials that are stored in your logins synchronized with your user accounts. For example, if a user's password for an operating system account that provides access to a workspace server changes, the login that contains credentials for that account must be updated to reflect the change.

Each user can reset his or her own passwords using either SAS Management Console or the SAS Personal Login Manager utility.* An *unrestricted user* can use SAS Management Console to reset a password for any user.

Updating Passwords That Are Included in Configuration Files

During installation, some of the user IDs and passwords that you specify for your pre-installation user accounts are written to various files in your configuration directories. For example, on UNIX platforms

- the OLAPServer.sh file in the OLAPServer directory includes the user ID and encoded form of the password for your *trusted user* account
- the OMRConfig.xml file in the ObjectSpawner directory includes the user ID and encoded form of the password for your *trusted user* account
- the ShareServer.sh file in the ShareServer directory includes the user ID and encoded form of the password for your General Servers account.

After you reset the password for an account that is referenced by one or more files in your configuration directories, you must manually update those files with the new password in its encoded form. If the password is included in a login for a user or group definition in the metadata repository, you must also update that login. To perform these tasks, complete the following steps:

1. Use your operating system's search facility to search your configuration directories for files that contain the user ID of the account that you updated. For information about configuration directories, see "Overview of Post-Configuration Tasks" on page 147.

   Note: If no occurrences of the user ID are found, open the configuration.properties file to determine which accounts your deployment has been configured to use. For example, if your configuration.properties file specifies **`OMATRST=MyComputer\\sastrust`** then the **`MyComputer\sastrust`** account is being used as the *trusted user* account for your deployment. The configuration.properties file is generated from the information that you provide during installation. △

2. In each file that is found, locate the user ID of the account that you updated. If the file includes a password for that account, complete the following steps to update that password:

   a. Use PROC PWENCODE to encode the new password. For example, to encode a password of "SAStrust1" you would submit the following SAS statements:

   ```
   proc pwencode in='SAStrust1';
   run;
   ```

   The encoded password is written to your SAS log.

   b. In the file, replace the old encoded password with the new encoded password.

3. Stop and then restart the servers that use the files that you have modified. This causes your changes to take effect.

4. Review your configuration instructions to determine whether the password that you changed is stored in a login in the metadata repository. If the password is stored in the metadata, use SAS Management Console to update the password on the Logins tab for the appropriate user definition or group definition.

   Note: You must log in to SAS Management Console as the *unrestricted user* (sasadm) in order to update a password for another user. When you are logged in

* For information about the SAS Personal Login Manager, contact your SAS Support Consultant.

to SAS Management Console as the *unrestricted user*, all rows on every Logins tab contain asterisks in the Password column—even if no password has been specified. △

5 If your deployment includes SAS Web Infrastructure Kit or SAS Information Delivery Portal, perform the following additional steps after you change the password for the SAS Guest User, the SAS Web Administrator, or the SAS Trusted User:

   a Run the configure_wik utility to create new service deployment configurations and new SASStoredProcess.war and Portal.war files.
   b Deploy the Portal.war and SASStoredProcess.war file to the servlet container on your portal Web application's Web server machine.

   For more information about these tasks, see "Understanding the Administration Tools—Configure.wik Utility" in the *SAS Web Infrastructure Kit Administrator's Guide* at **`support.sas.com/rnd/itech/doc9/portal_admin/admintools`**.

Adding Users to a Deployment

To add a user to a deployment, complete the following steps:

1 Create one or more user accounts (if the necessary account are not already in place).

   a Create an operating system, LDAP, or Active Directory account that enables the user to access the metadata server, as described in "Initial Authentication" on page 30.
   b Create any other user accounts that are needed to access servers such as workspace servers, stored process servers, or database servers as described in "Additional Authentication" on page 35.

   Note: On Windows platforms, assign user rights to these accounts as described in "Setting System Access Permissions" in the *SAS Metadata Server: Setup Guide* at **`support.sas.com/rnd/eai/openmeta/`**. △

2 If the new user is an administrator, give the user status as an *administrative user* of the metadata server by adding the user ID of the account that you created in step 1–a to the adminUsers.txt file. Changes that you make to the adminUsers.txt file will take effect after you stop and restart the metadata server.

   Note: For information about the location and syntax of the adminUsers.txt file, see "Configuring Special Users" in the *SAS Metadata Server: Setup Guide* at **`support.sas.com/rnd/eai/openmeta/`**. △

3 If the user's access needs cannot be met through membership in PUBLIC, create a metadata identity for the user.

   □ Log in to SAS Management Console by opening a metadata profile with an *administrative user* account.

      Note: Only an *unrestricted user* or an *administrative user* can create user definitions. △

   □ In the navigation panel of SAS Management Console, select User Manager.
   □ Open the New User properties dialog box by selecting the following path from the menu bar:

      Actions ► New ► User

   □ On the General tab, enter the user's name in the **`Name`** field. The other fields on the General tab are optional.

- On the Logins tab, perform these steps:
  - **a** Add the login that the metadata server will use to determine the user's metadata identity during initial authentication. This login must contain the fully qualified form of the user ID for the account that you created in step 1–a.
  - **b** Add other logins as needed to support additional authentication. These logins must contain the credentials for the user accounts that you created in step 1–b. The exact requirements will vary depending on whether you are using shared accounts, whether the applications that you are using cache credentials, and how many authentication domains contain resources that the user will access. For a summary of the requirements, see "User Planning Summary" on page 56.
- On the Groups tab, define the user's group memberships in accordance with your security plan. If the user is an administrator, make the user a member of the Administrators group. Each user can belong to multiple groups.
- Click **OK** to save and close the user definition.

  Note: It is not necessary to set any permissions on the Authorization tab. By default, only *administrative users*, *unrestricted users*, and the user who is represented by a particular user definition can make changes to that user definition. △

Importing User Information

As an alternative to creating user and group definitions in SAS Management Console, you can import user information from an external system into a metadata repository by using the MDUIMPC.SAS and MDUIMPL.SAS autocall macros that SAS provides in the autocall libraries.

You can use the macros to create user and group definitions from information that you extract from sources such as the following:

- Microsoft Active Directory
- UNIX Password Files
- RACF databases
- any other source that is used to store user and group information and can be read by a SAS DATA step.

For more information and examples, see "Registering Users Using SAS AutoCall Macros" in the *SAS Metadata Server: Setup Guide* at **support.sas.com/rnd/eai/openmeta/**.

Controlling Access to Resources

As you add resources to the repository, it is important to understand the initial, default access controls that apply to the resource. The initial controls come from the Users and Permissions tab of the repository ACT and from any access controls that are specified on the resource's parent objects.*

You can review the access controls for a particular resource by locating the resource in SAS Management Console and displaying the properties window for the resource. On the Authorization tab, examine the permissions that are assigned to each identity (user or group) that is listed in the **Names** box.

* For more information, see "Permissions from the Repository ACT" on page 48 and "Inherited Access Controls" on page 43.

When you select an identity in the **Names** box, the permissions list displays all applicable permissions settings except direct access controls that are assigned to a group to which the selected identity belongs. A directly assigned permission is an ACE or ACT that is set directly on the target resource (rather than on a parent object or on the repository ACT). Permissions that are directly assigned to a group are also directly assigned to all members of the group. However, the group's directly assigned permission is not displayed in the permissions lists of the members of the group. This means that *the permissions list that is displayed for a particular identity does not always indicate that identity's effective permissions for the current resource*.

For example, if an identity who has an inherited (gray background) grant of WriteMetadata permission for a particular resource belongs to a group that has a directly assigned (no background color or green background color) denial of the same permission for the same resource, the group's directly assigned denial will override the identity's inherited grant. However, there is no visual indication of the group's directly assigned denial in the permissions list that is displayed when the identity is selected in the **Names** list box.

You can determine whether an inherited or repository ACT permission (gray background color) is indicative of a identity's effective permissions by examining the other permission assignments on the Authorization tab. If a group to which the identity belongs is listed in the **Names** list box and has a directly assigned permission (green background color or no background color), the group's directly assigned permission has precedence that is not reflected in the permissions list for the identity. If the group has a conflicting permission, you can override that permission in the identity's permissions list. Select the identity in the **Names** list box and then click the identity's inherited permission (gray background color) to change it to a directly assigned permission (no background color).

Set additional access controls in accordance with your security goals. You can override the initial permission settings for a resource by using any of the following approaches:

- selecting check boxes on the resource's Authorization tab
- applying an ACT to the resource
- setting permissions on one of the resource's parent objects.

Note: Permissions that you assign to individual users for specific resources can be difficult to manage. To minimize the complexity of maintaining access controls, use more centralized approaches (such as using access control templates and assigning permissions to user groups) whenever possible. △

Managing Authentication for Added Servers

For each server that you register in the repository, complete the following steps:

1. Determine what authentication domain is associated with the new server.
2. Make sure that exactly one login that contains credentials for accessing the new server is available to every user who will access that server.

   Note: For an explanation of the relationships between servers, authentication domains, and logins, see "Authentication Domains" on page 37. △

A Minimal Security Implementation

This section demonstrates how you can use the PUBLIC group to create a very low security environment where users access servers using shared accounts and metadata

objects and computing resources have no additional protections in the metadata authorization layer.

Note: The minimal security implementation that is described in this section is not a recommended approach for a production environment. △

To establish a minimal security environment, use the following approach:

- Create accounts in the operating system or with alternative authentication providers as needed. In a minimal security environment, some or all of these accounts will be shared accounts.
- Do not take any steps to protect the metadata repository. On the Users and Permissions tab of the repository ACT, leave the initial grants of ReadMetadata and WriteMetadata permissions in place. In addition, grant Read, Write, Create, and Delete to PUBLIC. This enables all users to access OLAP data and to use the metadata LIBNAME engine to access SAS data and relational database data.
- Do not set up any administrators. Instead, use the unrestricted user (the SAS Administrator account) that you created during installation to perform all metadata administrative tasks.
- Do not create any user-defined groups. Users will access resources as members of PUBLIC.
- Create user definitions for only those users who log in with applications that require a metadata identity. For example, SAS Enterprise Miner requires users to have a metadata identity. On each user definition, include only the login that the metadata server needs to establish a metadata identity. Do not include a password or specify an authentication domain in these logins.
- Provide access to workspace, stored process, and database servers using shared accounts. Store the credentials for these accounts in the metadata as logins that are owned by the PUBLIC group. Each login should include a password and specify the authentication domain to which it provides access.
  - If you have homogeneous authentication, add a single login to the PUBLIC group definition.
  - If you have mixed authentication, add one login to the PUBLIC group definition for each authentication domain.

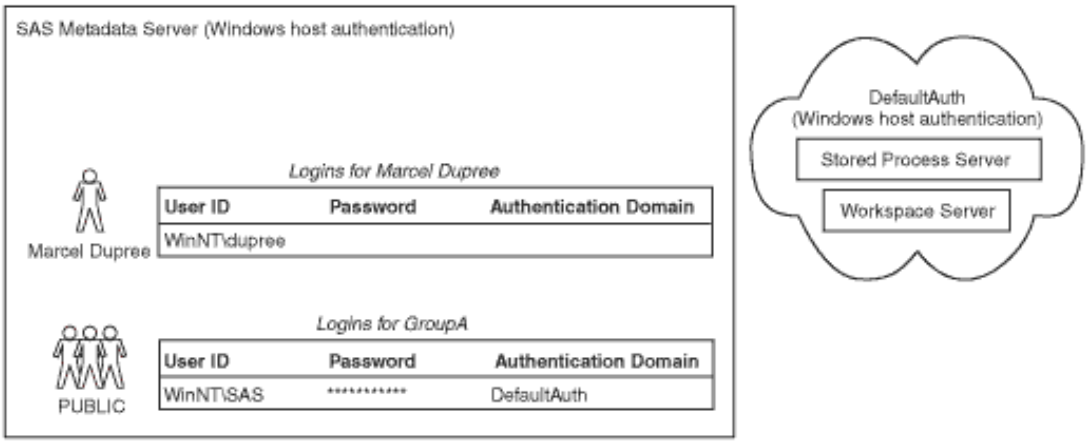
This minimalist approach drastically reduces the number of metadata identities, logins, and access controls that you create and maintain in the metadata. These benefits come at the price of an almost complete loss of individual accountability and an inability to define specific access controls for most users in the metadata authorization layer. You can only define group memberships and specific access controls for users who have a metadata identity.

Example of a Minimal Security Configuration

The following figure depicts a low security implementation for a homogeneous environment that has these characteristics:

- The metadata server, workspace server, and stored process server are running on a single platform and using host authentication.
- The workspace server and stored process server are registered in a single authentication domain that is named DefaultAuth.

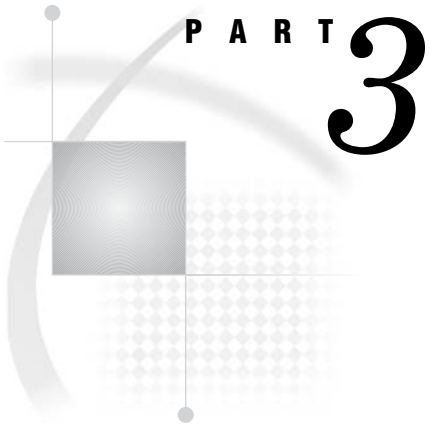
Figure 5.5 Logins for a Minimal Security Implementation



The figure depicts these things:

- Marcel Dupree must have a metadata identity because he uses SAS Enterprise Miner. Marcel's only login enables the metadata server to determine his metadata identity.
- The PUBLIC group has a login for the DefaultAuth authentication domain. This login includes the user ID and password for a Windows user account that all users share when accessing the workspace and stored process servers. As members of the PUBLIC group, all users can use this login to access the workspace and stored process servers during additional authentication.
- If you move the workspace server to a UNIX platform, you must create a shared account on UNIX and give the PUBLIC group an additional login that contains the credentials for that shared account.

In this example, the PUBLIC group's login for DefaultAuth makes credentials for accessing the stored process server and workspace server available to all users. If you also grant all permissions to PUBLIC on the repository ACT, then all users will have metadata layer authorization to access all resources in this very low security environment.



PART 3

Installation and Initial Configuration

CHAPTER

6

Pre-Installation Tasks

Overview of Pre-Installation Tasks

There are a set of tasks that you can perform before your SAS software arrives and you begin the process of installing it. These tasks include

- Setting up a project directory.
  You set up a network-accessible shared directory and place in that directory a planning file created by your SAS representative and your SAS Installation Data

(SID) files. The SAS Software Navigator and the SAS Configuration Wizard both use the planning file as input. For further information, see "Setting Up Your Project Directory" on page 94.

- Filling out, and performing the tasks listed on, one or more pre-installation checklists.
  You can find these checklists in "Pre-Installation Checklists" on page 95. Most of the tasks on these checklists involve setting up required operating environment user accounts and setting up an infrastructure for running Web applications such as SAS Information Delivery Portal. The checklists are largely self documenting; however, you should also read the following sections before filling out the required checklists and performing the tasks that the checklists tell you to perform:
  - "Setting Up Required User Accounts" on page 113
  - "Preparing to Run Web Applications" on page 116
- Determining whether any of the ports normally used by the SAS servers are in use. For a list of the ports that will be used by default, see "Default Ports" on page 119.

Note: This chapter is intended primarily for those of you who are preparing for a project install. However, if you are preparing for a basic install, which is covered in Appendix 2, "Basic Installs," on page 321, you will still need to read most of this material. You will not need to create a project directory, but you will need to use the pre-installation checklists and review the list of default ports. △

Setting Up Your Project Directory

Your project directory is a directory that

- is accessible from all of the machines on which you will install software
- will hold your planning file (and related support files) and your SAS Installation Data (SID) files.

You will receive all of the files that belong in this directory from SAS.

Your Planning File

When you and your SAS representative initially plan your intelligence system—what software you will need and what hardware that software will be installed on—you use a SAS planning tool (a Web application) to record your decisions. Subsequently, SAS will send you an email message containing either a set of files or a ZIP file that contains a set of files. One of these files will be a planning file, **`plan.xml`**, that contains information about which products should be installed and configured on each host in your system. This may be a custom planning file, created specifically for your site, or a standard SAS planning files. The email or ZIP file will also contain ancillary files, such as pre-installation checklists for different platforms.

Copy this set of files to your project directory. The **`plan.xml`** file will serve as input to both the SAS Software Navigator and the SAS Configuration wizard.

Your SID Files

Later, you will receive one or more Software Order Emails, which will contain a SID file. This file contains information about the SAS products you have licensed. The emails will instruct you to store these files in a particular location. Place a *copy* of each SID file in the project directory.

Pre-Installation Checklists

This section contains three pre-installation checklists: one for Windows systems, one for UNIX systems, and one for z/OS systems. You will need to use one checklist for the platform on which you will be installing your SAS servers and one for the platform on which you will be building your Web-application infrastructure (if you will be running any SAS Web applications). Of course, you might be installing your SAS servers and your Web-application infrastructure on the same platform. In that case, you will only need one checklist.

Before you begin to install any software (this is, before you go on to the next chapter), you will need to record information in one or more checklists and to perform a set of tasks described in those checklists. However, before you do that, you should

- read the appropriate checklists to see what information you will need to record and what tasks you will need to perform
- read the section "Setting Up Required User Accounts" on page 113 for additional information about setting up the user accounts required by the SAS Configuration Wizard
- read the section "Preparing to Run Web Applications" on page 116 for additional information about building the necessary infrastructure for the SAS Web applications that you will be running (if any).

The following sections contain the actual checklists.

Pre-Installation Checklist for Windows

Before you begin installing your software, you must create certain user accounts required by the SAS Configuration Wizard. *Print this document* and keep track of your progress by filling in the blanks and adding check marks, e.g., [x], in the Done columns below. You will need to enter the exact values you write down on this checklist into the SAS Configuration Wizard and the SAS Management Console. *Failure to accurately complete each step will prevent you from successfully configuring your SAS software correctly*.

For more information about the usage and requirements of these users and groups, see "Setting Up Required User Accounts" on page 113. For more information about platform-specific security options, see the SAS 9.1 Metadata Server: Setup Guide at: **`http://support.sas.com/rnd/eai/openmeta/`**.

When creating these users, you must:

- □ Deselect "User must change password at next logon"
- □ Select "User cannot change password"
- □ Select "Password never expires"

Note: In the Windows user manager, you cannot enter "domain\username" (you enter user name only), but you will need to enter "domain\username" in the SAS Configuration Wizard and SAS Management Console. △

Note: We strongly recommend that you create new user IDs for the users listed below. Although new user IDs are not mandatory, using existing user IDs will not work if the user IDs do not match one for one with the recommended configuration below. Do not collapse roles. This includes group membership requirements.

The new user IDs can conform to your site standards so the names do not have to match these documented user IDs. If you choose to vary from these documented user IDs, please ensure there is a new corresponding user ID for each user ID documented in the checklist. △

1. Create a *SAS Administrator* on your metadata server machine. This user has privileges to manage user accounts in metadata and administer the metadata server. The SAS Administrator has unrestricted access to the metadata and this user ID should be protected accordingly. This account should never be used for applications other than the SAS Management Console.

Table 6.1 SAS Administrator Information

SAS Administrator Information		Done
User Name:	_____ e.g., <domain>\sasadm, where <domain> is the Windows domain qualifier	[___]
Full Name:	_____ e.g., SAS Administrator	[___]
Password:	_____	[___]

2. Create a *SAS General Server* user on your server machine. This account is used by the object spawner to launch stored process servers. This account will need access to any OS resources required by running stored processes. The initial install sets up this single account for load balanced, stored process server usage. Additional server accounts can be created to give different levels of access as required.

Table 6.2 SAS General Server User Information

SAS General Server User Information		Done
User Name:	_____ e.g., <domain>\sassrv, where <domain> is the Windows domain qualifier	[___]
Full Name:	_____ e.g., SAS General Server	[___]
Password:	_____	[___]

3 Create a *SAS Guest* on your metadata server machine. This user is a generic user account and has the lowest level of security privileges (e.g., this account is used by the SAS Web Portal to log users into the public kiosk area).

Table 6.3 SAS Guest Information

SAS Guest Information		Done
User Name:	_____ e.g., <domain>\sasguest, where <domain> is the Windows domain qualifier	[___]
Full Name:	_____ e.g., SAS Guest User	[___]
Password:	_____	[___]

4 Create a *SAS Trusted User* on your metadata server machine. Because the user ID is a trusted ID, SAS servers such as the OLAP server and mid-tier applications can authenticate to the metadata server using this ID to impersonate authenticated clients of the metadata server; that is, the servers can communicate with the metadata server on behalf of the clients. This is a highly privileged account and should be protected accordingly.

Table 6.4 SAS Trusted User Information

SAS Trusted User Information		Done
User Name:	_____ e.g., <domain>\sastrust, where <domain> is the Windows domain qualifier	[___]
Full Name:	_____ e.g., SAS Trusted User	[___]
Password:	_____	[___]

5 Create a *SAS Demo User* on your metadata server machine. This user has permission to demonstrate the SAS software you have installed, verify the configuration, etc.

Table 6.5 SAS Demo User Information

SAS Demo User Information		Done
User Name:	_____ e.g., <domain>\sasdemo, where <domain> is the Windows domain qualifier	[___]
Full Name:	_____ e.g., SAS Demo User	[___]
Password:	_____	[___]

6 *Create group and set permissions.* On Windows, you must grant permissions to the users that can access servers. To simplify ongoing maintenance, we recommend that you create a group and then grant permissions to the group.

Table 6.6 Granting Permissions to Users

Task	Done
Create a new group called *SAS Server Users*.	[___]
Add all the user IDs listed above (steps 1 to 5) to the *SAS Servers Users* group.	[___]
Grant the "*Log on as a batch job*" permission/policy to the *SAS Server Users* group.	[___]
Windows NT and Windows 2000 only: Grant the *Act as part of the operating system* permission/policy to *SAS General Server*.	[___]

Note: The remaining steps are required only if you will be using any of the SAS Web software (such as SAS Information Delivery Portal) in your deployment. △

7 Create a *SAS Web Administrator* on your metadata server machine. You must create an account for a Web administrator if you will be installing any Web applications, such as SAS Information Delivery Portal. This user has permission to administer the SAS Web infrastructure.

Table 6.7 SAS Web Administrator Information

SAS Web Administrator Information		Done
User Name:	_____ e.g., <domain>\saswbadm, where <domain> is the Windows domain qualifier	[___]
Full Name:	_____ e.g., SAS Web Administrator	[___]
Password:	_____	[___]

8 Add the Web Administrator to the SAS Server Users group.

Table 6.8 Adding the SAS Web Administrator to a Group

Task	Done
Add *SAS Web Administrator* user ID (step 7) to the *SAS Server Users* group.	[___]

9 You must install JDK 1.4.1_02 on the machine where you will install the mid-tier Web software. *Your JDK installation directory must be in your PATH environment variable*. You can download it from the SAS Web site at **`http://support.sas.com/sasnavigator91`**. Verify an already installed version of Java by opening a Windows command prompt and entering "java –version".

Table 6.9 JDK 1.4.1_02 Information

JDK 1.4.1_02 Information		Done
JDK Installation Location:	_____ e.g., C:\j2sdk1.4.1_02	[___]
"java -version" Output:	_____ e.g., java version "1.4.1_02"	[___]

10 If you have already installed a Servlet Container, fill in the following details. If you have not already installed a Servlet Container, you can skip this step now and install Tomcat from the SAS Software Navigator. *Note:* If you use Tomcat as your Servlet Container, you must change the default Tomcat installation location to contain NO SPACES.

Table 6.10 Servlet Container Information

Servlet Container Information		Done
Server Container Provider:	_____ e.g., Tomcat	[___]
Servlet Container Version:	_____ e.g., 4.1.18	[___]
Server Container Machine:	_____ e.g., myServer.myCompany.com	[___]
Servlet Container Location:	_____ e.g., C:\Tomcat4.1 *Note: You must change the default Tomcat installation location to contain NO SPACES.*	[___]

11 If you have already installed a WebDAV (Web Based Distributed Authoring and Versioning) server, fill in the following details. If you have not already installed a WebDAV Server, skip this step now and install Apache or Xythos from the SAS Software Navigator. A WebDAV Server is required if you are installing the Information Delivery Portal or the Web Infrastructure Kit.

Table 6.11 WebDAV Server Information

WebDAV Server Information		Done
WebDAV Server Name:	_____ e.g., My WebDAV Server	[___]
WebDAV Server Provider:	_____ e.g., Apache	[___]
WebDAV Server Version:	_____ e.g., 2.0.45	[___]
WebDAV Server Machine:	_____ e.g., myServer.myCompany.com	[___]
WebDAV Server Location:	_____ e.g., C:\ProgramFiles\Apache Group\Apache 2.0	[___]

Pre-installation Checklist for UNIX

Before you begin installing your software, you must create certain user accounts required by the SAS Configuration Wizard. *Print this document* and keep track of your progress by filling in the blanks and adding check marks, e.g., [x], in the Done columns below. You will need to enter the exact values you write down on this checklist into the SAS Configuration Wizard and the SAS Management Console. *Failure to accurately complete each step will prevent you from successfully configuring your SAS software.*

For more information about the usage and requirements of these users, see the section "Setting Up Required User Accounts" on page 113. For more information about platform-specific security options, see the SAS 9.1 Metadata Server: Setup Guide at **`http://support.sas.com/rnd/eai/openmeta/`**.

Note: We strongly recommend that you create new user IDs for the users listed below. Although new user IDs are not mandatory, using existing user IDs will not work if the user IDs do not match one for one with the recommended configuration below. Do not collapse roles. This includes group membership requirements.

The new user IDs can conform to your site standards so the names do not have to match these documented user IDs. If you choose to vary from these documented user IDs, please ensure there is a new corresponding user ID for each user documented in this checklist. △

1 Create a *SAS* group. This group is used to control access to some directories and files in the Configuration Directory.

Table 6.12 SAS Group

Task	Done
Create a new group called *SAS*.	[___]

2 Create a *SAS* user. Make the SAS group this user's primary group. We recommend that you install and configure all SAS software under this user ID. By default, this user will also be the process owner for the servers (OLAP server, Metadata Server, Object Spawner, SAS/CONNECT Spawner, and SAS/SHARE Server) and the owner of the Configuration Directory structure. The Configuration Directory structure is protected such that only this ID has access to most of the directory hierarchy in the structure. For that reason, installing under one user ID and running the servers under a different user ID will prevent successful operation of the servers. This account should be protected to prevent unauthorized access to the Configuration Directory structure.

Table 6.13 SAS User Information

SAS User Information		Done
User ID:	_____ e.g., sas	[___]
Full Name:	_____ e.g., SAS	[___]

SAS User Information		Done
Password:	_____	[___]
Task:	Assign the *SAS group* as the primary group for this user.	[___]

3 Create a *SAS Administrator* on your metadata server machine. This user has privileges to manage user accounts in metadata and to administer the metadata server. The SAS Administrator has unrestricted access to the metadata and this user ID should be protected accordingly. This account should never be used for applications other than the SAS Management Console.

Table 6.14 SAS Administrator Information

SAS Administrator Information		Done
User ID:	_____ e.g., sasadm	[___]
Full Name:	_____ e.g., SAS Administrator	[___]
Password:	_____	[___]

4 Create a *SAS General Server* user on your server machine and make the SAS group its primary group. This account is used by the object spawner to launch stored process servers. This account will need access to any OS resources required by running stored processes. The initial install sets up this single account for load balanced, stored process server usage. Additional server accounts can be created to give different levels of access as required.

Table 6.15 SAS General Server User Information

SAS General Server User Information		Done
User ID:	_____ e.g., sassrv	[___]
Full Name:	_____ e.g., SAS General Server	[___]
Password:	_____	[___]
Task	Assign the *SAS group* as the primary group for this user.	[___]

5 Create a *SAS Guest* on your metadata server machine. This user is a generic user account and has the lowest level of security privileges (e.g., this account is used by the SAS Web Portal to log users into the public kiosk area).

Table 6.16 SAS Guest Information

SAS Guest Information		Done
User ID:	_____ e.g., sasguest	[___]
Full Name:	_____ e.g., SAS Guest User	[___]
Password:	_____	[___]

6 Create a *SAS Trusted User* on your metadata server machine. Because the user ID is a trusted ID, SAS servers such as the OLAP server and mid-tier applications can authenticate to the metadata server using this ID to impersonate authenticated clients on the metadata server; that is, the server can communicate with the metadata server on behalf of the clients. This is a highly privileged account and should be protected accordingly.

Table 6.17 SAS Trusted User Information

SAS Trusted User Information		Done
User ID:	_____ e.g., sastrust	[___]
Full Name:	_____ e.g., SAS Trusted User	[___]
Password:	_____	[___]

7 Create a *SAS Demo User* on your metadata server machine. This user has permission to demonstrate the SAS software you have installed, verify the configuration, etc.

Table 6.18 SAS Demo User Information

SAS Demo User Information		Done
User ID:	_____ e.g., sasdemo	[___]
Full Name:	_____ e.g., SAS Demo User	[___]
Password:	_____	[___]

Note: The remaining steps are required only if you will be using any of the SAS Web software (such as SAS Information Delivery Portal) in your deployment. △

8 Create a *SAS Web Administrator* on your metadata server machine. You must create an account for a Web administrator if you will be installing any Web applications, such as SAS Information Delivery Portal. This user has permission to administer the SAS Web infrastructure.

Table 6.19 SAS Web Administrator Information

SAS Web Administrator Information		Done
User ID:	_____ e.g., saswbadm	[___]
Full Name:	_____ e.g., SAS Web Administrator	[___]
Password:	_____	[___]

9 You must install JDK 1.4.1 on the machine where you will install the mid-tier Web software. *Your JDK installation directory must be in your PATH environment variable*. You can download it from the SAS web site at **`http://support.sas.com/sasnavigator91`**. Verify an already installed version of Java by opening a shell and entering "java –version".

Table 6.20 JDK 1.4.1 Information

JDK 1.4.1 Information		Done
JDK Installation Location:	_____ e.g., /usr/j2sdk1.4.1	[___]
"java -version" Output:	_____ e.g., java version "1.4.1_02"	[___]

10 If you have already installed a Servlet Container, fill in the following details. If you have not already installed a Servlet Container, you can skip this step now and install Tomcat from the SAS Software Navigator. *Note:* If you use Tomcat as your Servlet Container, you must change the default Tomcat installation location to contain NO SPACES.

Table 6.21 Servlet Container Information

Servlet Container Information		Done
Server Container Provider:	_____ e.g., Tomcat	[___]
Servlet Container Version:	_____ e.g., 4.1.18	[___]

Servlet Container Information		Done
Server Container Machine:	_____ e.g., myServer.myCompany.com	[___]
Servlet Container Location:	_____ e.g., /usr/local/Tomcat4.1 *Note: You must change the default Tomcat installation location to contain NO SPACES.*	[___]

11 If you have already installed a WebDAV (Web Based Distributed Authoring and Versioning) Server, fill in the following details. If you have not already installed a WebDAV Server, skip this step now and install Apache or Xythos from the SAS Software Navigator. A WebDAV server is required if you are installing the Information Delivery Portal or the Web Infrastructure Kit.

Table 6.22 WebDAV Server Information

WebDAV Server Information		Done
WebDAV Server Name:	_____ e.g., My WebDAV Server	[___]
WebDAV Server Provider:	_____ e.g., Apache	[___]
WebDAV Server Version:	_____ e.g., 2.0.45	[___]
WebDAV Server Machine:	_____ e.g., myServer.myCompany.com	[___]
WebDAV Server Location:	_____ e.g., /usr/local/apache or /opt/apache	[___]

Pre-installation Checklist for z/OS

Before you begin installing your software, you must create certain user accounts, a security group, a USS directory, and reserve several port numbers required by the SAS Configuration Wizard. *Print this document* and keep track of your progress (both *Installers* and *Systems Programmers*) by filling in the blanks and adding check marks in the Done columns below. You will need to enter the exact values you write down on this checklist into the SAS Configuration Wizard and the SAS Management Console. *Failure to accurately complete each step will prevent you from successfully configuring your SAS software.*

For more information about the usage of these users and groups, see the section "Setting Up Required User Accounts" on page 113. For more information about platform-specific security options, see the SAS 9.1 Metadata Server: Setup Guide at **`http://support.sas.com/rnd/eai/openmeta/`**.

Note: We strongly recommend that you create new user IDs for the users listed below. Although new user IDs are not mandatory, using existing user IDs will not work if the user IDs do not match one for one with the recommended configuration below. Do not collapse roles. This includes group membership requirements.

The new user IDs can conform to your site standards so the names do not have to match these documented user IDs. If you choose to vary from the documented user IDs, please ensure there is a new corresponding user ID for each user ID documented in the checklist. △

1 Create a RACF group named *SAS*. This group is used to control access to directories and files in the Configuration Directory created in the HFS file system. This group must be defined with an OMVS segment and must be set as the default group for the SAS and SAS General Server user IDs.

Table 6.23 RACF Group

Task	Group Name	Installer Done	Systems Programmer Done
Create a RACF group	_____ e.g., *SAS*	[___]	[___]

2 Create a *SAS* user. Make the SAS group this user's default group. We recommend that you install and configure all SAS software under this user ID. By default, this user will also be the Started Task owner for the servers (OLAP Server, Metadata Server, Object Spawner, SAS/CONNECT Spawner, SAS/SHARE Server) and the owner of the Configuration Directory structure. The Configuration Directory structure is protected such that only this ID has access to most of the directory hierarchy in the structure. For that reason, installing under one user ID and running the servers under a different user ID will prevent successful operation of the servers. This account should be protected to prevent unauthorized access to the Configuration Directory structure.

Note: Installer chooses the ID name; Systems Programmer defines user ID. △

Table 6.24 SAS User Information

SAS User Information		Installer Done	Systems Programmer Done
User ID:	_____ e.g., *sas*	[___]	[___]
Full Name:	_____ e.g., SAS	[___]	[___]
Password:	_____	[___]	[___]
Task:	Assign the *SAS group* as the default group for this user.	[___]	[___]

3 Create a *SAS Administrator* on your metadata server machine. This user has privileges to manage user accounts in metadata and to administer the metadata server. The SAS Administrator has unrestricted access to the metadata and this user ID should be protected accordingly. This ID should never be used for applications other than the SAS Management Console.

Note: Installer chooses the ID name; Systems Programmer defines user ID. △

Table 6.25 SAS Administrator Information

SAS Administrator Information		Installer Done	Systems Programmer Done
User ID:	_____ e.g., *sasadm*	[___]	[___]
Full Name:	_____ e.g., SAS Administrator	[___]	[___]
Password:	_____	[___]	[___]

4 Create a *SAS General Server* user on your server machine and make the SAS group its default group. This account is used by the object spawner to launch stored process servers. This account will need access to any OS resources required by running stored processes. The initial install sets up this single account for load balanced, stored process server usage. Additional server accounts can be created to give different levels of access as required.

Note: Installer chooses the ID name; Systems Programmer defines user ID. △

Table 6.26 SAS General Server User Information

SAS General Server User Information		Installer Done	Systems Programmer Done
User ID:	_____ e.g., *sassrv*	[___]	[___]
Full Name:	_____ e.g., SAS General Server	[___]	[___]
Password:	_____	[___]	[___]
Task:	Assign the *SAS group* as the default group for this user.	[___]	[___]

5 Create a *SAS Guest* on your metadata server machine. This user is a generic user account and has the lowest level of security privileges (e.g., this account is used by the SAS Web Portal to log users into the public kiosk area).

Note: Installer chooses the ID name; Systems Programmer defines user ID. △

Table 6.27 SAS Guest Information

SAS Guest Information		Installer Done	Systems Programmer Done
User ID:	_____ e.g., *sasguest*	[___]	[___]
Full Name:	_____ e.g., SAS Guest User	[___]	[___]
Password:	_____	[___]	[___]

6 Create a *SAS Trusted User* on your metadata server machine. Because the user ID is a trusted ID, SAS servers such as the OLAP server and mid-tier applications can authenticate to the metadata server using this ID to impersonate authenticated clients on the metadata server; that is, the servers can communicate with the metadata server on behalf of the clients. This is a highly privileged account and should be protected accordingly.

Note: Installer chooses the ID name; Systems Programmer defines user ID. △

Table 6.28 SAS Trusted User Information

SAS Trusted User Information		Installer Done	Systems Programmer Done
User ID:	_____ e.g., *sastrust*	[___]	[___]
Full Name:	_____ e.g., SAS Trusted User	[___]	[___]
Password:	_____	[___]	[___]

7 Create a *SAS Demo User* on your metadata server machine. This user has permission to demonstrate the SAS software you have installed, verify the configuration, etc.

Note: Installer chooses the ID name; Systems Programmer defines user ID. △

Table 6.29 SAS Demo User Information

SAS Demo User Information		Installer Done	Systems Programmer Done
User ID:	_____ e.g., *sasdemo*	[___]	[___]
Full Name:	_____ e.g., SAS Demo User	[___]	[___]
Password:	_____	[___]	[___]

8 Create a *SAS Web Administrator* on your metadata server machine. You must create an account for a Web administrator if you will be installing any Web applications, such as SAS Information Delivery Portal. This user has permission to administer the SAS Web infrastructure. NOTE: This user is required only if you will be using any of the SAS Web software (Information Delivery Portal, Web Infrastructure Kit) in your deployment.

Note: Installer chooses the ID name; Systems Programmer defines user ID. △

Table 6.30 SAS Web Administrator Information

SAS Web Administrator Information		Installer Done	Systems Programmer Done
User ID:	_____ e.g., *saswbadm*	[___]	[___]
Full Name:	_____ e.g., SAS Web Administrator	[___]	[___]
Password:	_____	[___]	[___]

9 Select the names and port numbers of the started tasks for each server. Define those started tasks on the system. It is recommended that all ports reserved for these servers to use be registered in your **`/etc/services`** file so that no other processes will attempt to use them.

Note: Installer chooses the names; Systems Programmer reserves ports and defines started tasks. △

Table 6.31 Names and Port Numbers for Started Tasks

Server Name	Started Task Name	Num of Ports Required	Service Name (Optional, for /etc/services)
Metadata Server	_____	1	_____
Object Spawner	_____	3	_____ _____ _____
OLAP Server	_____	1	_____
SAS/CONNECT Spawner	_____	1	_____
SAS/SHARE Server	_____	1	_____

Note: The table below contains additional columns for the this table. △

Table 6.32 Names and Port Numbers for Started Tasks (continued)

Server Name	Reserved Port Number	Installer Done	Systems Programmer Done
Metadata Server	_____ e.g. 8561	[___]	[___]
Object Spawner	_____ e.g. 8571 _____ e.g. 8581 _____ e.g. 8591	[___]	[___]

Server Name	Reserved Port Number	Installer Done	Systems Programmer Done
OLAP Server	_____ e.g. 5451	[___]	[___]
SAS/CONNECT Spawner	_____ e.g. 7551	[___]	[___]
SAS/SHARE Server	_____ e.g. 8551	[___]	[___]

10 Reserve the port numbers for the following spawned server. It is recommended that all ports reserved for this server's use be registered in your **`/etc/services`** file so that no other processes will attempt to use them.

Note: Installer and Systems Programmer work together to select port numbers. Systems Programmer reserves them in **`/etc/services`**. △

Table 6.33 Port Numbers for Spawned Servers

Spawner Server Name	Number of Ports Required	Service Name (Optional: for /etc/services)	Reserved Port Number	Installer Done	Systems Prog. Done
Stored process server	4	_____ _____ _____ _____	_____ e.g. 8601 _____ e.g. 8611 _____ e.g. 8621 _____ e.g. 8631	[___]	[___]

11 Define the name of the configuration directory where these servers will run and where the directory substructure can be defined.

Note: Installer selects directory name; Systems Programmer sets it up. △

Table 6.34 Configuration Directory for Servers

Configuration Directory		Installer Done	Systems Programmer Done
Configuration directory name:	_____	[___]	[___]

12 Define the SAS and SAS/C executable libraries to be program controlled. NOTE: The z/OS system considers the object spawner to be a daemon process. Therefore, if the BPX.DAEMON profile of the RACF Facility class is active and RACF program control is enabled, then the SAS and SAS/C load libraries specified in the STC procedure must be program controlled. However, the user ID under which the object spawner runs does not require RACF READ access to the BPX.DAEMON

profile. You may or may not be able to add these datasets to the program control list prior to dataset creation.

Note: Installer passes names of libraries to System Programmer; Systems Programmer adds to security profile. △

Table 6.35 Executable Libraries to Be Program Controlled

Task	Library Name	Installer Done	Systems Programmer Done
Define SAS executable library to be program controlled	_____	[___]	[___]
Define SAS/C executable library to be program controlled	_____	[___]	[___]

13 After completing this checklist, give it to your Systems Programmer. These tasks must be performed before you can configure the SAS 9.1 Business Intelligence Architecture.

Setting Up Required User Accounts

As indicated in the pre-installation checklists, before installing your software, you must create several user accounts. This section organizes these accounts according to where you should create them and provides a little more information about how the accounts are used. The final subsection explains where you can find information on setting user rights on Windows systems.

Local Accounts on the Metadata Server Host

You should create the following accounts on your metadata server host. The first four accounts are always required, and the last account is required if you will be running any of the SAS Web applications:

- SAS Administrator (**sasadm**)
- SAS Guest (**sasguest**)
- SAS Trusted User (**sastrust**)
- SAS Demo User (**sasdemo**)
- SAS Web Administrator (**saswbadm**)

SAS Administrator

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 on the metadata server (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.

SAS Guest

The SAS Guest User account is used to provide general access to your system's metadata. For example, if you have installed the SAS Web Infrastructure Kit or the SAS Information Delivery Portal, this user configures the Public Kiosk for the portal Web application.

SAS Trusted User

The SAS Trusted User account is used by servers such as the OLAP server and the Xythos WebFile Server to impersonate already authenticated clients on the metadata server. That is, these servers authenticate clients; then, if a client needs to interact with the metadata server, the servers 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 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

The SAS Demo User account is also used by the SAS Information Delivery Portal. If you have loaded the initial demo data, this account allows users to test the portal Web application implementation and to learn about its features.

SAS Web Administrator

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. (You might want to set up other administrative accounts as well. For instance, you might want set up a **sasoladm** account for an OLAP content administrator.)

Accounts on the SAS Server Host or Network Accounts

You also need to create a SAS General Server User account (**sassrv**) and might need to create a SAS user (**sas**). If your SAS and mid-tier servers will reside on more than one host, the **sas** account should be a network account. If you might run load balanced stored process servers on multiple hosts, the **sassrv** account should be a network account as well.

SAS General Server User

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.

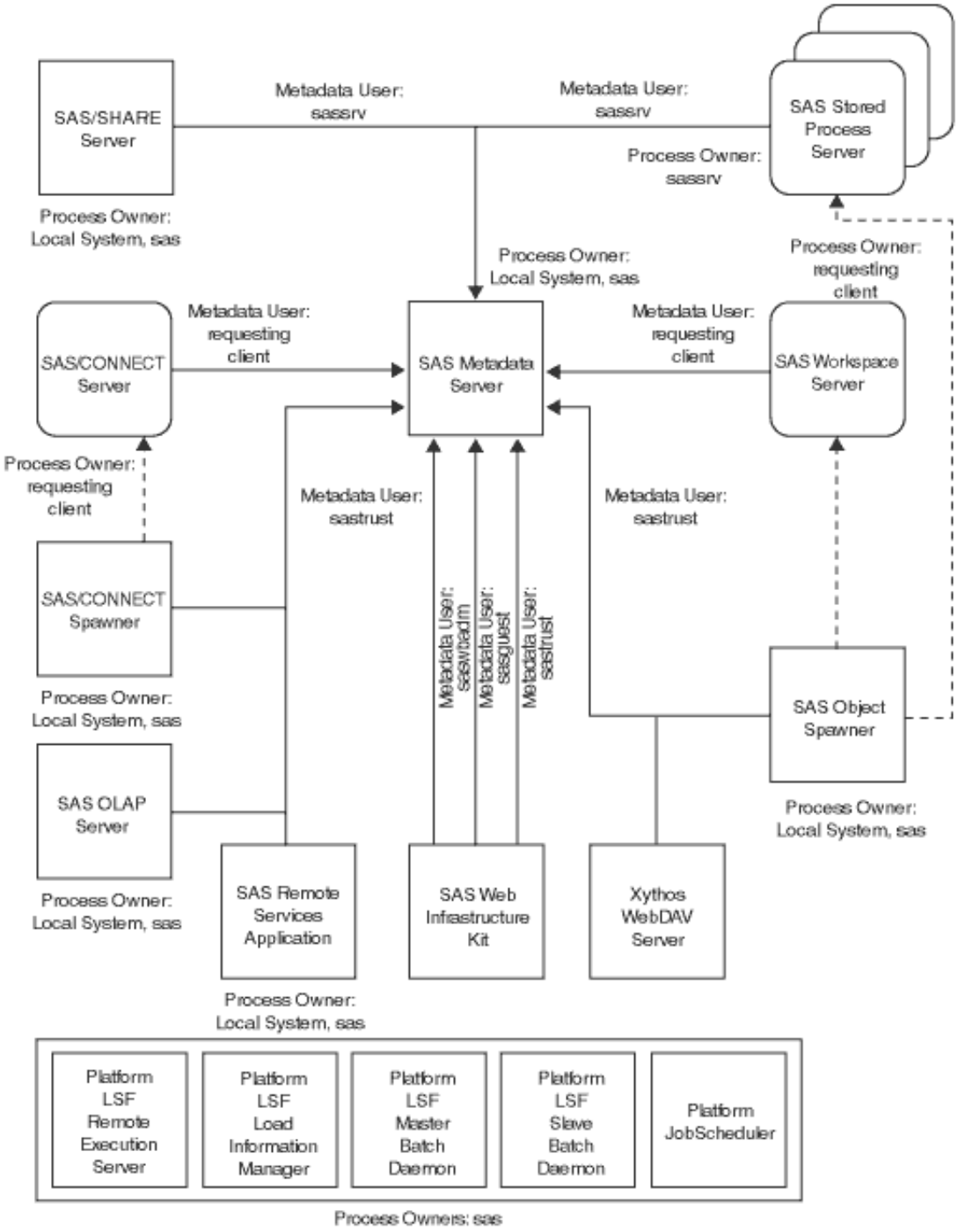
SAS Installer

You should be logged in as the SAS user (**sas**) when you install and configure software on UNIX and z/OS systems. The SAS user will be the owner of the configuration directory and its contents and will be the process owner for items such as the metadata server, the OLAP server, and the object spawner.

How the Accounts Are Used by the SAS Servers

The following figure depicts how the accounts you create during pre-installation are used by the SAS servers in an intelligence system. The figure shows who owns each server process and which account each server uses to communicate with the metadata server.

Figure 6.1 Process Owners



Setting User Rights (Windows Systems Only)

The Windows pre-installation checklist instructs you to assign certain user rights—such as “Log on as a batch job” or “Act as part of the operating system”—to particular users and groups. For information on how to set these user rights, see the

SAS Metadata Server: Setup Guide. In particular, see **support.sas.com/rnd/eai/openmeta/v9/setup/sysperms.html**.

Preparing to Run Web Applications

If you will be running any SAS Web applications at your site—such as SAS Information Delivery Portal—there are some pre-installation tasks you must perform on the host on which you will run these Web applications. These tasks include

- installing the Java 2 SDK
- installing a servlet container or a J2EE application server
- installing a WebDAV server.

See the following sections for information on how to perform these tasks.

Installing the Java 2 SDK

To run SAS Web applications, you must install the Java 2 SDK on your mid-tier host. The servlet container and application server discussed in the next section require the SDK's Java compiler to compile servlets derived from JavaServer Pages. The following table shows which version of the Java 2 SDK you need for each possible mid-tier platform.

Table 6.36 Java 2 SDK

Platform	Supported Version of Java 2 SDK
Windows NT, Windows 2000, and Windows XP	1.4.1_02
Solaris	1.4.1_02
HP-UX	1.4.1.01
AIX	1.4.0

You should download the required version of the SDK from **support.sas.com/sasnavigator91**.

Installing a Servlet Container or a J2EE Application Server

You must also install a servlet container or a J2EE application server on your mid-tier machine if you will be running any of the SAS business-intelligence Web applications. These applications are built using servlets and JavaServer Pages (JSPs), and they execute inside a servlet container.

Servlet Containers Versus J2EE Application Servers

A servlet container provides a subset of the capabilities of a J2EE application server (or J2EE platform). A servlet container provides an execution environment for servlets and for JSPs, which are translated to servlets. A J2EE application server includes a servlet container, but it also contains an Enterprise JavaBean container (for applications that use distributed objects) and a message server, and supports a host of other technologies. The most commonly used servlet container is Apache Tomcat.

Popular J2EE platforms include BEA's WebLogic Platform and IBM's WebSphere application server.

Which option is right for you? Well, SAS Web applications, like SAS Information Delivery Portal, can all run in a servlet container. They do not use technologies such as Enterprise JavaBeans or the Java Message Service. Therefore, a servlet container such as Apache Tomcat might be sufficient. In addition to providing the required execution environment, this product has the advantages that

- it is free
- it is the reference implementation for the servlet and JSP application programming interfaces (APIs). Therefore, Web applications that run in this environment will run correctly.

While you might not need all of the J2EE-related features of an application server, these servers generally provide a number of features that you might require, for example,

- scalability
- load balancing
- security
- persistent sessions.

Ultimately, you will need to base your decision on the number of users you need to support and the number of features your system requires.

Supported Servers

The business-intelligence Web applications have been validated against the servlet container and application server listed in the following table.

Table 6.37 Supported Application Servers

Platform (Version of Java 2 SDK)	Application Server
Windows NT, Windows 2000, Windows XP (J2SDK 1.4.1_02)	Apache Tomcat 4.1.18 BEA WebLogic 8.1
Solaris (J2SDK 1.4.1_02)	Apache Tomcat 4.1.18 BEA WebLogic 8.1
HP/UX (J2SDK 1.4.1.01)	Apache Tomcat 4.1.18 BEA WebLogic 8.1
AIX (J2SDK 1.4.0)	Apache Tomcat 4.1.18

Installing a Servlet Container or Application Server

If you choose to use the Apache Tomcat servlet container, you do not have to install Tomcat before you begin the installation of your SAS software. You will have the opportunity to install the product from a SAS CD when you install software on your mid-tier machine.

If you choose to use the BEA WebLogic Platform, install the application server on your mid-tier machine before you install your SAS software by following the instructions in the documentation that accompanied that product. In addition, be sure to record on your pre-installation checklist the location at which you installed the server.

CAUTION:
When you install your server, make sure that the installation directory contains no spaces. △

Setting Up a WebDAV Server

Any SAS business-intelligence system that workers will use to share documents requires that you install a WebDAV server. This server acts as a repository for these documents.

The sections below explain

- what WebDAV is
- what WebDAV servers can be used with the SAS business-intelligence tools
- how to install and configure your WebDAV server.

What Is WebDAV?

WebDAV stands for "Web-Based Distributed Authoring and Versioning" and is a set of extensions to the Hypertext Transfer Protocol (HTTP). These extensions enable multiple authors to collaborate on documents located on an HTTP server. Whereas HTTP without the WebDAV extensions enables you to read a document identified by a particular Uniform Resource Locator (URL), the WebDAV extensions enable you to write a document to a URL and to edit a document located at a URL. Many times, these are HTML or XML documents, but a WebDAV server can also host word-processing documents, images, and other types of information.

WebDAV supports these features:

- the locking of documents
- the association of properties (or metadata) with documents.

The ability of one author to lock a document ensures that only one author can modify a document at a time. The ability to associate properties with documents makes searching for particular documents much easier.

SAS business-intelligence products use a WebDAV server mainly as a document repository. For example, using the SAS Information Delivery Portal, you can run stored processes and save the results in the WebDAV repository or publish content to the repository.

Supported WebDAV Servers

Your SAS intelligence system can use any of the following WebDAV servers:

- the Apache HTTP Server, with its WebDAV modules enabled
- the Xythos WebFile Server
- the Microsoft Internet Information Server (IIS).

Installing Your WebDAV Server

If you plan to use the Apache HTTP Server for your WebDAV server, you do not need to do anything prior to installing your SAS software. You can install the HTTP server from the SAS Software Navigator when you install software on your mid-tier machine. You can then use the SAS Configuration wizard to configure the server and your environment so that the server functions as a WebDAV server.

Note: If you already have the Apache HTTP Server installed on your mid-tier machine, you should still install the version that is shipped with your SAS software. This is the officially supported version of the product. △

If you plan to use the Xythos WebFile Server, you can install the product from the SAS Software Navigator. However, you will need to configure the server using the documentation you received from Xythos Software Inc. For information on how to register this server in the metadata, see "Defining Your WebDAV Server in the Metadata Repository" on page 154.

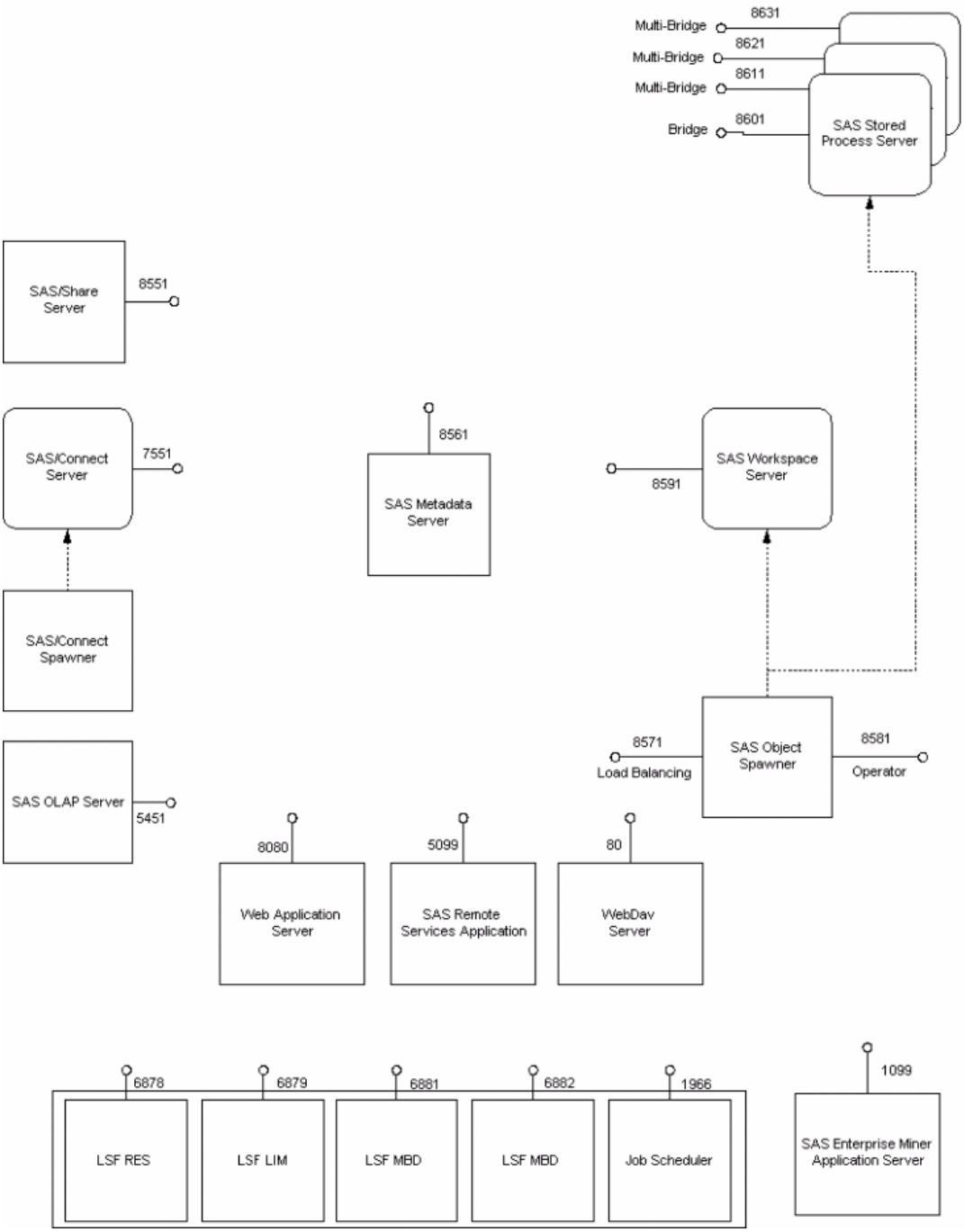
The Microsoft IIS is not shipped with the SAS software. To use it, you will need to obtain it from Microsoft Corporation. You will also have to install and configure product manually, using the product's documentation.

Default Ports

The servers in a SAS intelligence system 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, which you can read more about in Chapter 7, "Installing and Configuring Your Software," on page 123, 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. Review this set of ports to determine whether any of the standards ports are in use. If the standard ports are in use, you will need to decide on an alternate port to use for each port that is already in use.

Figure 6.2 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. This editor is discussed in "Running the Configuration Wizard on Windows and UNIX Systems" on page 135.)

Table 6.38 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
WEBSRV_PORT	8080	Web Application 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**
EM_APPSRV_PORT	1099	SAS Enterprise Miner application server 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.

What's Next

Before you leave this chapter, make sure that you have filled out all of the necessary pre-installation checklists and performed the tasks described on those checklists. Then, proceed to Chapter 7, "Installing and Configuring Your Software," on page 123.

CHAPTER

7

Installing and Configuring Your Software

Overview of Installing and Configuring Your Software

After you have completed the pre-installation tasks and your SAS Installation Kits have arrived, it is time to install your SAS software. This chapter, in conjunction with the online documentation, will lead you step by step through the installation and configuration of your system.

Note: This chapter explains how to perform project installs, installations for which SAS has provided a planning file (**`plan.xml`**). For information on basic installs, which do not require a planning file, see Appendix 2, "Basic Installs," on page 321. △

This chapter covers the following topics:

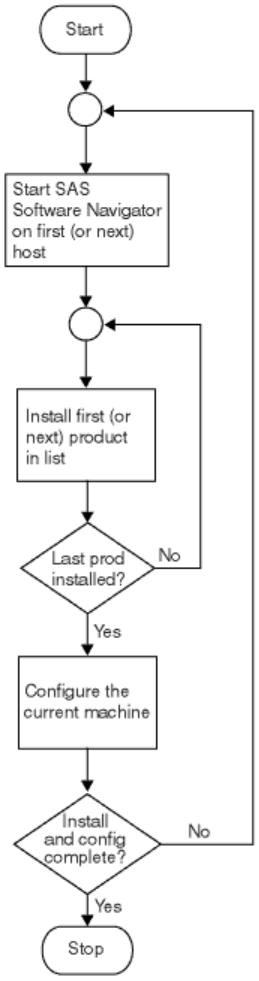
- The SAS Software Navigator.
  This is the tool that you use to install your software.
- The creation of a SAS Software Depot.
  You create this depot by using the SAS Software Navigator to copy the necessary CDs from your Installation Kits to a designated location on your network. This task is highly recommended. If you copy your CDs to the network, you will not have to insert a new CD each time you want to install a new product.
- The installation of your software.
  The SAS Software Navigator also installs each product on a machine. If you need to install three products on a machine, you install all three products before configuring any of the software.

- The configuration of the software on each machine in your system.
  Part of this configuration is performed automatically by the SAS Configuration Wizard. You perform the remainder of the configuration manually, following instructions generated by the wizard. Frequently, you use SAS Management Console to perform these manual configuration steps. The general procedure is to install all of the software on a given machine, and then configure all of the software on that machine at once.

Note: When you installs SAS 9.1 Foundation on a z/OS system and configure the SAS servers, you follow a different procedure. You install the software directly from the SAS media (as explained in *Installation Instructions for SAS 9.1 Foundation for z/OS*), and you configure the software on the machine using a script rather than a configuration wizard. The latter task is covered in "Configuring a Machine" on page 135. △

To summarize, the procedure is to create a SAS Software Depot and then to install and configure the software on one machine at a time, as shown in the following figure.

Figure 7.1 Installation and Configuration Flowchart



Note: You must begin by installing software on the machine on which your SAS Metadata Server will run. This server has to be running when you install and configure software on other machines. Also, you should install server-tier components before mid-tier components and mid-tier components before clients. △

The SAS Software Navigator

You use the SAS Software Navigator both to create a SAS Software Depot and to install software on all of the machines in your intelligence system. This section explains how to start the SAS Software Navigator for the first time, from a CD. After that—if you follow our recommendation—you will be able to start the navigator without a CD.

To start the SAS Software Navigator for the first time, insert the SAS Software Navigator CD into your CD-ROM drive, and then follow the appropriate instructions:

- On Windows, the SAS Software Navigator will automatically run if autorun is enabled. If the SAS Software Navigator does not start automatically, click the **Start** menu from your task bar and select **Run**; then, enter the following command in the **Open** text box:

  ```
  cd-rom:\setup.exe
  ```

  where *cd-rom* is your actual CD-ROM drive.

  Note: The SAS Software Navigator will not run on a 64-bit Windows machine. △

- On UNIX, the CD will be mounted automatically if you are running an automount program, such as **vold** on Solaris. Otherwise, you must manually mount the CD.

  Manual Mounting

  Manually mounting a CD on UNIX requires **root** privilege. Log in as **root** to mount the CD:

  ```
  $ su root
  ```

  The **mount** command on UNIX follows this format:

  ```
  # mount [options] device mount_point
  ```

  where *[options]* are valid mount options for the operating system, *device* is the name of the CD-ROM device, and *mount_point* is the directory used as the mount point for the media. The following commands are sample **mount** commands for each supported UNIX system (that is, UNIX systems where the SAS Software Navigator is supported). The device names listed are only for example; substitute your actual device names accordingly. These instructions assume that your mount point is **/cdrom**; however, you may choose another location.

  AIX

  ```
  # mount -r -v cdrfs /dev/cd0 /cdrom
  ```

  HP-UX and HP-UX for the Itanium Processor Family

  ```
  # mount -F cdfs -o rr,ro /dev/dsk/cd0 /cdrom
  ```

  Linux

  ```
  # mount -r /dev/cd0 /cdrom
  ```

Solaris

```
# mount -r -F hsfs /dev/cd0 /cdrom
```

Tru64 UNIX

```
# mount -rt cdfs -o noversion,rrip /dev/cd0 /cdrom
```

Remote Mounting

If your CD-ROM drive resides on another host and is properly exported, mount the CD using NFS by issuing a command similar to the following:

```
# mount -o ro <remotehost>:/<cd-rom-dir> /cdrom
```

where *<remotehost>* refers to the machine that owns the CD-ROM drive and *<cd-rom-dir>* is the actual mount point for the CD-ROM drive on the server.

Invoking the SAS Software Navigator on UNIX

Once mounted, launch the SAS Software Navigator with the command:

```
$ /cd-rom/setup.sh
```

where */cd-rom* is the mount point of your CD-ROM drive.

□ On OpenVMS Alpha, mount the CD using the command:

```
$ SET PROCESS/PRIVILEGE=BYPASS
$ MOUNT/OVER=ID/MEDIA=CDROM/UNDEF=(STREAM_LF:0) cd-rom
```

where *cd-rom* is the actual device name of your CD-ROM drive.

Once you have mounted the CD, launch the SAS Software Navigator with the command:

```
$ MCR cd-rom:[000000] SETUP_VMS.EXE
```

Note: If you run the SAS Software Navigator using a terminal emulator, it might not function properly. △

Creating a SAS Software Depot

A SAS Software Depot is simply a network copy of some or all of the CDs in your Installation Kits. If you want to lessen the time it takes to create the depot and to conserve disk space, you can copy to the depot only the CDs for the products listed in your planning file. If you are not sure which CDs you need to add to the depot or if you want to be prepared to install products that you might license in the future, you can copy all of your CDs to the depot.

If you create this depot before you begin installing your SAS software, you will not have to insert (and possibly mount) a new CD each time you want to install a new product. Your installation program, the SAS Software Navigator, will automatically find the contents of the necessary CD in the depot—regardless of whether you are installing software on a Windows or UNIX system. All you do is supply the physical path of the directory that holds the software library. We highly recommend that you create this depot.

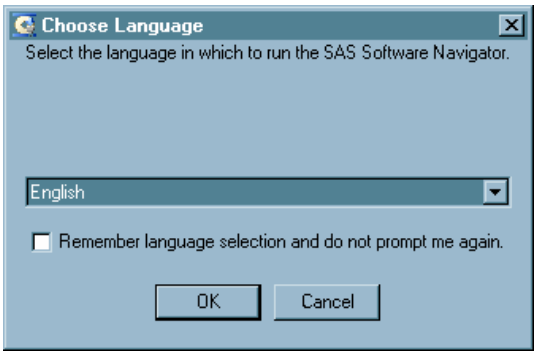
Note: Because z/OS software is distributed on cartridges, SAS 9.1 Foundation for z/OS cannot be a part of your SAS Software Depot. Also, a SAS Software Depot cannot be created on a machine running under the OpenVMS Alpha operating environment. △

To create your depot, follow these instructions:

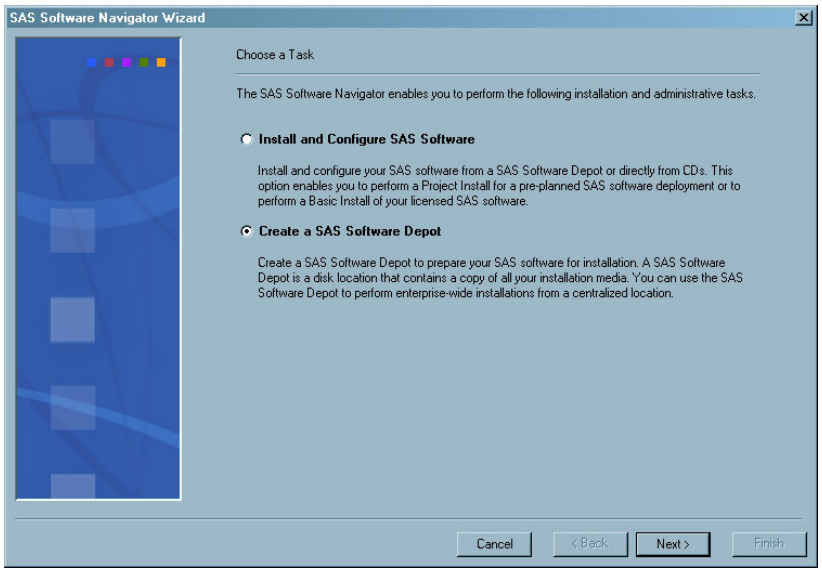
1. Determine the amount of disk space required by the depot. To do this, count the number of CDs that you intend to use to create your SAS Software Depot and multiply by 600 MB. The resulting product is the approximate space required. For example, if you intend to copy 12 CDs into your SAS Software Depot, multiply 12 by 600 MB to get 7200 MB or 7.2 GB. The 600 MB per CD is an approximation, which also accounts for room for swap space as you copy.

   Note: Duplicate CDs will not be copied to the SAS Software Depot. △

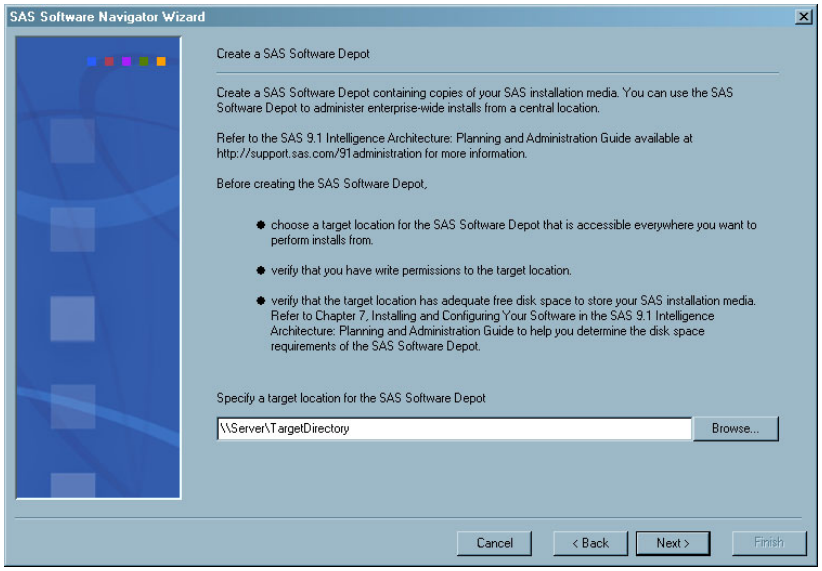
2. Determine the location of the SAS Software Depot. This must be a directory that contains the amount of space that you calculated previously and that is accessible from all of the machines on which you will be installing software.
3. Go to any of the hosts on which you will be installing software (the computer on which you will run the SAS Metadata Server if that is convenient), and find the CD labeled "SAS Software Navigator 9.1" in the Installation Kit for that machine. Insert this CD in the machine's CD drive. Then, follow the instructions in "The SAS Software Navigator" on page 125 to start the navigator. A Choose Language dialog box will appear.



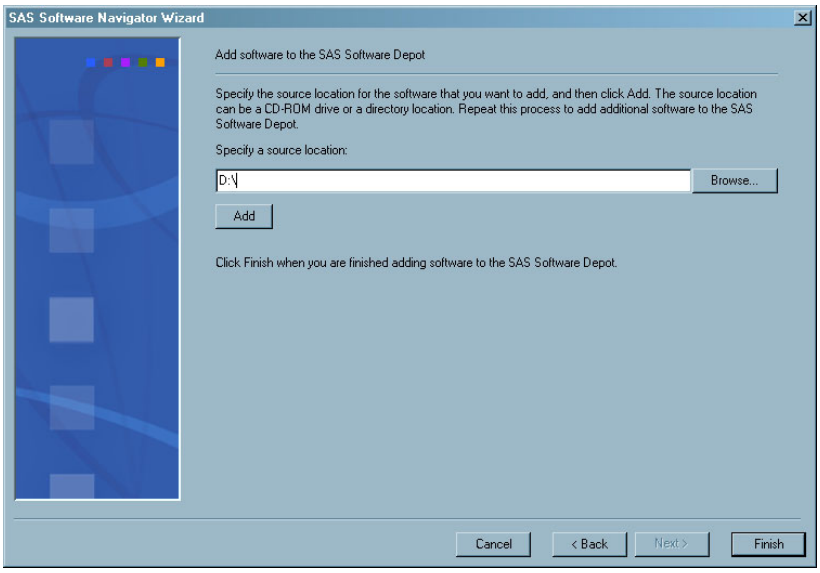
4. In the Choose Language dialog box, from the language drop-down list, select the language that you want the SAS Software Navigator to use. If you want the navigator to use this same language every time that you start the SAS Software Navigator on this machine, select the **`Remember language selection and do not prompt me again`** check box. Then click OK. The SAS Software Navigator Wizard starts.



5 Select the **`Create a SAS Software Depot`** radio button, and click Next. The following window appears.



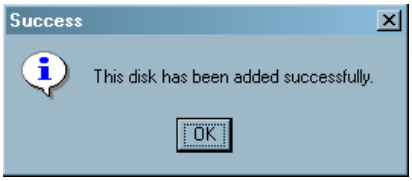
6 In the **`Specify a target location for the SAS Software Depot`** text box, enter the network location where you want to create the SAS Software Depot. You can also browse to the location by using the Browse button. After entering a value in the text box, click Next. The wizard's next screen appears.



7 Enter the location from which the application will be copying data. Most often, this will be your CD drive. Then insert the first or (next CD) to be copied into your drive, and click Add.

Note: For certain media, the SAS Software Navigator might respond with a message indicating that it does not recognize the media you inserted, followed by a prompt for a name for the directory in which you will store this media. You should enter a name for the directory that is intuitively connected to the media name. Later, while you are using the SAS Software Depot to install software, if you click on a link to this media, the depot will tell you it cannot find the media. From the window that the depot displays, browse to the directory that you named, and the SAS Software Navigator will continue as if it had found the media originally. △

8 A progress bar is displayed as the contents of the media are copied to your target directory. When the copy process is complete, you will see a dialog box that indicates the disk has been copied successfully.



Click OK in this dialog box. Then, continue building your SAS Software Depot by repeating steps 7 and 8 for each piece of media you want to add.

Note: You can use this process to copy any piece of media for any platform into the SAS Software Depot. It is not necessary to create different depots for various platforms. △

9 After you have copied the last piece of media, click Finish. The SAS Software Navigator closes. You can restart it from your new SAS Software Depot location. On Windows systems, run the **`setup.exe`** file, and on UNIX systems, run the **`setup.sh`** script.

Ensure that all interested parties are aware of the network location of your SAS Software Depot and of your project directory, which contains your planning file and SAS Installation Data files.

Note: If you would like to add content to an existing SAS Software Depot or create another SAS Software Depot, you must use the original SAS Software Navigator CD. If you start the SAS Software Navigator from a SAS Software Depot, then the opening dialog box, which contains the option to **`Create a SAS Software Depot`** will not be displayed. △

Installing Software on a Machine

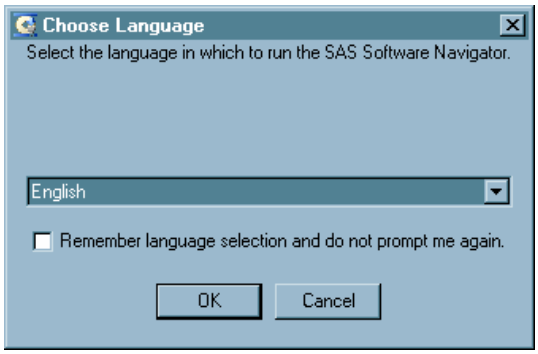
After you have set up your SAS Software Depot—or if you have chosen not to create such a depot—you are ready to begin installing your SAS software. Begin by installing the software that will run on the machine hosting your metadata server. Then, proceed to the next machine listed in your planning file. The procedure for installing software on each machine is the same.

Note: The following instructions explain how to install software on Windows and UNIX systems. For information on how to install SAS 9.1 Foundation for z/OS, consult the *Installation Instructions for SAS 9.1 Foundation for z/OS* in your Installation Kit. △

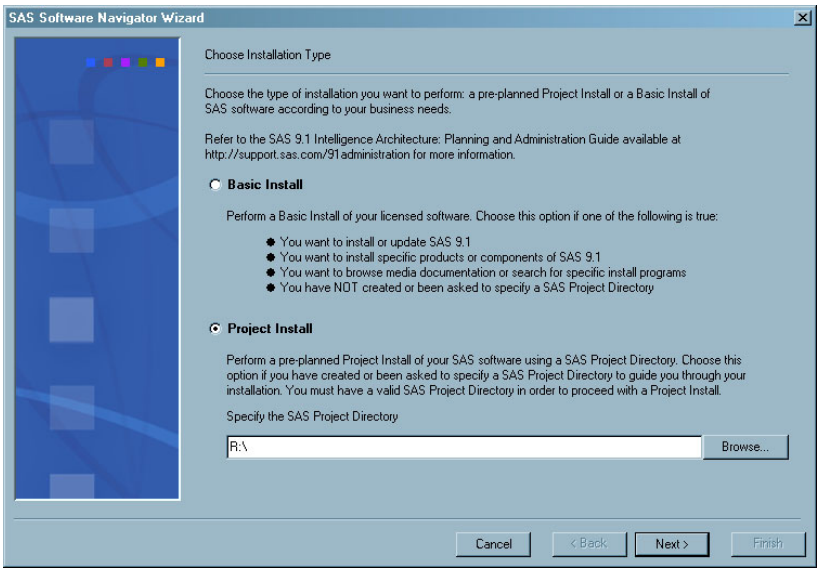
1. Log in to the computer on which you will be installing software. On Windows systems, you can log in as any user who belongs to the Administrators group. On UNIX systems, log in as the SAS user (recommended name **`sas`**), which you defined as one of your pre-installation tasks.

   Note: We recommend that you do not log in as **`root`** to install software on a UNIX system. △

2. Start the SAS Software Navigator, which is the installation program. If you have created a SAS Software Depot (recommended), you can run the navigator from its network location in this depot. Otherwise, you can use the CD in your Installation Kit to run the program, as explained in "The SAS Software Navigator" on page 125. A Choose Language dialog box appears, unless you have already asked not to be prompted for a language again on the current machine.



   From the language drop-down list, select the language that you want the SAS Software Navigator to use. If you want the navigator to use this same language each time you start the program on the current machine, select the **`Remember language selection and do not prompt me again`** check box. Then click OK. The SAS Software Navigator wizard will start. The purpose of this wizard is to gather information about which products are to be installed on the current machine.



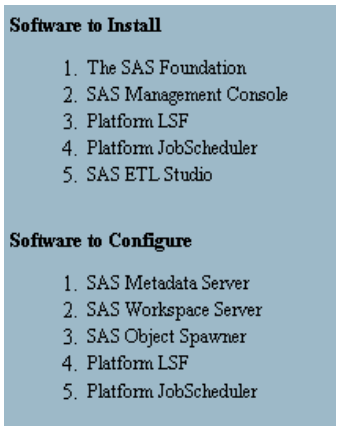
3 Enter information in the SAS Software Navigator Wizard.

a In the first screen, check the **`Project Install`** radio button. Also, enter the location of your project directory in the **`Specify the SAS Project Directory`** text box.

Either type in the complete path to your project directory, or use the available file browser to specify the directory. (This is the directory where you have stored your planning file, **`plan.xml`**, and your SAS Installation Data files as described in "Setting Up Your Project Directory" on page 94.) Then click Next.

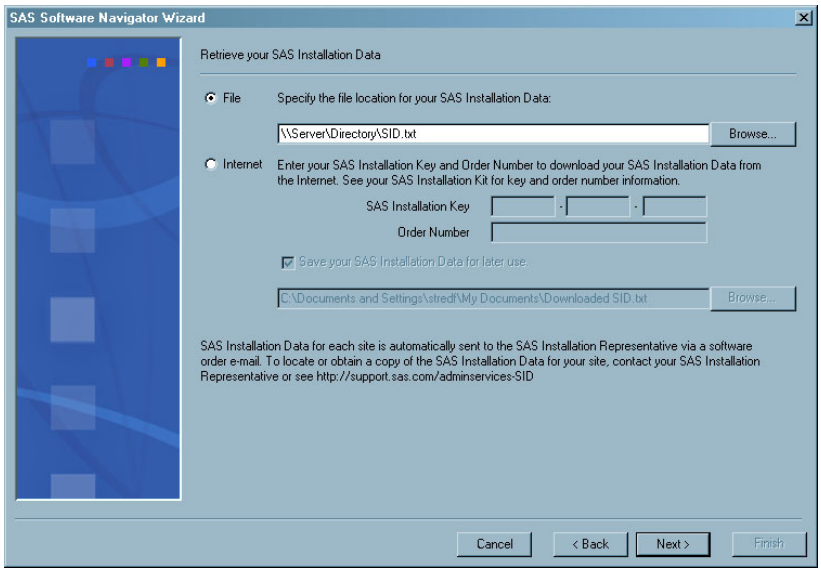
b The next screen in the wizard enables you to confirm that the wizard has detected which machine you are working on—or to specify that machine. The window contains a **`Machine`** list box.

If this box does not contain the correct machine name, select the appropriate name from the list box. (A choice can also be a machine type, such as Desktop, for cases where you want to install the same set of products on multiple machines.) At this point, you will see a list of the products to be installed on the machine and the items to be configured.



Click Next.

c The next window in the wizard asks you to specify the location of the SAS Installation Data for the current machine.



You can specify whether you want to obtain this data from a file (via a pathname) or download it from the Internet. If you have received your SAS Installation Data with your Software Order Email and have saved it on your system, use the **`File`** option; otherwise, select the **`Internet`** option.

- If you select **`File`**, you are prompted to indicate where the SAS Installation Data is located. If you do not know where the file is located, then use the Browse button to browse to the location. When you have located the file, click Next.
- If you select **`Internet`**, you are prompted to enter your SAS Installation Key and Order Number—from your SAS Order Information Letter or Software Order Email—as identifiers for your specific SAS Installation Data. Note that this connection to the Internet uses the SAS secure Web site. You are also able to use the SAS Software Navigator to save your SAS Installation Data to a file with this option. When you have downloaded your SAS Installation Data, click Next.

UNIX Proxy Information

If you choose to obtain the SAS Installation Data from the Internet and your site uses a UNIX proxy server to access the Internet, you need to verify that the system has your proxy information defined properly.

For proxy support of https requests, make sure that your **HTTPS_PROXY** environment variable is defined to point to your proxy server and port. The **HTTPS_PROXY** variable can be defined using either a host address or an IP address. For example, using the Bourne shell, you can define **HTTPS_PROXY** as follows:

```
$ HTTPS_PROXY = "http://proxy.server.com:8080"; export HTTPS_PROXY
```

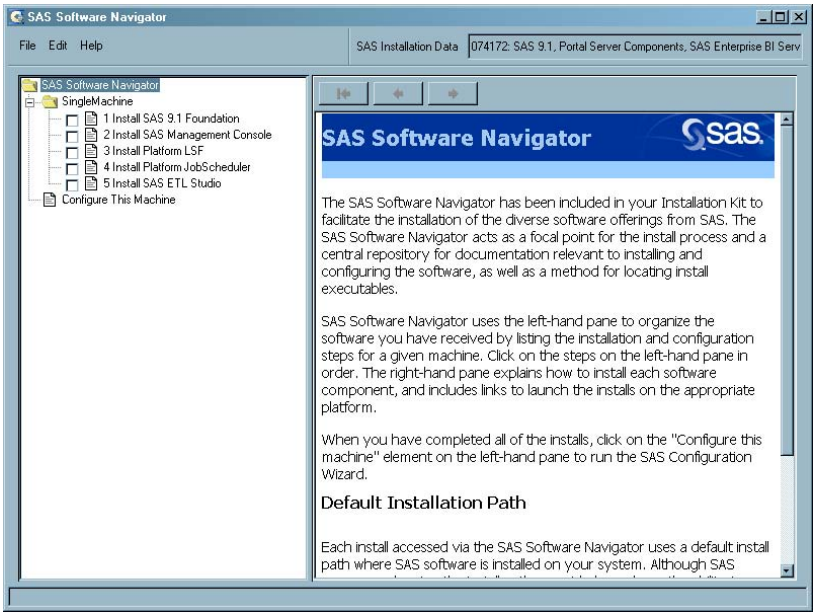
or

```
$ HTTPS_PROXY = "127.0.0.1:8080"; export HTTPS_PROXY
```

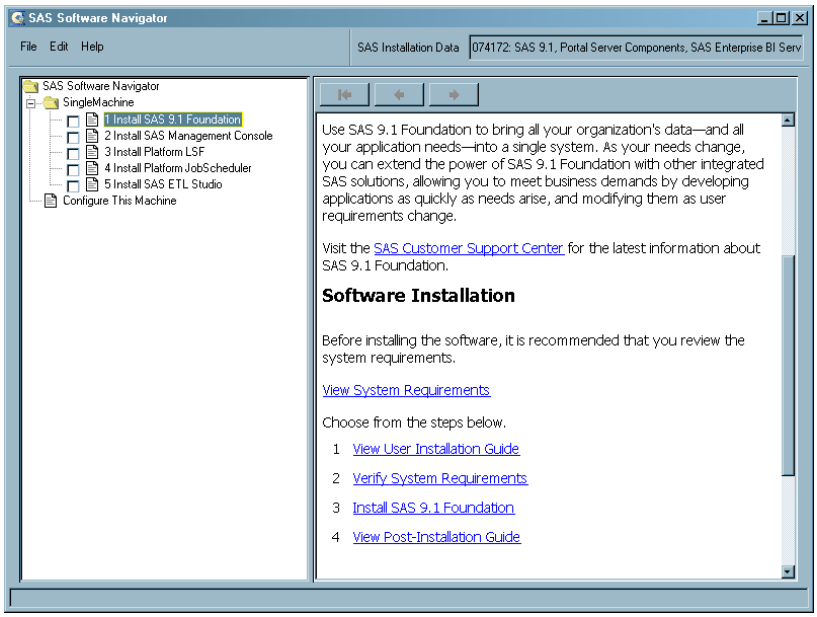
If your proxy server requires basic authentication credentials, you can also define the **HTTPS_USERNAME** and **HTTPS_PASSWORD** environment variables. For example, using the Bourne shell, you can define **HTTPS_USERNAME** and **HTTPS_PASSWORD** as follows:

```
$ HTTPS_USERNAME = "myaccount"; export HTTPS_USERNAME
$ HTTPS_PASSWORD = "mypasswd"; export HTTPS_PASSWORD
```

d The wizard's final screen indicates that the wizard has successfully retrieved your SAS Installation Data. Click [Finish]. You will now see the SAS Software Navigator GUI. In the left pane of the navigator, you will see a list of the products to install followed by the **Configure This Machine** tree node.



4 To begin installing the first product in the list, click the **1 Install** *product-name* link in the left pane. In this case, the first product is **1 Install SAS 9.1 Foundation**. The page in the right pane will be replaced with product-specific information. Somewhere on this page—you might need to scroll down the page—you will see a link to installation instructions for this product and a link to start the installation.



Note: On UNIX systems, if you are not logged in as **sas**, you will see a warning dialog box when you start to install the first product. If you see this dialog box, we recommend that you exit the SAS Software Navigator, create the **sas** account, log in using that account, and then restart the SAS Software Navigator. However, you can continue the installation using your current user ID. △

5 Click the install-software link to start the product's InstallShield wizard. For example, if you click the **Install SAS 9.1 Foundation** link that is shown in the previous display, the SAS Setup Wizard will start.

6 Run the wizard to install the product. When the installation is complete, the appropriate check box in the left pane of the SAS Software Navigator will be selected to indicate that the product is installed.

Note: On Windows system, an installation program could update system information and reboot the operating system. If this happens, restart the SAS Software Navigator, and pick up where you left off. △

Note: You might need to select or deselect a check box manually. For example, if the SAS Software Navigator does not detect that a third-party product is already installed when the navigator starts, you might want to select the box for that product. You can toggle the state of a check box by holding down the Shift key and clicking the left mouse button. △

After you have finished installing the first product on a machine, repeat steps 4 through 6 to install the second product. Continue this process until you have installed all of the products that are listed in the SAS Software Navigator. Then, configure the machine using the instructions in the section "Configuring a Machine" on page 135.

When you have installed all of the software on a machine and configured that machine, proceed to the next machine at your site. Repeat steps 1 through 6 of this procedure for each machine. After installing (and configuring) the software on each machine, starting your servers, and starting your applications, you will have a working system.

Configuring a Machine

After you have installed all of the products on a machine, you must configure the software on that machine. On Windows and UNIX systems, you run the SAS Configuration Wizard to perform some automated configuration tasks and to produce an HTML document that explains which manual configuration steps you also need to perform. On z/OS systems, you run a script that performs the same tasks that the SAS Configuration Wizard performs on the other platforms. For more information on these subjects, see the following sections.

Running the Configuration Wizard on Windows and UNIX Systems

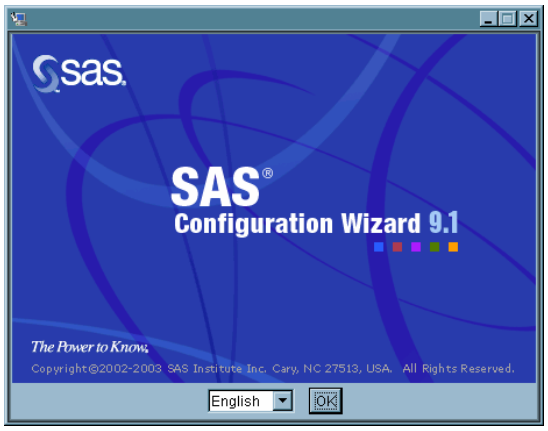
The information that the SAS Configuration Wizard prompts you for will vary depending on which software you have installed on the machine that you are configuring. The following steps detail how the process would go if you were configuring a machine on which you had installed SAS 9.1 Foundation software and were configuring the following servers:

- SAS Metadata Server
- SAS Workspace Server
- SAS Stored Process Server
- SAS Object Spawner.

Running the SAS Configuration Wizard on a different machine will be very similar. Keep your pre-installation checklist near by.

To start and run the SAS Configuration Wizard, follow these instructions:

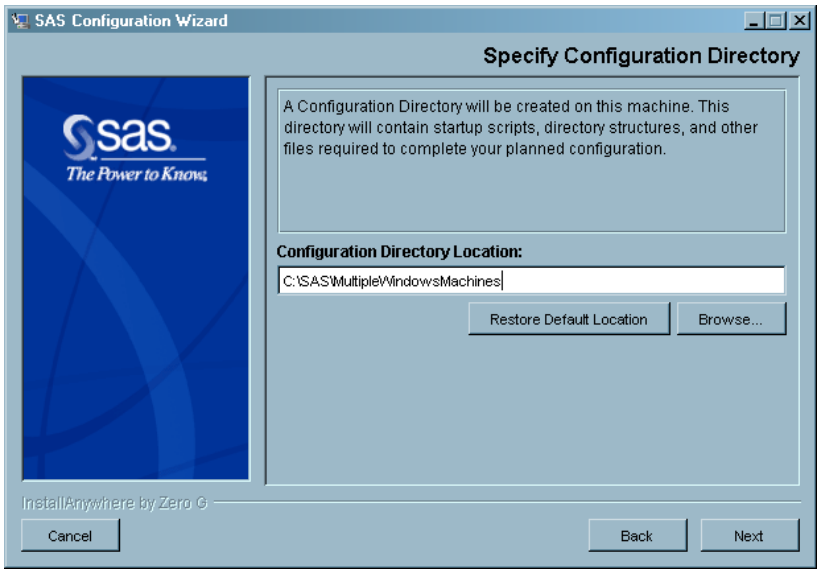
1. Click the **`Configure This Machine`** link in the left pane of the SAS Software Navigator. A new page of documentation appears in the right pane.
2. In the right pane of the SAS Software Navigator, click the **`Configure`** link that is appropriate for your operating system. The SAS Configuration Wizard starts.



3 In the wizard's splash–screen window, select a language from the text box at the bottom of the window, and click OK.

4 In the Introduction window, read the text in the window, and click Next.

5 You will see a specify Configuration Directory window.



Enter the path to a directory in the text box, or accept the default location. This will be the directory in which the SAS Configuration Wizard creates the configuration directory structure described in the following sections:

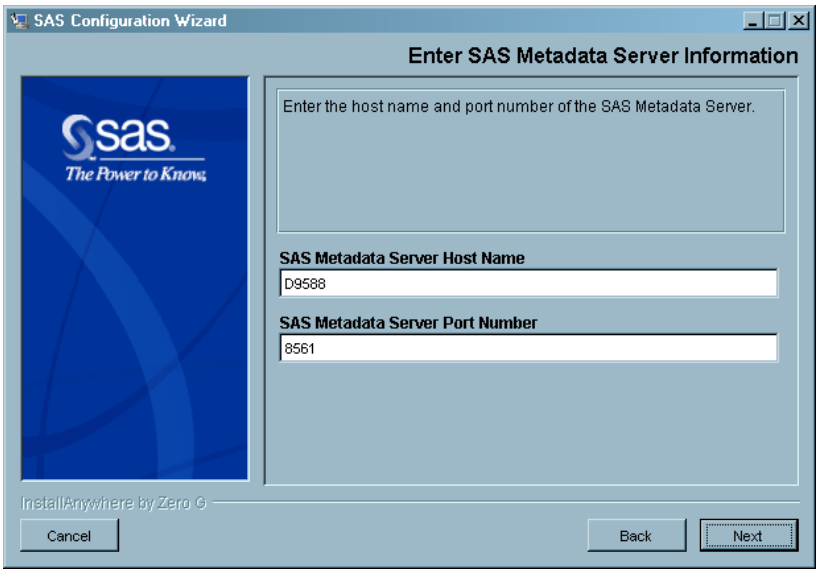
- "Configuration Directory: Server Tier Machines" on page 148
- "Configuration Directory: Mid-Tier Machines" on page 150
- Appendix 1, "Understanding the SAS Configuration Environment," on page 309

Click Next.

6 If you are configuring servers on a Windows machine, the SAS Server Configuration Options window appears.

This window enables you to specify whether you want your SAS servers and spawners to run as Windows services or whether you want to start the servers and spawners using scripts. We strongly recommend that you run the servers as services.

7 The Enter SAS Metadata Server Information window appears.



Provide the following information:

- In the **`SAS Metadata Server Host Name`** text box, enter the name of the machine that you are configuring. This box should have been automatically populated.
- In the **`SAS Metadata Server Port Name`** text box, enter *8561* (unless that port is in use). This box should have been automatically populated. (For a list of the ports that are used in the default configuration, see "Default Ports" on page 119.)

Click Next.

8 The SAS Administrator Information window appears.

Enter the following information:

- In the **User ID (domain/userID)** text box, enter the user ID of the SAS Administrator (**sasadm**). You created this user as one of your pre-installation tasks.
- In the **Password** text box, enter the user's password.
- In the **Confirm Password** text box, reenter the user's password.

Click Next.

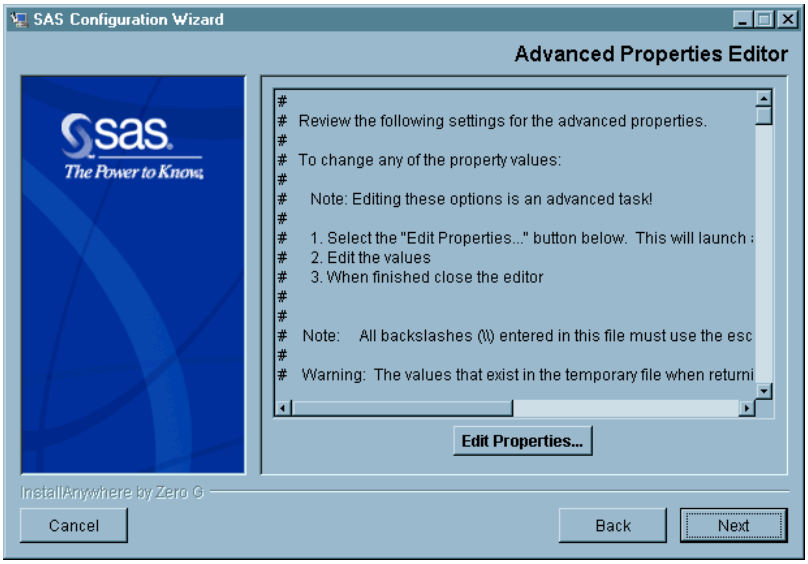
9 In the SAS General Server Information window, enter information about the SAS General Server User (**sassrv**). Click Next.

10 In the SAS Guest Information window, enter information about the SAS Guest account (**sasguest**). Click Next.

11 In the SAS Trusted User Information window, enter information about the SAS Trusted User (**sastrust**). Click Next.

12 In the SAS Demo User Information window, enter information about the SAS Demo User (**sasdemo**). Click Next.

13 The Advanced Properties Editor window appears.



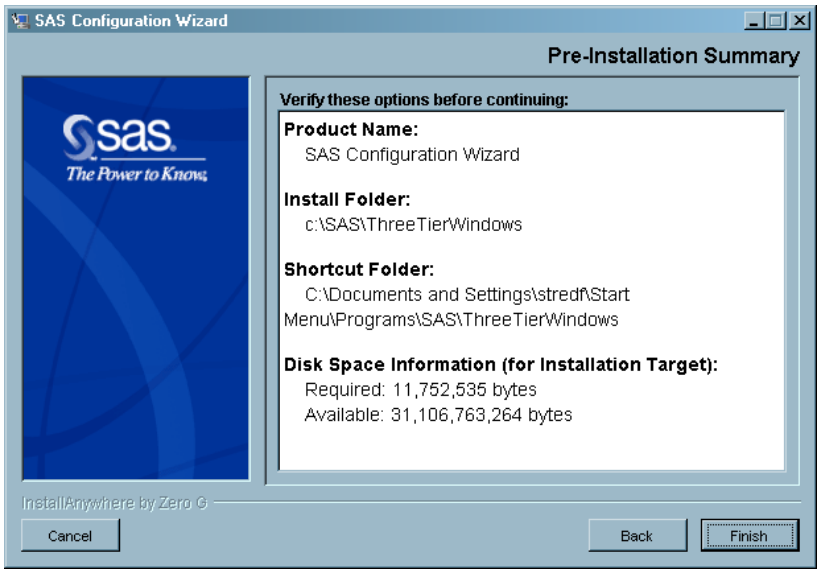
In this window, you can view the settings of a number of properties and change the values of those properties. The properties that contain default port numbers are most likely to change. If these port numbers are in use, you must change the values of the appropriate properties. You can also use the Advanced Properties Editor to specify the following:

- the name of the application server to be configured
- the name of the metadata repository to be created
- the name of the default authentication domain to be used during the configuration.

If you click Edit, you will see the contents of the properties file in a text editor window and can modify the properties file.

Note: This feature is primarily for use by expert installers. △

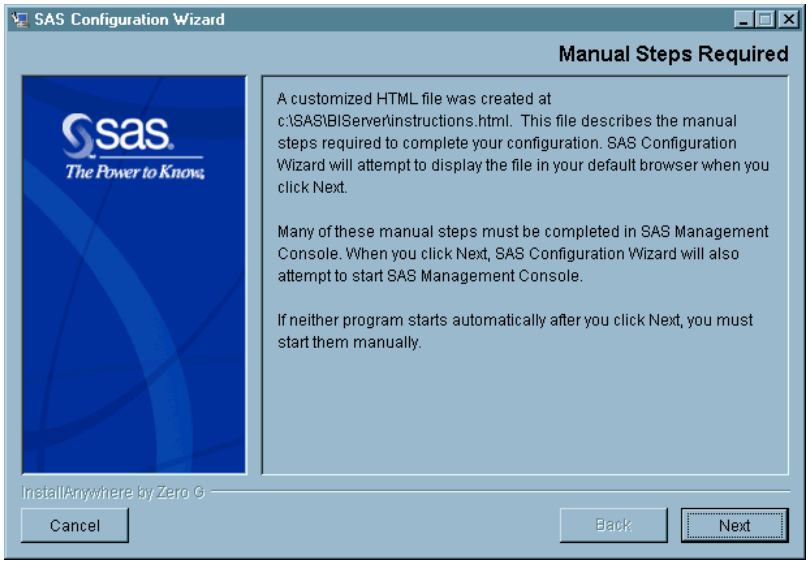
14 The Pre-Installation Summary window appears.



Click [Finish].

15 The Installing SAS Configuration Wizard window appears. While this window is displayed, the wizard is performing automated configuration tasks. For information about these tasks, see "Understanding the State of Your System" on page 148.

16 When the automated configuration is complete, the Manual Steps Required window appears.



Read the instructions about manual configuration. Then, click [Next].

17 The Configuration Complete window appears. Click [Done].

18 At this point, the SAS Configuration Wizard will perform the following tasks, if possible. If the SAS Configuration Wizard does not perform these tasks, you should perform these tasks manually. (Performing the first task below will supply you with the instructions for performing the other tasks.)

- Start a Web browser and display the set of instructions (**`instructions.html`**) that were generated by the SAS Configuration Wizard. These instructions

explain which manual configuration steps you need to take to finish setting up the current machine.

- □ Start the metadata server. (This is appropriate, of course, only if you have set up a metadata server on a machine.)
- □ Start SAS Management Console. You will need this application to perform many of the configuration tasks discussed in "Performing Manual Configuration Steps" on page 143.
- □ Create a metadata profile. This profile enables the user of SAS Management Console to connect to the metadata server.

See "Performing Manual Configuration Steps" on page 143 for information on how to perform the remaining configuration steps.

Configuring SAS Servers on z/OS Systems

Before you configure the SAS software on a z/OS system, you must have

1. filled out the pre-installation checklist shown in "Pre-installation Checklist for z/OS" on page 106.
2. submitted the pre-installation checklist to the data center staff so that they could perform the tasks described in the checklist
3. installed SAS 9.1 Foundation for z/OS by following the instructions in the *Installation Instructions for SAS 9.1 Foundation for z/OS*.

You can then configure the software on the system by performing the following steps:

1. edit and submit the $PREFIX.W0.SRVCNTL(COPYIA) job
2. log in to the USS shell
3. edit the **`configuration.properties`** file
4. run the **`deploy_IA.sh`** script
5. verify the results of running the script
6. follow the instructions in the customized **`instructions.html`**.

For details on how to perform each task, see the following sections.

Edit and Submit $PREFIX.W0.SRVCNTL(COPYIA) Job

The COPYIA job copies a server configuration PAX file to the USS **`/tmp`** directory and extracts the contents of this file into a configuration directory that you specify. This extraction process creates the directory structure and some of the files needed for the SAS Intelligence Architecture server deployment.

To edit and submit this job, perform the following tasks:

1. Starting around line 30, provide values for the following environment variables:
   a. Set **`CONFIG_DIR`** to the configuration directory recorded on your pre-installation checklist. Note that the COPYIA job will attempt to create the directory if it does not exist. Therefore, you must ensure that the user ID under which you are running the COPYIA job is the SAS user ID (**`sas`**) that you created during pre-installation and that this user can create and/or write to the **`CONFIG_DIR`** directory. If you are planning to use the default **`CONFIG_DIR=`***`path`*, you must run the HFSCREAT and HSFMOUNT jobs that were created in the $PREFIX.CNTL dataset during your SAS install, before you run COPYIA.
   b. Set **`LEVEL`** to an application server level, such as **`Lev1`**.
   c. Set **`APPNAME`** to the name of your SAS application server, such as **`SASMain`**.
2. Submit COPYIA.

3 Verify that the job ran successfully by

   a verifying that the COPYIA job's return code was 0
   b viewing the output from the COPYIA1, UNTAR, and SHELLOUT steps for possible problems.

Log In to the USS Shell

Invoke the UNIX System Services shell (or **rlogin** to your z/OS host).

Note: You must be logged in as the SAS user (**sas**), which you specified on your pre-installation checklist. △

Edit the configuration.properties File

Edit the file **configuration.properties** to add the values required by the **deploy_IA.sh** script:

1 Change directories (using the **cd** command) to the directory **CONFIG_DIR/Utilities/zOS_config**. (**CONFIG_DIR** is the environment variable that you set in step 1.)
2 Edit the **configuration.properties** file by entering the appropriate values from your pre-installation checklist.

Run the deploy_IA.sh Script

Run the script **deploy_IA.sh** to configure your SAS servers and spawners:

1 Change directories (using the **cd** command) to the directory **CONFIG_DIR/Utilities/zOS_config**. (**CONFIG_DIR** is the environment variable that you set in step 1.)
2 Run the script using the following command:

```
./deploy_IA.sh -properties configuration.properties
```

3 Review the output for potential errors.

Verify the Results

Check the &PREFIX.W0.SRV* datasets for customized server files (&PREFIX is the high-level qualifier to which SAS was installed):

- &PREFIX.W0.SRVCFG: SAS configuration files
- &PREFIX.W0.SRVCLIST: SAS CLISTs
- &PREFIX.W0.SRVENV: SAS TKMVSENV files
- &PREFIX.W0.SRVPARM: SAS Object Spawner parameter files
- &PREFIX.W0.SRVPROC: SAS started procedure JCL
- &PREFIX.W0.SRVREXX: SAS REXX execs

Follow the Instructions in instructions.html

The **deploy_IA.sh** script produces the **instructions.html** file, which explains the manual configuration steps that you must still perform. For information on how to perform this part of the configuration, see "Performing Manual Configuration Steps" on page 143.

Performing Manual Configuration Steps

When you run the SAS Configuration Wizard on Windows and UNIX systems or run the **`deploy_IA.sh`** script on z/OS systems, after performing as much automated configuration as possible, the wizard or script creates an HTML document called **`instructions.html`**. This document explains the steps that you must perform manually to complete the configuration of the current machine. Many of these steps require that you use SAS Management Console.

On Windows and UNIX systems, the SAS Configuration Wizard attempts to start SAS Management Console and to display **`instructions.html`** in a browser before the SAS Configuration Wizard exits. If the SAS Configuration Wizard is unsuccessful, you must start these applications yourself.

If you are configuring SAS servers on a z/OS system, the situation is a little different. SAS Management Console does not run on z/OS, so you must move your **`instructions.html`** file to a Windows or UNIX system from which you can communicate with your z/OS system. On that system, perform these steps:

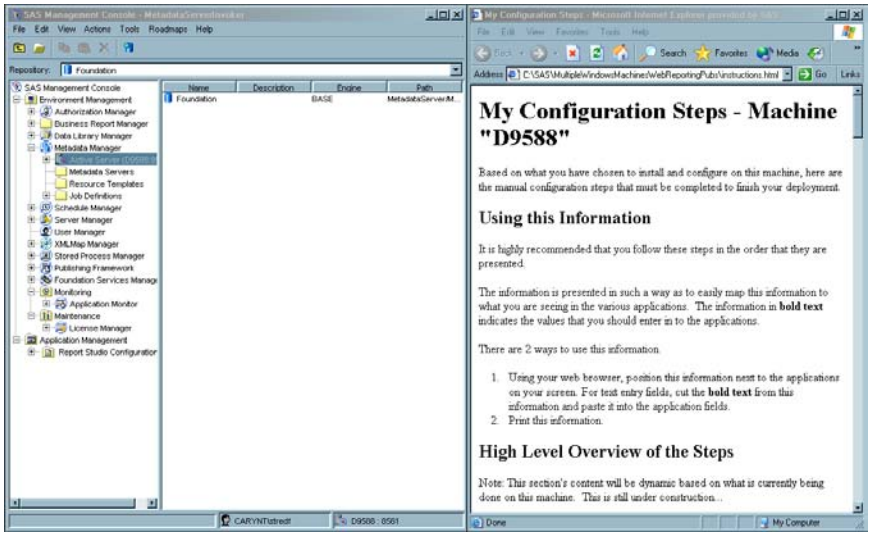
1. Install SAS Management Console. (For information on how to install a product without having a planning file, see Appendix 2, "Basic Installs," on page 321.)
2. Start SAS Management Console.
3. Use a Web browser to display **`instructions.html`**.

From this point on—on all systems—follow the HTML instructions precisely.

Note: Place the SAS Management Console and Web browser windows side by side. △

In this way, when you are prompted for information in SAS Management Console, you can copy the correct response from the HTML instructions and paste the response into SAS Management Console.

Display 7.1 SAS Management Console and Web Browser Windows



Note: Information that you might need to cut and paste is shown in bold type in the HTML instructions. △

When you have completed the last task in the HTML instructions, the configuration of the current machine is complete. You can exit SAS Management Console and close the Web browser window that is displaying the configuration instructions.

After configuring the first machine in your setup, you can proceed to install software on the second machine, and so forth. After you have configured the last machine, you have finished the initial configuration of your system.

CAUTION:
At this point, no metadata layer access controls have been set to protect the foundation metadata repository, the default ACT, or the group definitions that you created using SAS Management Console. See "Configuring Security" on page 156 for information about protecting these resources and performing other post-installation security configuration activities. △

Checking Your Metadata for Required Objects

After you have configured your system, certain metadata objects must exist in your metadata repository. This section lists the User and Group objects that you must have 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 7.1 Summary of Metadata Identities

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

Metadata Identities	Logins			Group Membership Information
	User ID*	Password**	Authentication Domain	
Group: Portal Admins***				members: SAS Web Administrator
Group: Portal Demos***				members: SAS Demo 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 in to SAS Management Console as an unrestricted user, you will always see ******** in the password column, even if no password was specified. To view a password in clear text, you must log in to SAS Management Console as the user who own the login.

***You only need this metadata identity if you have a mid-tier.

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 264.

CHAPTER

8

Post-Configuration Tasks

Overview of Post-Configuration Tasks

After you have run the SAS Configuration Wizard and completed any manual steps that it instructed you to perform, your basic system is in place. However, there is some post-configuration work that you should perform before your users begin using the system.

- "Understanding the State of Your System." When the configuration wizard ran, it performed some behind-the-scenes actions that you need to be aware of in order to administer your system effectively. This section explains those actions so that you understand the current state of your system.
- "Tasks That You Might (or Will) Need to Perform" on page 153, explains some tasks that you might—or in some cases, will—need to perform before the users of your system's applications can begin work. These tasks include such things as

deploying Web applications (if they were not deployed automatically), registering data sources, and configuring SAS/ACCESS products. Before you perform these tasks, Web applications might not run, and workers such as ETL specialists will not have access to the data sources that they need.

- "Configuring Security" on page 156, explains how to set up some basic security at your site. The tasks discussed here are not mandatory, but they are necessary if you want to take reasonable precautions to secure your data and metadata. The tasks discussed in this section include controlling access to your foundation metadata repository, preventing unauthorized access to your configuration directory and its subdirectories, and encrypting sensitive data that moves across a network.
- "Ongoing Administration and Maintenance" on page 159, introduces the subject of ongoing maintenance and provides pointers to many of the subsequent chapters in this document. It points you to information on administering the metadata server, optimizing data storage, performing administrative tasks in support of particular applications, and scaling and improving the performance of your system.

Understanding the State of Your System

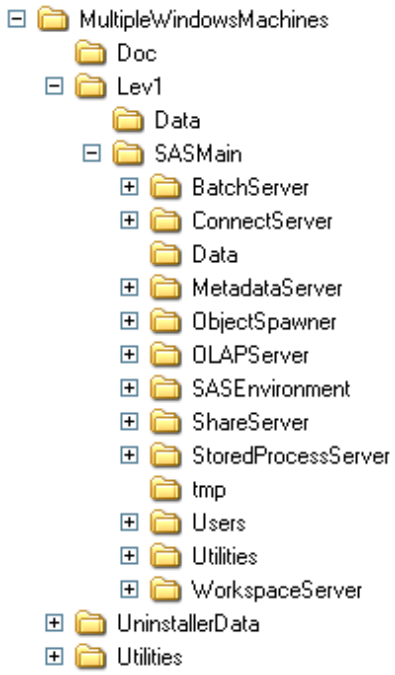
As mentioned previously, when you ran the SAS Configuration Wizard, that wizard did some behind-the-scenes work. Most important, on server tier and mid-tier machines, the wizard created a configuration directory. This directory and its subdirectories contain majors part of your system, such as your metadata repositories and data that is specific to a SAS application server. The configuration directory makes it possible for you to promote and replicate entire configurations.

This section also explains how your workspace and stored process servers were configured.

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.

Display 8.1 Configuration Directory



For a complete discussion of this directory structure, its purpose, and its contents, see Appendix 1, "Understanding the SAS Configuration Environment," on page 309. 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 "Special Users of the Metadata Server" on page 49.

The configuration wizard creates one unrestricted, administrative user: the SAS Administrator (`sasadm`). This user is listed in `adminUsers.txt`, and the ID is preceded by as 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 the section "Configuring Special Users" in the *SAS Metadata Server: Setup Guide* at `support.sas.com/rnd/eai/openmeta/v9/setup/`.)

By default, the `trustedPeers.xml` file contains only comments. However, you can add content to the file to create trusted peer servers, as explained in "Trusted Peer Servers" on page 34.

Server Start-Up Scripts and Logs

Each server or object spawner that you configure on a machine is represented by a directory inside the **SASMain** directory. For example, you might see a **MetadataServer** folder, an **ObjectSpawner** folder, and a **WorkspaceServer** folder. On UNIX systems and Windows systems on which you have chosen to start your servers using scripts, each such directory for a server that you can start directly contains a script called **start***server-type***.***extension*. On UNIX systems, you call these scripts directly to start servers and spawners. On Windows systems, you can call the scripts directly, you can use the Start menu, for example,

Start ► Programs ► SAS ► configuration-directory

► 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.

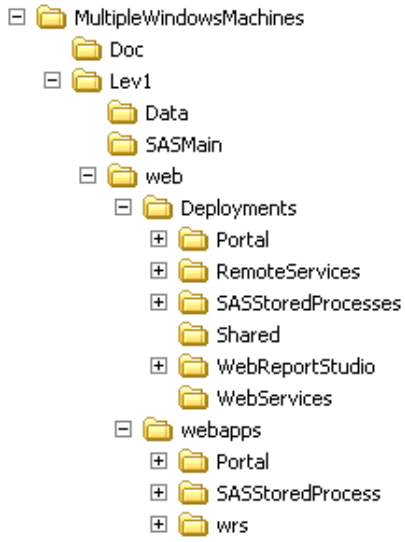
Metadata Repository

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 ETL Studio creates project repositories for ETL developers, as explained in "Setting Up Change Management" on page 209.

Configuration Directory: Mid-Tier Machines

The SAS Configuration Wizard also creates a special directory structure in a configuration directory on mid-tier machines.

Display 8.2 Configuration Directory (Mid-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 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.

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

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 WebLogic Platform for this purpose, you must deploy these files by hand to your server's **webapps** directory, as described in the section "Deploying Web Applications" on page 154.

SAS Application Servers

This section explains how the SAS Configuration Wizard configures workspace and stored process 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. A workspace server can also be configured so that clients use a connection from a pool of previously created connections to workspace server processes, in which case the server is said to be *pooled*. 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, see the *SAS Integration Technologies Administrator's Guide* at **`support.sas.com/rnd/itech/doc9/admin_oma/`**.

In cases where you have a workspace server configured on more than one host, it is also possible to create a load balanced logical workspace server. In this case, 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 ETL 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 260.

You can configure a stored process server to run in either of these modes:

- standard
- 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.

The SAS Configuration Wizard configures your initial stored process server to be load balanced. 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 266.

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

Tasks That You Might (or Will) Need to Perform

Although running the SAS Configuration Wizard brought your system to a point where it is functional, there are a few additional tasks you might need to perform (or will need to perform) before you roll the system out to your users. These tasks are discussed in the following sections.

Starting Your SAS Servers

You started some of your SAS servers when you performed the initial configuration of your system. However, you also need to know how to start those servers later, after you have stopped a server or rebooted a server host. How you start the server depends on the platform on which it is running on and how the server was configured. See the appropriate subsection.

Windows

If your metadata server, object spawner, or OLAP server is running on a Windows machine and you have chosen to run servers as services, your servers will start automatically when you restart your machine. You can also stop and start services using the Services window at

Settings ► Control Panel ► Administrative Tools ► Services

(These services generally run under the account **`Local System`**.)

If your server is running on a Windows machine and you have chosen to start your servers using scripts, follow these directions:

1. Log in as a member of the **`Administrators`** group.
2. Use the Start menu to start the server or object spawner, for example,

   Start ► Programs ► SAS ► configuration-directory

   ► Start SAS Metadata Server

(You can also start a server by executing a BAT file. You will find the BAT file for a particular server in the folder *path-to-config-dir***\Lev1\SASMain***server-type*.)

UNIX

On a UNIX system, you start a server by following these steps:

1. Log in as the SAS User (recommended user name **sas**).
2. Change directories to *path-to-config-dir***/Lev1/SASMain/***server-type*.
3. Execute the start-server script in that directory.

It is also possible to configure your systems so that certain servers or spawners run as daemons. For example, to make the metadata server run as a daemon, you could copy the **MetadataServer.sh** start-up 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 start a server by following these steps:

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

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

   Each server is associated with a different started task.

Deploying Web Applications

This section is pertinent only if you have installed one or more Web applications, such as SAS Information Delivery Portal. If you have installed one more such applications and will use the Tomcat servlet container to execute those applications, the SAS Configuration Wizard will have already deployed these applications to your servlet container. If you are using the WebLogic Platform to execute your Web applications, you must deploy these applications yourself.

To deploy the applications, perform the following tasks:

1. Locate the Web application archive (WAR) files for your applications. They will be located in the directory *configuration-directory***/web/webapps**. Each WAR file has the filename extension **.war**.
2. Copy these WAR files to your application server's **webapps** directory. Consult your BEA Systems, Inc. documentation for the location of this directory.
3. Start or restart your application server. This will cause your Web applications to be deployed.

Defining Your WebDAV Server in the Metadata Repository

If you are using the Apache HTTP Server for your WebDAV server, that server should already be registered in the metadata. If you installed the Xythos WebFile Server (WFS), your server will not be registered. The following paragraph explains under what circumstances you need to register the server and refers you to instructions for performing this registration.

To enable the portal Web application shell to search for files, packages, and reports in the Xythos WFS WebDAV server's repository, the portal Web application does not require a definition for the WebDAV server on the SAS Metadata Server. However, in the following cases, you must ensure that you have a WebDAV server definition (for the Xythos WFS WebDAV server) on the SAS Metadata Server:

- when you run SAS Stored Processes that publish to a Xythos WFS WebDAV server
- when you configure WebDAV-based SAS publication channels or WebDAV package subscribers
- when you run other applications that require a WebDAV server definition.

For information about using SAS Management Console to define WebDAV server on the SAS Metadata Server, see "Administering HTTP Servers and WebDAV" in the *SAS Integration Technologies Administrator's Guide* at **support.sas.com/rnd/itech/doc9/admin_oma/**.

Registering Data Sources

Before your coworkers can use such products as SAS ETL Studio, you must have created metadata objects to represent such items as your database servers, your SAS and database libraries, your database schema, and your data sets and tables. For information about how to register these items, see "Defining Metadata about the Data" on page 167.

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 ETL Studio:

- The preferred solution is to name the format catalog **formats.sas7bcat** and to place the catalog in the directory *path-to-config-dir***\Lev1\SASMain\SASEnvironment\SASFormats**.
- An alternative method of making user-defined formats "visible" is to follow this procedure:
  1. Add a line to the configuration file *path-to-config-dir***\Lev1\SASMain\sasv9.cfg** that points to a configuration file for handling user-defined format catalogs. For example, you might add the line

     ```
     -config path-to-config-dir\Lev1\SASMain\userfmt.cfg
     ```

  2. Then, in the file **userfmt.cfg**, enter a **set** statement and a **fmtsearch** statement. For example,

     ```
     -set fmtlib1 "path-to-config-dir\Lev1\Data\orformat"
     -fmtsearch (work fmtlib1.orionfmt library)
     ```

     This will make the format catalog **orformat.orionfmt** available.

Configuring SAS/ACCESS Products

If you have installed one of the SAS/ACCESS products on a UNIX host in order to access a DBMS, you need to edit a script that was created by the SAS Configuration Wizard so that the SAS servers that call the script can find the necessary shared libraries for your DBMS. This script is called **sas.sh** and is located in the directory *path-to-config-dir***/Lev1/SASMain**.

For information on what you need to add to the script, see Chapter 3, "Post-Installation Configuration for SAS/ACCESS Software" in the *Configuration Guide for SAS 9.1 Foundation for UNIX Environments*, which is included with your Installation Kit. That chapter explains what environment variables you need to set in the script for your product and your operating system.

Configuring Security

This section outlines post-installation security configuration activities including the following:

- protecting the metadata repository and the configuration directories
- encrypting data sent over the network
- modifying the default security configuration.

After you complete these security configuration tasks, you will be ready to implement the rest of your security plan. A step-by-step process for implementing security is provided in "Overview of Implementing Security" on page 71.

Protecting the Metadata Repository and Configuration Directories

At the end of the installation and configuration process, no metadata layer access controls have been set to protect the foundation metadata repository, the default ACT, or the group definitions that you created with SAS Management Console during installation. Unless you intend to have an extremely low-security environment, you should set some initial access controls in the metadata authorization layer to secure the repository and its contents. As your implementation progresses, you can selectively expand access to the repository. See "Protecting the Foundation Repository" on page 72 for more information and instructions on how to quickly set these initial controls.

It is also important that the operating system permissions be set correctly on the directories in your configuration directory. On UNIX systems, the SAS Configuration Wizard will have set these permissions correctly. For information about how these permissions have been set, see "Default Directory Permissions" on page 318.

On Windows systems, you must set these permissions yourself. Assuming that your SAS server and spawners are set up to run as services under the **`Local System`** account, set folder permissions as follows:

- For folders that contain vital information such as repository data sets and encoded passwords, give Full Control to **`SYSTEM`**. Such folders include the **`MetadataServer`**, **`OLAPServer`**, and **`ObjectSpawner`** folders.

  Note: Remove all other users and groups from the list of users and groups for whom security permissions have been defined—instead of denying access to those users and groups. △

- Grant Full Control to **`SYSTEM`** and Read to **`Everyone`** for the following folders:
  - **`BatchServer`**
  - **`SASEnvironment`**
  - **`Users`**
  - **`Utilities`**
  - **`WorkspaceServer`**
- For the **`StoredProcessServer`** folder and the folder **`StoredProcessServer\logs`**, grant Full Control to **`SYSTEM`** and the SAS General System User (**`sassrv`**).

Setting an Encryption Method

By default, only user credentials sent from a client to a server, or from one server to another, are encrypted, and they are encrypted using an algorithm called SAS Proprietary. SAS Proprietary is a fixed encoding algorithm that is provided with Base SAS software and is supported on the Windows, UNIX, and z/OS platforms. It requires no additional SAS product licenses. The SAS Proprietary algorithm is appropriate for use in applications where you want to prevent accidental exposure of information while it is being transmitted over a network.

If you want to prevent the exposure of secret information, you should use the RC2, RC4, DES, or TripleDES algorithm. Using one of these algorithms makes it extremely difficult for anyone to discover the content of messages sent over the network. To use these algorithms, you must license SAS/SECURE software. This software must be installed on your SAS-server hosts, your application-server host, and your client machines. In addition, each host must specify the same algorithm.

Each host must also specify the same level of encryption. Your options are the following:

- **NONE** - Nothing is encrypted.
- **CREDENTIALS** - Log-in credentials are encrypted.
- **EVERYTHING** - All client-server and server-to-server communications are encrypted.

To change the default encryption setup, follow these instructions:

1. Install SAS/SECURE software on your server hosts in its default location.
2. Install SAS/SECURE software on your mid-tier machine.

   Install the client-side version of the product. As part of that installation, two JAR files, **sas.core.jar** and **sas.rutil.jar** will be written to your machine. Copy the JAR files mentioned previously to the **WEB-INF\lib** directory for each of your Web applications. For example, if you have the SAS Information Delivery Portal installed on the machine, you might copy these files to **C:\Tomcat4.1\webapps\Portal\WEB-INF\lib**.

   Note: If your SAS servers and your mid-tier server are running on the same machine, you will need to perform both a server-side and a client-side install of SAS/SECURE software. △

3. Install SAS/SECURE software on your client machines. Install the Java or Windows version of the product, or both, depending on which clients are running of the particular machine.

   Once you have installed SAS/SECURE software on a host where Java clients are running, you must make a copy of the file **sas.rutil.jar** for each client. The following list indicates the location to which should copy the file for each application:

   - For SAS Management Console, copy the file to *drive*:**\Program Files\SAS\SAS Management Console\9.1** (Windows) or *install-location*/**SASManagementConsole/9.1** (UNIX).
   - For SAS ETL Studio, copy the file to *drive*:**\Program Files\SAS\SAS ETL Studio\9.1**.
   - For SAS OLAP Cube Studio, copy the file to *drive*:**\Program Files\SAS\SAS OLAP Cube Studio\9.1**.

4. Edit the start-up commands for your metadata and OLAP servers.

   You need to make two changes to the command that starts the noninteractive SAS session. First, you need to add the option

```
-netencralg "algorithm-identifier"
```

where *algorithm-identifier* is **RC2**, **RC4**, **DES**, or **TripleDES**. Second, you need to add to the **objectserverparms** string the argument

```
cel=encryption-level
```

where *encryption-level* is **NONE**, **CREDENTIALS**, or **EVERYTHING**.

How you make these changes depends on your environment and configuration. If the servers are running as services on a Windows system, you should make the changes in the configuration file *path-to-config-dir**level**application-server**server-type***sasv9**_*server-type*.**cfg**. For example, to change the configuration file for a metadata server, you might edit the file **C:\SAS\ETLServerMin\Lev1\SASMain\MetadataServer\sasv9_MetadataServer.cfg**.

If the servers are running on a UNIX system or are started via scripts on a Windows system, you must edit the start-up scripts described in "Server Start-Up Scripts and Logs" on page 150.

5 Edit the metadata objects representing your SAS servers, such as your workspace, stored process, and OLAP servers.

In SAS Management Console, expand the Server Manager node in the tree, and completely expand the application-server portion of the tree (**SASMain**). Perform the following for each physical server:

a Select the server. This causes the server's connection(s) to appear in the right side of the interface.

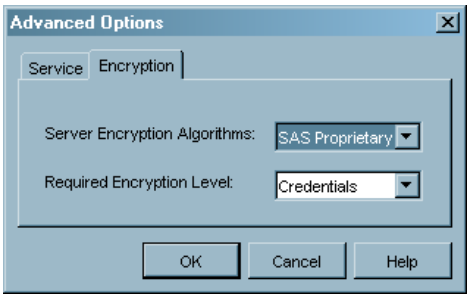
Note: If the server has more than one connection associated with it, you must perform the following steps for each connection. △

b Right-click the connection, and select **Properties** from the pop-up menu that appears. A Properties dialog box displays.

c Select the Options tab.

d Click Advanced Options. An Advanced Options dialog box appears.

e Select the Encryption tab.



f Use the list boxes to select an encryption algorithm and an encryption level. The algorithm and the level should match those specified in your server start-up scripts.

Modifying the Default Security Configuration

Depending on your architecture and security goals, you might need to

- create more authentication domains. After you have run the SAS Configuration Wizard, only one authentication domain exists: DefaultAuth. All of your servers

are associated with this authentication domain. You can create additional authentication domains using SAS Management Console. For information about authentication domains, see "Authentication Domains" on page 37. For instructions on how to create new authentication domains, see the *SAS Management Console User's Guide*.

- create additional administrative users, unrestricted users, or trusted users. Because of the work you did during pre-installation, you have one user who is an unrestricted user (**sasadm**) and one user who is a trusted user (**sastrust**). You can create additional unrestricted and trusted users by editing the files **adminUsers.txt** and **trustedUsers.txt** on your metadata-server host. For more information on how to define these users, see "Server Administrative Privileges" in the *SAS Metadata Server: Setup Guide* at **support.sas.com/rnd/eai/openmeta/v9/setup/**.

  Note: For definitions of these users and explanations of what they can do, see "Special Users of the Metadata Server" on page 49. △

- change the way users of Web applications that are authenticated. Any Web applications that you have installed have been set up to authenticate users using the metadata server's authentication provider, as opposed to one of your servlet container or application server's authentication mechanisms.

  If you want users of the SAS Information Delivery Portal to be authenticated by Apache Tomcat or the WebLogic Platform, see "Setting Up Web Server Authentication" in the *SAS Integration Technologies Web Infrastructure Kit Administrator's Guide* at **support.sas.com/rnd/itech/doc9/portal_admin/**.

- restrict access to the SAS Workspace Server (if you do not have a separate, unrestricted license to use SAS Integration Technologies software on the machine on which the SAS OLAP Server is deployed). A SAS OLAP Server license includes limited permission to use a SAS Workspace Server, which is part of the SAS Integration Technologies product. In this situation, only those SAS OLAP Cube Studio users who are authorized to build and maintain cubes should have access to the SAS Workspace Server.

Ongoing Administration and Maintenance

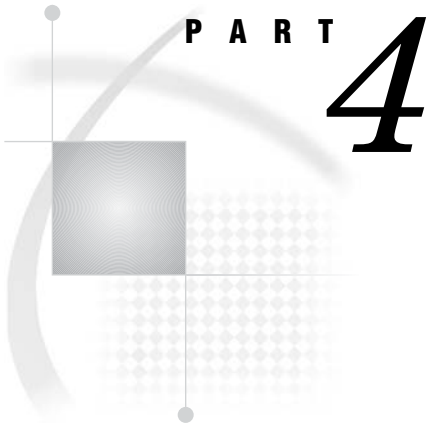
After you have completed the initial configuration of your system, you might need to perform the following tasks:

- You will need to administer the metadata server and your metadata repositories. The tasks in this area include the following:
  - starting and stopping the metadata server
  - creating and deleting metadata repositories
  - invoking repository audit trails
  - backing up the metadata server
  - checking the status of the server and repositories
  - moving or copying a repository.

  See "Administering the Server" in the *SAS Metadata Server: Setup Guide* at **support.sas.com/rnd/eai/openmeta/v9/setup/**.

- You will need to optimize your data storage to facilitate the querying and analysis of that data. This subject is discussed in Chapter 10, "Optimizing Data Storage," on page 181.
- You will need administer the applications being used at your site, such as SAS ETL Studio and the SAS Information Delivery Portal. See Chapter 11, "Administering SAS ETL Studio," on page 205.

- You might want to reconfigure your SAS servers or add new SAS servers to improve the performance of your system. For information on this subject, see Chapter 14, "Configuring SAS Servers for Better Performance," on page 259.
- You might want to perform the metadata-related tasks described in Chapter 15, "Promoting and Replicating Metadata," on page 273.



PART 4

Data Storage Management

CHAPTER

9

Preparing Data for Use

Overview of Preparing Data for Use

In a SAS Intelligence Architecture deployment, you can use one or more of these SAS storage options:

- default SAS storage in the form of SAS tables
- third-party hierarchical and relational database tables such as DB2, Oracle, SQL Server, and NCR Teradata
- parallel storage from the SAS Scalable Performance Data Engine (SPD Engine)

  Note: The SPD Engine is available with Base SAS. The SAS Scalable Performance Data (SPD) Server is an additional, individually licensed storage mechanism. The SPD Server is a stand-alone client/server product that provides much of the functionality of the SPD Engine plus additional features. For more information, see the SPD Server documentation at **`support.sas.com/scalability/spds`**. △
- multidimensional databases (cubes).

All four data sources provide input to reporting applications. The first three sources also are used as input for these data structures:

- cubes, which are created with either SAS ETL Studio or SAS OLAP Cube Studio
- data marts and data warehouses, which are created with SAS ETL Studio. A data mart is a collection of data that is optimized for a specialized set of users who have a finite set of questions and reports. A data warehouse is a collection of data that is extracted from one or more sources for the purpose of querying and analysis.

This chapter provides information that helps you with these tasks:

- understand the data storage options
- define metadata about the data
- prepare for cube loading
- secure the metadata about the data.

Understanding the Data Storage Options

SAS storage options include SAS data tables, parallel storage, multidimensional databases, and third-party hierarchical and relational databases such as DB2 and Oracle. You also can combine any of these storage structures to satisfy unique business requirements.

Default SAS Storage

You can use SAS data sets (tables), the default SAS storage format, 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*.

Third-Party Relational Data Storage

Warehoused data also can be stored in third-party party hierarchical and relational databases such as DB2, Oracle, SQL Server, and NCR Teradata. SAS/ACCESS interfaces provide fast, efficient loading of data to these facilities and allow SAS to work directly from these sources without making a copy.

Several of the SAS/ACCESS engines use an I/O subsystem that enables you to read entire blocks of data instead of reading data just one record at a time. This feature reduces I/O bottlenecks and enables procedures to read data as fast as they can process it. The following SAS/ACCESS engines support this functionality:

- Oracle
- Sybase
- DB2 (UNIX and PC)

- ODBC
- SQL Server
- Teradata.

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. Coupling the threaded SAS procedures with these SAS/ACCESS engines provides even greater gains in performance.

Note: One of the limitations to the amount of scalability that can be seen with the SAS/ACCESS engines is the efficiency of parallelization implemented in the DBMS itself. There are options available on the LIBNAME statement that enable tuning of the threaded implementation within the SAS/ACCESS engines themselves. For more information, see Chapter 10, "Optimizing Data Storage," on page 181. △

For more information about using the SAS/ACCESS interfaces, see *SAS/ACCESS for Relational Databases: Reference*.

Parallel Storage

Shipped with Base SAS, the SAS Scalable Performance Data Engine (SPD Engine) is designed for high-performance data delivery. It enables rapid access to SAS data for intensive processing by the application. The SPD Engine delivers data to applications rapidly because it organizes the data into a streamlined file format that takes advantage of multiple CPUs and I/O channels to perform parallel input/output functions.

The SPD Engine uses 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 the SPD Engine functionality, the real power of the SPD Engine comes from the way that the software structures SAS data. The SPD Engine can read and write partitioned files and, in addition, comprise a new file format. This data structure permits threads, running in parallel, to perform I/O tasks efficiently.

Although it is not intended to replace the default Base SAS engine for most tables that do not span volumes, the SPD Engine is a high-speed alternative for processing very large tables. It reads and writes tables that contain millions of observations, tables that expand beyond the 2-GB size limit imposed by some operating systems, and tables that SAS analytic software and procedures must process faster.

The SPD Engine performance is boosted in these ways:

- support for gigabytes of data
- scalability on symmetric multiprocessing (SMP) machines
- parallel WHERE selections
- parallel loads
- parallel index creation
- parallel I/O data delivery to applications
- implicit sorting on BY statements.

The SPD Engine runs on UNIX, Windows, z/OS (on HFS and zFS file systems only), and OpenVMS Alpha (on ODS-5 file systems only).

Symmetric Multiprocessing

The 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 controllers and multiple disk drives per controller. When the SPD Engine reads a data file, it launches one or more threads for each CPU; these threads then read data in parallel from multiple disk drives, driven by one or more controllers per CPU. The SPD Engine running on an SMP machine provides the capability to read and deliver much more data to an application in a given elapsed time.

For example, in a perfectly tuned system, reading a table with an SMP machine that has 5 CPUs and 10 disk drives could be as much as 5 times faster than I/O on a single-CPU machine. In addition to threaded I/O, an SMP machine enables threading of application processes.

The exact number of CPUs on an SMP machine varies by manufacturer and model. The operating system of the machine is also specialized; it must be capable of scheduling code segments so that they execute in parallel. If the operating system kernel is threaded, performance is further enhanced because it prevents contention between the executing threads.

For more information about using the SPD Engine, see *SAS Scalable Performance Data Engine: Reference* and **`support.sas.com/rnd/scalability/spde`**.

Multidimensional Storage

Multidimensional databases (or cubes) are another storage option, and they are derived from source data such as SAS tables, SPD Engine tables, and SAS/ACCESS database tables by using tools such as the Cube Designer wizard, which is available from SAS ETL Studio and SAS OLAP Cube Studio. Cubes are managed by the SAS OLAP Server, which is a multi-user, scalable, online analytical processing server that can be used to store and access large volumes of data while maintaining system performance.

The SAS OLAP Server uses a SAS engine that organizes data into a streamlined file format, which enables the engine to rapidly deliver data to client applications. The engine also reads and writes partitioned tables, which enables it to use multiple CPUs to perform parallel I/O functions. The multi-threaded model enables the SAS OLAP Server to create and query aggregations in parallel for fastest performance.

Cubes are especially useful when providing business users with multiple views of their data through drill-down capabilities. Queries against the cubes are performed by using the multidimensional expressions (MDX) query language.

Cubes can be accessed by client applications that are connected to the SAS OLAP Server with the following tools:

- the SQL Pass-Through Facility for OLAP, which is designed to process MDX queries within the PROC SQL environment. For information about using the SQL Pass-Through Facility for OLAP, see the *SAS OLAP Server: Administrator's Guide*.
- open access technologies such as OLE DB for OLAP and ADO MD. For more information, see the *SAS Data Providers: ADO/OLE DB Cookbook*.

The planned SAS installation defines and configures a SAS OLAP Server and creates one or more start-up scripts that are appropriate for the operating system. There might be a script that starts the server as a service, as well as a script that can be used to start the server manually. For more information about installation and configuration, see Chapter 7, "Installing and Configuring Your Software," on page 123.

For detailed information about managing multidimensional storage and a SAS OLAP Server, see the *SAS OLAP Server: Administrator's Guide*.

Defining Metadata about the Data

In order for the SAS Intelligence Architecture applications such as SAS ETL Studio to be able to use SAS tables, SPD Engine tables, and SAS/ACCESS databases, you must define metadata about these items:

- database servers (for SAS/ACCESS database tables)
- database schemas (for SAS/ACCESS database tables)
- libraries
- data sources.

Note: Cubes are registered in the metadata with the Cube Designer wizard, and, rather than libraries, cubes are members of OLAP schemas, which are assigned to SAS OLAP Servers. For more information, see "Preparing for Cube Loading" on page 173. △

For information about the SAS Management Console tasks discussed in this section, see the *SAS Management Console: User's Guide*.

Preliminary Tasks

Before you can define the metadata for servers, schemas, and libraries, these tasks must have been completed.

1. A SAS Metadata Server is started.
2. At least one metadata repository is defined.
3. You have ReadMetadata and WriteMetadata permissions to the SAS Metadata Repository in which you want to save the metadata.

   Note: If your environment is change managed, you must be granted CheckInMetadata permission, rather than WriteMetadata permission, in order to register data sources in the repository. △

   Note: No metadata layer permissions to a data source are required in order to register a data source. However, in order to access the data source, you must have the adequate permissions in the data source and operating system authorization layers. △
4. You have a metadata profile for accessing the SAS Metadata Server and the SAS Metadata Repository that you want to use.

In order to define the metadata for data sources with a source designer, these tasks also must have been completed:

1. An object spawner is listening for requests for services from a SAS Workspace Server.
2. A SAS Workspace Server is available for services.

You also can define metadata about data sources with the metadata LIBNAME engine, which does not require a running object spawner or SAS Workspace Server.

During a planned installation, a SAS Metadata Server, an Object Spawner, and a SAS Workspace Server are defined and configured. On Windows machines, the servers and the spawner are usually started as services. For other platforms, the planned installation creates start-up scripts. The installation process also creates a foundation repository.

Note: For information about a planned installation and configuration, see Chapter 7, "Installing and Configuring Your Software," on page 123. △

Defining a Database Server

A database server provides relational database services to a client. Before you can define a database schema and database tables, you must define a database server to match the schema type and the library type. Although the information required for each type of database server is slightly different, the servers all require the same basic information:

- server name
- machine on which the server runs
- location of the data
- credentials for logging into the server.

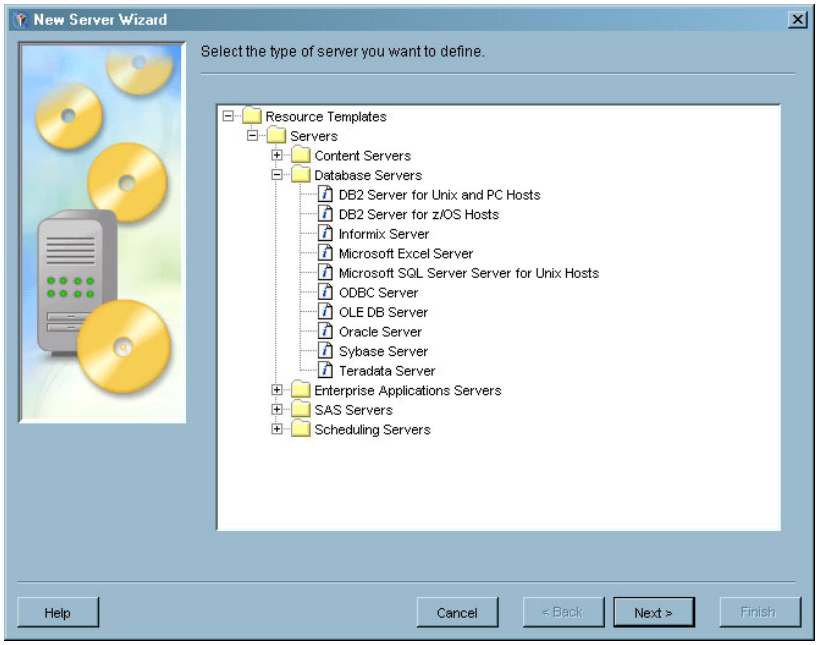
You define the server with the Server Manager plug-in to SAS Management Console. To launch the New Server wizard from SAS Management Console, complete these steps:

1. Use your metadata profile to log in 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 the applicable server type, then click Next to continue. The balance of the information that you must enter depends on the type of server that you select.

For more information about how to define a server, see the *SAS Management Console: User's Guide*.

Defining a Database Schema

A database schema is a pointer to an existing schema, which is a map or model of the structure of a database. Before you can define a database schema, you must have a

database server of the same type. In addition, both the database server and the database schema definitions are required in order to define a database library of the same type.

You define database schemas with the Data Library Manager plug-in to SAS Management Console. The plug-in provides support for a wide variety of schema types through the use of resource templates. A resource template is an XML file that specifies the information required to define a certain type of resource (such as a database schema).

Note: During a planned installation, a foundation metadata repository is created. When the foundation repository is created, all available resource templates are also loaded into the repository. △

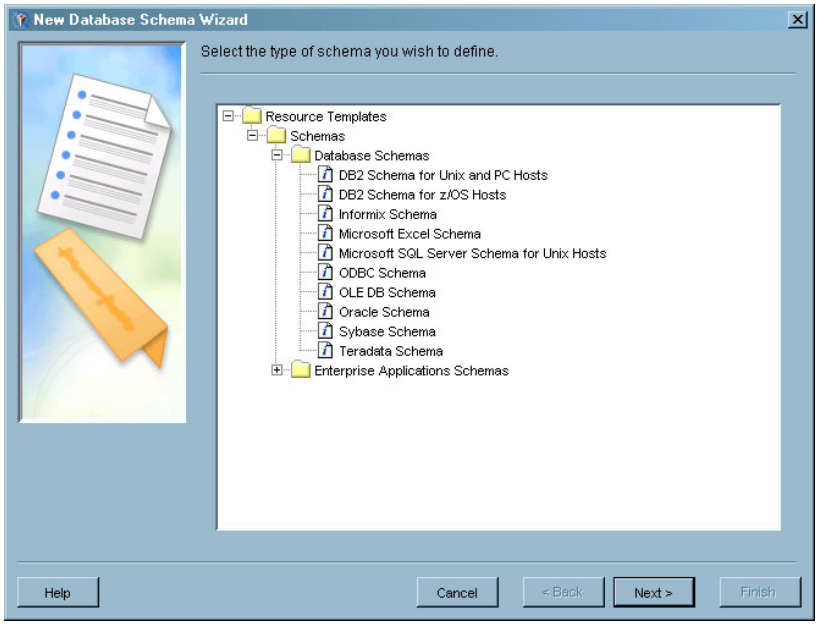
To launch the New Database Schema wizard from SAS Management Console, complete these steps:

1. Use your metadata profile to log in 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 ► Data Library Manager ► Database Schemas

3. To display the first wizard window, select

   Actions ► New Database Schema



4. Select the applicable schema type, then click Next to continue. The balance of the information that you must enter depends on the type of schema that you select.

For more information about how to define a database schema, see the *SAS Management Console: User's Guide*.

Defining a Library

A library is a collection of one or more files that are recognized by SAS and that are referenced and stored as a unit. Each file in the library, such as a SAS table, is a

member of the library. To define libraries in a SAS Metadata Repository, you use the Data Library Manager plug-in to SAS Management Console. After the definitions are stored in the SAS Metadata Repository, they are available for other applications, such as SAS ETL Studio, to use.

Note: In order to define a database library, you also must have defined a database server and a database schema that match the library type. See "Defining a Database Server" on page 168 and "Defining a Database Schema" on page 168. △

Note: In SAS ETL Studio and SAS OLAP Cube Studio, users can launch the New Library wizard from a source designer wizard or the Cube Designer wizard, so that they can define a new library when they are defining a data source or creating a cube. △

You use the Data Library Manager to manage SAS data libraries, libraries that contain data from other applications, and libraries that are used directly by other applications. The plug-in provides support for a wide variety of library types through the use of resource templates. A resource template is an XML file that specifies the information required to define a certain type of resource (such as a library).

Note: During a planned installation, a foundation metadata repository is created. When the foundation repository is created, all available resource templates are also loaded into the repository. △

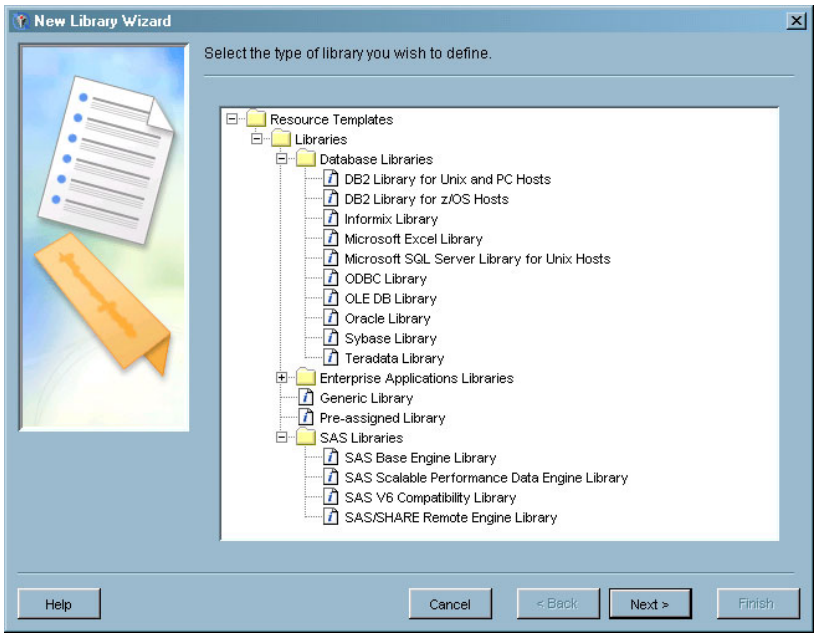
To launch the New Library wizard from SAS Management Console, complete these steps:

1. Use your metadata profile to log in 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 ▶ Data Library Manager ▶ SAS Libraries

3. To display the first wizard window, select

   Actions ▶ New Library



4. Select the applicable resource template, then click Next to continue. The balance of the information that you must enter depends on the template that you select.

Many of the library types correspond to the engine types specified on the SAS LIBNAME statement, with the options available for the library definition corresponding to the LIBNAME options for the engine. Some of those options can be used to optimize use of the tables within the libraries. For more information, see Chapter 10, "Optimizing Data Storage," on page 181.

For more information about how to define a library, including how to pre-assign a library to a server, see the *SAS Management Console: User's Guide*.

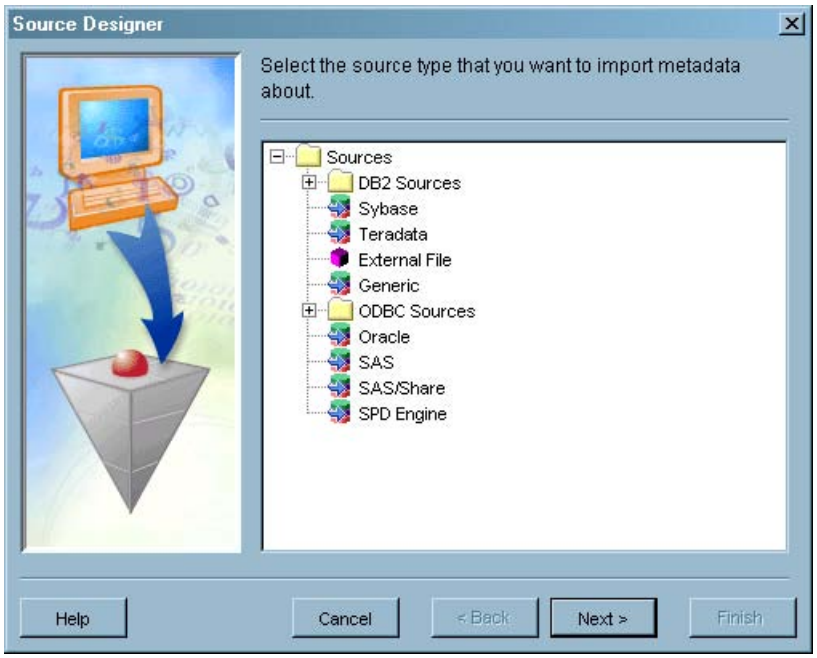
Defining Data Sources with a Source Designer

Library members (data sources) also must be defined in the metadata. You can define data sources by using a source designer wizard, which can be launched independently from within SAS OLAP Cube Studio and SAS ETL Studio, or from within the Cube Designer wizard when you are choosing the input data source for the cube.

To launch a source designer wizard from SAS ETL Studio, complete these steps:

1. Use your metadata profile to log in to the SAS Metadata Server that contains the SAS Metadata Repository that you want to use.
2. Select

   Tools ► Source Designer



3. Select the wizard for the type of data source that you want to define, then click Next to continue. The balance of the information that you must enter depends on the wizard that you select.

In SAS ETL Studio, you can include the metadata for a data source in a job that extracts information from one or more sources and writes it to one or more targets. You then run the job to create the specified targets on the file system.

For more information about using source designers in SAS ETL Studio, see the SAS ETL Studio Help, which is available from within the product. For more information about using source designers in SAS OLAP Cube Studio, see the SAS OLAP Cube Studio Help.

Defining Data Sources with the Metadata LIBNAME Engine

If available, you should use a source designer (described in "Defining Data Sources with a Source Designer" on page 171) to define metadata about your data sources. However, as an alternative, you can use the metadata LIBNAME engine.

The metadata engine works much like other SAS engines. That is, you execute a LIBNAME statement in order to assign a libref and to specify an engine. You then use that libref throughout the SAS session where a libref is valid. However, instead of the libref being associated with the physical location of a SAS data library, the metadata libref is associated with a set of metadata objects. The metadata objects identify the SAS engine that provides access to the data and options that are necessary to process the SAS data library and its members.

Here is an example of a LIBNAME statement for the metadata engine and a description of what happens when you execute the statement:

```
libname oralib meta libid=A8000001 repid=AWPKT800
   userid=metaid pw=metapw
   ipaddr=myip.us.org.com port=6401
   protocol=bridge liboptset=myopts;
```

1. The metadata engine retrieves information about the target SAS data library from the metadata.
2. The metadata engine uses the retrieved information to construct a LIBNAME statement for the engine that is specified in the metadata (referred to as the underlying engine) and assigns it the appropriate options.
3. Then, when the metadata engine needs to access data, the metadata engine uses the underlying engine to process the data.

Note: For more information about the metadata LIBNAME engine, see the *SAS Metadata LIBNAME Engine User's Guide*. △

Creating Metadata for an Existing Data Source

The METAOUT= option for the metadata LIBNAME engine controls the results of the output processing. Here are the output choices:

- ALL creates a new table and registers corresponding metadata.
- META registers just metadata for a specified table.
- DATA creates a new table but does not register metadata.

You can use the METAOUT= option either as a LIBNAME statement option or as a data set option. If you want to specify behavior for a library, use the LIBNAME statement option. (Note that the behavior applies to all members in the library and exists for the duration of the library.) To specify behavior for a specific table, use the data set option.

Note: In order to register table metadata by using the metadata LIBNAME engine, you must have Create permission for the table's library. For more information about permissions, see "Planning Your Access Controls" on page 62. △

The following code illustrates how to create metadata for an existing table with the METAOUT=META option on the LIBNAME statement. For this example, the Sales table exists in an Oracle library.

```
libname oralib oracle user=myuser pw=mypw
   path=ora_dbms preserve_tab_names=yes
```

```
   connection=sharedread schema=myschema;  ❶

libname metaeng meta libid=A8000001 repid=AWPKT800  ❷
   userid=metaid pw=metapw
   ipaddr=myip.us.org.com port=6401
   protocol=bridge (metaout=meta);   ❸

data metaeng.New ;
   set oralib.Sales (obs=0);   ❹
   stop;
run;   ❺
```

1. The LIBNAME statement for the Oracle SAS/ACCESS engine directly accesses the Oracle library that contains the Oracle table Sales.
2. The LIBNAME statement for the metadata engine uses the argument LIBID= in order to identify the existing SASLibrary object that defines information about the Oracle library and serves as an anchor point for obtaining other metadata. The REPID= option identifies the metadata repository in which the library resides. You can find these ID values by viewing the properties for the library in SAS Management Console. The ID appears in the form *repositoryID.libraryID*.

   As an alternative to LIBID= and REPID=, you can use LIBRARY="*library-name*" and REPNAME="*repository-name*".
3. With the METAOUT=META option specified, the behavior for the library will be to create only metadata for output processing.
4. OBS=0 is used to prevent rows from being inserted, or use the STOP statement to stop processing the DATA step. If you do not use either technique, unnecessary Oracle processing occurs.
5. Using the Oracle SAS/ACCESS engine, the DATA step creates metadata in the repository based on the existing table Sales.

You also can specify metadata connection information with SAS system options.

Table 9.1 Metadata LIBNAME Options and Corresponding SAS System Options

Metadata LIBNAME Option	SAS System Option
IPADDR	METASERVER
PORT	METAPORT
USER	METAUSER
PASSWORD	METAPASS
PROTOCOL	METAPROTOCOL
REPNAME	METAREPOSITORY

Preparing for Cube Loading

You create cubes with the Cube Designer wizard, which is available from SAS ETL Studio and SAS OLAP Cube Studio. The Cube Designer wizard helps you perform these tasks:

- create and edit cube definitions that are stored in the active metadata repository

□ build cubes based on stored definitions.

Preliminary Tasks

In order to use the Cube Designer wizard to define metadata for cubes, these tasks must have been completed:

1. The SAS Metadata Server is started.
2. At least one metadata repository is defined.
3. An OLAP schema is defined in the SAS Metadata Repository. See "Making Cubes Available to a SAS OLAP Server" on page 174.
4. A database server is defined, if users are loading cubes from a SAS/ACCESS database table.
5. A database schema is defined, if users are loading cubes from a SAS/ACCESS database table.
6. One or more libraries are defined.
7. One or more data sources are defined.
8. Cube builders have ReadMetadata and WriteMetadata permissions to the SAS Metadata Repository in which they want to save the metadata.
9. Cube builders have a metadata profile that they can use to log in to the SAS Metadata Server that contains the SAS Metadata Repository connection that they want to use.

Note: Users can define the libraries and data sources as part of completing the Cube Designer wizard. △

In order to create the physical cube in addition to defining its metadata, these tasks also must have been completed:

1. An Object Spawner is listening for requests for services from a SAS Workspace Server.
2. A SAS Workspace Server is available for services. See "Testing a SAS Workspace Server Connection with SAS OLAP Cube Studio" on page 178.

During a planned installation, a SAS Metadata Server, an Object Spawner, and a SAS Workspace Server are defined and configured. A SAS OLAP Server is also defined and configured, although the server does not have to be running in order to build a cube. On Windows machines, the servers and the spawner are usually started as services. For other platforms, the planned installation creates start-up scripts. The installation process also creates a foundation metadata repository.

Note: For information about a planned installation and configuration, see Chapter 7, "Installing and Configuring Your Software," on page 123. △

Making Cubes Available to a SAS OLAP Server

Part of a SAS OLAP Server definition is an OLAP schema assignment. An OLAP schema specifies which group of cubes that a SAS OLAP Server can access. A cube is assigned to a schema when it is created with either SAS ETL Studio or SAS OLAP Cube Studio. The schema must be in the active metadata repository or in a repository that is dependent on the active repository.

If the SAS OLAP Server is installed on a machine that is hosting the SAS Metadata Server, a default OLAP schema named **`SASMain - OLAP Schema`** is assigned to the SAS OLAP Server. If the SAS OLAP Server is installed on a machine that is not hosting a SAS Metadata Server, then the application server might have a user-defined name that

will be used instead of **SASMain** to form the default OLAP schema name. For example, if the application server is named **SASApp**, then the OLAP schema will be named **SASApp - OLAP Schema**.

You can assign a different schema to a SAS OLAP Server by using any of these methods:

- In SAS OLAP Cube Studio, create a new OLAP schema with the OLAP Schema wizard. You can assign the new schema to one or more servers that are available in the current repository and in any respositories that depend on the current repository (see "Defining a New OLAP Schema with SAS OLAP Cube Studio" on page 175).
- In SAS Management Console, edit the definition of the SAS application server that contains the logical SAS OLAP Server. You can select an existing OLAP schema or launch the OLAP Schema wizard to define a new schema.

Note: Although you can have multiple schemas in a repository, a server can only access the cubes in one schema, so you do not need to create more OLAP schemas than there are OLAP servers at your site. △

Note: When defining a new SAS OLAP Server, if you accept the default definition settings, then an OLAP schema automatically is created and assigned to the server. To change that assignment, you edit the server definition. △

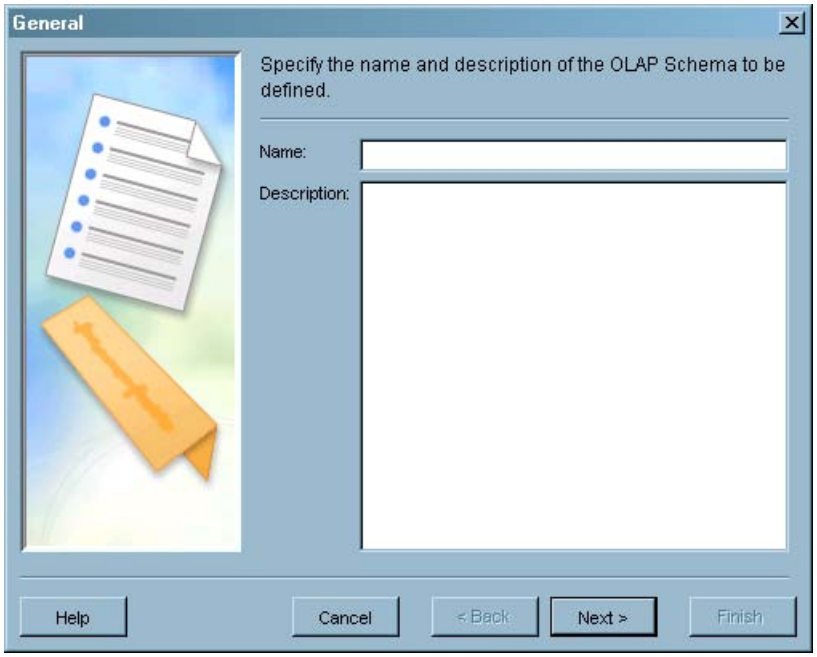
Defining a New OLAP Schema with SAS OLAP Cube Studio

To define a new schema with SAS OLAP Cube Studio, complete these steps:

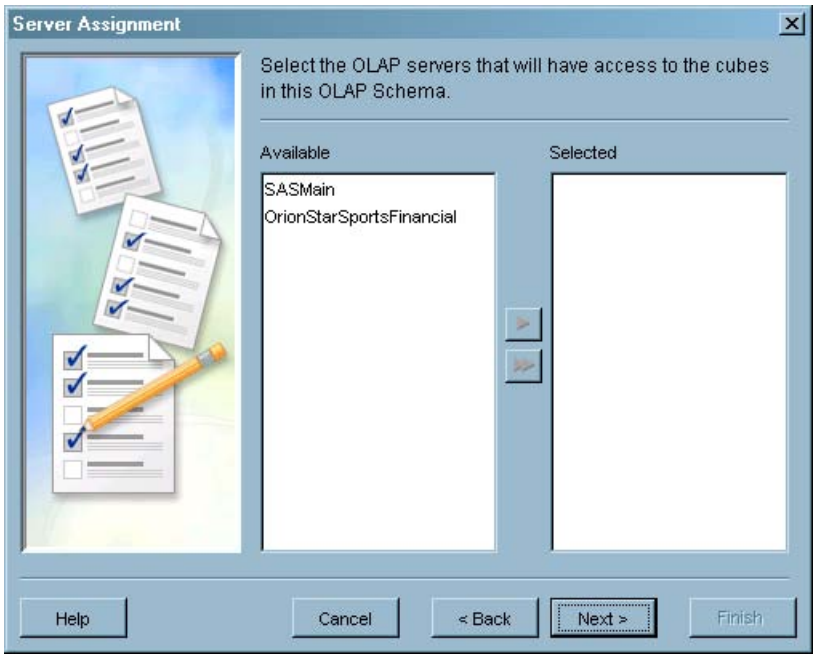
1. Use your metadata profile to log in to the SAS Metadata Server that contains the SAS Metadata Repository connection that you want to use.
2. In SAS OLAP Cube Studio, select

   File ► New OLAP Schema

   to launch the OLAP Schema wizard.
3. On the General window, enter the schema **Name** and **Description**, then click Next.



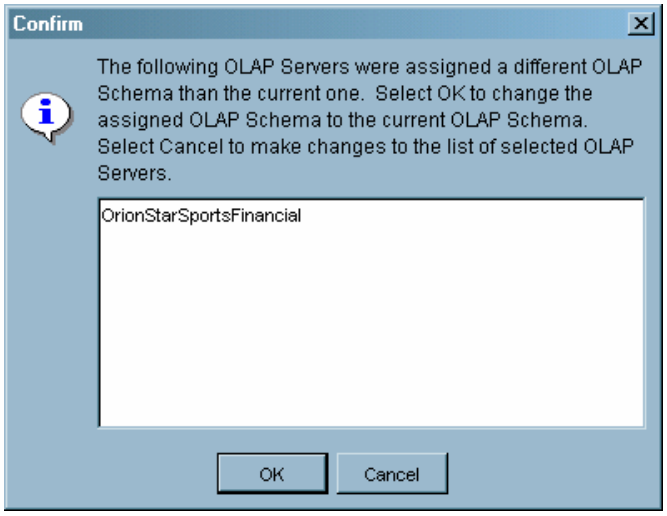
4 On the Server Assignment window, specify the OLAP servers that should have access to the group of cubes assigned to the OLAP schema. The **Available** box lists the OLAP servers defined in the current metadata repository and in any repositories that are dependent on the current repository. To add a server to the schema, select server names in the **Available** list box and move them to the **Selected** list box.



Click [Next] to display the Finish window.

Note: This step is optional. If you choose not to specify the servers by using the wizard, you can add that information later by modifying the schema's property sheet. △

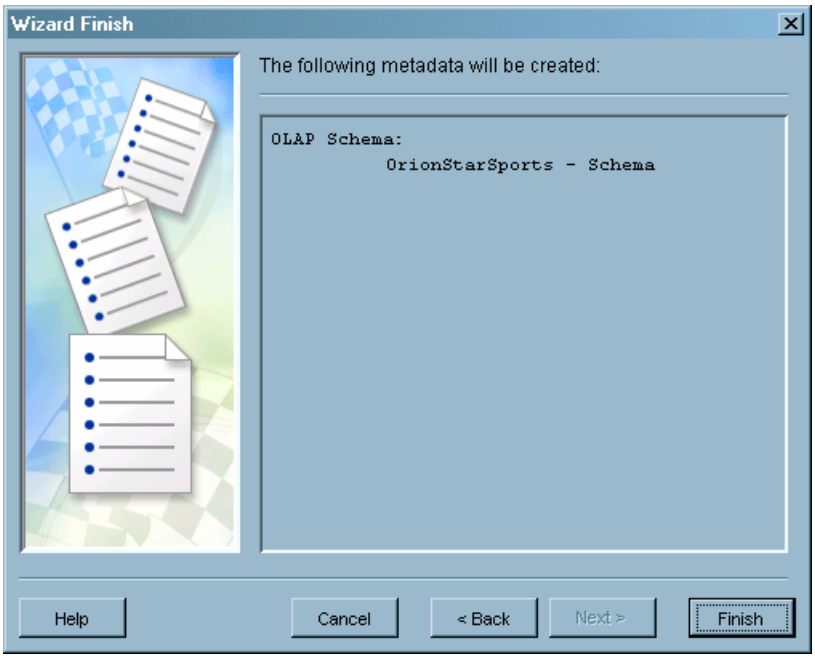
5 If you select one or more servers that have already been assigned to a different OLAP schema, you see a confirmation message box when you click [Next].



□ Click [OK] to accept the reassignment, close the message box, and display the Wizard Finish window.

- To disregard the reassignment, click Cancel to remain on the Server Assignment window.

6 On the Wizard Finish window, review the selections that you made on the General and Server Assignment windows.



To change a selection on a previous window, click Back. Otherwise, click Finish to save the new schema and exit the wizard.

Editing a SAS OLAP Server Definition to Change the Schema Assignment

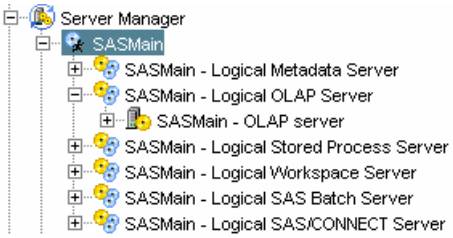
To edit a SAS OLAP Server definition to change its OLAP schema assignment, complete these steps in SAS Management Console:

1 Use your metadata profile to log in to the SAS Metadata Server that contains the SAS Metadata Repository that you want to use.

2 In the SAS Management Console navigation tree, select

Environment Management ► Server Manager

3 Select a SAS application server that contains the SAS OLAP Server that you want to edit.



4 Select

File ► Properties

5 In the Properties dialog box, select the OLAP Schema tab.

6 From the drop-down list, select the existing OLAP schema to which you want to assign this server, or click [New] to define a new schema.

Note: For information about defining a new schema, see "Defining a New OLAP Schema with SAS OLAP Cube Studio" on page 175. From within SAS Management Console, the steps are the same, except that the Server Assignment window does not appear. △

7 Click [OK] to save your changes and exit the dialog box.

Testing a SAS Workspace Server Connection with SAS OLAP Cube Studio

If users plan to create the physical cube in addition to registering its metadata, then a SAS Workspace Server must be available. To check availability within SAS OLAP Cube Studio, complete these steps:

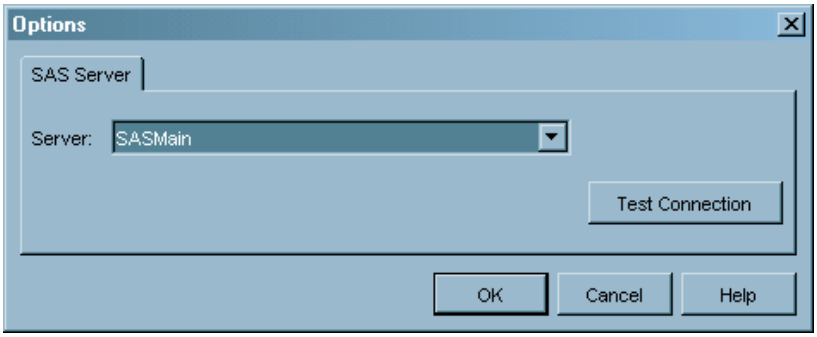
Note: Every SAS Intelligence Architecture product that uses a SAS Workspace Server has a facility for testing the connection to the server. For details, see the individual product documentation. △

1 Use your metadata profile to log in to the SAS Metadata Server that contains the SAS Metadata Repository connection that you want to use.

2 Select

[Tools] ▶ [Options]

to display the Options dialog box.



3 From the drop-down list, select the name of the SAS application server that contains the SAS Workspace Server to which you want to connect.

4 Click [Test Connection].

Note: If the Log On To dialog box appears, enter your **User Name** and **Password**, then click [OK]. △

5 If the SAS Workspace Server successfully connects to the SAS Metadata Server, then you will see the message "Connection to the server was successful!" Click [OK] in the message box.

Note: If the connection is not successful, contact the administrator who defined the SAS application server that contains the SAS Workspace Server that you are using. △

6 Click [OK] to close the Options dialog box.

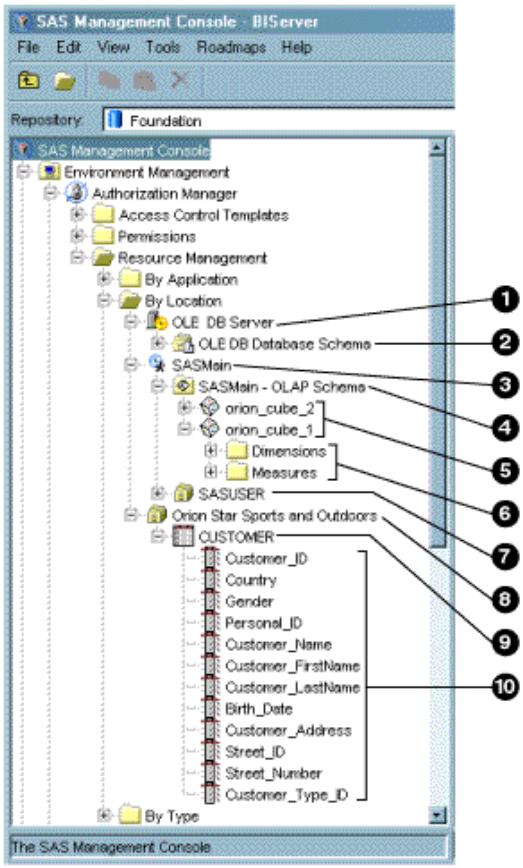
Securing Access to the Metadata That Defines the Data

To secure access to the metadata objects that you just defined, you use the Authorization Manager plug-in to SAS Management Console. To locate the metadata objects in the SAS Management Console navigation tree, complete these steps:

1. Use your metadata profile to log in 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

Display 9.1 Location of Metadata Objects under the Authorization Manager



1. Database server
2. Database schema
3. SAS application server
4. OLAP schema
5. Cubes
6. Dimensions and measures in a cube
7. Library that is assigned to a SAS application server
8. Library that is not assigned to a SAS application server

9 Data source contained within a library

Note: Securing the metadata object that represents the data source is not the same as securing access to the underlying data. In the current release, most SAS Intelligence Architecture applications enable users to view the underlying data if the users have access to the metadata object that represents the data source and all of its parent objects. △

10 Columns in a data source

For information about how to grant permissions for metadata objects, see the help for the Authorization Manager plug-in. For information about how permissions are inherited, see

- "Inheritance in SAS Data" on page 45
- "Inheritance in Relational Database Data" on page 45
- "Inheritance in OLAP Data" on page 46.

Note: Because all SAS Intelligence Architecture 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 "Planning Your Access Controls" on page 62. △

CHAPTER

10

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 9, "Preparing Data for Use," on page 163) can be optimized to improve performance. Some optimization can be achieved, for example, by specifying transformation options in SAS ETL 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 182
- "Indexing Data" on page 183
- "Sorting Data" on page 185
- "Buffering Data" on page 187
- "Using Threaded Reads" on page 188

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.

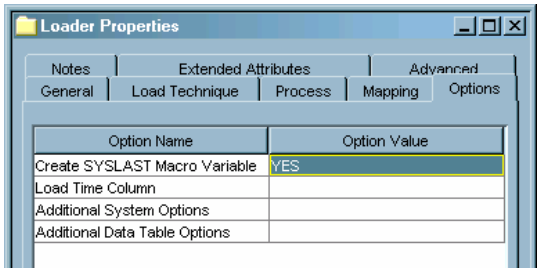
These are the types of compression that you can specify:

- 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 ETL 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 10.1 The Options Tab in a Loader Properties Dialog Box in SAS ETL Studio



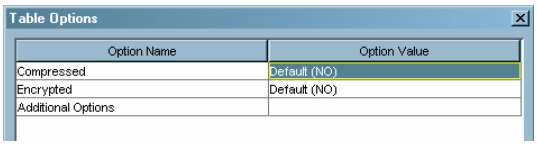
- all tables for a particular SAS data library. For example, when you define a Base SAS Engine library in the metadata, you can specify the COMPRESS= option in

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 195). 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 196).

Note: You cannot specify compression for a SPD Engine data library. △

- an individual table. In SAS ETL 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 10.2 The Table Options Dialog Box in SAS ETL 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 ETL 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:

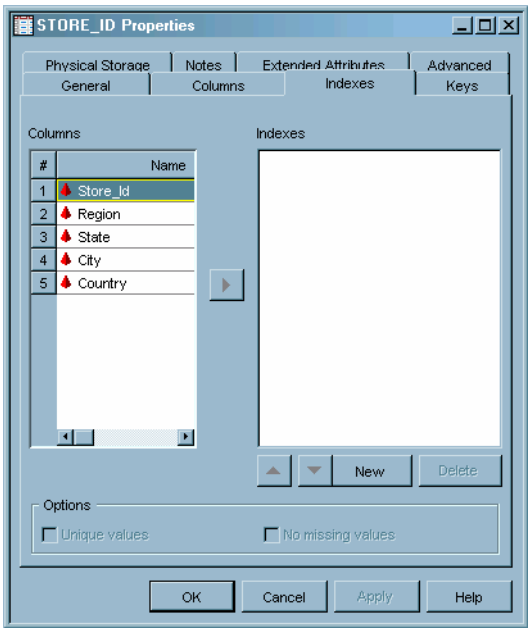
- A simple index, which consists of the values of one column.
- A composite index, which consists of the values of more than one column, with the values concatenated to form a single value.

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.

Besides writing SAS code to create indexes, you can create indexes by using SAS ETL Studio. In SAS ETL 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 10.3 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 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.

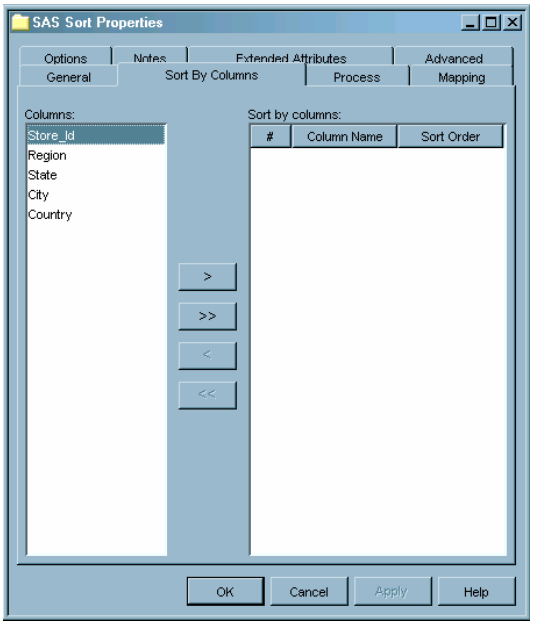
For more information about creating indexes, see *SAS Language Reference: Concepts*.

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:

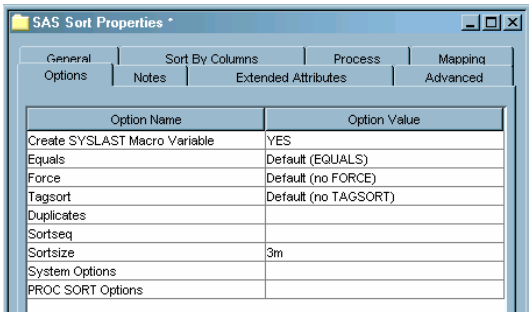
- using the SAS SORT procedure.
- setting properties for a SAS sort template in SAS ETL Studio.

Display 10.4 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 ETL Studio, access the Options tab in the properties window for the sort template and enter a value in the **`Sortsize`** field.

Display 10.5 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 199.

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

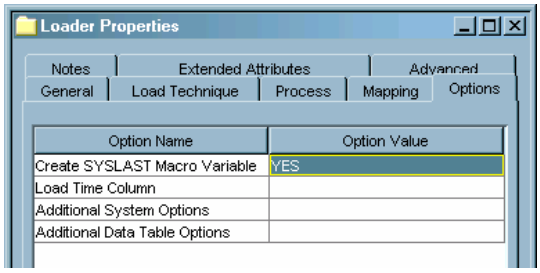
For Base SAS tables and some relational database tables, you can adjust page buffer settings to optimize CPU and I/O use. Different options are used for each type of table.

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 ETL 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 196) 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 196.

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 ETL 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.

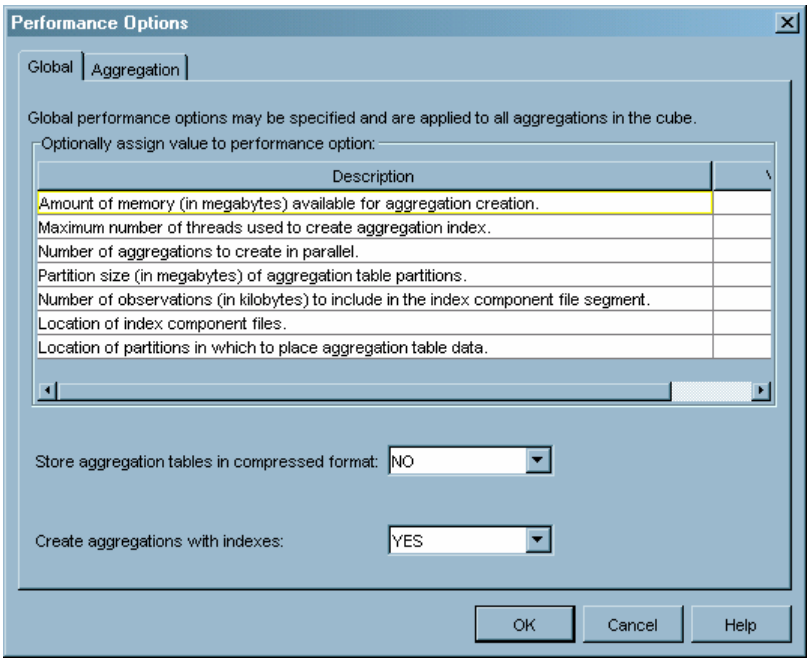
The resulting log file shows timing statistics for each cycle. You can examine this information to determine whether your system is configured correctly. The program is available at **`support.sas.com/rnd/scalability/spde/valid.html`**.

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 ETL Studio or from SAS OLAP Cube Studio.

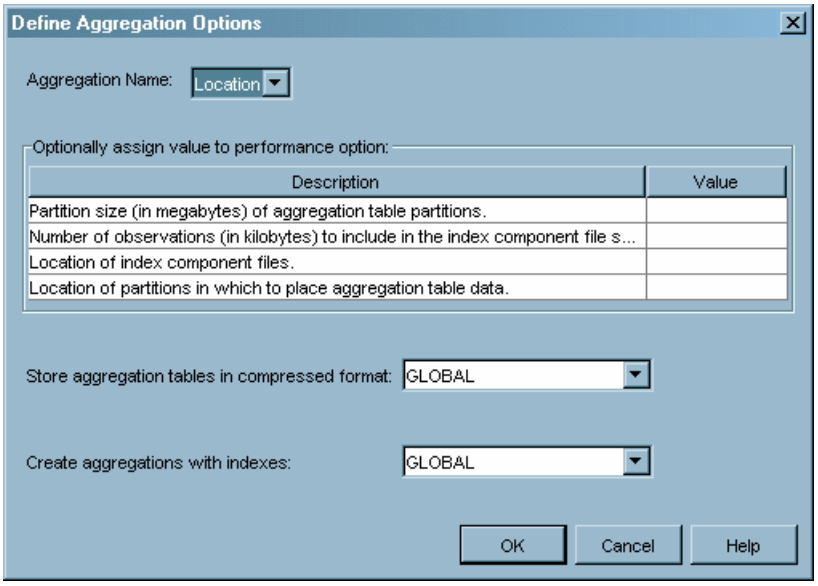
In the Performance Options dialog box, the Global tab enables you to set options that apply to all aggregations in the cube.

Display 10.6 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 10.7 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.

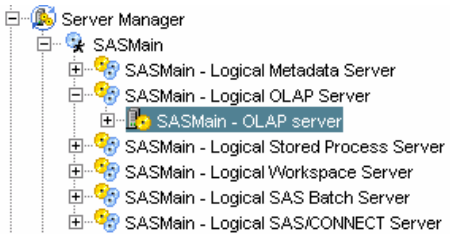
Optimizing Performance of a SAS OLAP Server

During a planned SAS 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 193. △

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.

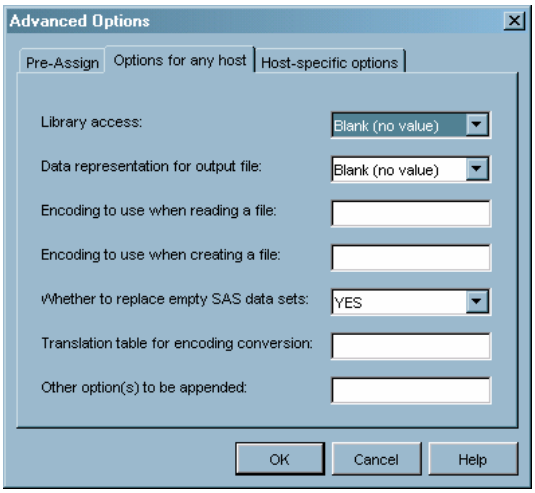
Setting LIBNAME Options That Affect Performance

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 10.8 The Options for Any Host Tab in the Advanced Options Dialog Box for a Base SAS Library



Here are some examples of options that may 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 182).

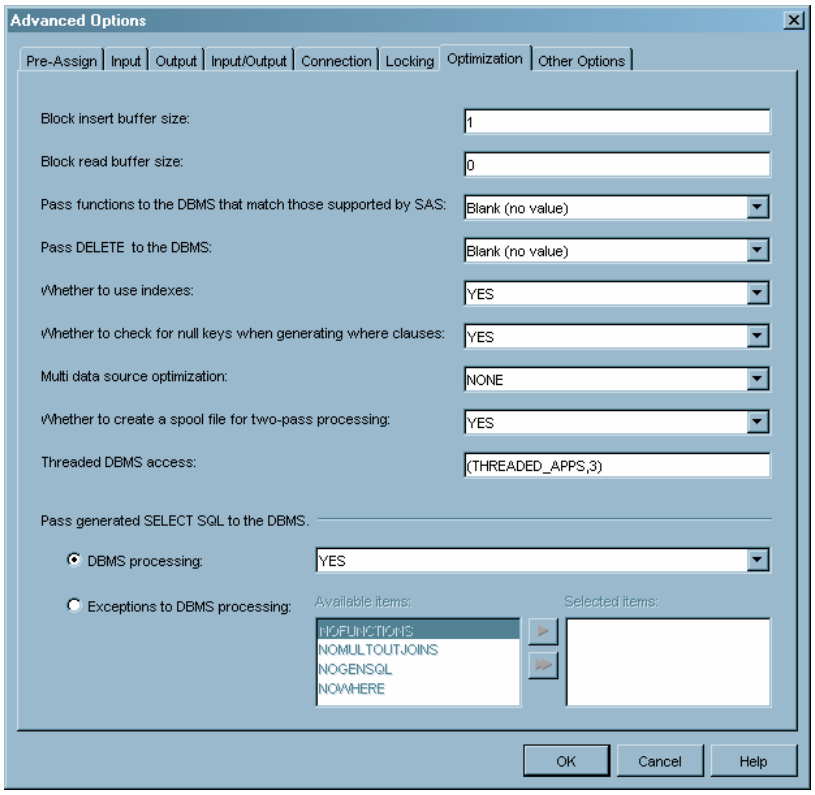
For information about each of the LIBNAME options in the Advanced Options dialog box, click the Help button.

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 10.9 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 187.

Block read buffer size (READBUFF=)
specifies the number of rows of DBMS data to read into the buffer. See "Buffering Data" on page 187.

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 183.

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 188.

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.

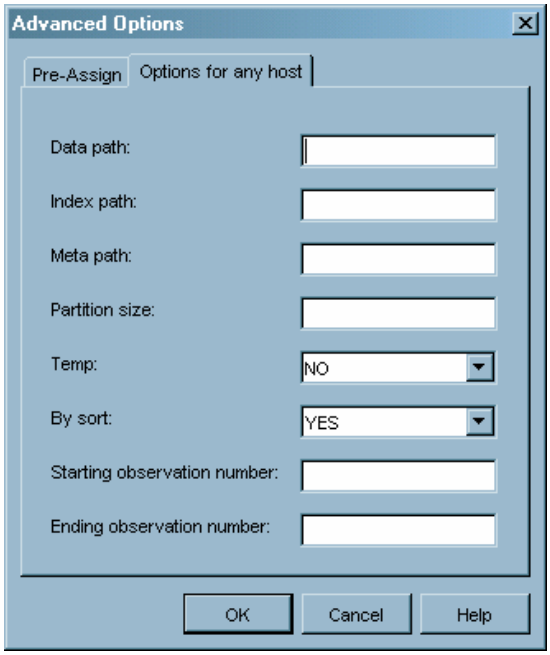
The Other Options tab, which is available for all database management systems, can be used to specify LIBNAME options such as COMPRESS= (see "Compressing Data" on page 182).

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 10.10 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 **`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 5

Application Administration

CHAPTER

11

Administering SAS ETL Studio

Overview of Administering SAS ETL Studio

SAS ETL Studio is a Java application that ETL specialists can use to build data warehouses and data marts from existing operational data. For information on how to build warehouses and marts, these users should consult the *SAS ETL Studio User's Guide*, which is available at support.sas.com. To access the book, go to **`support.sas.com`**, then select

Documentation ► Products & Solutions Documentation

In the **`Select a Product`** box on the Products & Solutions page, select **`SAS ETL 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 can or must perform the following tasks:

- Make sure that your ETL 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.
- Make sure that your ETL specialists can connect to the necessary data servers.
- 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.
- Set up a folder structure for metadata objects that enables you to control access to those objects.
- Control access to SAS ETL Studio SAS code transformations, which are a class of user-written transformation.
- Import metadata from data modeling tools such as the AllFusion ERwin Data Modeler.
- Test the servers (and clients) that enable your users to schedule sets of SAS ETL Studio jobs.
- Set up the infrastructure necessary for your users to employ data-quality transformations.

All of these topics are covered in the remaining sections of this chapter.

Connecting to SAS Servers

In order to use SAS ETL Studio, data warehousing 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.

Metadata Server

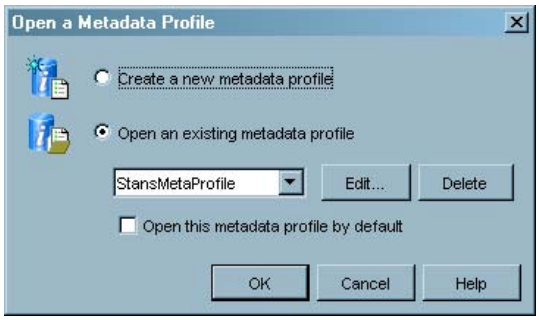
The easiest way to ensure that ETL specialists can connect to a running metadata server is to use SAS ETL Studio to open a *metadata profile*, which contains information about the metadata server, a metadata repository, and a user. When SAS ETL Studio opens this profile, it attempts to connect to the metadata server. If it connects successfully, SAS ETL 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 ETL Studio has been installed and start the application. You do this by selecting

Start ► Programs ► SAS ► SAS ETL 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 ETL 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 online help, if necessary, to create a profile. (Do not save your password in the profile.) When you have finished defining the profile, SAS ETL Studio will automatically try to use the profile to connect to the metadata server.

Workspace Server

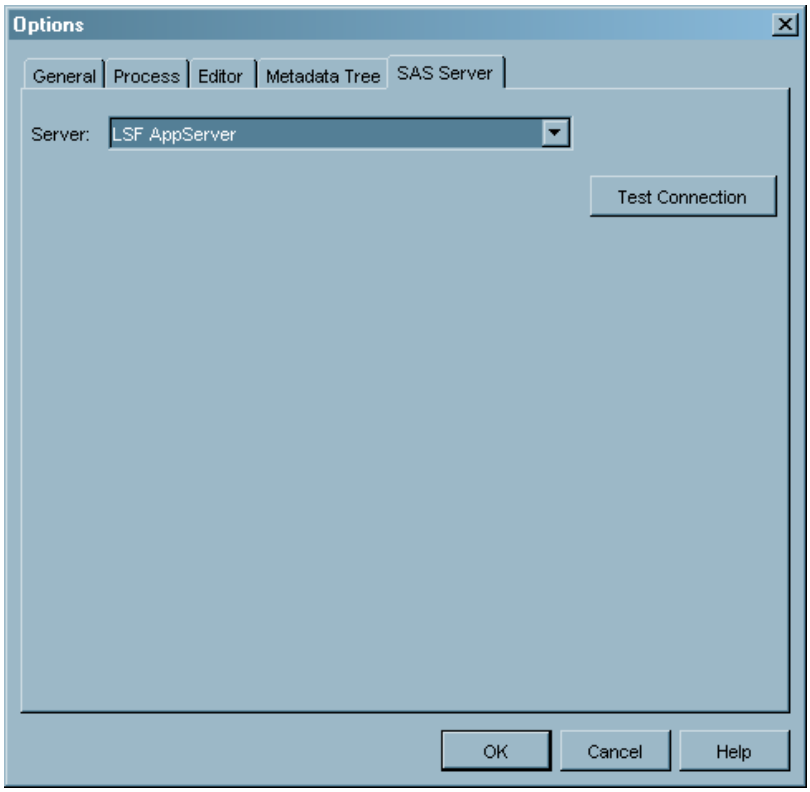
When you execute a job in SAS ETL Studio, the application submits generated SAS code to a workspace server, which executes the code. Therefore, it is imperative that an object spawner be up and running and that SAS ETL Studio be able to use the spawner to start a workspace server. To test a connection 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

Once 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 206), it is a good idea to make sure that users can get to the data sources that will provide the input to SAS ETL Studio jobs, such as

- 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 "Defining Metadata about the Data" on page 167.

- Use SAS ETL 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 ETL Studio's Inventory tree.

To determine whether your ETL 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 should appear and show you the data in the table.

View Data: CITY

#	City ID	City Name	Country
1	3500000001	Acheres	France
2	3500000002	Aix En Provence	France
3	3500000003	Alencon	France
4	3500000004	Amiens	France
5	3500000005	Amilly	France
6	3500000006	Angers	France
7	3500000007	Angouleme	France
8	3500000008	Annecy	France
9	3500000009	Annonay	France
10	3500000010	Antibes	France
11	3500000011	Antony	France
12	3500000012	Arcueil	France
13	3500000013	Argenteuil	France
14	3500000014	Armees	France
15	3500000015	Arras	France
16	3500000016	Asnieres Sur Seine	France
17	3500000017	Athis Mons	France
18	3500000018	Aubagne	France
19	3500000019	Aubervilliers	France
20	3500000020	Aulnay Sous Bois	France
21	3500000021	Aurillac	France
22	3500000022	Aussonne	France
23	3500000023	Auxerre	France
24	3500000024	Avranches	France

Note: You can quickly determine the type of a table in the Tables folder by bringing up the table's Properties dialog 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 ETL Studio contains a feature called change management that enables ETL 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 ETL 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 tasks you must perform to set up change management in the case where there is one project repository per

developer. The following sections provide details about how to perform these tasks, where necessary:

1. Set up an operating system user account for each ETL specialist. On Windows systems, you must give the user or group 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 representing 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 invoked the metadata server has write access to this directory. For details on how to perform this step, see "Creating a Repository Directory" on page 210.
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 211.
5. In SAS Management Console, set the permissions on the foundation and project repositories so that the user 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 211.
6. Have the ETL 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 213.
7. Have the ETL 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 214.

As previously mentioned, you should already have performed steps 1 and 2. If you have not performed those steps, return to "Setting Up Security for Regular Users" on page 77 and 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 *configuration-directory***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 ETL specialist. This directory functions as the user's project repository. In addition, set the ownership and permissions on this new directory so that only the user who started 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 ETL 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 up.

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 area 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—Accept the default value, Base. This setting indicates that Base SAS will be used to access your metadata repository.
   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 your user will check metadata objects out of the foundation repository into his or her project directory to work on them. 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 button. The foundation repository icon will move to the list on the right, entitled "Repository will depend on." 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.

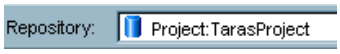
Setting Metadata Permissions for Your User

Once 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 in to SAS Management Console as an unrestricted user. For example, you could be logged in 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.

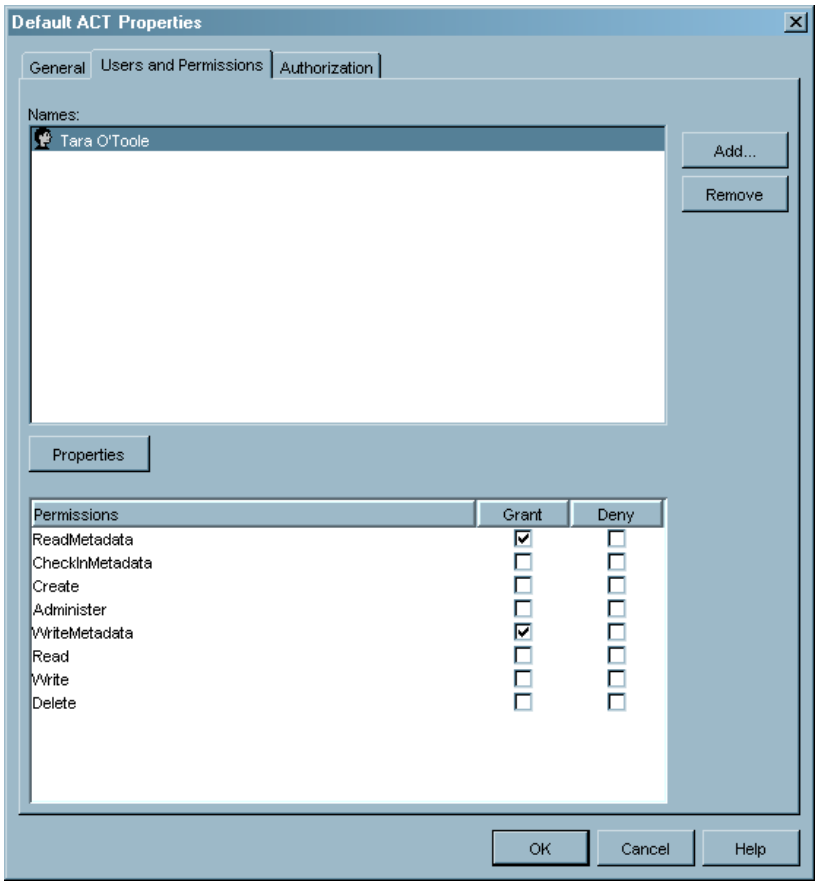


2. In the left pane, expand the Authorization Manager section of the tree and, within that section, the Access Control Templates section. You will see an icon named Default ACT (Access Control Template). Select this icon and select

   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 ETL 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 ETL 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 detailed instructions about how to create a metadata profile, here they are:

1 Start SAS ETL Studio. You will see a dialog box named Open a Metadata Profile.

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 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 ETL 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. Once 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 *windows-domain\user-name* or *host-name\user-name*.
- Password—Enter your password.

There is also a **Save username and password in this profile** check box. We recommend that you do not select this check box. If you do, any user can connect to the metadata server by simply starting SAS ETL 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.)

7 In the Finish window, click Finish.

Using the Metadata Profile

When the user finishes running the Metadata Profile wizard, SAS ETL 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 ETL Studio interface appears, there is a Project tab at the bottom of the tree pane. If the user clicks on this tab, he or she will see an icon representing his or her project repository.

Note: For information about how to work in a change-managed environment, see the *SAS ETL Studio User's Guide*. △

Changing the Default umask for ETL Developers (UNIX only)

By default, a user's umask is set to 022, meaning that any file the user creates will have the following file permissions:

```
rwxr-xr-x
```

Because the workspace server runs under the user's credentials, any target SAS data set that an ETL developer creates using SAS ETL Studio will have these permissions as well. This situation can become problematic if you want multiple developers to be able to run the same job and create or update the same target file. For example, if Developer 1 runs a job that creates a file called **Target.sas** and then goes on vacation, when

Developer 2 tries to run that same job, Developer 2 will get an error because Developer 2 does not have permission to write to the target file.

There is a simple workaround for this problem:

1. Create a group for SAS ETL specialists, and make all ETL specialists members of that group.
2. In each user's login file, set the user's default umask to 002, for example,

```
umask 002
```

Now, files created by any member of the group will have the following permissions:

```
rwxrwxr-w
```

Therefore, any member of the group can update a file such as a target SAS data set.

Using Custom-Tree Folders for Security

When your ETL developers are working in SAS ETL 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 ETL 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 ETL 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 ETL developer and one for the administrator.

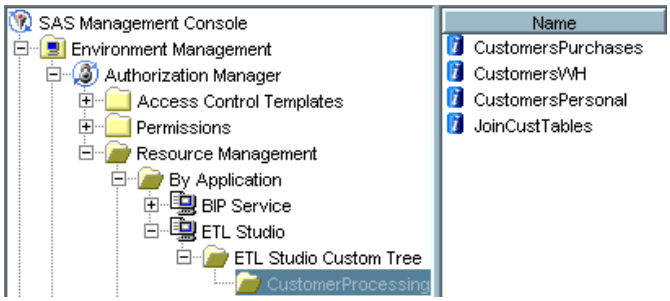
- The Custom tree enables developers to organize their metadata objects in any manner they see fit. For example, they might want to create a folder that contains all of the metadata objects that are related to a particular set of jobs.
- The Custom tree also displays in the SAS Management Console's Authorization 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 an ETL 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.

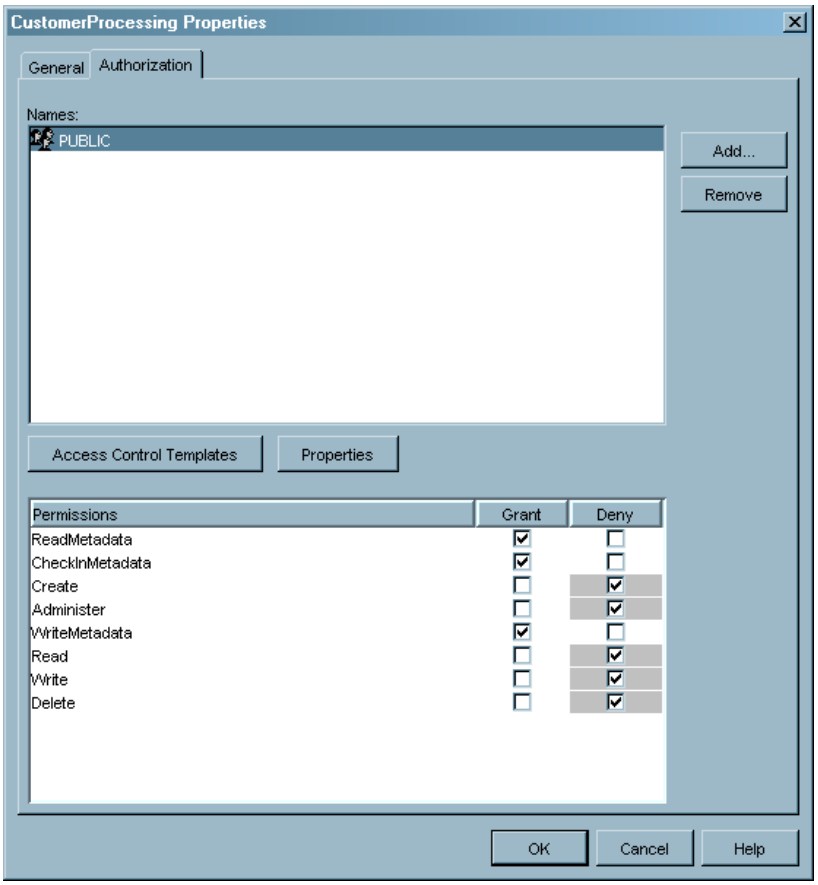
   At this point, you will see the top-level groups in the Custom tree, and if you select a group/folder, you will see the objects in that folder in the pane to the right of the tree structure.



   In this case, a user has created a group/folder called CustomerProcessing, and this group contains several tables and a job.

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. By default, you will see the settings 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 11.1 Custom Tree Object Types

Object Type	Description	Inherits from Folder
Cubes	OLAP cubes including dimensions, hierarchy, and level.	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

Object Type	Description	Inherits from Folder
Jobs	Processes that create output. In SAS ETL Studio, the job is illustrated by a process flow diagram.	Yes
Libraries	SAS 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 Schema	OLAP schema 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. △

Multiple Inheritance of Access Controls

As is explained in "Inherited Access Controls" on page 43, 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 SAS Code Transformations

SAS ETL 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 SAS code transformations. ETL specialists can write additional Java plug-ins and SAS code transformations. However, this section deals only with SAS code transformations because it is these transformations that you might need to secure.

SAS ETL Studio users can use a Transformation Generator wizard to create SAS code transformations. They can then export these transformations. In their exported state, the transformations are stored as XML files. Other SAS ETL Studio users can import these files, at which point those users can employ the SAS code transformations in their jobs.

The potential problem with this arrangement is this: suppose that multiple ETL 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 SAS code 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 ETL users can import SAS code 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 ETL developers is the ability to

- design and generate 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 ETL Studio or SAS Management Console to import metadata of this type. Both applications enable you to export metadata as well.

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

SAS ETL Studio's Metadata Import Wizard uses converters installed in the **`plugins`** directory (of SAS ETL Studio or SAS Management Console) to import metadata. By default, you will 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 "Supported Metadata Formats" on page 219.

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 ETL Studio, connect to the metadata server 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 on 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 detailed 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 ETL Studio or SAS Management Console **plugins** directory.

There are two restrictions on the export function:

- You can export only relational data, for example, data from a SAS library or a DBMS schema. The section "Importing Metadata" on page 219 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 you either of the following to start the export wizard:

- In SAS ETL 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 detailed step-by-step instructions on how to export metadata.

Testing the Job Scheduler

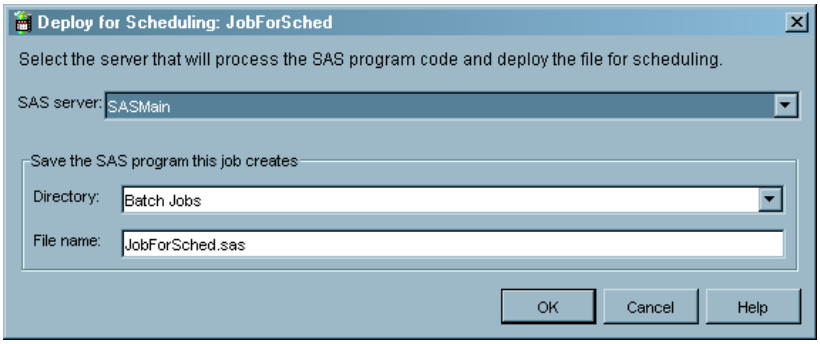
ETL specialists can create sets of SAS ETL Studio jobs, called flows, and execute each job within a flow

- at a certain time
- depending on the state of the file system
- depending on the status of another job within the flow.

The software that supports this scheduling includes the Schedule Manager plug-in to SAS Management Console and two products from Platform Computing: Platform LSF and Platform JobScheduler. 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.

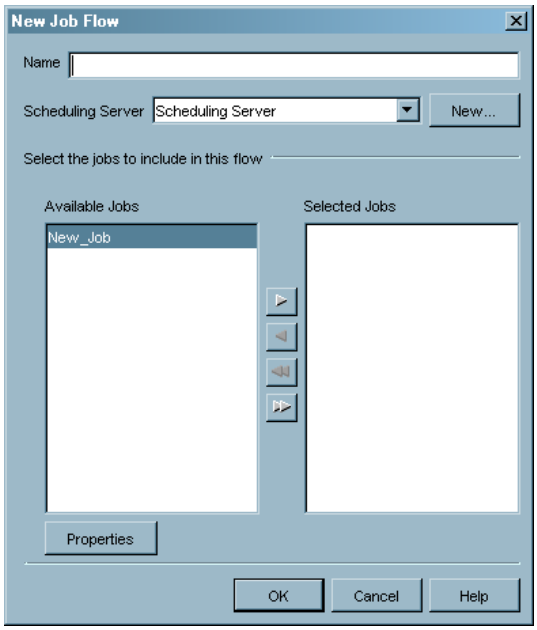
1. Create an empty job in SAS ETL Studio. You can do this by starting the New Job Wizard, entering a name (such as JobForSched) in the **Name** text box, and clicking Finish. A new job will appear in the Jobs folder of the Inventory tree.
2. Deploy the new job for scheduling by right-clicking the icon for the new job and selecting **Deploy for Scheduling**. The Deploy for Scheduling dialog box will appear.



   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 will appear.



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 Scheduler Server you that created earlier (JobScheduler by default).
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 will appear 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 JobScheduler).

Setting Up a SAS Data Quality Server

By installing the SAS Data Quality Server and configuring a SAS application server to read a Quality Knowledge Base, you enable ETL developers to use the data cleansing transformations Create Match Code and Apply Lookup Standardization. This section explains how to install the SAS Data Quality Server, how to configure an application server appropriately, and how to perform a simple test to ensure that the system is working. The section also covers several administrative tasks associated with data quality, including

- downloading new locales
- creating new schemes
- setting SAS ETL Studio's data quality options.

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. △

How the Data-Quality Software Has Been Configured

If you have installed the SAS 9.1 Foundation software, including the SAS Data Quality Server, on a machine and have run the Configuration wizard to configure the software on that machine, everything should be set up for your ETL developers to use the data cleansing transformations. 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 was installed in *sas-root*/**dquality**. Items such as locales and schemes are located in directories subordinate to **dquality**: **sasmisc/content/locale** and **sasmisc/content/scheme**.

The configuration file *configuration-directory*/**Lev1/SASMain/sasv9.cfg** sets up the SAS environment for the SASMain application server. It includes the following lines:

```
-dqlocale (ENUSA)
-dqsetuploc ''dqsetup.txt''
```

The first line indicates that the default locale is English (USA) and that this locale will be loaded into memory when the application server starts. The second line indicates that the **dqsetup.txt** file specifies the storage locations that make up the Quality Knowledge Base. The **dqsetup.txt** file is located in the same directory as the **sasv9.cfg** file.

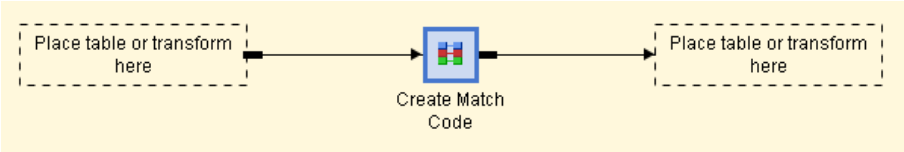
Testing the SAS Data Quality Server

A simple procedure to verify that your SAS Data Quality Server is working is to create a job in SAS ETL Studio that contains a Create Match Code transformation. Follow these steps to create such a job.

1. From the SAS ETL Studio desktop, select

   Tools ▶ Process Designer

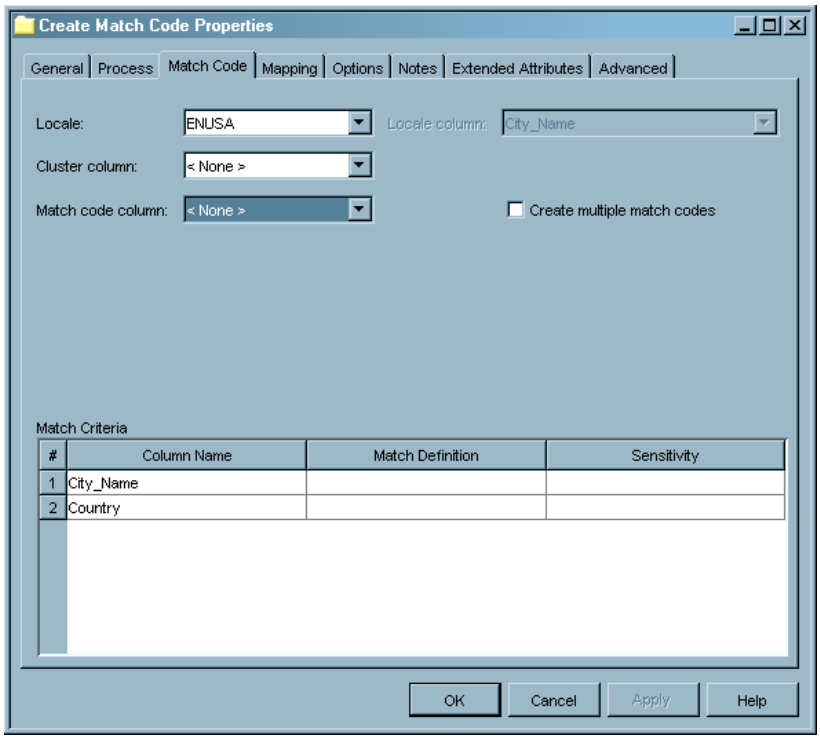
   to start the New Job Wizard.
2. In the New Job Wizard, enter a name for the job—such as Create Database Match Codes—in the Name text box. Then, click the Finish button. A new Process Designer window appears on the right side of your workspace.
3. From the Process Library tree, select and drag the Create 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 Right-click the icon for the Create Match Code transformation, and select **`Properties`** from the pop-up menu that appears. A Create Match Code Properties dialog box is displayed.

7 In the Create Match Code Properties dialog box, select the Match Code tab. You will know that the SAS Data Quality Server is set up correctly if (1) you see a graphical indicator that SAS ETL Studio is **`Reading [the] Quality Knowledge Base`** and (2) the lists in the Create Match Code Properties dialog box are populated, as shown in this 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: **`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

Before your ETL developers can use the Apply Lookup Standardizations template, you must create schemes using the SAS Data Quality Server or dfPower Studio. For information on how to create schemes using the SAS software, see *SAS Data Quality Server: Reference* for information on PROC DQSCHEME. For information about creating schemes using dfPower Studio, see the documentation for that product.

Normally, you should save these schemes to the directory *sas-root*/**dquality/sasmisc/content/scheme**. However, there is also a **scheme** directory in your configuration directory: *configuration-directory*/**Lev1/SASMain/SASEnvironment/QltyKB/scheme**. Use the latter location for schemes that are specific to an application server context. As explained in the section "Setting SAS ETL Studio's Data Quality Options" on page 225, you can specify which scheme repository SAS ETL Studio should use by setting a value in SAS ETL Studio's Options dialog box.

Setting SAS ETL Studio's Data Quality Options

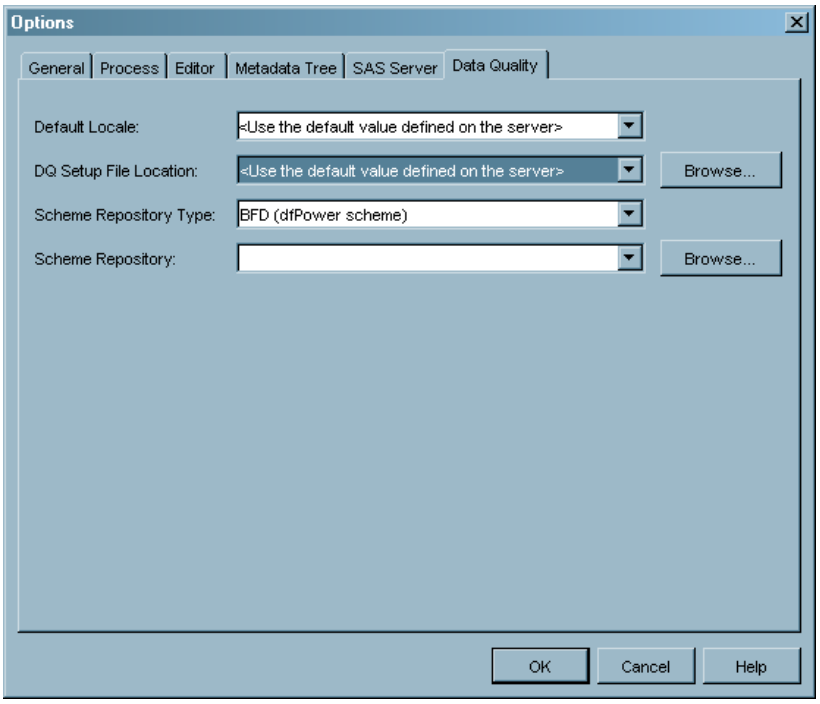
You can set several options related to data quality using SAS ETL Studio's Options dialog box.

1 Select

Tools ▶ Options

The Options dialog appears.

2 Select the Data Quality tab. The dialog box should now look like this:



The following table explains how to use the controls in this dialog box.

Table 11.2 Data Quality Options

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 list box.
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 list box or the **Browse** button to specify another setup file.
Scheme Repository Type	By default, the repository type is BFD because the schemes that are supplied with the SAS Data Quality Server are in this format. If you later create schemes that are SAS data sets, you can change the value here to NOBFD.
Scheme Repository	The default scheme repository is *sas-root*/**dquality/sasmisc/content/scheme**. Use the list box or the **Browse** button to specify a different repository.

CHAPTER

12

Analyzing and Accessing Your Intelligent Business Data

Overview of Intelligent Business Data Analysis and Delivery

Whether your organization requires electronic distribution of business data, interactive query and analysis environments, delivery of content through a Web-based portal, or publish-and-subscribe channel distribution, the SAS Intelligence Architecture can be tailored to meet your organization's needs. SAS provides many ways for your organization to analyze your business data and surface your business intelligence content. One way to accomplish this is to use stored processes to generate and analyze business data, and then deliver the data using the SAS Information Delivery Portal.

This chapter explains what stored processes are and how they can be used in the SAS Intelligence Architecture to develop, analyze, and deliver your intelligent business content. Specifically, this chapter provides information about how to use the following products to create, register, run, and deliver your stored processes:

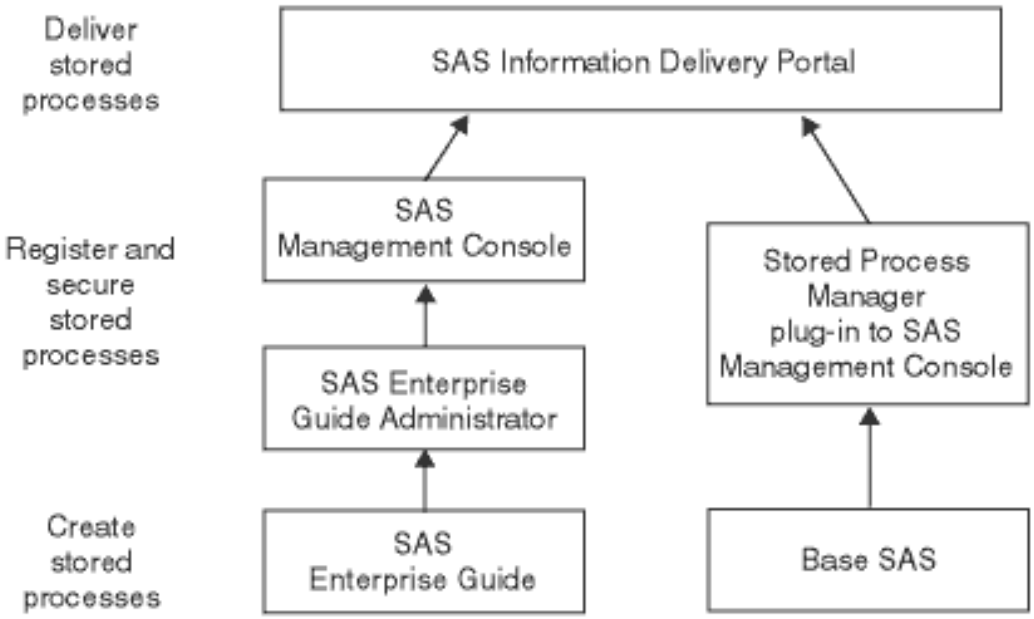
SAS Enterprise Guide — enables business analysts, programmers, and content providers within your organization to develop SAS Stored Processes and register them in the metadata repository. Stored processes are SAS programs that are stored on a server and can be executed as required by requesting applications. Stored processes can be used for analytics, building Web applications, delivering result packages to clients or the mid-tier, and publishing results to channels or repositories. Stored processes can also access any SAS data source or external file and create new data sets, files, or other data targets supported by SAS.

SAS Enterprise Guide Administrator — enables administrators to identify the workspace and stored process servers that will be used to run stored processes and to integrate SAS Enterprise Guide with your SAS metadata repository. Using the metadata repository rather than an LDAP repository or the

	standard SAS Enterprise Guide repository enables you to take full advantage of the security that can be implemented in SAS Management Console.
SAS Management Console	provides a single point of control for SAS administrative tasks by creating metadata definitions for a wide variety of resources, defining servers that are used to run stored processes and metadata repositories that are used to store metadata, and securing servers, stored processes, and the underlying data that is accessed or created by stored processes.
SAS Information Delivery Portal	enables report writers to share reports and information securely and selectively throughout the enterprise. The SAS Information Delivery Portal provides a single point of access for all users in the organization while ensuring that sensitive data and applications are secured. The SAS Information Delivery Portal enables report consumers within your organization to view reports, access business documents, and launch Web-based applications.

There are two primary strategies for creating stored processes, registering the metadata for those stored processes, securing them, and preparing them for delivery. These strategies are illustrated in Figure 12.1 on page 228.

Figure 12.1 Stored Process Development and Delivery



This chapter provides information you will need for the following:

- “Understanding Stored Processes” on page 229
- “Developing Stored Processes with SAS Enterprise Guide” on page 233
- “Manually Developing Stored Processes” on page 238.

Understanding the different strategies for developing and delivering your stored processes and the advantages of each strategy will ensure that your organization takes

advantage of the strategy that best meets your needs. Strategies and procedures for delivering your stored processes using the SAS Information Delivery Portal are available at **`support.sas.com/rnd/web/portal/library/library20.html`**.

Understanding Stored Processes

Within the context of the SAS Intelligence Architecture, stored processes are one way to generate and analyze intelligent business data. More specifically, a stored process is a SAS program that is stored on a server and can be executed as required by requesting applications. Stored processes can access any SAS data source or external file and create new data sets, files, or other data targets supported by the SAS System. Additionally, stored processes can be used to generate various types of output that can be used for reporting, analytics, building Web applications, delivering result packages to clients or the mid-tier, and publishing results to channels or repositories.

Stored processes offer the following benefits:

- Centralized storage of SAS code, which simplifies the process of managing and sharing code.
- The ability to use parameters in the code, which increases your ability to reuse code.
- The ability to deploy SAS programs in variety of clients, for example, a client tool such as SAS Enterprise Guide or the SAS Information Delivery Portal, a Web service, a Java applet, or on an application server using Enterprise JavaBeans technology.
- The opportunity for users with little to no SAS programming experience to take advantage of content generated through SAS analytics.
- Improved security, which is accomplished by storing the code on a central server.
- The ability to limit who has permission to view or edit the original SAS code, while providing users with permission to execute the program.

Developing Stored Processes

Stored processes consist of two distinct parts: the SAS code and a stored process definition that resides in a metadata server. The basic steps for developing a stored process are

1. writing the stored process code (SAS code)
2. choosing or defining a server
3. registering the stored process metadata.

You can either create your stored processes using SAS Enterprise Guide or you can begin with traditional SAS code and then convert that code into a stored process. Table 12.1 on page 230 identifies the primary advantages of each of these strategies for developing your stored processes.

Table 12.1 Strategies for Developing Stored Processes

Advantages of Developing Stored Processes with SAS Enterprise Guide	Advantages of Developing Stored Processes Manually
□ Business analysts and developers who are not experienced SAS programmers can use the Create New Stored Process wizard, which simplifies the process of developing the stored process code and enables business analysts to concentrate on the analytical and statistical goals of the stored process rather than on the technical components of the SAS code. □ SAS Enterprise Guide makes it easy to parameterize your stored process code, which increases your ability to reuse code and also enables the code to be customized at run time. Stored process parameters typically correspond to macro variables that are included in the SAS code. □ You can embed libraries in your stored process code and perform library logic checking, which enables you to develop efficient and effective stored process code. □ SAS Enterprise Guide enables you to easily register your stored process in the SAS metadata repository when this repository is selected as the default repository for your SAS Enterprise Guide application. □ You can synchronize your stored processes with your stored process server, which ensures that your stored process server always accesses the most recently updated stored process code. □ You can publish and run your stored process code from within SAS Enterprise Guide.	□ Experienced SAS programmers can quickly convert existing SAS code into stored processes by changing hard coded variables to parameters and adding the necessary macro variables without having to use a graphical user interface. □ You can register stored processes in the metadata repository using the Stored Process Manager plug-in to SAS Management Console. □ The only software products that you need to manually create and register stored processes are Base SAS or a text editor, SAS Management Console, and the Stored Process Manager plug-in to SAS Management Console. □ The Stored Process Manager plug-in to SAS Management Console makes it easy to parameterize your stored process code, which increases your ability to reuse code and also enables the code to be customized at run time. Stored process parameters typically correspond to macro variables that are included in the SAS code.

For more detailed information about each of these strategies for creating and registering your stored processes, see “Developing Stored Processes with SAS Enterprise Guide” on page 233 and “Manually Developing Stored Processes” on page 238.

Types of Stored Process Output

In some cases, stored process output (or a "result") is delivered to the client application executing the stored process. In other cases, the output is generated only on the server. Stored processes can be used to produce four different types of output.

Table 12.2 Stored Process Output Types

Output Type	Description
None	The client receives no output from the stored process. The stored process is still able to create or update data sets, external files or other objects, but this output remains on the server.
Streaming	The client receives a data stream, such as an HTML page or XML document. The data stream can be textual or binary data and is visible to the stored process program. Streaming output is supported only on the stored process server. Stored processes executing on a workspace server cannot use streaming output.
Transient Result Package	The client receives a temporary package. The package can contain multiple entries, including SAS data sets, HTML files, image files, or any other text or binary files. The package exists only as long as the client is connected to the server. This result type is a convenient way to deliver multiple output objects (such as an HTML page with associated GIF or PNG images) to a client application. Transient result package output is available on both stored process and workspace servers, but the implementations differ.
Permanent Result Package	The client receives a package that is stored in a permanent location on a WebDAV server or in the server file system. The package is immediately accessible to the stored process client, but is also permanently accessible to any client with access to Web Distributed Authoring and Versioning (WebDAV) or the server file system. This result type is a convenient way to publish output for permanent access.

Streaming output is much more suitable for interactive use in the SAS Information Delivery Portal because streaming output is displayed directly in the browser window. Result packages are better for saving results and asynchronous execution because packages can be saved or published to a number of destinations unlike streaming output, which is not saved. For more information about stored process outputs, see "SAS Stored Processes" in the *SAS Integration Technologies: Developer's Guide* at **`support.sas.com/rnd/itech/library/library9.html`**.

Running Stored Processes

Stored processes can be run on either workspace servers or stored process servers. The type of server you use depends on the type of results that you want the stored process to return and on the security requirements associated with the data that is referenced or analyzed in the stored process. Table 12.3 on page 232 identifies the primary advantages associated with running stored processes on each type of server.

Table 12.3 Selecting a Server for Your Stored Processes

Advantages of Using a Workspace Server	Advantages of Using a Stored Process Server
A workspace server is a single-user server that runs under a user ID specific to the client (assuming it is not configured as a pooled server).	A stored process server is a multi-user server that runs under a fixed user ID that must act as a proxy for all clients.
A new server session is started for each stored process.	Stored process servers are more scalable than the workspace server because they are always load-balanced and because they are capable of simultaneously running multiple processes.
	A stored process server enables you to produce streaming results as well as result packages from your stored process.

Note: Because stored process servers use a multi-user login to authenticate, it is recommended that you define a separate stored process server for each content area within your organization. A content area describes a collection of data that is sensitive and should be accessed only by a specific group of users. For example, if you define content areas for Human Resources, Finance, and Sales, then you should configure a stored process server for each of these content areas. Each server should be associated with a group login on the metadata server that is created for the set of users who access the data in that content area. △

For more information about the types of output that stored processes can generate, see "Types of Stored Process Output" on page 230. For more information about security considerations associated with selecting a stored process server or a workspace server, see "Planning Security on Workspace and Stored Process Servers" in the "Security" chapter of the *SAS Integration Technologies: Administrator's Guide* at **`support.sas.com/rnd/itech/library/library9.html`**. For more information about defining additional stored process servers, see "Administering SAS Servers" in the *SAS Integration Technologies: Administrator's Guide* at **`support.sas.com/rnd/itech/library/library9.html`**.

Securing Stored Processes and Business Content

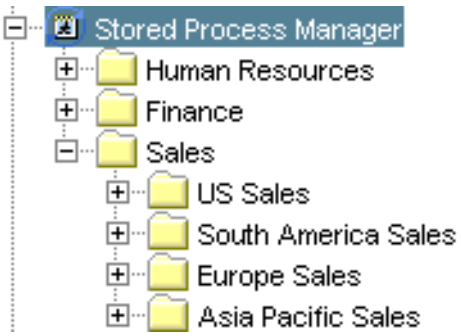
When developing and managing stored processes in the SAS Intelligence Architecture environment, it is important to consider how you want to secure

- the SAS code in a stored process
- the metadata definition that describes the stored process
- the underlying data that is accessed by the stored process code.

Typically, you will secure the SAS code (or stored process file) through the security layer for your operating system or through your file system. Additionally, if you save your stored process results to a mid-tier WebDAV server, then you can also take advantage of the WebDAV security.

Securing the metadata definition that describes the stored process is easier if you group your stored process definitions in folders within the metadata repository because stored processes inherit access controls and permissions from the folder in which they are stored. Therefore, it is recommended that you create a customized folder structure similar to the one shown in Display 12.1 on page 233 to organize your stored process definitions. This folder structure enables you to apply access controls to the folders rather than to each individual stored process.

Display 12.1 Stored Process Content Areas



It is recommended that you identify specific content areas for stored processes that will be used by groups of similar users or departments within your organization. For example, your organization might have stored processes that are associated with content that pertains to personnel data, financial data, and sales data. You can create folders within the metadata repository that correspond to these content areas. For example, you might want to create a parent folder for the entire organization and then create subfolders for Human Resources, Finance, Sales, and so on, and then create additional subfolders for the different sales regions. You could then set access controls for the group of users that needs access to those folders. For more information about defining user groups, see "Planning Your User Groups" on page 59. For more information about securing stored processes with access controls, see "Access Requirements for Working with Stored Processes" on page 66.

You can secure the underlying data that is accessed by stored process code in many ways. You can apply security through the

- operating system
- file system
- access controls placed on the library that is used (if the library is secured in SAS Management Console)
- data source control (such as DBMS credentials).

If you are running your stored processes on a stored process server or on a pooled workspace server, then a multi-user or puddle login is used to authenticate, and you must ensure that the login has the appropriate access to the data. If you are using a workspace server to run your stored processes, then you must ensure that each individual user has the appropriate access to the data.

Developing Stored Processes with SAS Enterprise Guide

Overview of SAS Enterprise Guide

SAS Enterprise Guide is a Windows application designed for statisticians, business analysts, and SAS programmers that enables them to analyze and produce intelligent business data. It provides wizard-driven access to much of the reporting, graphing, and analytic capabilities of SAS without the need to write SAS code. Business analysts, statisticians, and programmers within your organization can use SAS Enterprise Guide to author sophisticated stored processes that are fundamental in migrating SAS content

from the traditional SAS programs to the new SAS Intelligence Architecture. SAS Enterprise Guide enables you to

- seamlessly integrate the best of traditional SAS programming with analytic tasks through use of the Process Flow Builder
- have open access to SAS code developed with SAS Enterprise Guide, which facilitates code reuse and deployment
- access and manipulate data locally or on SAS servers
- integrate SAS Enterprise Guide with your SAS metadata repository, which enables you to secure your stored processes using access controls that can be implemented in SAS Management Console
- perform basic reporting and summaries
- perform complex data analyses
- utilize the highest quality SAS graphics
- export or publish results to SAS servers and other Windows or server-based applications
- publish to a stored process server for use in SAS Enterprise Guide or SAS Information Delivery Portal
- customize applications and Windows OLE automation using VB, C++, Visual Basic .Net, and C# .Net.

SAS Enterprise Guide enables you to select one of three different types of repositories as your active repository: Enterprise Guide repository, SAS metadata repository, and LDAP repository. In order to integrate SAS Enterprise Guide with the other SAS Intelligence Architecture products that SAS offers and in order to take full advantage of the security that SAS provides, you should select your metadata repository as your active repository. This ensures that when you create a stored processes the metadata that describes that stored process is automatically registered in your metadata repository. The metadata repository that you select can be either local or remote. SAS Enterprise Guide Administrator enables you to specify the active (or default) repository. For instructions on setting the metadata repository as the active repository in SAS Enterprise Guide, see "Integrating SAS Enterprise Guide and Your SAS Metadata Repository" on page 234.

In some cases when using SAS Enterprise Guide in the SAS Intelligence Architecture environment, the SAS Workspace Server or SAS Stored Process Server that you will use to run your stored process code resides on a remote machine in the back tier rather than on the client machine. In this scenario, SAS Enterprise Guide must be configured to use a remote workspace server or stored process server to fulfill client requests for SAS sessions. SAS Enterprise Guide Administrator enables you to define the remote server that you will use for running code in SAS Enterprise Guide. For detailed instructions, see "Identifying and Using a Remote Workspace Server or Stored Process Server" on page 237.

Integrating SAS Enterprise Guide and Your SAS Metadata Repository

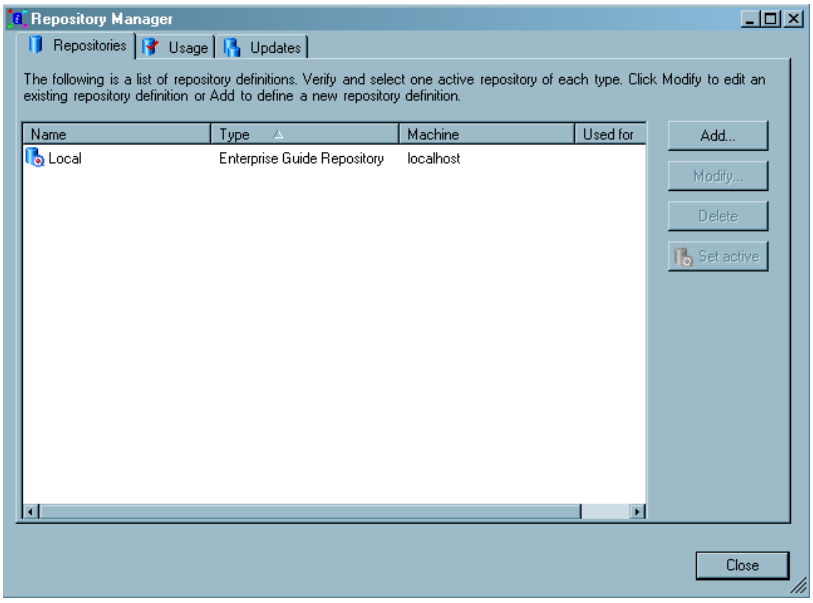
To specify your SAS metadata repository as the active repository in SAS Enterprise Guide:

1. Open SAS Enterprise Guide Administrator.
2. Select

   File ► Manage Repositories

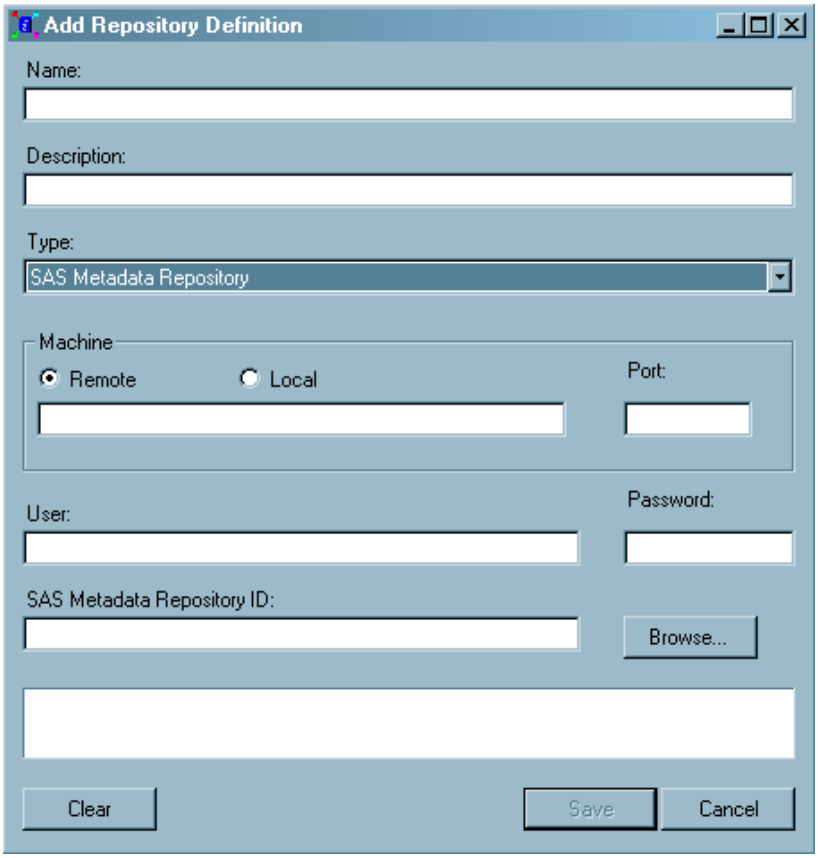
   The Repository Manager window displays.

Display 12.2 Repository Manager Window–Repositories Tab



3 From the Repositories tab in the Repository Manager window, click Add. The Add Repository Definition window displays.

Display 12.3 Add Repository Definition Window in SAS Enterprise Guide Administrator



4 Complete the fields as follows:

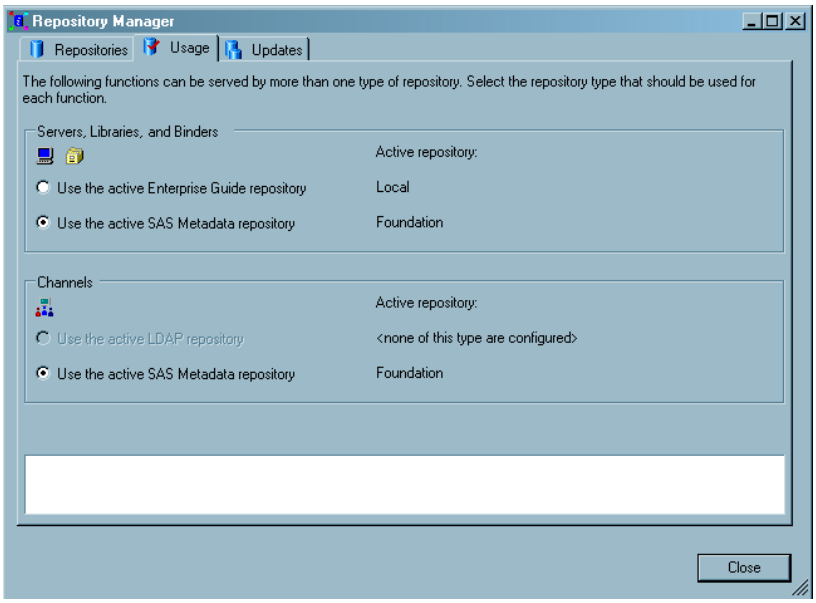
Name	identifies the name of the repository that you want to add. Enter the exact name of the metadata repository.
Description	provides a text description of the repository. This field is optional.
Type	specifies the type of repository. Select **`SAS Metadata Repository`**.
Machine	identifies the name, location, and port number that are associated with the metadata server that contains the repository. Machine parameters include the following:
	machine location — identifies whether the metadata server host is **`Local`** or **`Remote`**. Select the appropriate location for the machine on which the repository resides.
	machine ID — identifies the metadata server host on which the repository resides. Enter either the host name or the IP address for the machine.
	Port — identifies the port number that is assigned to the metadata server host on which the repository resides. Enter the appropriate port number.
User	identifies the user ID that is required to log on to the metadata server on which the repository resides. Enter the appropriate user ID.
Password	identifies the password for the user ID that is required to log on to the metadata server on which the repository resides. Enter the appropriate password.
SAS Metadata Repository ID	identifies the ID that is assigned to the metadata repository. The format of the repository ID is *A00000001.nnnnnnnn*. You can either enter the appropriate ID or click Browse to display a list of the repositories that are defined on the metadata server. When you select a repository from the list, the ID is automatically entered into the **`SAS Metadata Repository ID`** field.

Note: The Metadata Manager plug-in to SAS Management Console enables you to view all of the metadata repository parameters that you need to specify. To view the metadata repository parameters, use the Metadata Manager plug-in to identify the appropriate metadata server, and then display the properties of the metadata repository. For more information on using the Metadata Manager, see the Help that is available within SAS Management Console. △

5 Click Save.

6 Select the Usage tab in the Repository Manager window. The Usage tab enables you to specify the active repository to use when working with channels, servers, libraries, and binders. If a repository type has not been defined, any radio buttons corresponding to that type are disabled.

Display 12.4 Repository Manager Window–Usage Tab



7 Select **`Use the active SAS Metadata repository`** in both the **`Servers, Libraries, and Binders`** group box and the **`Channels`** group box to specify that you want to use the metadata repository rather than the standard SAS Enterprise Guide repository, and then click [Close].

Identifying and Using a Remote Workspace Server or Stored Process Server

To use a remote workspace server or stored process server to run the stored process code that you generate with SAS Enterprise Guide:

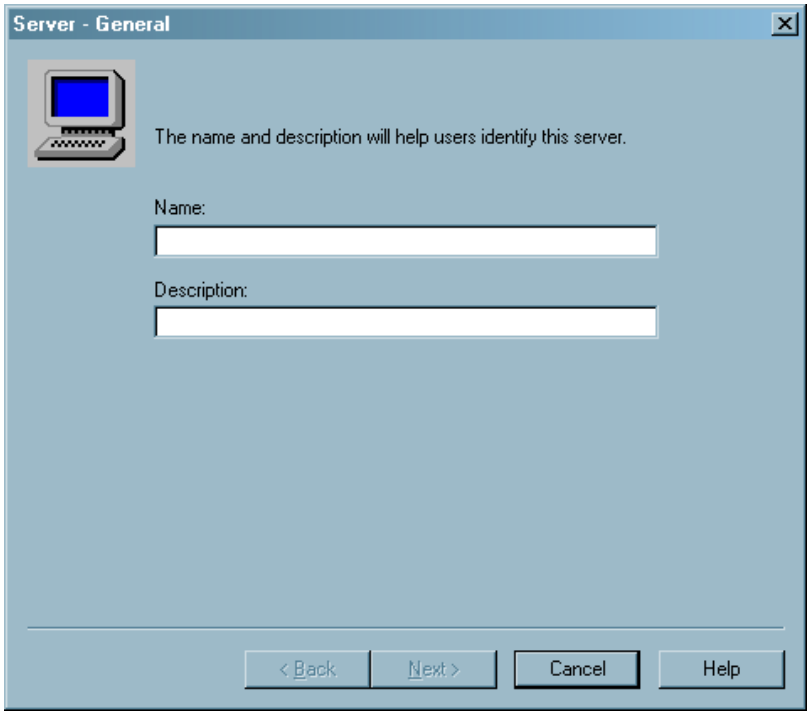
1 Open SAS Enterprise Guide Administrator.

2 Select the following and then click [OK]:

[File] ► [New] ► [Server]

The Server General window in the New Server wizard displays.

Display 12.5 New Server Wizard–Server General Window



3 Complete the Server wizard, selecting **IOM** as the type of server connection. It is recommended that you test the server before you click Finish. For more information on adding a server with the Server wizard, see "Managing Servers" in the SAS Enterprise Guide Administrator Help.

4 Close SAS Enterprise Guide Administrator.

5 Open SAS Enterprise Guide and create your code.

6 Choose the appropriate remote server by selecting the following, and then click OK:

Code ► Select Server

7 Select

Code ► Run

Manually Developing Stored Processes

SAS Enterprise Guide is not required to develop or modify stored processes. For SAS programmers and business analysts who are experienced in developing SAS code, it is sometimes faster and easier to manually develop SAS code and use the Stored Process Manager plug-in to SAS Management Console to register or modify a stored process.

You can manually create stored process programs in traditional interactive SAS or any text editor and save them directly to an appropriate location on the SAS server. When you manually create your stored processes, you are responsible for generating

- all source code, including the standard begin and end macros
- library definitions
- parameter macro variables.

For more information about manually developing stored processes, see "SAS Stored Processes" in the *SAS Information Technologies: Developer's Guide* and "SAS Stored Processes" in the *SAS Information Technologies: Administrator's Guide* at **`support.sas.com/rnd/itech/library/library9.html`**.

The Stored Process Manager plug-in enables you to register new stored processes in the metadata repository and modify stored processes that have already been registered. It can be used to create or modify the same stored process metadata that can be created by SAS Enterprise Guide. For more information about using the Stored Process Manager plug-in, see "Registering a New Stored Process" in the *SAS Information Technologies: Administrator's Guide* at **`support.sas.com/rnd/itech/library/library9.html`** or see the Stored Process Manager Help.

CHAPTER

13

Preparing SAS Enterprise Miner for Use

Overview of Preparing SAS Enterprise Miner for Use

SAS Enterprise Miner 5.1 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 secure a desired effect.

There are two ways to deploy SAS Enterprise Miner:

- *Complete client.* 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 complete-client installation is appropriate for single-user configurations.
- *Thin client and application server.* The SAS Enterprise Miner thin-client installation includes only the Java files needed for display. All other files are installed on a SAS Enterprise Miner application server. Thin clients connect directly to the application server process, which handles all communication with the SAS Metadata Server and the SAS Workspace Server. This installation 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 by using Java Web Start, which enables you to deliver thin-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.

Note: For more information about how to deploy SAS Enterprise Miner, see the SAS Enterprise Miner Help. △

During a planned SAS installation, a SAS Metadata Server, an Object Spawner, and a SAS Workspace Server are defined and configured and are available for use by SAS Enterprise Miner. The deployment process also creates a foundation metadata repository and some initial users (see Chapter 7, "Installing and Configuring Your Software," on page 123 for more information).

In addition, you must perform these tasks:

- Create additional SAS Enterprise Miner users (see "Planning Your Users" on page 53)
- Complete the SAS Enterprise Miner configuration wizard.
- Secure the metadata definitions for projects and models that are created by SAS Enterprise Miner users.

And, you might need to perform these tasks:

- Set extended attributes on the SAS Workspace Server.
- Add variables to the shell scripts used for UNIX systems.
- Configure the Apache Tomcat HTTP Server for use with SAS Enterprise Miner.

Configuring SAS Enterprise Miner

The SAS Enterprise Miner configuration wizard must be completed on these computers:

- each computer that hosts a complete-client installation
- the computer that hosts the SAS Enterprise Miner application server for thin-client installations deployed, for example, with Java Web Start.

Launching the SAS Enterprise Miner Configuration Wizard

To launch the SAS Enterprise Miner configuration wizard, complete the tasks that are applicable for your operating system.

Table 13.1 Tasks for Launching the SAS Enterprise Miner Configuration Wizard in Different Operating Systems

To launch the configuration wizard from here	Perform these tasks
UNIX	Execute the **`emconfigure`** script by entering **`[emroot]/bin/emconfigure`**
Windows	Select Start ▶ Programs ▶ SAS ▶ SAS Enterprise Miner 5.1 then click the Configure* icon in the SAS Enterprise Miner Launch Control browser window.

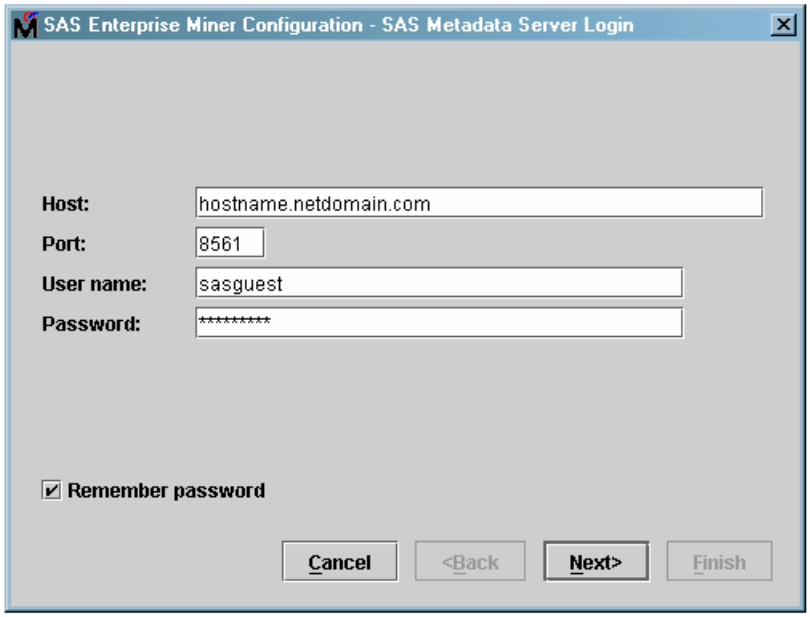
* After initial configuration, the *Configure Enterprise Miner on this computer* icon is relabeled as *Reconfigure Enterprise Miner on this computer*, as shown in the following illustration.

Display 13.1 SAS Enterprise Miner Launch Control Window as It Appears After Initial Configuration on a Windows System for a Complete-Client Installation

Completing the SAS Enterprise Miner Configuration Wizard

Here are the steps that you must follow to complete the SAS Enterprise Miner configuration wizard:

1 In the first wizard window, you enter the login information for the SAS Metadata Server.



a Enter the network address of the **Host** on which the SAS Metadata Server is running.

b Enter the **Port** on the host on which the SAS Metadata Server is listening. The default port is 8561.

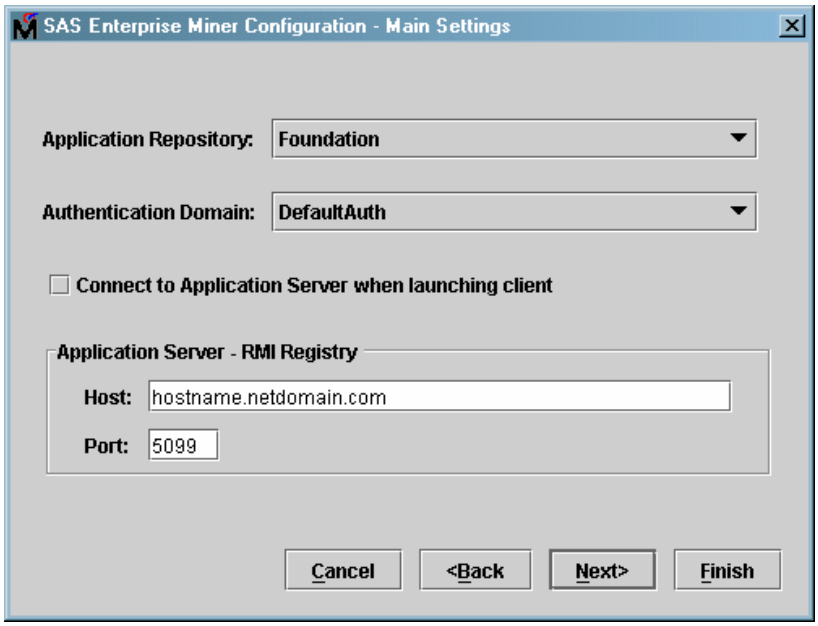
c Enter the **User name** of the user of this installation of SAS Enterprise Miner. The user must have a login for the specified SAS Metadata Server. For a Windows platform, if SAS Enterprise Miner is installed on a different computer than the SAS Metadata Server, then enter the fully qualified domain name. For example, **mydomain\sasguest**.

d Enter the **Password** for the user name that you entered.

e Select the **Remember password** check box to save the user's password in encoded format with the SAS Enterprise Miner login properties. If you do not select this check box, then the user will be prompted to supply a password each time that the user accesses the specified SAS Metadata Server through SAS Enterprise Miner.

When you are done, click Next.

2 Enter the information needed to validate the user's login on the SAS Metadata Server. If this is a thin-client installation, then also enter information about the SAS Enterprise Miner application server.



a If the SAS Metadata Server that you specified on the first wizard window has multiple repositories, then, from the **Application Repository** drop-down list, select the SAS Metadata Repository that you want to store this SAS Enterprise Miner configuration information.

b Select the **Authentication Domain** that the user specified on the first wizard window. Typically, all SAS Enterprise Miner users are assigned to the same authentication domain.

c Use these guidelines to determine whether to select the **Connect to Application Server when launching client** check box:

- If this SAS Enterprise Miner installation will connect to a SAS Workspace Server defined in a metadata repository that is dedicated to a SAS Enterprise Miner application server that is running on another computer, then select this check box.
- If this SAS Enterprise Miner installation is using a metadata repository that is defined on the same computer, then do not select this check box. Do not select the check box even if you will allow other clients to use the repository on this computer, although, in that case, you must provide the **Host** and **Port** numbers.

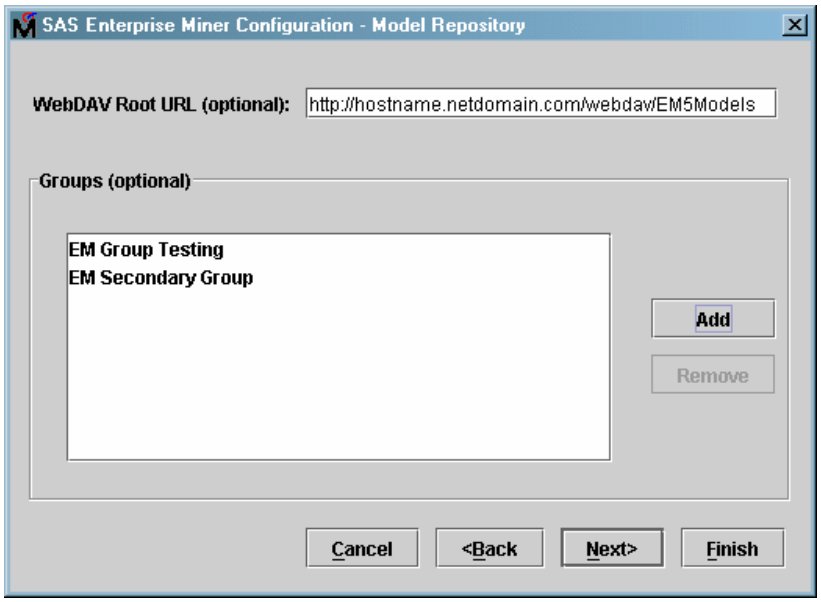
d Enter the **Application Server, Host**, which is the network address of the computer on which the SAS Enterprise Miner application server is or will be running.

e Enter the **Application Server, Port**, which is the port on which the RMI registry is or will be listening. This port is typically 5099.

Note: Remote method invocation (RMI) means that the server is listening for requests for a specific invocation of classes that should be remotely instantiated. △

When you are done, click Next.

3 For thin-client installations, you can enter modeling parameters. All items on this window are optional.



a Users can save model results for later examination by the SAS Enterprise Miner Model Viewer. To store the model, you write a package of data to a WebDAV area that is managed by an HTTP server (see the SAS Enterprise Miner 5.1 Help for more information). The **WEBDAV Root URL (Optional)** field contains the URL of the WebDAV storage application. The URL should contain a port number when applicable, such as **http://hostname.netdomain.com:8080/webdav/EM5Models**.

b Users can group saved models for the operational convenience of investigation and study. To add a group, click the Add button and specify the name and description of the group. To delete a group, select it in the list and then click Remove.

When you are done, click Next.

4 Verify the settings on Summary window, then click Finish.

Customizing SAS Workspace Server Settings

You can set extended attributes on each logical SAS Workspace Server that you are using at your site.

Table 13.2 Optional SAS Enterprise Miner Extended Attributes for the SAS Workspace Server

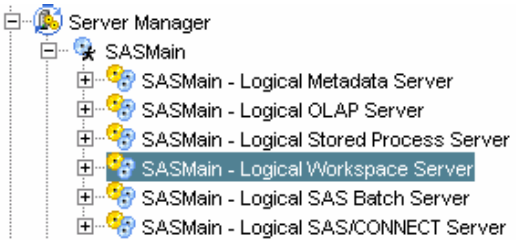
Extended Attribute	Description
EM_PROJECT_ROOT	The path on the server in which projects should be stored. The default is to let users specify any path when they create projects; however, in a production environment, the best practice is to specify the path.
EM_ENFORCE_PROJECT_LOCATION	Indicates whether to enforce the specified project root path. As a best practice in a production environment, set this attribute to `Y` to prevent users from changing the path.
EM_SERVER_INIT_CODE	The path on the server from which SAS Enterprise Miner retrieves SAS code to execute when a project is opened. For example, you might write code to force users to log their use into an audit trail.
EM_MAX_CPU	The number of CPUs to exercise during a process flow. The default is -1, which enables each project to use all available processors. To restrict the number of CPUs, enter an integer. For example, if you limit the number of CPUs that each project can use, then more projects can execute at the same time with less stress on the computer.

To add these attributes to a logical SAS Workspace server, complete these steps in SAS Management Console:

1. Use your metadata profile to log in 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. Select a SAS application server context that contains a logical SAS Workspace Server.
4. Select the name of the logical SAS Workspace server.

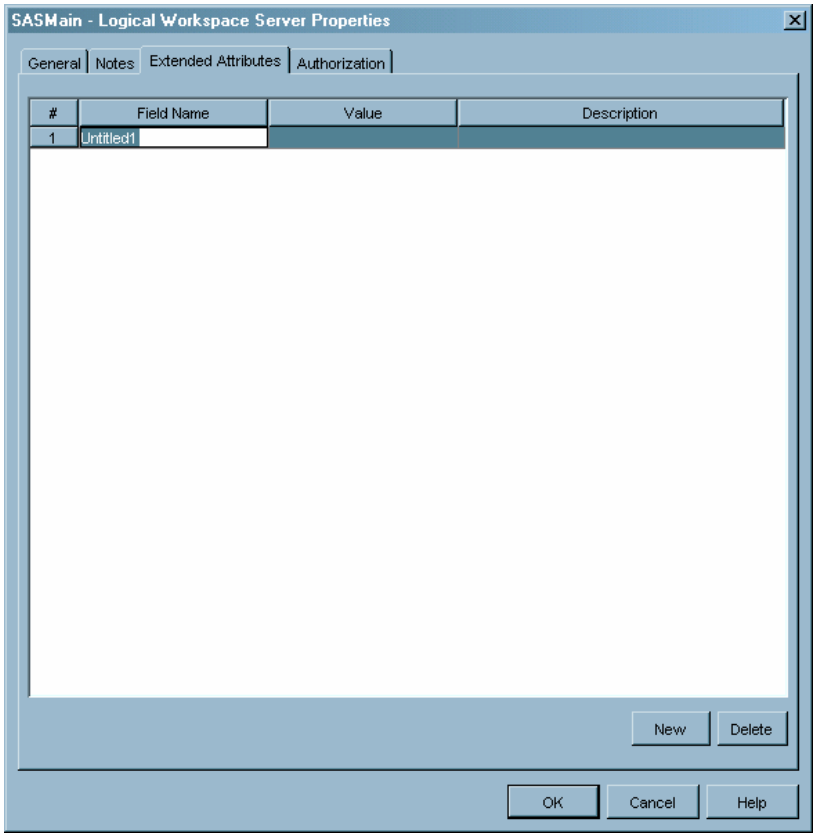


5. Select

   File ► Properties

6. In the Properties dialog box, select the Extended Attributes tab.

7 Click the New button. The first line is highlighted and **Untitled1** appears in the **Field Name** field.



8 Enter the information for each attribute that you want to set. After you complete each line, click New to insert a new line.

Table 13.3 Sample Values for SAS Enterprise Miner Extended Attributes on the SAS Workspace Server

Field Name	Sample Value*	Description
EM_PROJECT_ROOT	*d:\my\project\path*	Project tree storage path
EM_ENFORCE_PROJECT_LOCATION	Y	User cannot change the project storage path
EM_SERVER_INIT_CODE	*d:\my\server\initialCode.sas*	Path to server initialization code
EM_MAX_CPU	-1	Use all available processors

* User-specified values appear in italic.

9 When you are done, click OK to save the attributes and close the dialog box.

Setting Required Variables in UNIX Shell Scripts

For UNIX systems, there are shell scripts that must contain variables that identify the default Java and SAS Enterprise Miner directory paths. The variables are EM_HOME and JAVA_HOME. Here are their default settings:

EM_HOME=/opt/SAS/SASEMiner/5.1/EM51

JAVA_HOME=/opt/java1.4.1

If these variables are not already present, then add them to the following shell scripts.

Table 13.4 UNIX Shell Scripts

Shell Script	Description
$EM_HOME/bin/em	Java client script
$EM_HOME/bin/emhelp	Java client script
$EM_HOME/bin/startserver	Java application server (mid-tier) script
$EM_HOME/bin/emconfigure	SAS Enterprise Miner configuration wizard script

Also, set the execute permission for each script file.

Customizing the Apache Tomcat HTTP Server

The HTTP server recommended for use with SAS Enterprise Miner 5.1 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 EM Model Viewer application for Tomcat.

Note: Servers other than Tomcat 4.1.18 might have different instructions. △

Note: These instructions assume that the default Windows and UNIX installation paths were used. The default path for Windows is **c:\Program Files\SAS\SASEminer\5.1\EM51**. The default path for UNIX is **/opt/SAS/SASEminer/5.1/EM51**. △

1 To enable the Tomcat write functions, remove the comment delineators from the following block of code in the WebDAV configuration XML file. For example, on Windows the file is typically found in **c:\Program Files\Apache Group\Tomcat 4.1\webapps\webdav\WEB-INF\web.xml**.

   Note: If the comment delineators that are shown in the following code are not present, then the Tomcat WebDAV write functions are already enabled. △

```
<!--
    <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 UNIX, you can launch the tool from the machine where Tomcat is installed by entering **http://localhost:8080/admin** into a Web browser (like Mozilla). (From a remote machine, enter the network address instead of **localhost**.) 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

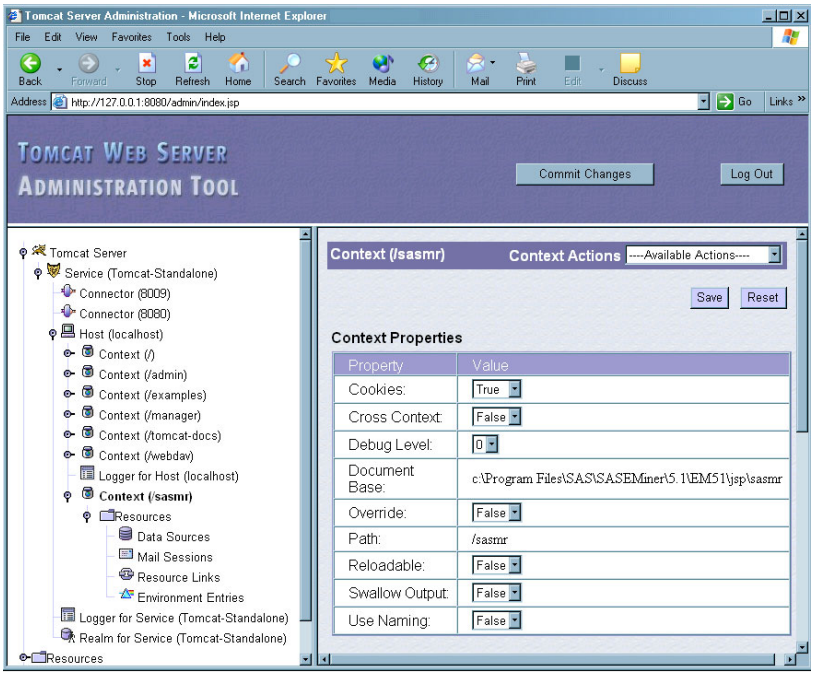
Tomcat Server ► Service ► Host

b In the display area to the right of the navigation tree, select **Create New Context** from the Host Actions drop-down list.

c In the **Document Base** field, enter the path to the viewer. Typically, the path is **c:\Program Files\SAS\SASEMiner\5.1\EM51\jsp\sasmr**

d In the **Path** field, enter the context path. Typically, the path is **/sasmr**.

e Click the Save button. The context is added to the selected Host as shown in the following illustration.



As an alternative to using 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 5
    <Context path=''/sasmr''
      docBase=''c:\Program Files\SAS\SASEMiner\5.1\EM51\jsp\sasmr''
      crossContext=''true''
      debug=''0''
      reloadable=''true''>
    </Context>
```

Note: If the SAS Enterprise Miner installation path is other than that shown, change the location of the Document Base (**docBase=** value). △

3 Use the Tomcat Web Server Administration Tool to define the SAS Enterprise Miner Java Web start client deployment to Tomcat.

a In the navigation tree on the left, select

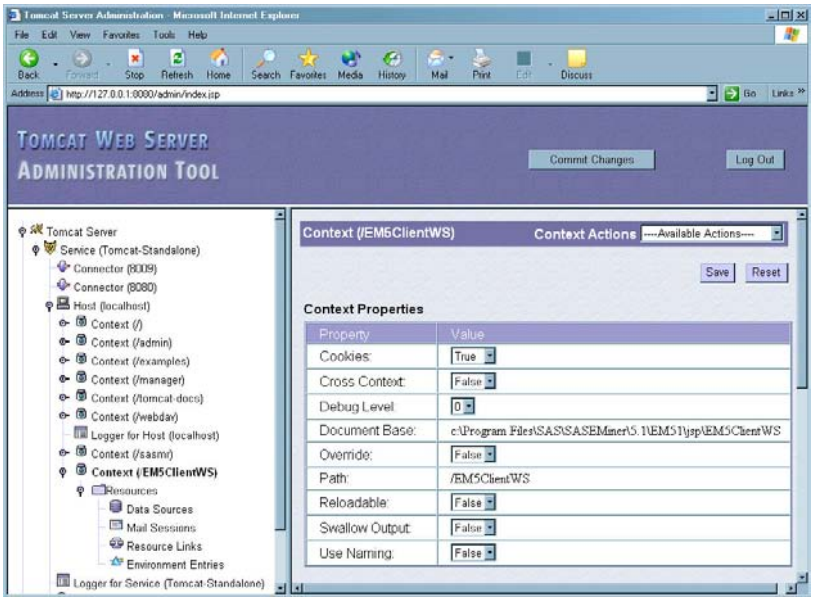
Tomcat Server ► Service ► Host

b In the display area to the right of the navigation tree, select **Create New Context** from the **Host Actions** drop-down list.

c In the **Document Base** field, enter the path to the viewer. Typically, the path is **c:\Program Files\SAS\SASEMiner\5.1\EM51\jsp\EM5ClientWS**

d In the **Path** field, enter the context path. Typically, the path is **/EM5ClientWS**.

e Click the Save button. The context is added to the selected Host as shown in the following illustration.



As an alternative to using the Tomcat Web Server Administration Tool, you can manually add the following context coding beneath the EM Model Viewer in the Tomcat server configuration XML file:

```
<!-- EM Model Repository 55
    <Context path=''/EM5ClientWS''
      docBase=''c:\Program Files\SAS\SASEMiner\5.1\EM51\jsp\EM5ClientWS''
      crossContext=''true''
      debug=''0''
      reloadable=''true''>
    </Context>
```

Note: If the SAS Enterprise Miner installation path is other than that shown, change the location of the Document Base (**docBase=** value). △

4 If you used the Tomcat Web Server Administration Tool, complete these tasks to exit the application:

a Click the Commit Changes button.

b Click the Log Out button.

5 Customize the members **c:\Program Files\SAS\SASEMiner\5.1\EM51\jsp\EM5ClientWS*.jnlp** and change all occurrences of the example network address and HTTP server port from **mycomputer.mydomain.com:8080** to the proper network address and HTTP server port number of the computer on which the SAS Enterprise Miner Java application server is installed. There are two occurrences in **em5clientws.jnlp** and one occurrence each in **em5static.jnlp**, **em5un3rd.jnlp**, and **em5jhall.jnlp**.

6 Stop and restart the Tomcat HTTP server to apply the changes.

7 Access the SAS Enterprise Miner client that was deployed by using Java Web Start at **`http://mycomputer.mydomain.com:8080/EM5ClientWS`**.

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 these metadata objects:

- 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 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 "Planning Your Access Controls" on page 62. △

Note: Currently, you cannot secure SAS Enterprise Miner models. △

Securing Access at the SAS Enterprise Miner Folder Level

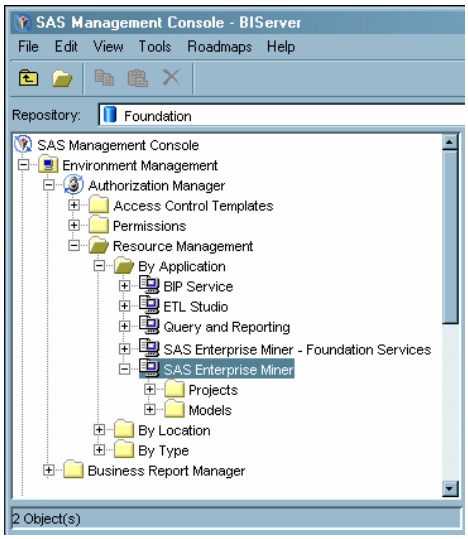
To access the Authorization tab for the **`SAS Enterprise Miner`** folder in the SAS Management Console navigation tree, complete these steps:

1 Use your metadata profile to log in 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

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 UserA, then UserA will not be able to see any projects in SAS Enterprise Miner, unless you explicitly grant ReadMetadata permission to UserA on the **Projects** folder (see "Securing Access at the Projects Folder Level" on page 253).

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:

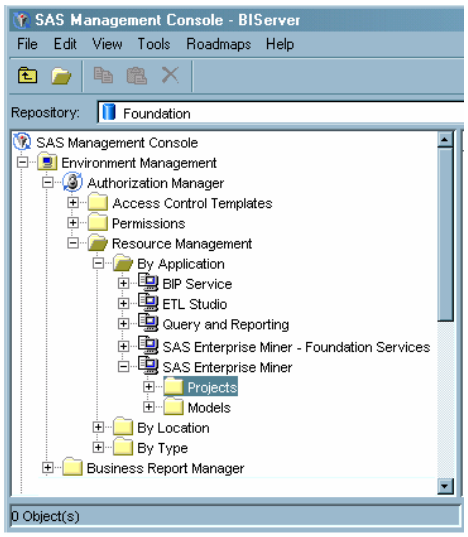
1 Use your metadata profile to log in to the SAS Metadata Server that contains the SAS Metadata Repository connection that you want to use.

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 UserA, then UserA 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 in to the SAS Metadata Server that contains the SAS Metadata Repository connection that you want to use.

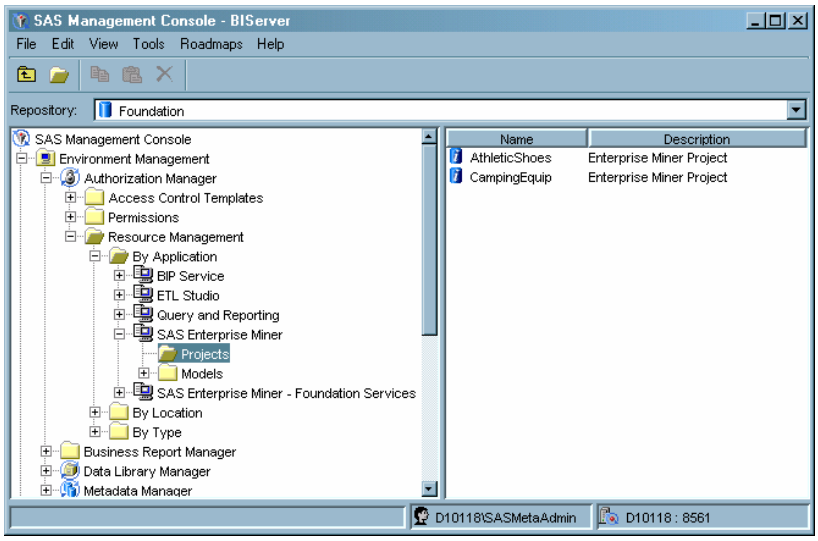
2 Select

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 UserA, then UserA will not be able to see the **CampingEquip** project in SAS Enterprise Miner. If you grant ReadMetadata but deny WriteMetadata, then UserA will be able to open the project but will not be able to save changes or delete it.

Note: If you have been denied ReadMetadata permission to the **Projects** folder that contains the project that you want to use, then you will not be able to physically navigate to the project even if you have been explicitly granted ReadMetadata permission to the project. △

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:

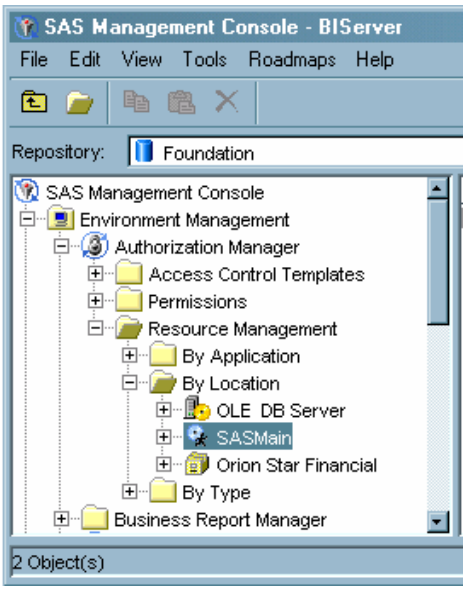
1 Use your metadata profile to log in to the SAS Metadata Server that contains the SAS Metadata Repository connection that you want to use.

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.



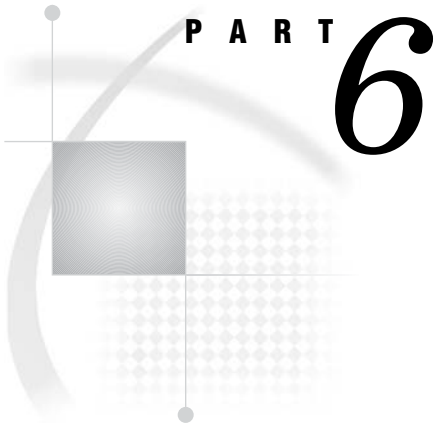
4 Select

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 UserA, then UserA will not have access to any projects associated with that server.



PART 6

Advanced Topics

CHAPTER

14

Configuring SAS Servers for Better Performance

Overview of Configuring SAS Servers for Better Performance

SAS Integration Technologies supports a load-balancing feature that enables you to scale and improve the performance of your system. It can be used with both workspace servers and stored process servers.

After a typical project install, your workspace server setup is very simple. You have an object spawner that is listening for requests for a workspace server. When such a request arrives, the object spawner creates a new workspace server process. This workspace server process handles the client's request and then terminates.

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 260.

The situation with stored process servers is a little different. 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 project install. 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 264.

You can scale your system up to some extent by adding MultiBridge connections to the existing stored process server. However, at some point, you will have 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 266.

Note: All of the servers in a load balanced cluster must belong to the same SAS authentication domain. △

Load Balancing Workspace Servers for Desktop Applications

Users of desktop applications, such as SAS ETL Studio, can place a heavy load on a workspace server. For example, in the case of SAS ETL 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 a workspace server on that host, and balance the workload across your new and old servers.

Installing the Software

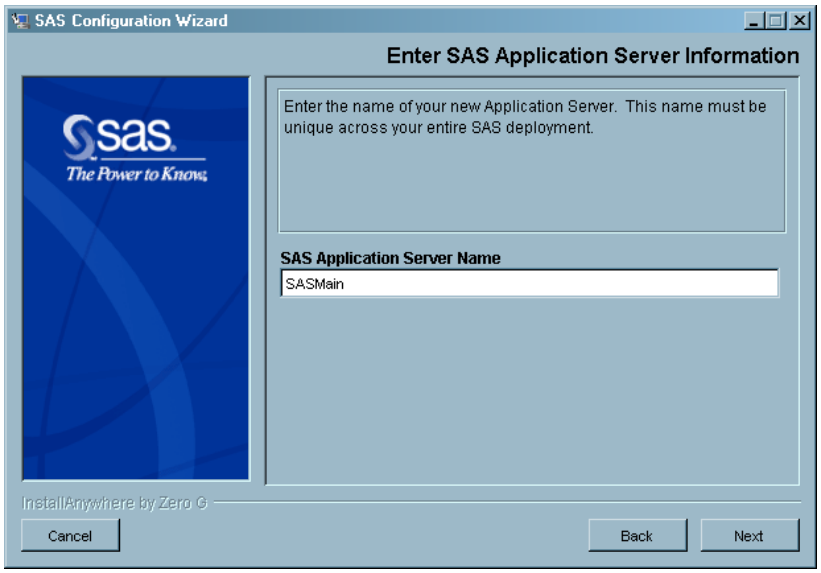
As when you first set up your system, you use the SAS Software Navigator to install your software. Just let your SAS representative know that you want to add a new host to your intelligence system and that you want to run a second workspace server on that host. Your SAS representative will be able to get you the software you need—SAS 9.1 Foundation and SAS Management Console—and to prepare the planning file you need for the installation.

For information on how to perform a project installation using the SAS Software Navigator, see “Installing Software on a Machine” on page 130.

Configuring the Workspace Server and Object Spawner

After you have installed SAS 9.1 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 14.1 Enter SAS Application Server Information Window



The assumption behind this question 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.

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 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**.

Follow these instructions instead:

1 Select "+" to expand the Server Manager node. Fully expand all three levels of **SASMain**.

2 Highlight the 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** "*path-to-config-dir***Lev1\SASMain\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 you have supplied, and click Finish. You will see a new workspace-server icon appear 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.

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 detailed in the sections "Converting the Logical Workspace Server to Load Balancing" on page 262 and "Setting Load Balancing Parameters for Each Workspace Server" on page 263.

Converting the Logical Workspace Server to Load Balancing

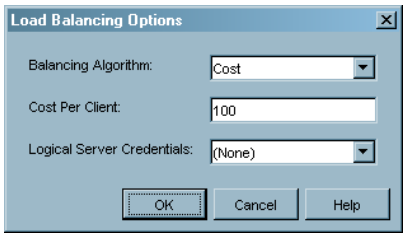
After you have defined your new workspace server and object spawner in the metadata, you can use SAS Management Console to convert your logical workspace server to load balancing. During this process, you set some options that apply to all of the physical workspace servers in the logical workspace server.

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 will appears.



2 Set the parameters in this dialog box using the explanations in the following table, and click OK.

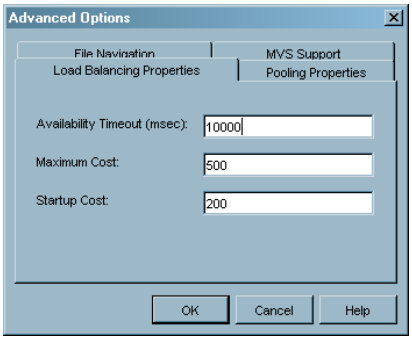
Table 14.1 Load Balancing Options

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 startup 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 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	The credentials used for communication between the object spawners handling the load balancing. 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:

1. In SAS Management Console, right-click the icon for the physical 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.



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 14.2 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 required for a server to start or the time required for a running server to become available.
Recycle Activation Limit	Specifies the number of times a connection to the server is reused before it is disconnected (recycled). If the value is 0, then there is no limit to the number of times a connection to the server can be reused. This field is optional.
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 startup cost of the server. When a request is made to the load balancer, the load balancer assigns this startup cost value to inactive servers. A new server is not started unless it is determined that its cost (the startup 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.

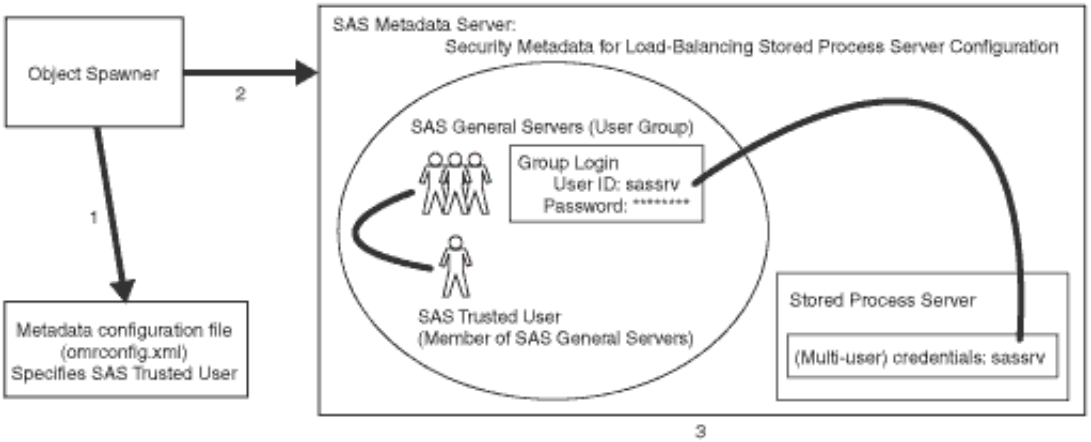
You can now start your object spawner.

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 14.1 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:

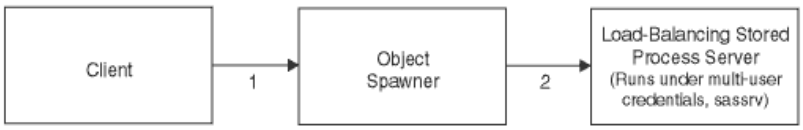
1. When the spawner is started, it reads a metadata configuration file named **`omrconfig.xml`** that contains information required to access the metadata server. This metadata configuration file specifies
   - the location of the metadata server
   - the user ID that the spawner will use to connect 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 14.2 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 266.

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 system 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 you need—SAS 9.1 Foundation and SAS Management Console—and to prepare the planning file you need for the installation

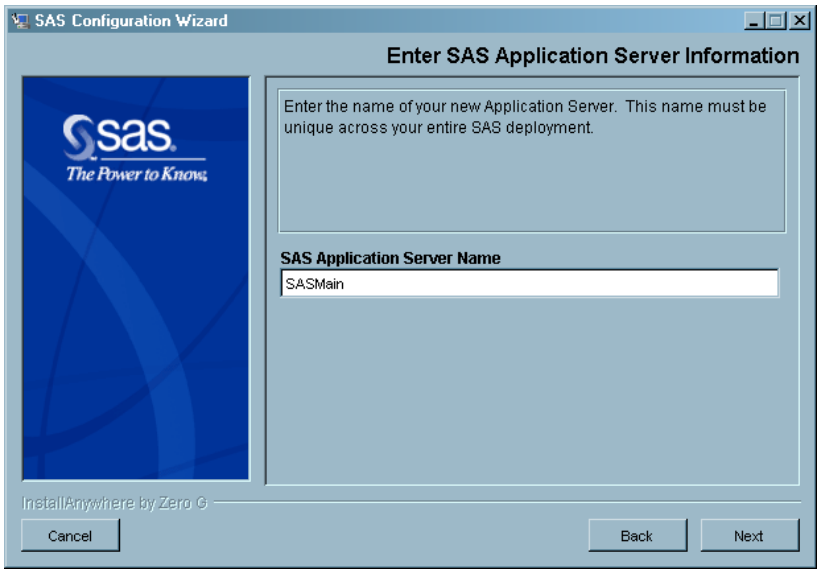
For information on how to perform a project installation using the SAS Software Navigator, see "Installing Software on a Machine" on page 130.

Configuring the Stored Process Server and Object Spawner

After you have installed SAS 9.1 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 14.2 Enter SAS Application Server Information Window



The assumption behind this question 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 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 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.

In the section "Edit the Properties of the Stored Process Server Component," replace steps 1 to 4 with the following instructions:

1. Select "+" to expand the Server Manager node. Fully expand all three levels of SASMain.
2. Highlight the **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\SASMain\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. You will see a new stored process server icon appear 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**" for 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 268 and "Setting the Load Balancing Properties for Each Stored Process Server" on page 270.

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.

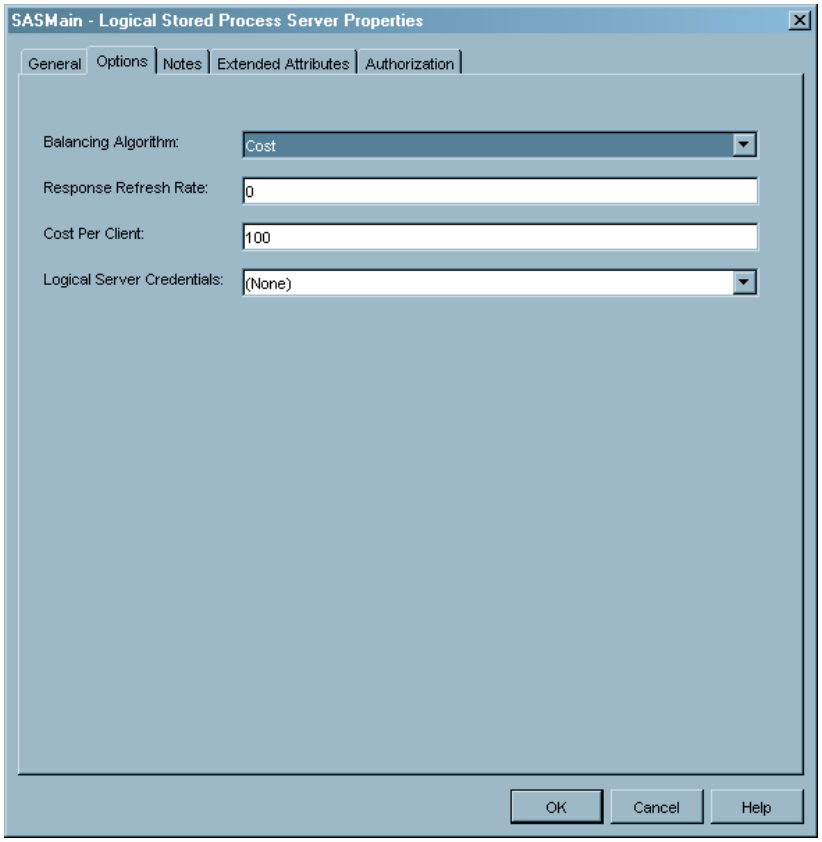
   a Create a network account for the SAS General Server User (**sassrv**). If you are creating a Windows domain account, be sure to grant the user the user right "Log on a batch job."

   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 Options tab. This tab contains the parameters that you can set.



4 Set these parameters using the explanations in the following table, and click OK.

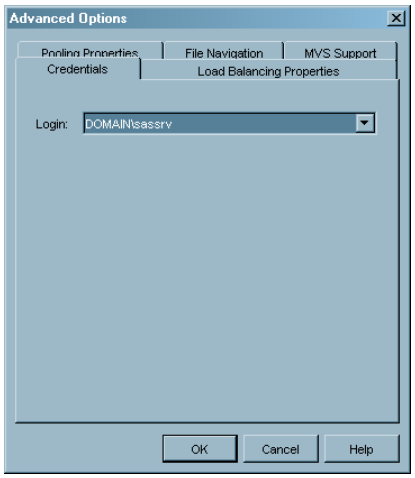
Table 14.3 Load Balancing Parameters

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 startup 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 Adminstrator'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`**.
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.

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. To set these properties:

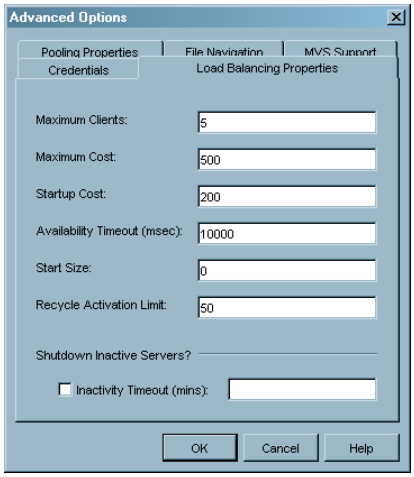
1. In SAS Management Console, right-click the icon for the stored process server, and select **`Properties`** from the pop-up menu that appears. A Properties dialog box appears.
2. Select the Options tab.
3. Click Advanced Options. An Advanced Options dialog box appears, and the Credentials tab displays.



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 14.4 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. Check 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.

CHAPTER

15

Promoting and Replicating Metadata

Overview of Promoting and Replicating Metadata

In a SAS ETL Studio environment, development, test, and production metadata repositories contain the same metadata at different points in their development cycle. If the contents are copied without any changes, then the process is referred to as replication. If the copying process includes the ability to change metadata values, then the process is called promotion.

You can only promote or replicate between servers that are running on the same platform. For example, promotion between two Windows servers is allowed, but promotion between a UNIX server and a Windows server is not permitted.

In addition, you should only promote or replicate foundation and custom repositories. You should not promote or replicate project repositories.

Note: For more information about promotion and replication, see the *SAS Management Console: User's Guide*. △

Preparing for Replication and Promotion

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. The SAS Metadata Server is one of four SAS application server components that are Integrated Object Model (IOM) Servers.

Note: For more information about SAS application servers, see Chapter 2, "Understanding the SAS Application Servers," on page 19. △

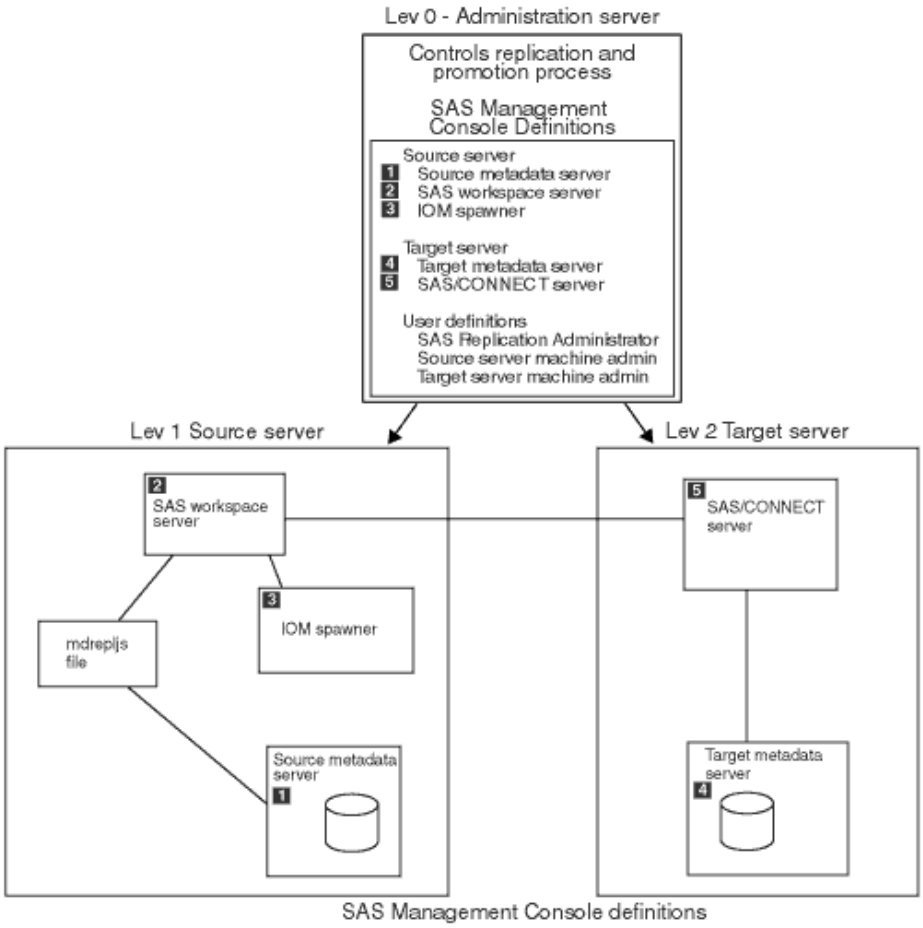
Each promotion or replication job uses three SAS Metadata Servers, one at each level in your SAS configuration environment (see Appendix 1, "Understanding the SAS Configuration Environment," on page 309):

- the administration metadata server from which the replication or promotion job is run. The production level identified as Lev0 represents the administration metadata server in the metadata and in the directory structure that the SAS Configuration wizard creates.
- the source metadata server from which metadata is promoted or replicated. The production level identified as Lev1 represents the source metadata server in the metadata and in the directory structure that the SAS Configuration wizard creates. Lev1 is created by default when you run the SAS Configuration wizard. Typically, Lev1 is the production server and it is the first level that you create when you install your SAS software.
- the target metadata server to which the metadata is promoted or replicated. The production level identified as Lev2 represents the target metadata server in the metadata and in the directory structure that the SAS Configuration wizard creates. Typically, Lev2 is the test server.

The metadata representation contains all references to servers needed during the promotion and replication processes. The directory structure contains scripts and log files that the physical servers use.

The following figure shows the relationship between the major components involved in replication and promotion.

Figure 15.1 Replication and Promotion Components



An Overview of the Required Steps

Here are the major steps that you must perform in order to prepare your system for promoting and replicating metadata. This procedure assumes that you are promoting or replicating the default foundation repository created on the source metadata server:

1. Verify SAS 9.1 installations.
2. Configure the administration metadata server.
3. Create the SAS Replication Administrator operating system account.
4. Define the SAS Replication Administrator in SAS Management Console.
5. Define the source metadata server in SAS Management Console.
6. Add a SAS Workspace Server component to the source metadata server definition.
7. Define an IOM Object Spawner for the SAS Workspace Server component.
8. Define the target metadata server in SAS Management Console.
9. Add a SAS/CONNECT server component to the target metadata server definition.
10. Create the metadata access file on the source metadata server machine.
11. Configure and start the target metadata server.
12. Define and start the SAS/CONNECT server for the target metadata server machine.

Assumptions

The tasks that you are instructed to perform assume that these conditions are true:

- You have created Lev1 by using the SAS Configuration wizard, including completing the appropriate pre-installation checklist. Any additions or modifications to the checklist will be explained.
- The repository that you are migrating is the default foundation repository that you created on the source metadata server by following the steps provided in the **`instructions.html`** file that the SAS Configuration wizard displayed during the configuration process.
- You are using the Windows operating system. If you are using a different operating system, then you must adjust the instructions accordingly. For example, the instructions for creating a new user account apply to Windows. UNIX and z/OS will have different instructions for creating new user accounts.

Step 1. Verifying SAS 9.1 Installations

Verify that SAS 9.1 software is installed on the machines for both the source and target metadata servers. The installations must include SAS Integration Technologies software and SAS/CONNECT software. Make note of the directories where SAS is installed, because you will need to know the path for the SAS installation in later steps.

Step 2. Configuring the Administration Metadata Server

To configure the administration metadata server, you complete the SAS Configuration wizard by entering basically the same information that you entered to create the source metadata server, including using the same configuration name and the same paths. The modifications are listed in these steps:

1. On the Select Install Set window, select **`Custom`** and then choose to install only the metadata server.
2. On the Enter SAS Metadata Server Information window, enter **`8560`** instead of **`8561`** as the port number.
3. On the Advanced Properties Editor window, click [Edit Properties]. In the text file that opens, make these changes:

Table 15.1 Values for the Administration Metadata Server

Replace occurrences of these	With this value
8561	8560
Lev1	Lev0

4. Save the changes to the file and close it to return to the SAS Configuration wizard.
5. Click [Next] to display the Finish window, then click [Finish].
6. When the **`instructions.html`** file appears in your browser, close the window. Instead, you must follow the instructions in this chapter.
7. If SAS Management Console launches, exit the application.
8. Click [Done] to exit the SAS Configuration wizard.

If you elected to start the server as service, then the administration metadata server is started automatically.

On Windows systems, you can manually start the server by selecting the applicable start-up task from

Start ► Programs ► Sas ► name of your configuration directory

Step 3. Creating the SAS Replication Administrator Operating System Account

On the machine that is hosting the administration metadata server, create a user account for the SAS Replication Administrator user. The SAS Replication Administrator will manage the replication and promotion processes.

1. Create a new user account with the name **sasrpadm**, the description **SAS Replication Administrator**, and a password. Set these properties for the user:
   a. Deselect **User must change password at next logon**.
   b. Select **User cannot change password**.
   c. Select **Password never expires**.
2. Add the **sasrpadm** user to the SAS Server Users group in the operating system. The SAS Server Users group was created as part of the pre-installation procedure.
3. Stop the administration metadata server.
4. Add **sasrpadm** to the adminUsers.txt file. Enter the name in the form *hostname***sasrpadm**. The default Windows location for the adminUsers.txt file is **C:\SAS\9.1\Lev0\SASMain\MetadataServer**.
5. Restart the administration metadata server.

Step 4. Defining the SAS Replication Administrator in SAS Management Console

In SAS Management Console, you must create a user definition for the SAS Replication Administrator that you just created in the operating system. The SAS Replication Administrator will have three logins, one for each server. For each log in, the user ID must belong to an administrative user on the associated machine. Each administrative user must have the security permissions that are required to perform these tasks:

- write to the applicable directories on the associated machine
- stop, start, and pause servers on the associated machine.

Complete these steps to define the SAS Replication Administrator:

1. Use SAS Management Console to connect to the administration metadata server with the SAS Replication Administrator login (**sasrpadm**).
2. Follow the on-screen instructions to establish a foundation repository for the new administration metadata server.
3. In the SAS Management Console navigation tree, select

   Environment Management ► User Manager

4. Select

   Actions ► New ► User

5. On the General tab, enter **SAS Replication Administrator** as the name.
6. Click the Logins tab.

7 Click New and complete the New Login Properties dialog box to create the login for the administration metadata server. Do not specify an authentication domain.

8 Click New and complete the New Login Properties dialog box to create the login for the source metadata server machine. Assign a new authentication domain named **ReplicationSourceAuth**. On the source metadata server machine, the user ID that you enter must

- have write and modify permissions to the subdirectories beneath the **ReplicationWorkArea** directory on the source metadata server machine
- have write and modify permissions to the directory in which the metadata repository will reside
- be in the adminUsers.txt file.

9 Click New and complete the New Login Properties dialog box to create the login for the target metadata server machine. Assign a new authentication domain named **ReplicationTargetAuth**. On the target metadata server machine, the user ID that you enter must

- have write and modify permissions to the subdirectories beneath the **ReplicationWorkArea** directory
- have write and modify permissions to the directory in which the metadata repository will reside
- be in the adminUsers.txt file.

Note: For detailed help in creating a new user, click Help in the User Manager application. △

Step 5. Defining the Source Metadata Server in SAS Management Console

In SAS Management Console, use the New Server wizard to define the source metadata server in the foundation metadata repository for the administration metadata server. Complete these steps to launch the wizard:

1 In the SAS Management Console navigation tree, select

Environment Management ► Server Manager

2 Select

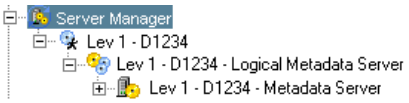
Actions ► New Server

When defining the server in the New Server wizard, specify these properties:

Server type	SAS Application Server
Name	Lev 1 - *hostname* (for example, **Lev 1 - D1234**)
SAS server type	Metadata Server
Authentication Domain	ReplicationSourceAuth
Host Name	Source server host name
Port	8561

The following display illustrates how the Server Manager navigation tree appears after the source metadata server has been defined.

Display 15.1 Server Manager Tree—SourceMetadata Server Defined



For more information about defining a server, see the *SAS Management Console: User's Guide*.

Step 6. Adding a SAS Workspace Server Component to the Source Metadata Server Definition

In SAS Management Console, use the New Application Server Component wizard to add a SAS Workspace Server component to the source metadata server that you defined in "Step 5. Defining the Source Metadata Server in SAS Management Console" on page 278. Complete these steps to launch the wizard:

1 In the SAS Management Console navigation tree, select

Environment Management ► Server Manager

► source metadata server name

For example, select

Environment Management ► Server Manager ► Lev 1 - D1234

2 Select

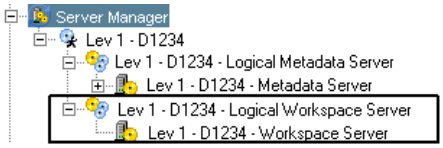
Actions ► Add Application Server Component

When defining the server component in the New Application Server Component wizard, specify these properties:

SAS server type	Workspace Server
Configuration	Basic
Authentication Domain	ReplicationSourceAuth
Host Name	Source server host name
Port	8591

The following display illustrates how the Server Manager navigation tree appears after the SAS Workspace Server component has been added.

Display 15.2 Server Manager Tree—SASWorkspace Server Component Added



For more information about defining an application server component, see the *SAS Management Console: User's Guide*.

Step 7. Defining an IOM Object Spawner for the SAS Workspace Server Component

In SAS Management Console, use the New Server wizard to define the IOM Object Spawner that will be used to launch the SAS Workspace Server that you defined in "Step 6. Adding a SAS Workspace Server Component to the Source Metadata Server Definition" on page 279. Complete these steps to launch the wizard:

1 In the SAS Management Console navigation tree, select

Environment Management ▶ Server Manager

2 Select

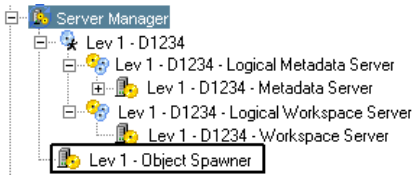
Actions ▶ New Server

When creating the spawner definition in the New Server wizard, specify these properties:

Server type	Object Spawner
Name	**`Lev 1 - Object Spawner`**
Associated Machine	Source server machine
Selected Servers	**`Lev 1 -`** *hostname* **`- Workspace Server`**
Authentication Domain	ReplicationSourceAuth
Host Name	Source server host name
Port	8581

The following display illustrates how the Server Manager navigation tree appears after the object spawner has been defined.

Display 15.3 Server Manager Tree—ObjectSpawner Defined



For more information about defining an IOM Object Spawner, see the *SAS Management Console: User's Guide*.

Step 8. Defining the Target Metadata Server in SAS Management Console

In SAS Management Console, use the New Server wizard to define the target metadata server in the foundation metadata repository in the administration metadata server.

Complete these steps to launch the wizard:

1 In the SAS Management Console navigation tree, select

Environment Management ► Server Manager

2 Select

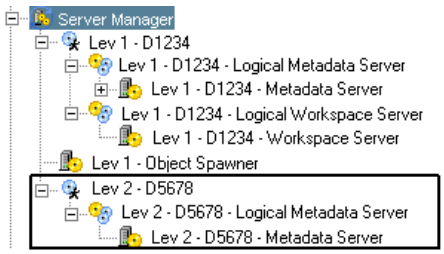
Actions ► New Server

When defining the server in the New Server wizard, specify these properties:

Name	Lev 2 - *hostname* (for example, **`Lev 2 - D5678`**)
SAS server type	Metadata Server
Authentication Domain	ReplicationTargetAuth
Host Name	Target server host name
Port	8562

The following display illustrates how the Server Manager navigation tree appears after the target metadata server has been defined.

Display 15.4 Server Manager Tree–Source Metadata Server Defined



For more information about defining a server, see the *SAS Management Console: User's Guide*.

Step 9. Adding a SAS/CONNECT Server Component to the Target Metadata Server Definition

In SAS Management Console, use the New Application Server Component wizard to add a SAS/CONNECT server component to the target metadata server that you defined in "Step 8. Defining the Target Metadata Server in SAS Management Console" on page 280. Complete these steps to launch the wizard:

1 In the SAS Management Console navigation tree, select

Environment Management ► Server Manager

► target metadata server name

For example, select

Environment Management ► Server Manager ► Lev 2 - D5678

2 Select

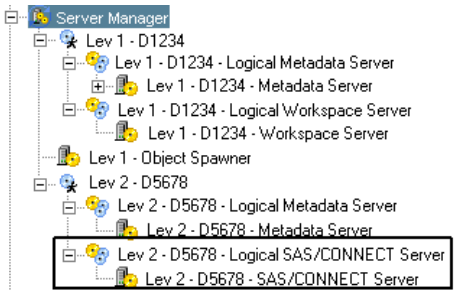
Actions ► Add Application Server Component

When defining the server component in the New Application Server Component wizard, specify these properties:

SAS server type	SAS/CONNECT Server
Configuration Type	Basic
Authentication Domain	ReplicationTargetAuth
Host Name	Target server host name
Port Number	7552

The following display illustrates how the Server Manager navigation tree appears after the SAS/CONNECT server component has been added to the target metadata server.

Display 15.5 Server Manager Tree—SAS/CONNECTServer Component Added



For more information about defining an application server component, see the *SAS Management Console: User's Guide*.

Step 10. Creating the Metadata Access File on the Source Metadata Server Machine

The metadata access file is used to establish communications between the IOM Object Spawner and the source metadata server. On the machine that is hosting the source metadata server, create a metadata server access file named **mdrepljs.sas** with the following content:

```
options metaserver='administration server name'
        metaport=administration server port
        metaprotocol=BRIDGE
        metauser='domain\sasrpadm'
        metapass='pw';
```

where

administration server name	specifies the machine name or DNS of the machine that is hosting the administration metadata server.
administration server port	specifies the port number of the machine that is hosting the administration metadata server. The default value is **8560**.
*domain***sasrpadm**	specifies the domain (if necessary) and the **sasrpadm** user ID used to start the administration metadata server.

pw specifies the encoded password for **sasrpadm**. To determine the encoded form of the password, start a SAS session and submit the following code in the Program Editor:

```
proc pwencode in='xxxxxx';
run;
```

where *xxxxx* is the unencoded password. Copy the resulting text from the SAS log to the metadata access file.

Save the file in the directory from which you started the source metadata server. On Windows, that directory is probably **C:\SAS***configuration-directory***\Lev1\SASMain***.

Note: At this point, make sure that the source metadata server and the IOM Object Spawner are both running on the source metadata server machine. △

Step 11. Configuring the Target Metadata Server

To configure the target metadata server, you complete the SAS Configuration wizard by entering basically the same information that you entered to create the source metadata server, including using the same configuration name and the same paths. The modifications are listed in these steps:

1. On the Select Install Set window, select **Custom** and then choose to install only the metadata server.
2. On the Enter SAS Metadata Server Information window, enter **8562** instead of **8561** as the port number.
3. When the Advanced Properties Editor window appears, click [Edit Properties]. In the text file that opens, make these changes.

Table 15.2 Values for the Target Metadata Server

Replace occurrences of these	With this value
8561	8562
7551	7552
Lev1	Lev2

4. Save the changes to the file and close it to return to the SAS Configuration wizard.
5. Click [Next] to display the Finish window, then click [Finish].
6. When the **instructions.html** file appears in your browser, close the window. Instead, you must follow the instructions in this chapter.
7. If SAS Management Console launches, exit the application.
8. Click [Done] to exit the SAS Configuration wizard.

If you elected to start the server as a service, then the target metadata server is started automatically.

Note: On Windows systems, you can manually start the server by selecting the applicable start-up task from

[Start] ► [Programs] ► [Sas] ► [name of your configuration directory]

△

Step 12. Defining a SAS/CONNECT Server for the Target Metadata Server Machine

On the machine that is hosting the target metadata server, you must define and start a SAS/CONNECT server. The SAS/CONNECT server enables the source and target metadata servers to communicate and copy data.

1. Open the file **ConnectServer.bat**, which, on Windows, is typically located in **C:\SAS***configuration directory name***\Lev2\SASMain\ConnectServer**.
2. Locate the **USEMETADATA** option and set the line to

   ```
   set USEMETADATA=0;
   ```

3. Locate the **isService** option and set the line to

   ```
   set isService=0;
   ```

4. Save your changes.
5. Run the **StartConnectServer.bat** file to start the SAS/CONNECT server.

Creating a Promotion Job

Complete these steps to create a promotion job:

1. Use SAS Management Console to connect to the administration metadata server with the SAS Replication Administrator login (**sasrpadm**).
2. In the SAS Management Console navigation tree, select

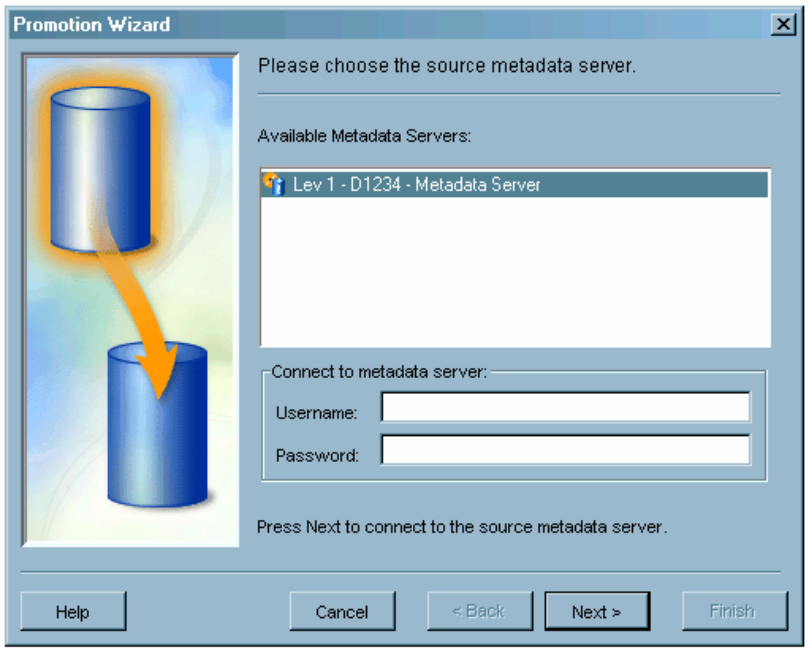
   Metadata Manager ► Job Definitions ► Promotion

   Note: You will create a promotion job to move metadata with substitutions from level 1 to level 2. Technically, however, moving metadata from a production level to a test level is a demotion. △
3. Select

   Actions ► New Definition

4. In the Source Metadata Server Definition window, select the source metadata server.

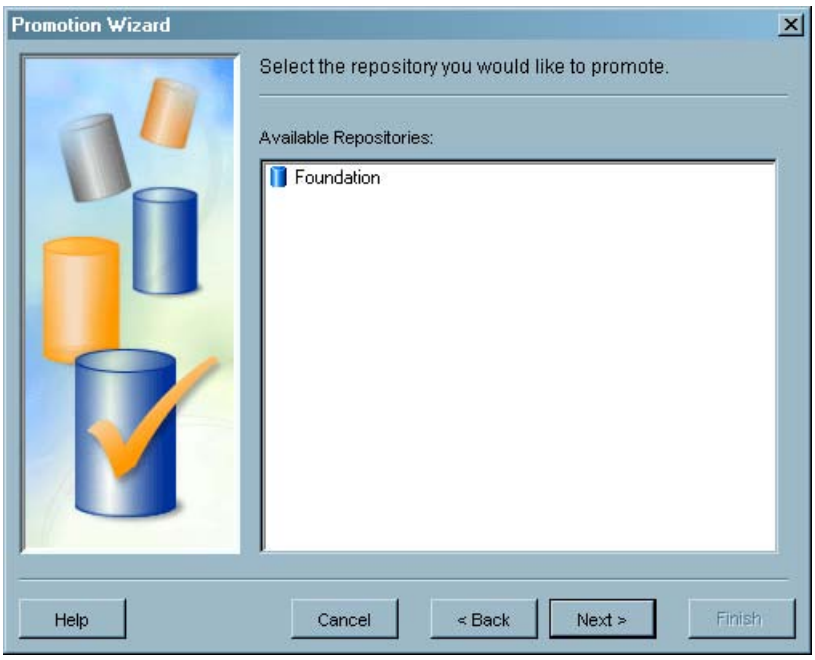
Display 15.6 Promotion Wizard—SourceMetadata Server Definition Window



Enter the source metadata server login for the SAS Replication Administrator. This login was defined in "Step 4. Defining the SAS Replication Administrator in SAS Management Console" on page 277. Click Next to continue.

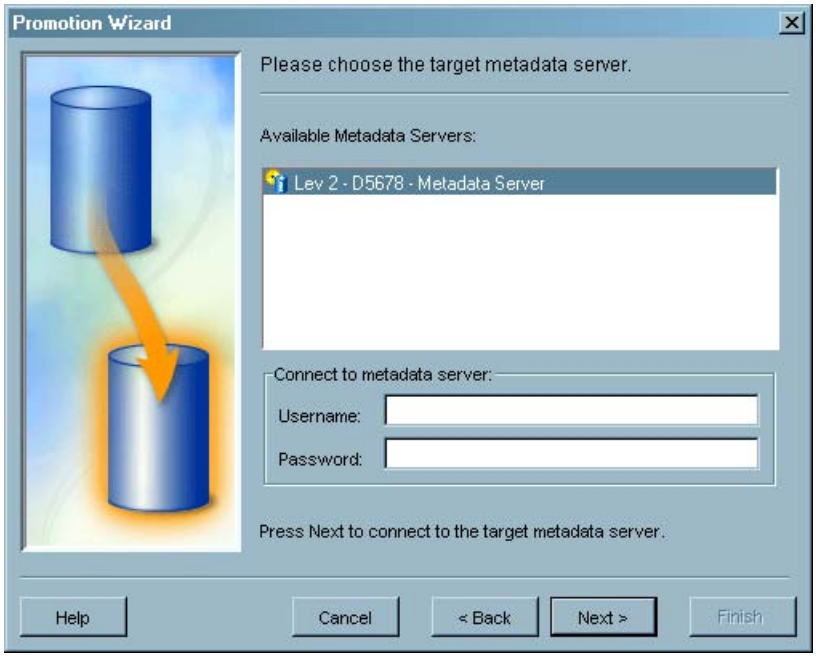
5 Select the **`Foundation`** repository and click Next to continue.

Display 15.7 Promotion Wizard—SelectRepository Window



6 In the Connect to Target Metadata Server window, select the target metadata server.

Display 15.8 Promotion Wizard—Connectto Target Metadata Server Window



Enter the target metadata server login for the SAS Replication Administrator. This login was defined in "Step 4. Defining the SAS Replication Administrator in SAS Management Console" on page 277. Click Next to continue.

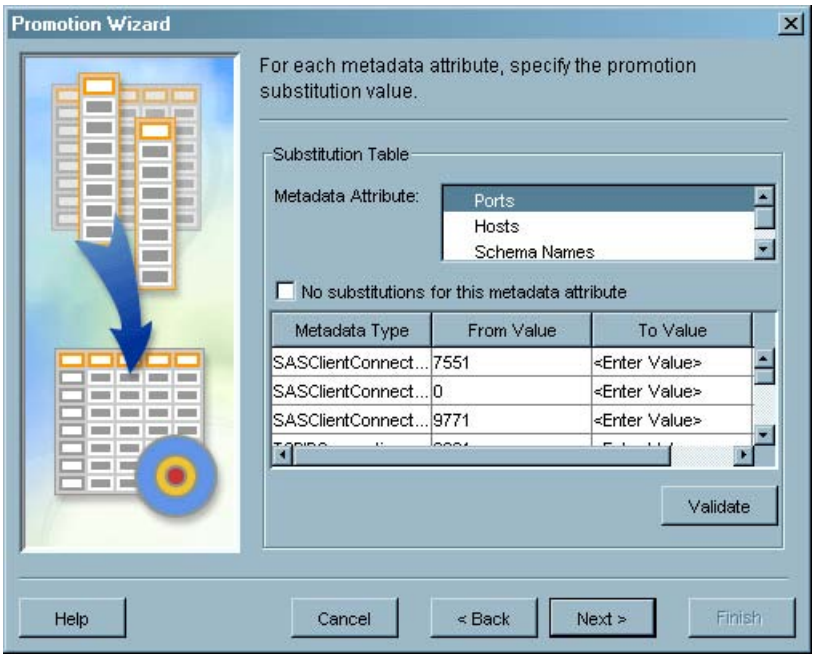
7 If the repository that you selected has not been defined on the target metadata server, you must specify the engine type and path for the repository:

- For a SAS repository, select **Base** as the engine type.
- For other repositories, if you select **DB2** or **Oracle** as the engine type, the **Options** field contains the options required to access the repository. Some options require you to specify additional information. For example, you might have to specify a user ID and password.

Click Next to continue.

8 The Substitutions window enables you to specify modifications that will be made to the metadata attribute values when the metadata is promoted.

Display 15.9 Promotion Wizard—SubstitutionsWindow



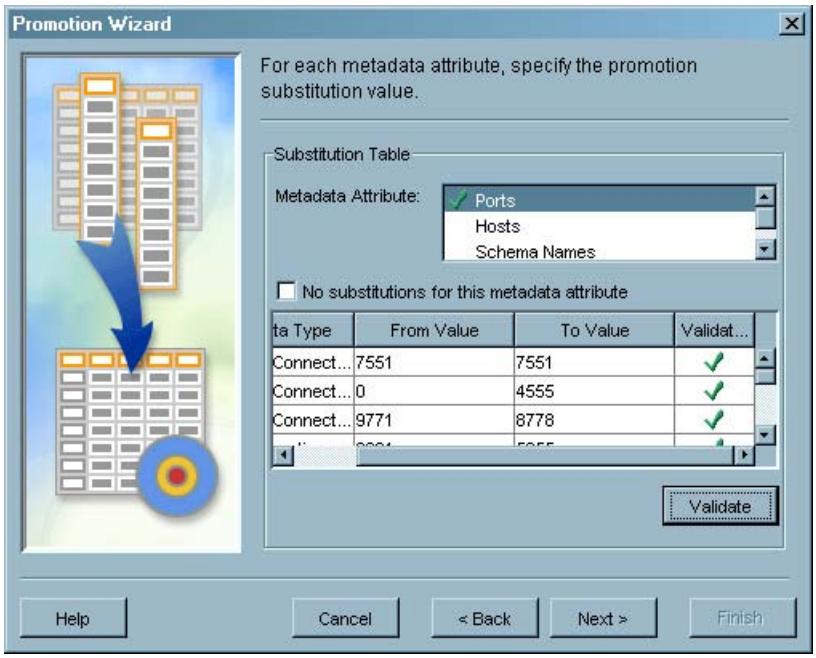
Note: If you do not want to make substitutions for an attribute, select the attribute name, then select the **`No substitutions for this metadata attribute`** check box. △

Table 15.3 Substitutions to Make for the Target Metadata Server

For this metadata attribute	Make this change to the associated values
Ports	Add 1 to each port number than ends in **`1`**. For example, if a port number is **`4321`**, change the value to **`4322`**.
Hosts	Change the source metadata server host names to the target metadata server host names. Use fully qualified host names. For example, even if the host name is displayed as **`D1234`**, you should enter the fully qualified name, which might be **`D1234.na.abc.com`**.
Paths	Change any paths that do not exist on the target metadata server to paths that are valid on the target server. Leave all relative paths unchanged.
Schema Names	Change any schema names to ones that will be appropriate on the target server.

9 When you are done specifying substitute values, click Validate. Attribute values for which you specified a substitution are marked with a check mark in the **`Validation`** column.

Display 15.10 Substitutions Window—ValidatedAttributes



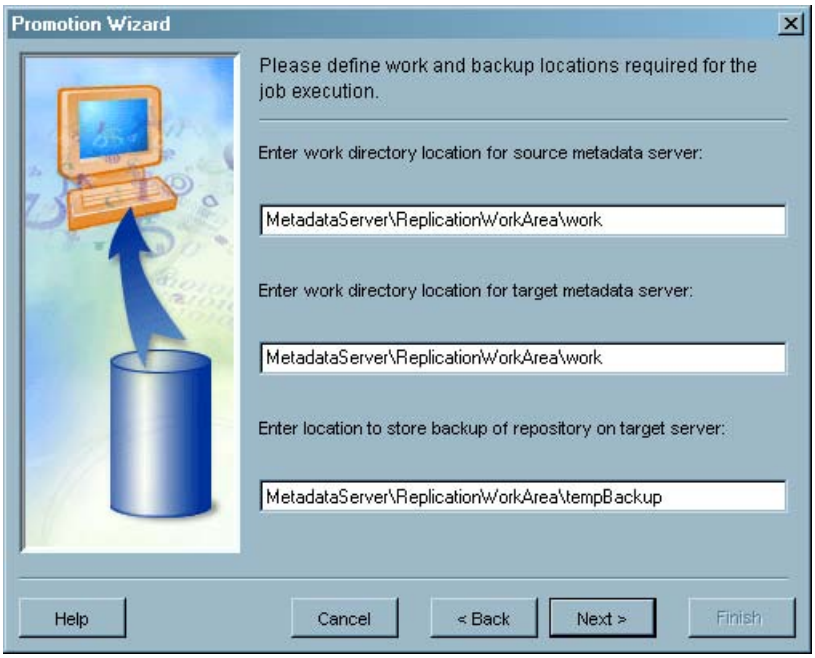
You must successfully validate all instances of all attributes before you can continue with the wizard.

Note: The validation process does not check whether substitution values are correct or appropriate, only that they are present. You must enter and verify appropriate values. △

Click Next to continue.

10 In the Define Work Directories window, specify the directories on machines that are hosting the source and target metadata servers where work files and a backup copy of the repository will be stored.

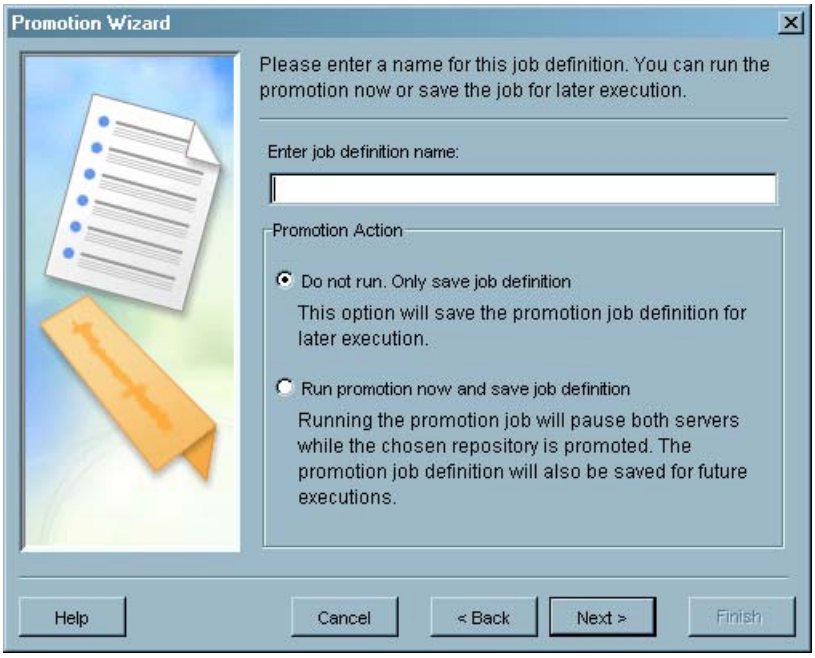
Display 15.11 Promotion Wizard—Define Work Directories Window



Click Next to continue.

11 Enter a job definition name, then specify the applicable option (save the job definition and run, or save the definition only). Click Next to continue.

Display 15.12 Promotion Wizard–Run or Save Job Definition Window



12 The Current Settings window displays all of the information that you specified in the wizard. To make changes, click Back until you reach the appropriate window.

If all of the information is correct, then click Finish to create the replication job definition and run the job, if you elected to do so.

Running a Promotion Job

Note: If you add any new metadata objects that have attributes that can be modified by a promotion job, then you must create a new job definition rather than rerun an existing job. Promotion jobs do not automatically identify new metadata attributes and supply substitute attribute values. △

Complete these steps to run a saved promotion job outside of the Promotion wizard:

1. In the SAS Management Console navigation tree, on the machine that is hosting the administration metadata server, select

   Metadata Manager ► Job Definitions ► Promotion

2. In the display area to the right, select a job to run.
3. Select

   Actions ► Run Job

Note: To save the SAS code for the promotion job to a file, select **Save to File** instead of **Run Job**. △

Modifying Metadata After Promotion to the Target Metadata Server

After you run the promotion job for the first time, you must make changes to some of the promoted metadata in order to make it valid in the target metadata.

Note: Some of these steps require that you locate a server definition in the SAS Management Console navigation tree. For information about the composition of a SAS application server, see Chapter 2, "Understanding the SAS Application Servers," on page 19. △

1. Modify the **ConnectServer.bat** file, which, on Windows, is typically located in **C:\SAS***configuration directory name***\Lev2\SASMain\ConnectServer**.
   a. Locate the **USEMETADATA** option and set the line to

      ```
      set USEMETADATA=1;
      ```

   b. If you are going to install the SAS/CONNECT server as a service on the target machine, then locate the **isService** option and set the line to

      ```
      set isService=1;
      ```

   c. Save your changes.
2. To install the SAS/CONNECT server as a service, run the **InstallConnectServer.bat** file once. Typically, the file is located in **C:\SAS***configuration directory name***\Lev2\SASMain\ConnectServer**.
3. Start the SAS/CONNECT server, depending on how you installed the server on the target machine:
   - If you did not install the server as a service, then run the **StartConnectServer.bat** file.
   - If you installed the SAS/CONNECT server as a service, then start the service.
4. Use SAS Management Console to connect to the target metadata server with the login that you assigned to the SAS Replication Administrator in "Step 4. Defining the SAS Replication Administrator in SAS Management Console" on page 277.

5 If necessary, modify logins for local users who were promoted to the target machine:

   a In the SAS Management Console navigation tree, select

   Environment Management ► User Manager

   b Select a user in the display area to the right.
   c Select

   Actions ► Properties

   d Click the Logins tab.
   e Locate any logins for the selected user that contain the source metadata server domain as part of the user ID and replace that domain information with the target metadata server domain.

6 In the SAS Management Console navigation tree, select

   Environment Management ► Server Manager

7 If a SAS Workspace Server is present in the list of servers, then select its definition and complete these steps:

   a Select

   File ► Properties

   b On the Options tab, enter the following in the **Command** field:

   ```
   sas -config ''C:\SAS\configuration directory name\Lev2\SASMain\sasv9.cfg''
   ```

   c Click OK to save the changes.

8 If a SAS Stored Process Server is present in the list of servers, then select its definition and complete these steps:

   a Select

   File ► Properties

   b On the Options tab, enter the following in the **Command** field:

   ```
   sas -config ''C:\SAS\configuration directory name\Lev2\SASMain\
   StoredProcessServer\sasv9_StorProcSrv.cfg''
   ```

   c Click OK to save the changes.

9 If a SAS OLAP Server is present in the list of servers, then select its definition and complete these steps:

   a Select

   File ► Properties

   b On the Options tab, click Advanced Options to open the Advanced Options dialog box.
   c On the Performance tab, enter the following in the **Path for temporary working files** field:

   ```
   sas -config ''C:\SAS\configuration directory name\Lev2\SASMain\sasv9.cfg''
   ```

   d Click OK to save the changes and return to the Properties dialog box.
   e Click OK to save the changes.

10 If a SAS/CONNECT server is present in the list of servers, then select its definition and complete these steps:

a Select

File ► Properties

b On the Options tab, enter the following in the **SASCMD** field:

```
C:\SAS\configuration directory name\Lev2\SASMain\sasconnect.bat
```

c Click OK to save the changes.

11 If a SAS batch server is present in the list of servers, then select its definition and complete these steps:

a Select

File ► Properties

b On the Server Properties tab, enter the following in the **Command line** field:

```
C:\SAS\configuration directory name\Lev2\SASMain\BatchServer\sasbatch
```

c Enter the following in the **Logs directory** field:

```
C:\SAS\configuration directory name\Lev2\SASMain\BatchServer\logs
```

d Click OK to save the changes.

12 If a SAS/SHARE server is present in the list of servers, then select its definition and complete these steps:

a Select the SAS/SHARE server connection name in the display area to the right.

b Select

File ► Properties

c On the Options tab, change the values for the **Server Host**, **Remote Session ID**, and the **Server ID** to the machine that is hosting the target metadata server.

d 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.

e Click OK to save the changes.

13 Exit the SAS Management Console application.

14 If you promoted cube metadata from the source metadata server to the target metadata server, then you must modify the work path, index path, and data path for each promoted cube. The following instructions explain how to use SAS ETL Studio to edit a cube's structure:

a Use SAS ETL Studio to connect to the target metadata server. You can use the login information for any user who has WriteMetadata access to the metadata repository that contains the cubes that you need to edit, as well as WriteMetadata access to the cube metadata that must be modified.

b In the SAS ETL Studio inventory tree, select the cube name.

c To launch the Cube Designer wizard, right-click on the cube name and select **Edit Cube Structure**.

d On the General window, change the **Work Path** to the correct path for the target machine.

e Click Next until the Generated Aggregations window appears. On that window, click Advanced to open the Performance Options dialog box. On the Performance Options dialog box, make these changes on the Global tab:

i Change the value of the **Location of index component files** to the correct path for the target machine.

ii Change the value of the **`Location of partitions in which to place aggregation table data`** to the correct path for the target machine.

iii Click [OK] to close the dialog box and return to the Cube Designer wizard.

f Click [Next] until the Finish window appears, then click [Finish] to save your changes.

Creating a Replication Job

Complete these steps to replicate a repository:

1 Use SAS Management Console to connect to the administration metadata server with the SAS Replication Administrator login (**`sasrpadm`**).

2 In the SAS Management Console navigation tree, select

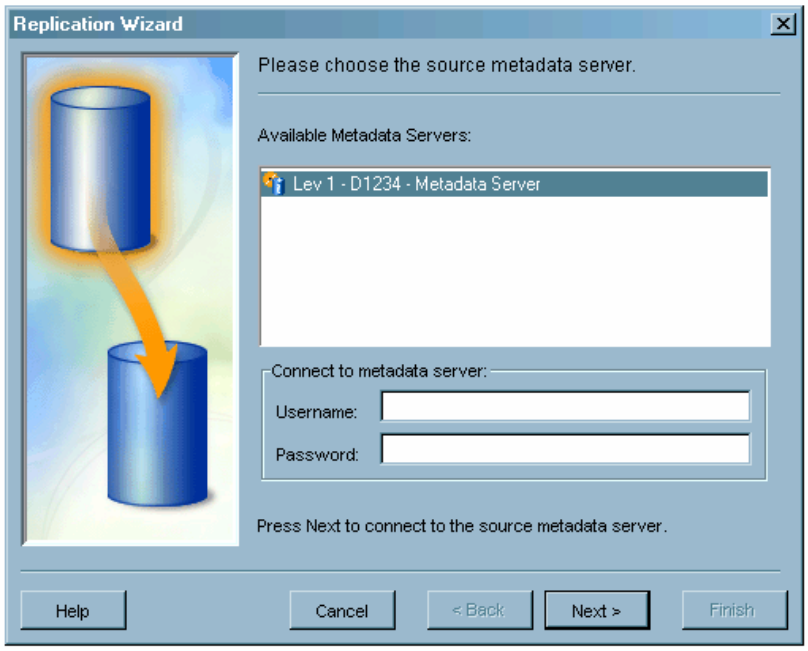
[Metadata Manager] ► [Job Definitions] ► [Replication]

3 Select

[Actions] ► [New Definition]

4 In the Source Metadata Server Definition window, select the source metadata server.

Display 15.13 Replication Wizard—SourceMetadata Server Definition Window

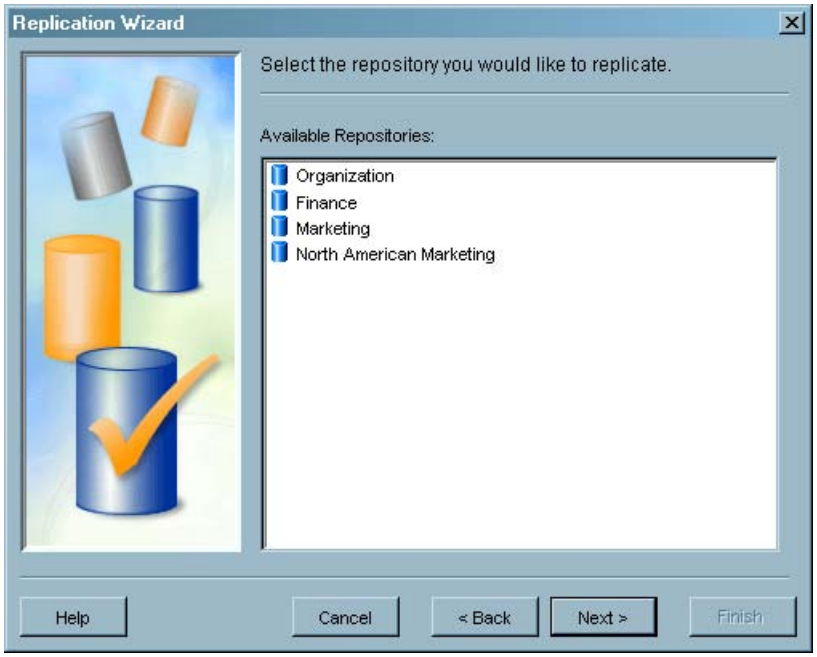


Enter the source metadata server login for the SAS Replication Administrator. This login was defined in "Step 4. Defining the SAS Replication Administrator in SAS Management Console" on page 277. Click [Next] to continue.

5 Use the Select Repository window to select the repository that you want to replicate.

Note: The target metadata server cannot contain an existing repository with the same name as the selected repository. △

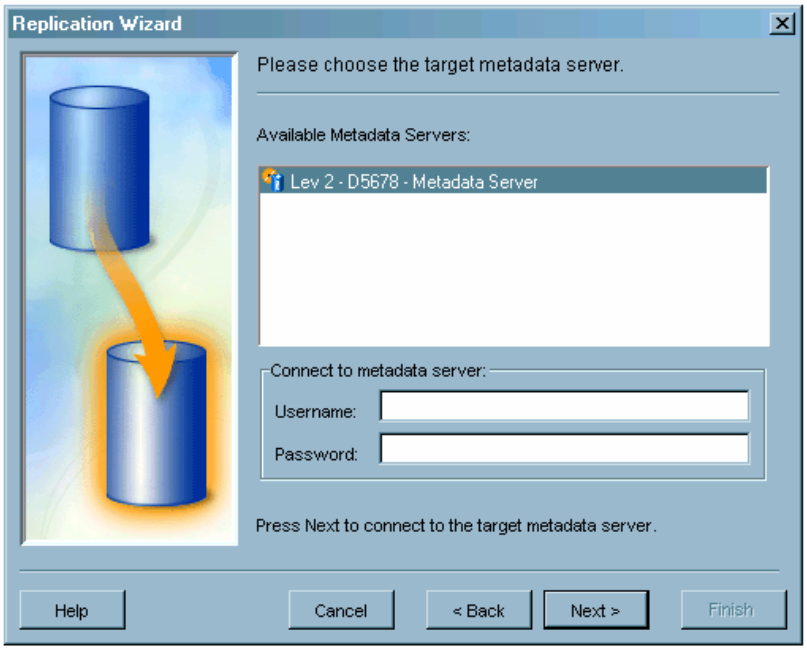
Display 15.14 Replication Wizard—SelectRepository Window



Click Next to continue.

6 In the Connect to Target Metadata Server window, select the target metadata server.

Display 15.15 Replication Wizard—Connectto Target Metadata Server Window

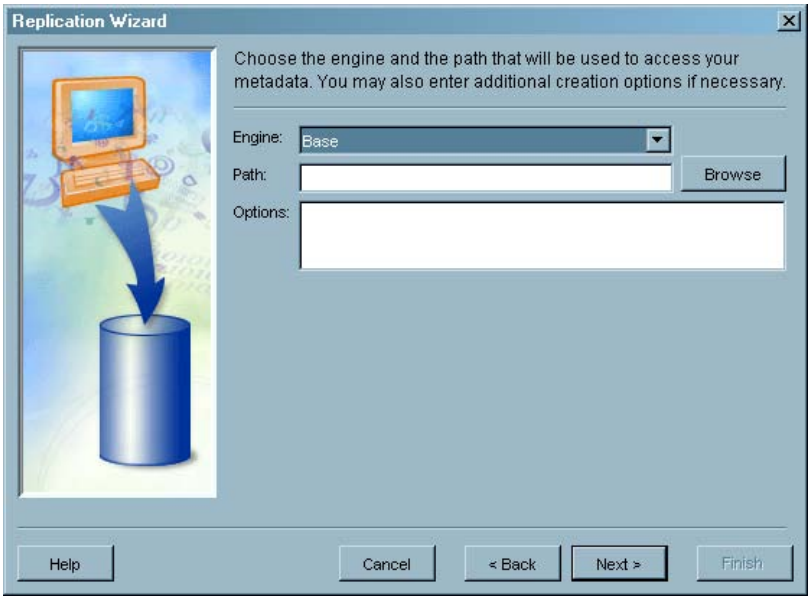


Enter the target metadata server login for the SAS Replication Administrator. This login was defined in "Step 4. Defining the SAS Replication Administrator in SAS Management Console" on page 277. Click Next to continue.

7 If the repository you selected has not been defined on the target metadata server, then you must specify the engine type and path for the repository:

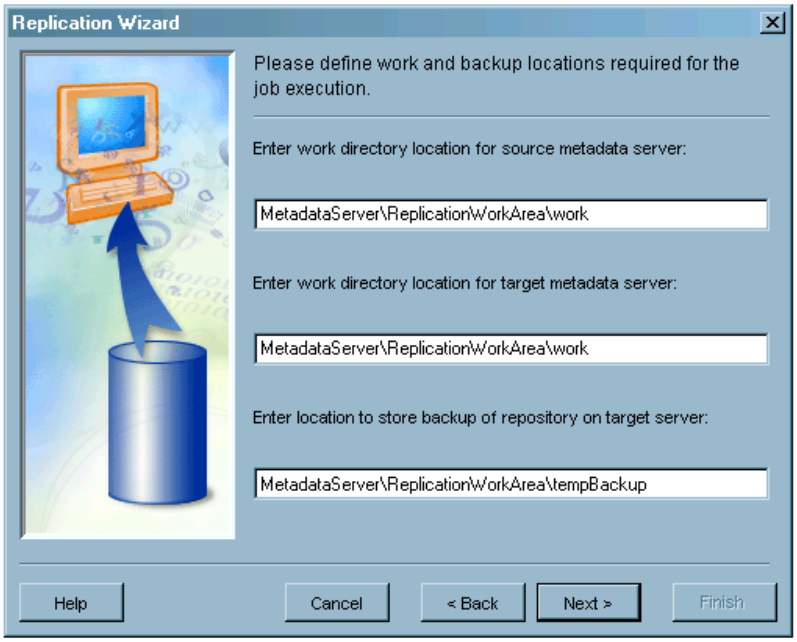
- For a SAS repository, select **`Base`** as the engine type.
- For other repositories, if you select **`DB2`** or **`Oracle`** as the engine type, the **`Options`** field contains the options required to access the repository. Some options require you to specify additional information. For example, you might have to specify a user ID and password.

Display 15.16 Replication Wizard—RepositoryAccess Window



8 In the Define Work Directories window, specify the directories on the machines that host the source and target metadata servers where work files and a backup copy of the repository are stored.

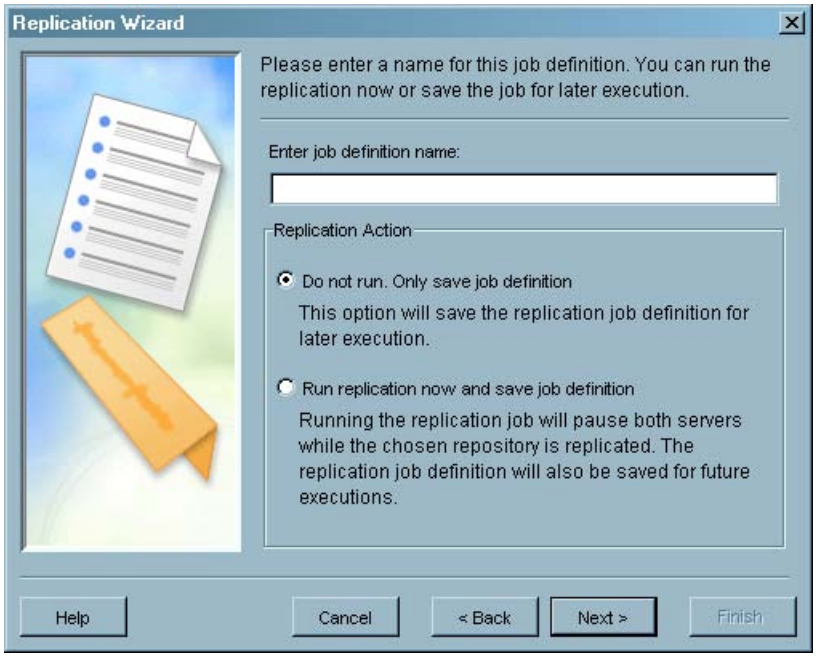
Display 15.17 Replication Wizard—Define Work Directories Window



Click Next to continue.

9 Enter a job definition name, then specify the applicable option (save the job definition and run, or save the definition only). Click Next to continue.

Display 15.18 Replication Wizard—Runof Save Job Definition Window



10 The Current Settings window displays all of the information that you specified in the wizard. To make changes, click Back until you reach the appropriate window.

If all of the information is correct, then click [Finish] to create the replication job definition and run the job, if you elected to do so.

Running a Replication Job

Complete these steps to run a saved replication job outside of the Replication wizard:

1. In the SAS Management Console navigation tree, on the machine that is hosting the administration metadata server, select

   [Metadata Manager] ► [Job Definitions] ► [Replication]

2. In the display area to the right, select a job to run.
3. Select

   [Actions] ► [Run Job]

Note: To save the SAS code for the replication job to a file, select **Save to File** instead of **Run Job**. △

Troubleshooting Replication and Promotion

Whenever a replication or promotion job runs, it writes any error messages to the SAS Management Console error log. This log is named **errorlog.txt**, and located by default in the directory from which SAS Management Console runs.

For example, an entry in the error log similar to the following text indicates that the **mdrepjls.sas** file was not found or that there are errors in the options that are specified in the file. See "Step 10. Creating the Metadata Access File on the Source Metadata Server Machine" on page 282.

```
Set META* options needed to connect to Job definition server
WARNING: Physical files does not exist, C:\Program Files\SAS\omasvr\mdrepljs.sas
```

If the entries in the error log are not sufficient to identify the source of the error, then you can save the replication or promotion job to a file and run the saved file in a SAS session. To identify problems, you can then view the error messages written to the SAS message log.

To save a promotion or replication job to a file:

1. In the SAS Management Console navigation tree, on the machine that is hosting the administration metadata server, select

   [Metadata Manager] ► [Job Definitions]

2. Depending on which type of job that you want to save, select **Replication** or **Promotion**.
3. In the display area to the right, select a job to save.
4. Select

   [Actions] ► [Save to File]

5. Specify a name for the job in the Save window and click [OK].

CHAPTER

16

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

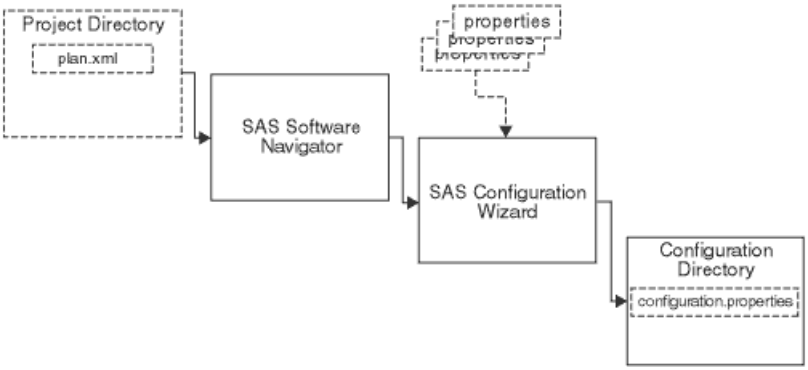
What Is a Property?

A 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. For more information about planned installations, see Chapter 7, "Installing and Configuring Your Software," on page 123. △

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 16.1 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.

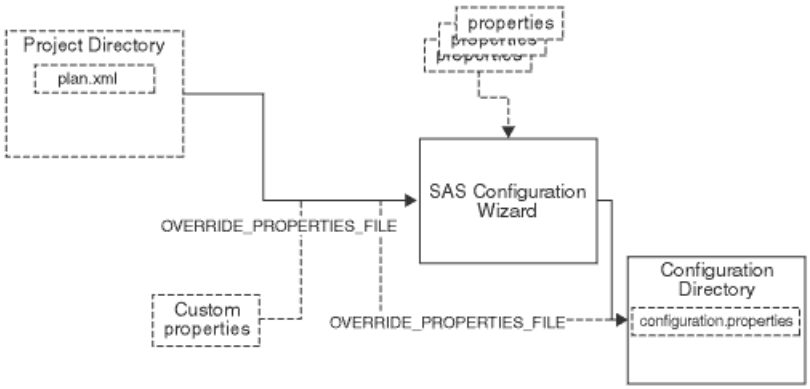
To use this option, you launch the SAS Configuration Wizard by using a command such as

```
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 16.2 Using the OVERRIDE_PROPERTIES_FILE Command to Specify a Custom Property File

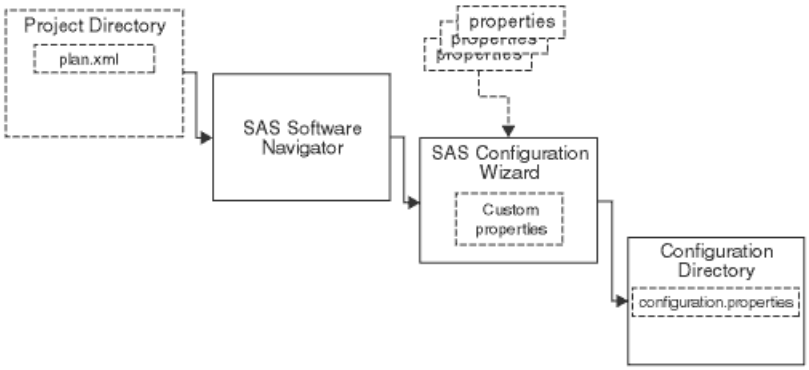


Editing the Configuration File from Within the SAS Configuration Wizard

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 16.3 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 "Resetting Passwords" on page 83. △

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**.

Note: Information about adding a level to an existing environment can be found in Chapter 15, "Promoting and Replicating Metadata," on page 273. △

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

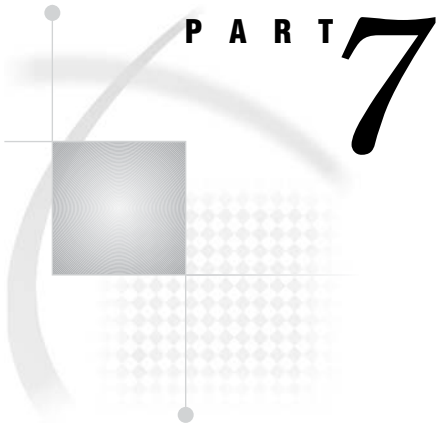
Complete these steps to remove a configuration directory that was created by the SAS Configuration Wizard:

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
     ```



PART 7

Appendix

APPENDIX

1

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 ETL 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 only supported by SAS ETL Studio, which maintains an unchanged production environment. For SAS ETL Studio, you might have Lev1for 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 209.

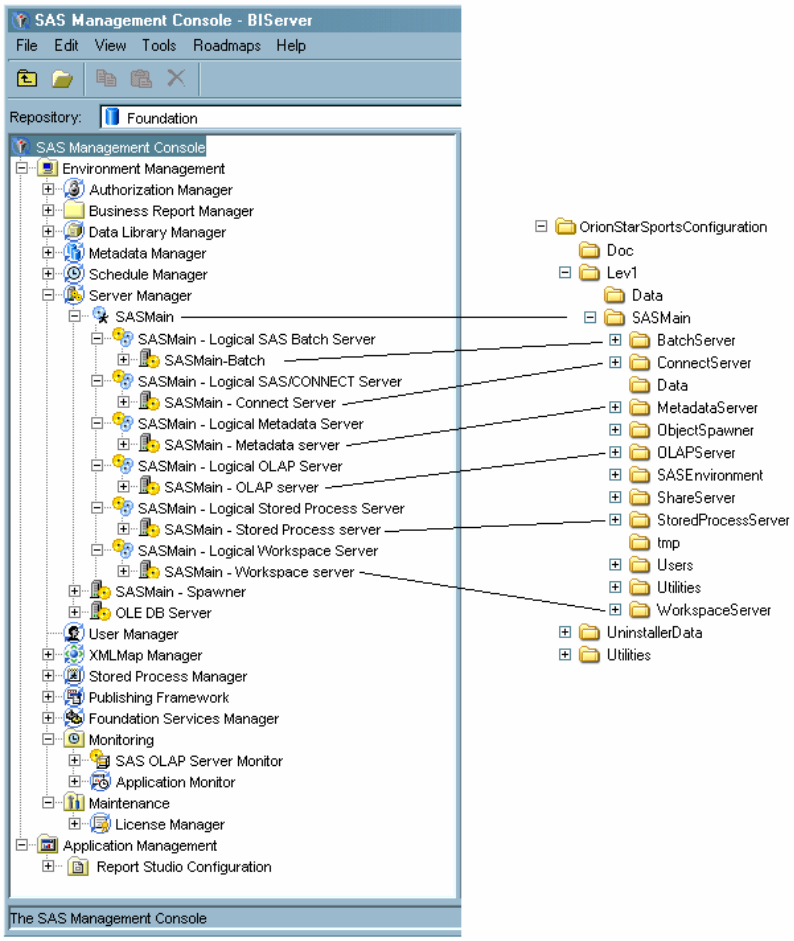
SAS Application Server — a logical framework under which SAS applications execute. A SAS application server enables you to specify metadata that applies to all of the logical servers and servers that the SAS application server contains. A SAS application server provides a place to attach libraries, schemas, 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 A1.1 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 313.

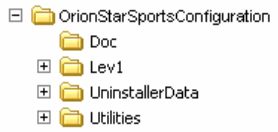
Note: **`SASMain`** is the name of the application server on the machine that hosts the SAS Metadata Server. Machines that are not hosting a SAS Metadata Server can have user-defined application server names; the default name is **`SASApp`**. This appendix only discusses **`SASMain`**. △

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 **`OrionStarSportsConfiguration`**.

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 A1.2 Example of the First Level of the Configuration Directory on a Windows System



At the same level in the structure shown in the illustration are

Doc provides a location for you to save your own documents related to the current environment.

Lev1 as explained in "The Lev1 Directory Structure" on page 312, 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 ETL Studio, which maintains an unchanged production environment. For SAS ETL 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 mid-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 A1.3 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 313.

Note: On Web-tier only machines, this directory is empty. △

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 **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 318.

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 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 **OrionStarSportsConfiguration** sample environment might have this content on a Windows system:

```
-config "C:\Program Files\SAS\SAS 9.1\sasv9.cfg"

-sasinitialfolder "c:\SAS\OrionStarSportsConfiguration\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>
```

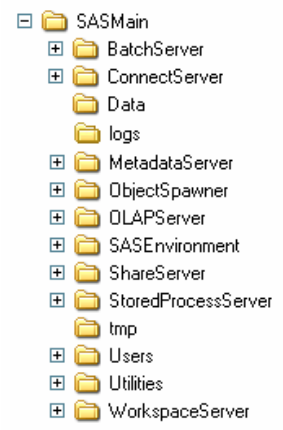
Note: For information about the **omaconfig.xml** file, see the *SAS Metadata Server: Setup Guide* available at **support.sas.com/rnd/eai/openmeta/v9/setup**. △

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 A1.4 Example of a SASMain Directory Structure on a Windows System



Here are descriptions of the folders 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.txt (see "Security-Related Files" on page 149).
- 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.
- 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 313):

- 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 **Step** 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:

- a sample SAS application named **LoadPlannedStoredProcessSamples.sas** that can be used to load stored process samples into the metadata repository.
- the **logs** subdirectory that stores any log 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. △

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. △

Note: For information about the users and groups that are created during a planned SAS deployment, see "Setting Up Required User Accounts" on page 113. △

Table A1.1 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

2

Basic Installs

Overview of Basic Installs

In this document, we have recommended the use of project installs, those installs for which you have a planning file. Because the planning file contains information about which software components are to be installed and configured on each host, this type of installation is the most foolproof. However, it is also possible to perform what is called a basic install, an install for which you have no planning file. Obviously, during a basic install, you have to know which software to install and configure on each host.

There are two types of basic installs: licensed software installs and CD Index installs. In a licensed software install, you install the required components of a technology package on a machine. In this case, the SAS Software Navigator gives you guidance as to the order in which you should install products. In a CD Index install, you select individual products to install from a set of CDs.

When would you perform a basic install? Well, there are two main cases. First, an expert installer might use a licensed software install, followed by a CD Index install of a couple of optional products, to install the software on a machine. If you have the knowledge to do this, you can save yourself the trouble of creating a planning file. Second, an installer might want to use a CD Index install to install a product after the initial installation and configuration of a system. For example, the installer might need to go back and install a JDBC driver.

Following a basic install, you can use the SAS Configuration Wizard to configure the software you have installed. Of course, you will have to specify which software you want the configuration wizard to configure. In addition, before you can run the SAS Configuration Wizard on a host where you have installed SAS or mid-tier servers, you must perform the pre-installation tasks described in Chapter 6, "Pre-Installation Tasks," on page 93—with the exception of setting up a project directory.

For more information about the topics introduced previously, see the following sections:

- "Licensed Software Installs" on page 322
- "CD Index Installs" on page 325
- "Using the SAS Configuration Wizard in Basic Installs" on page 326

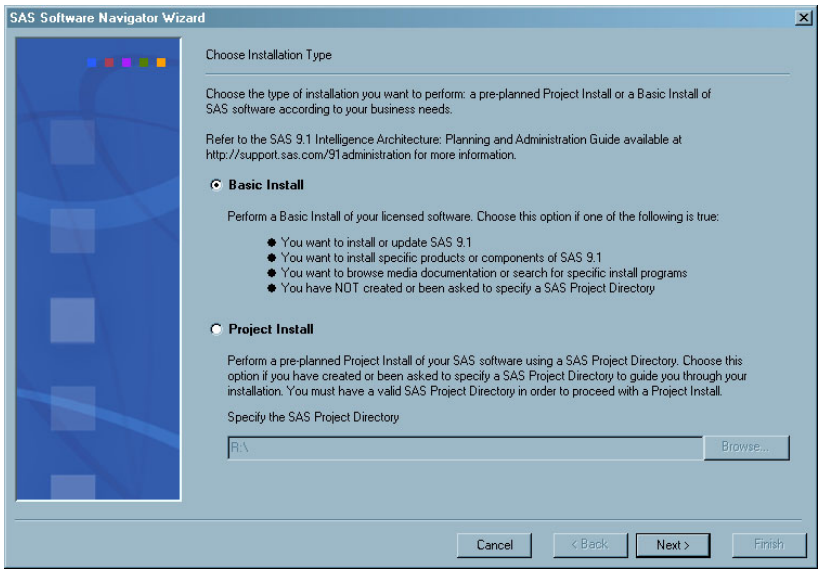
Note: Several SAS software products have features that allow for an installation that includes Quiet Mode (does not require user interaction, which is useful for administrators who need to deploy software across their enterprises in an unattended

manner) and Console Mode (installation from a text-only window, which is useful when displaying the graphical user interface is not possible). For a complete list of those products and a description of these features, go to the Install Center (**`support.sas.com/installcenter`**) and select the "Enhanced Features of Basic Installs" link on the right side under "More Documentation." △

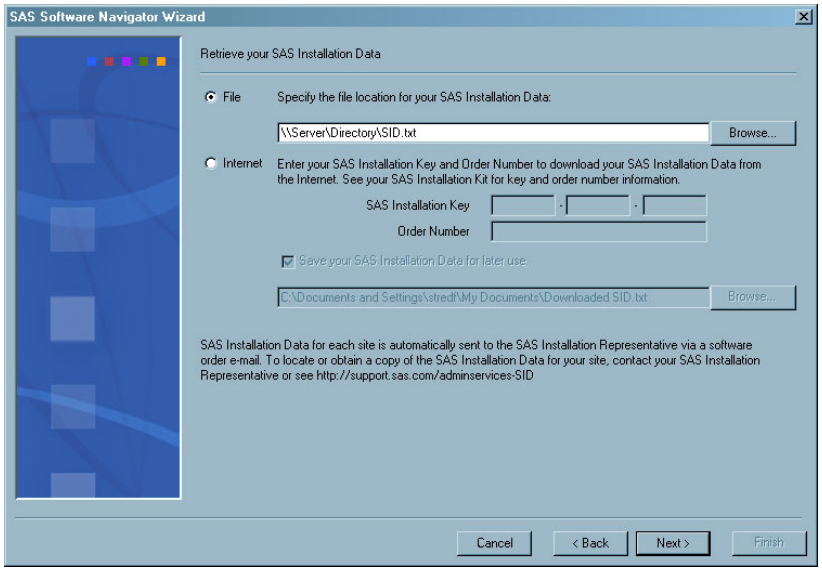
Licensed Software Installs

If you decide that you want to install the required products from a technology package, follow these steps:

1 Start the SAS Software Navigator from your SAS Software Depot or by using the CD that contains the navigator. The SAS Software Navigator Wizard starts.



2 Select the **`Basic Install`** radio button, and click Next. The next screen in the wizard prompts you for your SAS Installation Data.



3 You can specify whether you want to obtain this data from a file (via a pathname) or download it from the Internet. If you have received your SAS Installation Data with your Software Order Email and have saved it on your system, use the **File** option; otherwise, select the **Internet** option.

- If you select **File**, you are prompted to indicate where the SAS Installation Data is located. If you are not sure where the file is located, use the [Browse] button to browse to the location. When you have located the file, click [Next].
- If you select **Internet**, you are prompted to enter your SAS Installation Key and Order Number—from your SAS Order Information Letter or Software Order Email—as identifiers for your specific SAS Installation Data. Note that this connection to the Internet uses the SAS secure Web site. You are also able to use the SAS Software Navigator to save your SAS Installation Data to a file with this option. When you have downloaded your SAS Installation Data, click [Next].

UNIX Proxy Information

If you choose to obtain the SAS Installation Data from the Internet and your site uses a UNIX proxy server to access the Internet, you need to verify that the system has your proxy information defined properly.

For proxy support of **https** requests, you need to make sure that your **HTTPS_PROXY** environment variable is defined to point to your proxy server and port. The **HTTPS_PROXY** variable can be defined using either a host address or an IP address. For example, using the Bourne shell, you can define **HTTPS_PROXY** as follows:

```
$ HTTPS_PROXY = "http://proxy.server.com:8080"; export HTTPS_PROXY
```

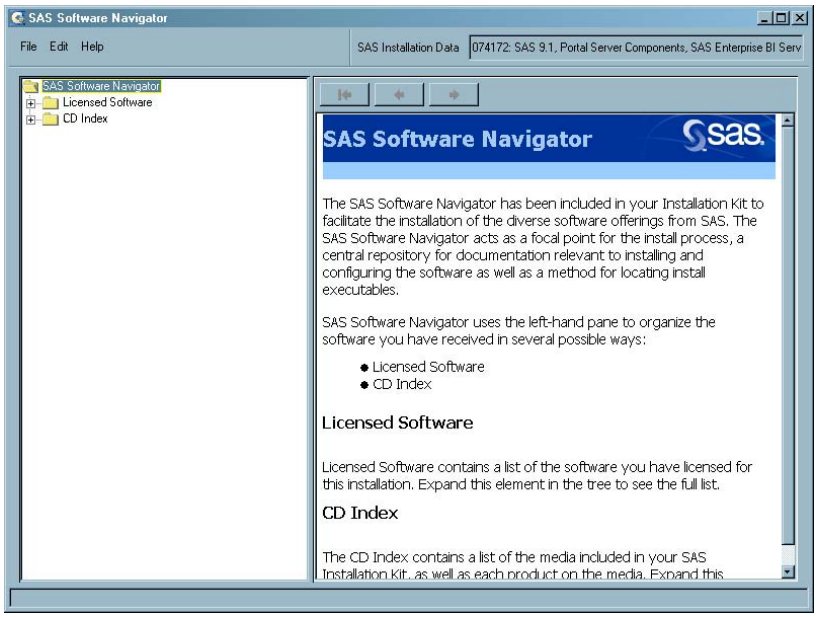
or

```
$ HTTPS_PROXY = "127.0.0.1:8080"; export HTTPS_PROXY
```

If your proxy server requires basic authentication credentials, you can also define the **HTTPS_USERNAME** and **HTTPS_PASSWORD** environment variables. For example, using the Bourne shell, you can define **HTTPS_USERNAME** and **HTTPS_PASSWORD** as follows:

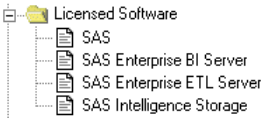
```
$ HTTPS_USERNAME = "myaccount"; export HTTPS_USERNAME
$ HTTPS_PASSWORD = "mypasswd"; export HTTPS_PASSWORD
```

4 Click [Finish] in the licensed software window. The SAS Software Navigator window appears.

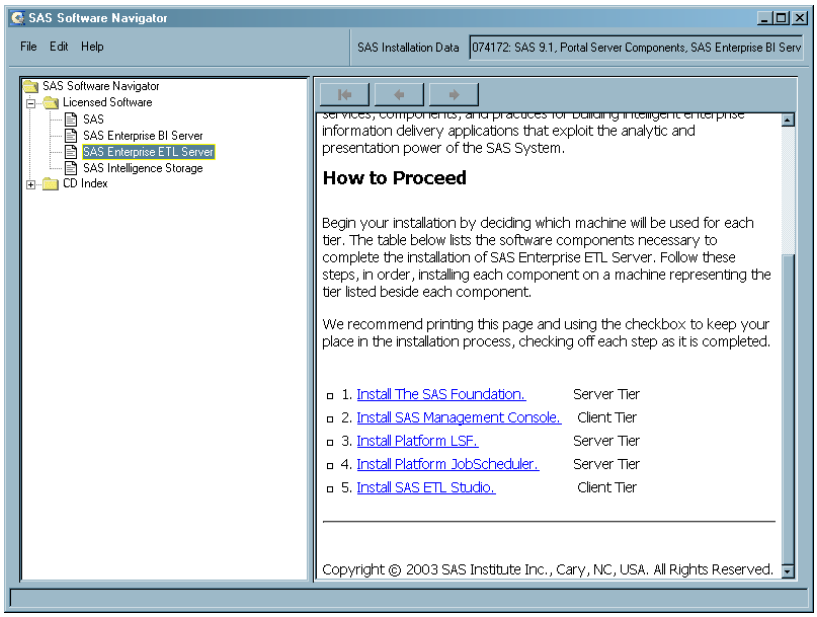


Note: The SAS Software Navigator folder in the left pane contains two subfolders: a Licensed Software folder and a CD Index folder. △

5 Expand the Licensed Software folder to display a list of the technology packages that you have licensed for the current machine.



6 Select the name of the technology package from which you want to install the required products. In the right pane, you will see a list of the required products in the order in which you should install them.



Note: To install additional products on this machine, you can perform a CD Index install, as explained in "CD Index Installs" on page 325. △

7 Click the link for the product that you want to install, and a new HTML page will appear in the right pane. This is the same page that you would see during a project installation at this point. It contains a description of the product, a link to installation instructions for the product, and a link that starts the product's installation program.

8 Read the installation instructions (if necessary), and run the installation program.

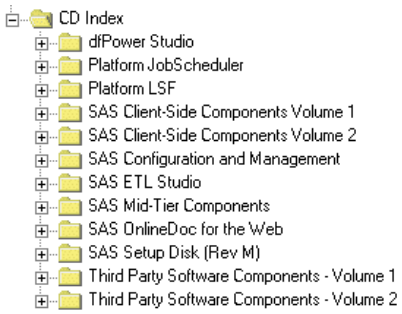
Repeat steps 7 and 8 for each product in the list.

For information about using the SAS Configuration Wizard to configure products that you have installed using a licensed-software basic install, see "Using the SAS Configuration Wizard in Basic Installs" on page 326.

CD Index Installs

The procedure for installing products using the CD Index is similar to the procedure for installing products using the Licensed Software approach. In fact, you should perform the first four steps listed in the preceding section to start a CD Index install. At the end of the fourth step, you should see the Licensed Software and CD Index folders in the left pane of the SAS Software Navigator GUI. From that point, perform the following steps:

1 Expand the CD Index folder. You should see a list of subfolders, each of which represents a CD in your Installation Kit.



2 Open a CD folder to display a list of the products on that CD. (If you do not know which CD a particular product is on, you might have to expand the folders. The names of the folders should guide your search.)

3 Select the product that you want to install. In the right pane of the SAS Software Navigator, you will see a familiar looking HTML page, the one that contains a description of the product, a link to installation instructions, and a link that starts an installation program.

4 To install the product, click the Install link and run the installation wizard.

You can install additional products in the same way.

For information about using the SAS Configuration Wizard to configure products that you have installed using a CD Index basic install, see "Using the SAS Configuration Wizard in Basic Installs" on page 326.

Using the SAS Configuration Wizard in Basic Installs

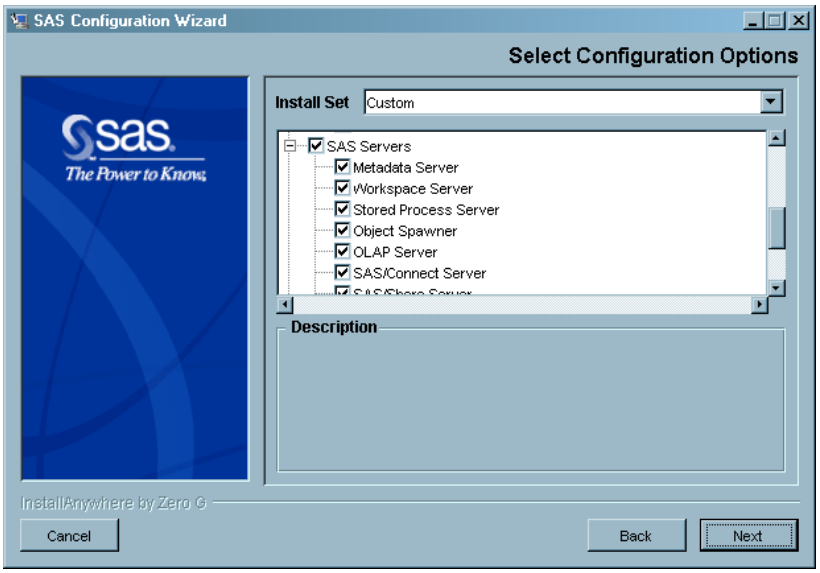
Although the SAS Configuration Wizard is designed primarily to be used in project installations, it can also be used to configure software after a basic installation. If you use the wizard in this way, it will create a configuration directory on all server hosts and provide you with instructions for configuring your software just as it would in a project installation. (For information about configuration directories, see Appendix 1, "Understanding the SAS Configuration Environment," on page 309.)

To use the SAS Configuration Wizard following a basic installation, perform these steps:

1. Start the SAS Configuration Wizard by looking in the CD Index for the CD titled **SAS Configuration and Management**, opening the folder for that CD, selecting **SAS Configuration Wizard**, and then clicking the **Configure** link in the right pane of the navigator. The SAS Configuration Wizard starts and displays a splash screen.
2. Select a language on the splash screen, and click OK. An Introduction window appears.
3. Click Next in the Introduction screen. The **Specify SAS Project Directory Location** screen appears.
4. Because you did not use a project directory for your installation, you do not need to supply any information in the Specify SAS Project Directory Location window. However, you cannot leave the **Specify Directory Location** text box empty. In that field, enter any path that is not a full path to a directory that contains a file named **plan.xml**, and click Next. You will see a message saying that there is no planning file at the location that you indicated. The message will also ask if you want to continue. Click Yes. The Specify Configuration Name window displays.
5. In the Specify Configuration Name window, enter in the **Configuration Name** text box the name you want the SAS Configuration Wizard to use for your configuration directory. Then click Next. The Select Install Set window appears.

6. In the Select Install Set window, you choose how you want to specify which products and servers you want the SAS Configuration Wizard to configure. In the Select Install Set window, select Custom. Selecting this option indicates that you

want to choose a set of products to configure from a list of all configurable products. Then click **Next**. The Select Configuration Options window displays.



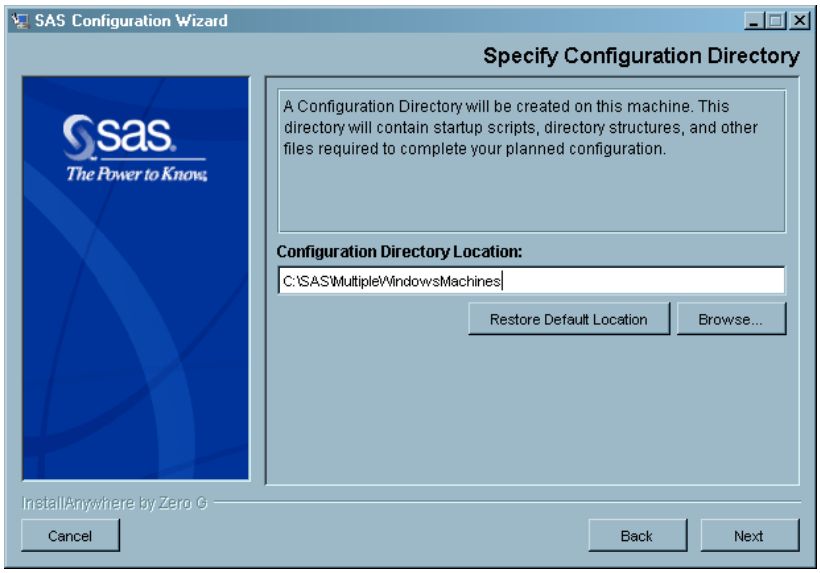
This window displays a list of the components that the SAS Configuration Wizard can configure:

- SAS Clients
  - Enterprise Miner
- SAS Mid-Tier
  - SAS Web Infrastructure Kit
  - Web Services for Java
  - Web Services for .NET
  - Apache Tomcat
- SAS Servers
  - Metadata Server
  - Workspace Server
  - Stored Process Server
  - Object Spawner
  - OLAP Server
  - SAS/CONNECT Server
  - SAS/SHARE Server
  - Batch Server
  - Job Scheduler
- SAS Services
  - SAS Foundation Services

7 Deselect the check boxes for the products and servers that you do not want to configure on the current machine.

Note: Always deselect the **Planned** check box because you are not doing a planned configuration. △

Then, click Next. The Specify Configuration Directory window displays.



From this point on, your interaction with the SAS Configuration Wizard is the same as it is for a project installation. Go to step 5 in "Running the Configuration Wizard on Windows and UNIX Systems" on page 135 for further step-by-step instructions.

APPENDIX

3

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 ETL Studio: User's Guide*
- *SAS Integration Technologies: Administrator's Guide*
- *SAS Integration Technologies: Developer'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 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 logins 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 evaluating rules to determine which users have which permissions for which resources. For example, an authorization rule can specify that a particular user has read and write permissions for a specific database table.

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.

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 (DBMS)
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.

database server
a server that provides relational database services to a client. Oracle, DB/2 and Teradata are examples of relational databases.

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 and 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 does not depend on other repositories. A foundation repository is often used to specify metadata for global resources. For example, a foundation repository could define most of the servers that are used in several data warehouses in a test environment.

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 set of protocols that are 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.

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 sizes, 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.

mode of execution
a method of executing or interacting with SAS software, which can include noninteractive mode, batch mode, interactive mode (using the SAS windowing environment or other graphical user interfaces), and interactive line mode.

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.

n-tier architecture
a type of network architecture that is used in the development of relational business applications. N-tier architecture separates an application's interface, its business logic, and its databases into components, or tiers. This approach enables the tiers to interact with each other in multiple configurations. The tiers can easily be used and reused in new combinations in order to meet dynamic business requirements.

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 Universal Unique Identifier (UUID).

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 domain and must include credentials that are appropriate for the systems to which the login provides access.

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 result 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 result packages remain in existence even after the stored process completes execution and the client disconnects from the server. See also transient result package.

pool
a group of server connections that can be shared and reused by multiple client applications.

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.

process flow
See process flow diagram.

process flow diagram
in SAS ETL 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 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 will participate in the system, about the software to be installed on each host, and about which SAS servers 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 .

project repository
in the SAS Open Metadata Architecture, a metadata repository that enables a specified person to add or update metadata that is under metadata source control.

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 set package
in the Publishing Framework, a container for content that was produced by a SAS program or by a third-party application. Package contents can include SAS files (SAS catalogs; SAS data sets; various types of SAS databases, including cubes; and SAS SQL views), binary files (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).

result type
the kind of output that is produced by a SAS Stored Process. Result types include none, streaming, permanent result package, and transient result 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 stored 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 IOM object
in the IOM object hierarchy, a root object that represents a single session within SAS. In function, a SAS IOM object is similar to the Application object, which is available in many Microsoft products.

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 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. Store d 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 ETL Studio generates code for a job, it uses SAS/CONNECT software to submit code to remote computers. SAS ETL 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 context
in SAS Management Console, the top-level object in the metadata for SAS servers. The server context specifies metadata that applies to all of the logical servers and server components that it contains. See also logical server, server component.

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.

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
in SAS ETL Studio, a table, view, or file from which you will extract information. Sources can be in any format that SAS can access, on any supported hardware platform. The metadata for a source is typically an input to a job.

source designer
in applications that support the SAS Open Metadata Architecture, a type of wizard that enables you to enter the metadata for a source.

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 SAS Stored Process result in which 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
in SAS ETL Studio, a table, view, or file that contains information that has been extracted from a source. Targets can be in any format that SAS can access, on any supported hardware platform. A target is an output of a job.

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 ETL 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 result 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 result package disappears. See also permanent result 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 type 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).

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

P

R

S

T

U

W

Z

Your Turn

If you have comments or suggestions about *SAS 9.1 Intelligence Architecture: Planning and Administration Guide*, 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
email: `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
email: `suggest@sas.com`