



SAS Publishing



SAS[®] 9.1.3 Intelligence Platform

Planning and Administration Guide

The correct bibliographic citation for this manual is as follows: SAS Institute Inc. 2004. *SAS® 9.1.3 Intelligence Platform: Planning and Administration Guide*. Cary, NC: SAS Institute Inc.

SAS® 9.1.3 Intelligence Platform: Planning and Administration Guide

Copyright © 2004, SAS Institute Inc., Cary, NC, USA

All rights reserved. Produced in the United States of America. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, or otherwise, without the prior written permission of the publisher, SAS Institute Inc.

U.S. Government Restricted Rights Notice. Use, duplication, or disclosure of this software and related documentation by the U.S. government is subject to the Agreement with SAS Institute and the restrictions set forth in FAR 52.227-19 Commercial Computer Software-Restricted Rights (June 1987).

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

1st printing, August 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.

SAS® and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Contents

PART 1 Before You Begin 1

Chapter 1 \triangle SAS Intelligence Platform 3

- Introduction to the SAS Intelligence Platform 3
- SAS Intelligence Value Chain 4
- Architecture of the SAS Intelligence Platform 9
- Advantages of Using the SAS Intelligence Platform 12

Chapter 2 \triangle Servers in the SAS Intelligence Platform 15

- Overview of Servers 15
- The SAS Application Server 15
- The Java Application Server 18

Chapter 3 \triangle Clients in the SAS Intelligence Platform 21

- Overview of Clients 21
- SAS Add-In for Microsoft Office 21
- SAS Enterprise Guide 22
- SAS Enterprise Miner 23
- SAS ETL Studio 23
- SAS Information Delivery Portal 24
- SAS Information Map Studio 25
- SAS OLAP Cube Studio 25
- SAS Web Report Studio 26

Chapter 4 \triangle Deployment Process 27

- Overview of the Deployment Process 27
- Deployed Architecture 27
- Types of Deployment 30
- Deployment Process 31

Chapter 5 \triangle Security Overview 35

- Introduction to the Security Overview 35
- Authentication in the SAS Intelligence Platform 35
- Authorization in the SAS Intelligence Platform 38
- Guide to Security Administration Activities 39

PART 2 Installation and Configuration 45

Chapter 6 \triangle Pre-Installation Tasks 47

- Overview of Pre-Installation Tasks 47
- Setting Up Your Project Directory 48
- Pre-Installation Checklists 49

Setting Up Required User Accounts	68
Servers Required to Run SAS Web Applications	71
Default Ports	74
What's Next?	77
Chapter 7 △ Installing and Configuring Your Software	79
Overview of Installing and Configuring Your Software	79
Starting the SAS Software Navigator	81
Creating a SAS Software Depot	84
Installing Software on a Machine	87
Configuring a Machine	100
Checking Your Metadata for Required Objects	111
Chapter 8 △ Troubleshooting Your Initial Setup	113
Overview of Troubleshooting Your Initial Setup	114
Troubleshooting SAS Servers	114
Troubleshooting Web Servers and Web Applications	122
Chapter 9 △ Post-Configuration Tasks	129
Overview of Post-Configuration Tasks	129
Understanding the State of Your System	130
Tasks That You Might (or Will) Need to Perform	134
Establishing Basic Protections	140
Ongoing Administration and Maintenance	143

PART 3 **Security Administration** **145**

Chapter 10 △ Understanding Authentication	147
Introduction to Understanding Authentication	147
Authentication Concepts and Terminology	148
The Authentication Process	152
Examples: Using Authentication Domains	161
Examples: Accessing SAS Servers	165
Examples: Accessing Third-Party Servers	170
Chapter 11 △ Understanding Authorization	175
Introduction to Understanding Authorization	175
Authorization Concepts and Terminology	176
Authorization Layers	176
Permissions in the Metadata Layer	177
Access Controls in the Metadata Layer	179
Identity Hierarchy in the Metadata Layer	186
Principles of Access Control Precedence	187
Authorization Decision Process	189
Chapter 12 △ Developing Your Security Plan	193
Overview of Security Planning	193

Defining Your Security Goals	194
Making Preliminary Decisions about Your Security Architecture	194
Planning Your Users	195
Planning Your User Groups	201
Planning Your Access Controls	204

Chapter 13 △ **Implementing Security** 213

Overview of Implementing Security	213
Protecting the Foundation Repository	214
Setting Up Security for Administrators	215
Securing ACTs and User-Defined Groups	217
Setting Up Security for Regular Users	218
Security Maintenance Activities	225

PART 4 **Data Administration** 233

Chapter 14 △ **Preparing Data for Use** 235

Overview of Preparing Data for Use	235
Understanding the Data Storage Options	236
Defining Metadata about the Data	239
Preparing for Cube Loading	248
Securing Access to the Metadata That Defines the Data	253

Chapter 15 △ **Optimizing Data Storage** 257

Overview of Optimizing Data Storage	257
Compressing Data	258
Indexing Data	259
Sorting Data	261
Buffering Data	263
Using Threaded Reads	264
Building Cubes from Star Schemas	265
Validating SPD Engine Hardware Configuration	265
Building Optimized Cube Aggregations	265
Optimizing Performance of a SAS OLAP Server	269
Setting LIBNAME Options That Affect Performance	271

PART 5 **Application Administration** 279

Chapter 16 △ **Administering SAS ETL Studio** 281

Overview of Administering SAS ETL Studio	281
Connecting to SAS Servers	282
Connecting to Data Servers	286
Setting Up Change Management	287
Using Custom-Tree Folders for Security	292
Importing and Exporting SAS Code Transformations	295
Importing and Exporting Metadata	295

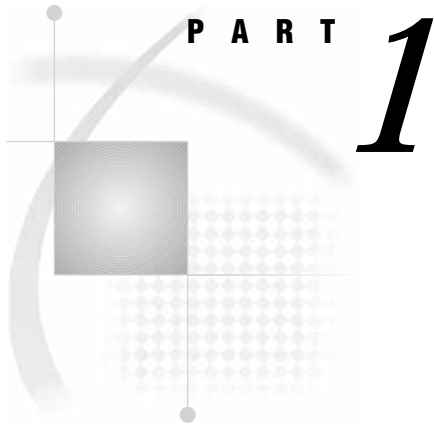
Testing the Job Scheduler	297
Setting Up a SAS Data Quality Server	298
Enabling Status Code Handling	302
Chapter 17 △ Managing the Reporting Environment	305
Overview of Reporting in the SAS Intelligence Platform	306
Managing Information Maps	308
Managing Reports	310
Managing Stored Processes	332
Securing Your Reporting Environment	335
Delivering Reports	338
Chapter 18 △ Preparing SAS Enterprise Miner for Use	341
Overview of Preparing SAS Enterprise Miner for Use	341
Configuring SAS Enterprise Miner	342
Customizing SAS Workspace Server Settings	348
Setting Required Variables in UNIX Shell Scripts	351
Customizing the Apache Tomcat HTTP Server	351
Securing SAS Enterprise Miner Metadata	354

PART 6 **Advanced Topics** **359**

Chapter 19 △ Configuring Your Servers for Better Performance	361
Overview of Configuring Your Servers for Better Performance	362
Tuning a Workspace Server for Use with SAS Web Report Studio	363
Workspace Server Pooling for SAS Web Report Studio and SAS Information Delivery Portal	366
Tuning Your J2EE Server or Servlet Container for Use with SAS Web Report Studio and SAS Information Delivery Portal	373
Load Balancing Workspace Servers for Desktop Applications	379
Overview of the Initial Load Balancing Setup for Stored Process Servers	383
Load Balancing Stored Process Servers on Multiple Hosts	385
Chapter 20 △ Promoting and Replicating Metadata	393
Overview of Promoting and Replicating Metadata	393
Preparing for Replication and Promotion	394
Creating a Promotion Job	407
Creating a Replication Job	415
Troubleshooting Replication and Promotion	420
Chapter 21 △ Managing an Environment	421
Overview of Managing an Environment	421
Customizing the Properties of a New Environment	421
Manually Changing the Properties of an Existing Environment	423
Adding to an Environment	425
Re-Creating an Existing Environment	426
Uninstalling an Environment	427

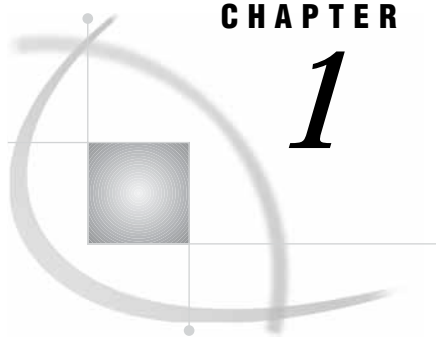
PART 7 **Appendixes** **429**

Appendix 1 △ Understanding the SAS Configuration Environment	431
Overview of Understanding the SAS Configuration Environment	431
The Basic Concepts	431
The Main Directory Structure	433
The Lev1 Directory Structure	434
Default Directory Permissions	440
Appendix 2 △ Software Index Installations	443
Overview of Software Index Installations	443
Software Index Installations	444
Using the SAS Configuration Wizard in Software Index Installations	448
Appendix 3 △ Upgrading a SAS 9.1 or 9.1.2 System to a SAS 9.1.3 System	453
Overview of Upgrading a SAS 9.1 or 9.1.2 System to a SAS 9.1.3 System	454
Upgrading in Place	454
Upgrading After Testing in a Test Environment	456
Upgrading Each Machine	458
Upgrading a SAS 9.1 or 9.1.2 z/OS System to a SAS 9.1.3 System	467
Appendix 4 △ Recommended Reading	473
Recommended Reading	473
Glossary	475
Index	TBD



Before You Begin

<i>Chapter 1</i>	SAS Intelligence Platform	<i>3</i>
<i>Chapter 2</i>	Servers in the SAS Intelligence Platform	<i>15</i>
<i>Chapter 3</i>	Clients in the SAS Intelligence Platform	<i>21</i>
<i>Chapter 4</i>	Deployment Process	<i>27</i>
<i>Chapter 5</i>	Security Overview	<i>35</i>



CHAPTER

1

SAS Intelligence Platform

<i>Introduction to the SAS Intelligence Platform</i>	3
<i>SAS Intelligence Value Chain</i>	4
<i>Products in Each Link</i>	5
<i>Plan Link</i>	5
<i>ETLQ Link</i>	6
<i>Intelligent Storage Link</i>	6
<i>Business Intelligence Link</i>	6
<i>Analytic Intelligence Link</i>	7
<i>SAS Management Console</i>	7
<i>Sample Business Intelligence Value Chains</i>	8
<i>Example 1: Building a Data Warehouse and Creating Reports</i>	8
<i>Example 2: Embedding SAS Reports in Microsoft Word Documents</i>	9
<i>Architecture of the SAS Intelligence Platform</i>	9
<i>SAS Foundation</i>	10
<i>SAS Business Intelligence Infrastructure</i>	10
<i>SAS Foundation Servers</i>	11
<i>SAS Foundation Services</i>	11
<i>SAS Application Services</i>	11
<i>SAS Client Services</i>	11
<i>Advantages of Using the SAS Intelligence Platform</i>	12
<i>Manageability</i>	12
<i>Scalability</i>	12
<i>Interoperability</i>	12
<i>Usability</i>	13

Introduction to the SAS Intelligence Platform

Your job is to build 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” below.

Second, all of these applications are part of what SAS calls the SAS Intelligence Platform. This platform includes 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 “Architecture of the SAS Intelligence Platform” on page 9.

SAS Intelligence Value Chain

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

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

Figure 1.1 SAS Intelligence Value Chain



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

The *Plan* link in the SAS Intelligence Value Chain delivers proven best-practice roadmaps that are supported by integrated industry-specific analytic models, project methodologies, and consulting expertise. 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 in order 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 a SAS data warehousing solution.

The *ETL*^o link is the stage at which you create a data warehouse from existing 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 reviewing and cleansing the data so that it is accurate, up-to-date, and consistently represented.

The *Intelligent Storage* link is the stage at 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 Information Technology retains control over the quality and consistency of the data that they are using. The products in this area also enable you to build Web portals that guide users to the information that they need.

Finally, the *Analytic Intelligence* link provides capabilities such as predictive and descriptive modeling, forecasting, optimization, simulation, and experimental design. 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. △

Products in Each Link

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 have to make decisions about the platforms on which you will host your solution and about the SAS software technology packages and products that your solution will require. Your needs might be met by a standard deployment plan—one of a set of plans that describe typical installations—or you and your SAS representative might use a SAS planning application to create a customized deployment plan. Both types of plans are represented by a planning file called `plan.xml`, which 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 *software index installations*. △

If your planning activities include data modeling, you will probably use a third-party product such as AllFusion ERwin Data Modeler to build your 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 such as 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. Δ

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 Surveyor applications enable you to build SAS ETL Studio jobs that help you read data in ERP systems from other vendors such as SAP, Siebel, and Oracle.

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 such as 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 Platform JobScheduler product enables 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
- SAS 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. SAS SPD Engine tables are managed by SAS 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: SAS SPD Engine is part of Base SAS. A higher-performance multi-user version of this product is available as a separate product and is included in the Intelligent Storage technology package. It is called the SAS Scalable Performance Data Server (SPD Server). Δ

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 and data visualization. The main products that fall into this category are listed in Table 1.1.

Table 1.1 Business Intelligence Products

Product	Description
SAS Information Map Studio	A Java application that creates information maps. These maps are user-friendly metadata definitions of physical data sources. These metadata definitions enable your business users to query a data warehouse to meet specific business needs, without needing to know how to retrieve the data from various sources.
SAS Web Report Studio	A Java 2 Enterprise Edition (J2EE) Web application that enables users to create reports and view reports from a Web browser. The input to the Web-reporting application is an information map.
SAS Information Delivery Portal	A J2EE Web application that enables users to aggregate data from a variety of sources and present the data in a Web browser. The content might include the output of SAS Stored Processes, links to Web addresses, documents, syndicated content from information providers, SAS information maps, and SAS reports.
SAS Enterprise Guide	A Microsoft Windows application for analyzing data that also enables users to create SAS Stored Processes and to store that code in a repository available to a SAS Stored Process Server. (Stored processes are SAS programs that are stored on a server and executed by client applications.) Stored processes are used for Web reporting and analytics, among other things.
SAS Add-In for Microsoft Office	A Windows application that enables you to embed SAS reports and analyses in Microsoft Word documents and Microsoft Excel spreadsheets.

Analytic Intelligence Link

SAS has multiple analytic-intelligence products, for areas such as the following:

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

In this document, we deal primarily with the SAS Enterprise Miner product. 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 objects that represent entities such as those mentioned previously. Among other tasks, SAS Management Console enables you to

- create metadata repositories and operate the metadata server
- create users and groups (in the metadata), 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.

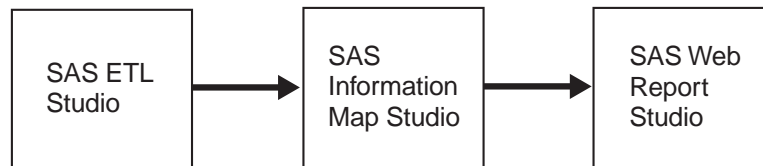
Sample Business Intelligence Value Chains

The best way to understand the benefit of the SAS Intelligence Value Chain is to look at a couple of examples of product chains that you might create. The first example below illustrates the chain that you might employ if you wanted to build a data warehouse and a set of data marts that were based on that warehouse. You could then create reports that were based on information in the data marts. The second example assumes that you have already built a data warehouse and data marts, and that you want to use SAS Stored Processes to create reports that users can view in Microsoft Word documents.

Example 1: Building a Data Warehouse and Creating Reports

To create the first example, you might use the links that are shown in the following figure in your chain of solutions.

Figure 1.2 Building a Data Warehouse and Creating Reports



This list explains what each product does:

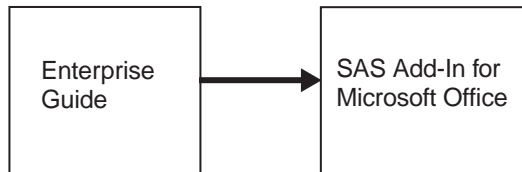
- 1 At the beginning of the process, your company has operational data that is stored in a variety of data sources. After your SAS applications are in place, a data warehousing specialist can use SAS ETL Studio to extract data from those data sources, transform the data as necessary, and load it into the warehouse, which could be a library of SAS data sets. While performing this work, the warehouse specialist defines metadata for the data sources, for the target library, and for the target data sets. (The other applications in the chain use metadata in an analogous way.) A data mart designer can then use SAS ETL Studio to read data from the data warehouse and to store a subset of the data in an OLAP cube.
- 2 The OLAP cube then serves as input to the business analyst who is using SAS Information Map Studio. Using this product, the analyst creates a view of the cube that serves as the basis for reports. (This information map is represented only by metadata because it is a logical entity.)
- 3 A report creator then uses SAS Web Report Studio to create reports that are based on an information map. This user can design the layout of the report.

At the end of the process, what was operational data has been distilled into a concise report that can be used for decision support.

Example 2: Embedding SAS Reports in Microsoft Word Documents

Here is an example of a short solution chain. Assume that your data warehouse and data marts are already in place and that you want to run a SAS program to read data from the warehouse or from a mart and to create a report. You then want to display that report in a Microsoft Word document. The following figure shows the chain of SAS solutions that you might employ.

Figure 1.3 Embedding SAS Reports in Microsoft Word Documents



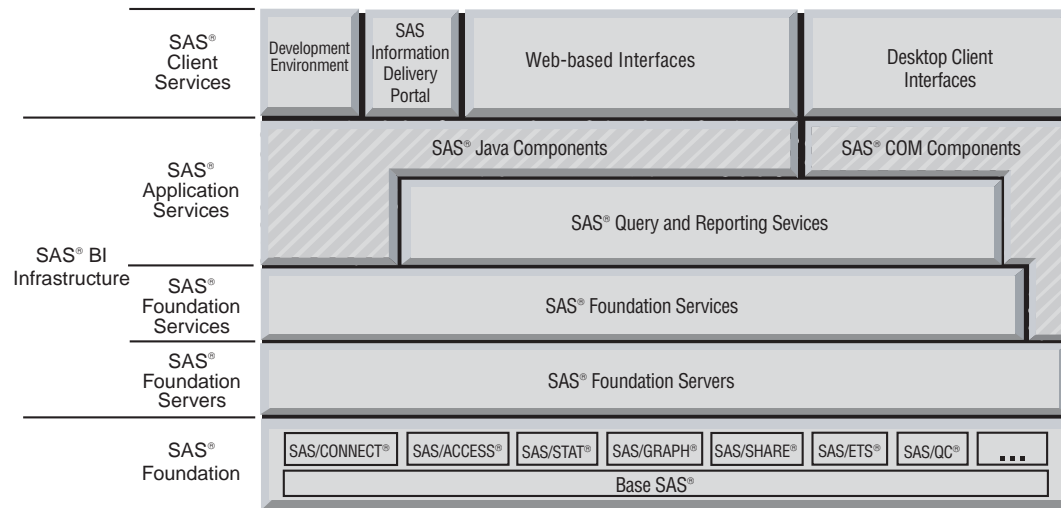
This list explains what each product contributes to this solution:

- 1 SAS Enterprise Guide enables users to write SAS Stored Processes. A stored process is a SAS program that
 - resides on a server
 - can be parameterized
 - can stream its output to another application.
- 2 SAS Add-In for Microsoft Office enables users to execute stored processes from a Microsoft Word document or from a Microsoft Excel spreadsheet and to include the output of those stored processes in the document or spreadsheet.

Architecture of the SAS Intelligence Platform

SAS creates and delivers enterprise intelligence through the SAS Intelligence Platform. This software 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 that is necessary to ensure the consistent and reliable enterprise-wide intelligence that is needed for exploring, analyzing, optimizing, and understanding your data.

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

Figure 1.4 Software Architecture for the SAS Intelligence Platform

SAS Foundation

The SAS Foundation layer consists of SAS products such as 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 Business Intelligence Infrastructure

The SAS Business Intelligence (BI) Infrastructure layer provides a suite of servers and services that deliver SAS computing power 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 such as SAS Enterprise Guide by 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 enables client applications to submit SAS code to a SAS session via an API.

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 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 Client Services

The SAS 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.

Advantages of Using the SAS Intelligence Platform

If you have built intelligence solutions using earlier versions of SAS software, you will notice a number of improvements in SAS 9 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 SAS 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 execute SAS code. In addition, an object spawner can balance a workload across multiple workspace or stored process servers.

Interoperability

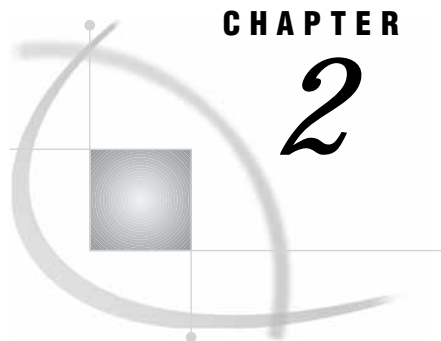
SAS 9.1 provides interoperability between SAS and third-party applications through the following:

- the SAS Open Metadata Architecture and the Common Warehouse Metamodel (CWM), which provide 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.

- the Integrated Object Model (IOM), which enables the use of industry-standard languages, programming tools, and communication protocols in the development of client programs that access services on IOM servers.
- SAS Foundation Services, which enables Java programmers to write distributed applications that are integrated with the SAS platform.
- SAS Publishing Framework, which enables the user to publish information and events proactively using a subscription channel model.
- the Application Messaging interface, which enables the user to incorporate messaging services into SAS programs.
- the SAS Web Infrastructure Kit, which enables Web applications and components to be developed using portal technology.
- the Directory Services interface, which enables LDAP directory services functions to be incorporated into client SAS programs.
- SAS/ACCESS software, which provides an interface between the SAS System and relational database management systems (DBMSs).

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 only a single interface to do his or her work. For instance, the ETL specialist might need only to learn the SAS ETL Studio interface. Using this client, the ETL specialist can create data warehouses and data marts. Likewise, portal designers will need only the SAS Information Delivery Portal to perform their work.



CHAPTER

2

Servers in the SAS Intelligence Platform

<i>Overview of Servers</i>	15
<i>The SAS Application Server</i>	15
<i>SAS Servers</i>	17
<i>Logical SAS Servers</i>	17
<i>Purpose of the SAS Application Server</i>	18
<i>The Java Application Server</i>	18
<i>Servlet Container</i>	19
<i>J2EE Server</i>	19

Overview of Servers

For high-level depictions of how servers participate in the SAS Intelligence Platform, see “Architecture of the SAS Intelligence Platform” on page 9 and “Deployed Architecture” on page 27. As these figures suggest, even a relatively simple SAS Business Intelligence system contains a number of servers. For example, a typical deployment includes these components:

- A metadata server that writes metadata objects to, and reads metadata objects from, SAS metadata repositories. These metadata objects contain information about all of the components of your system, such as users, groups, data libraries, and servers.
- One or more SAS data servers, such as the SAS Base Engine, the SAS Scalable Performance Data Server, or the SAS OLAP Server. The data servers can also be third-party products. For detailed information about data servers, see “Understanding the Data Storage Options” on page 236.
- At least one SAS application server, which consists of a set of SAS servers that execute SAS code. For example, a SAS application server can contain a stored process server, a workspace server, and a SAS OLAP Server.
- A Java application server, which supports SAS Web applications by providing an execution environment for servlets and Enterprise JavaBeans.

The following topics describe the purpose and components of the SAS application server and the Java application server.

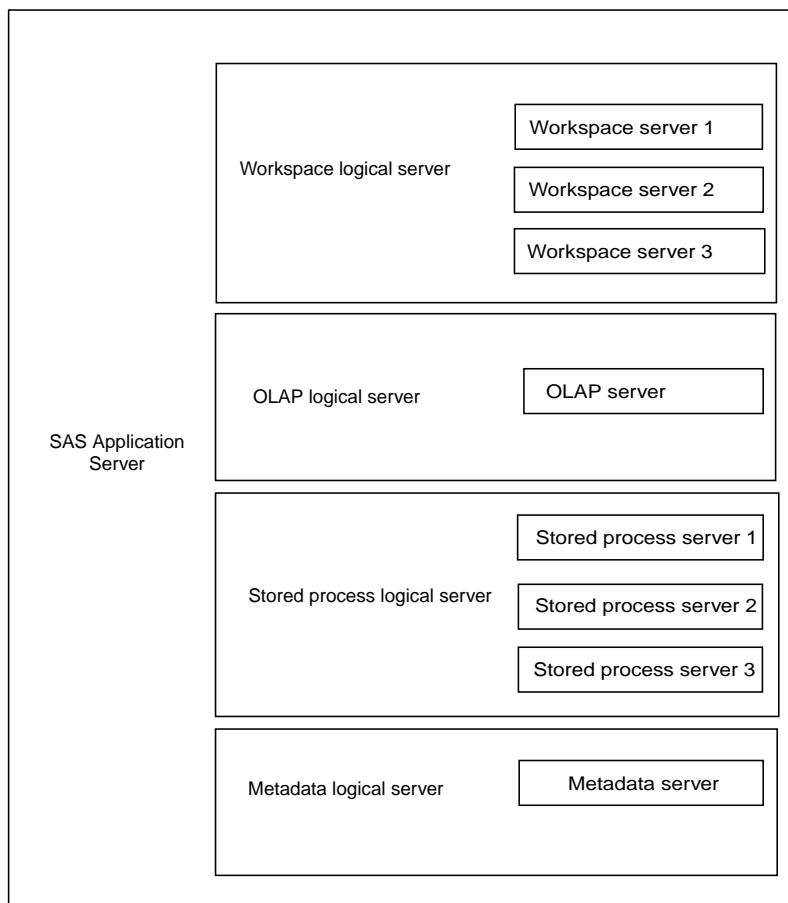
The SAS Application Server

A SAS application server is a logical entity: it is a collection of SAS servers that perform special types of processing. For example, the workspace server and stored

process server—which were introduced in “SAS Foundation Servers” on page 11—might belong to such a collection. When a SAS ETL Studio job generates the code that is necessary to perform an ETL operation, the application submits that code to a workspace server for execution. Similarly, if a SAS Web Report Studio user creates a report and the creation of the report requires the execution of a stored process—SAS code stored in a repository—that code is executed by a stored process server.

There is also an intermediate level of organization called a *logical server*. SAS servers of a particular type are grouped in a logical server of the corresponding type. For example, one or more workspace servers might be grouped in a logical workspace server. The logical servers are then grouped into a SAS application server. See the following figure.

Figure 2.1 SAS Application Server Components



The SAS application server might contain one or more logical servers, up to one for each type of server. For a list of the SAS server types, see “SAS Servers.”

The following subsections explain why each level in this hierarchy exists:

- “SAS Servers” on page 17 enumerates the different types of SAS servers and explains how these servers are represented in the metadata.
- “Logical SAS Servers” on page 17 explains why servers are organized into logical servers.
- “Purpose of the SAS Application Server” on page 18 explains why logical servers are organized into application servers.

SAS Servers

As mentioned previously, a SAS application server is an aggregation of specialized SAS servers. The types of servers that might be part of a SAS application server are shown in the following table.

Table 2.1 SAS Servers

Server Type	Description
Workspace server	Fulfills client requests for a specific SAS session. For example, SAS ETL Studio submits requests to a SAS Workspace Server to initiate SAS sessions for building OLAP cubes that are based on metadata in a SAS Metadata Repository.
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.
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.
SAS/CONNECT server	Provides computing resources on remote machines where SAS Integration Technologies is not installed.
Metadata server	Provides access to the metadata in one or more metadata repositories. <i>Note:</i> When you install your system, you will configure a metadata server. You do <i>not</i> need to specify that this server is a component of a particular application server at that point. That step becomes necessary only when you prepare to copy metadata from one repository to another.
Batch server	Gives you the ability to execute code in batch mode. There are three types of batch servers: DATA Step batch servers, Java batch servers, and Generic batch servers. The DATA Step server enables you to run SAS DATA steps and procedures in batch mode. The Java server enables you to schedule the execution of Java code, for example, the code that creates a SAS Marketing Automation marketing campaign. The Generic server supports the execution of any other type of code.

Because these SAS servers actually execute code, they are represented in the metadata by objects that contain information such as

- the name of the machine that is hosting the server
- the TCP/IP port or ports on which the server will listen for requests
- the SAS command that will be used to start the server.

Logical SAS Servers

When you define the first SAS server of a particular type, such as a workspace server, a logical server of the same type is created automatically. You can later define additional servers of the same type within the original logical server (with the exception that a logical OLAP server can contain only one OLAP server). But why is the logical server created in the first place?

There are a number of reasons, but here are a couple of the most important. Probably the most important reason—at least for readers of this guide—is that a logical

server provides a place for you to set up load balancing (for workspace and stored process servers). That is, a logical workspace server might contain several physical workspace servers, and you have the ability to specify that a workload be distributed across these servers according to an algorithm that you specify.

Note: The load balancer runs in the *object spawners* that are used to start the workspace (or stored process) servers. For further information about object spawners, see “Spawner Overview” in the *SAS Integration Technologies: Server Administrator’s Guide* at support.sas.com/rnd/itech/doc9/admin_oma/sasserver/iombridge/sp_ovrvw.html. \triangle

A logical workspace server also enables you to create a pool of workspace server processes, and connections to those processes, for use with SAS Information Delivery Portal or SAS Web Report Studio. If more than a few users will be working with these Web applications, it is essential for performance reasons that you set up workspace pooling.

A logical server is also a place at which you can define access controls for a set of servers. That is, by granting a group of users access to a logical stored process server, you might be granting those users access to several stored process servers running on different hosts.

Purpose of the SAS Application Server

We have already seen that an application server includes a set of logical servers. But why is the application server—the outermost container—necessary? The first part of the answer is that you might want to have two or more sets of logical servers, and you need names by which you can refer to these sets of servers.

Who uses these names? First, SAS applications can specify that they want a particular application server to execute their SAS code. For instance, SAS ETL Studio enables an ETL developer to define a default SAS application server. After a developer defines this default server, when he or she runs an ETL job, the generated SAS code for the job will be executed by a workspace server that belongs to the default application server.

This brings up another question. “Why should the ETL developer care which application server runs a job?” The answer is that, as the system administrator for a business intelligence system, you define the resources that will be available to a particular SAS application server. For example, when you define the metadata for a SAS library, you *assign that library* to an application server. When you do this, you are guaranteeing that the resource will be available if a client uses the application server that you specified to access this resource. The same rule applies to other resources as well, such as database schema. Note that assigning a resource to an application server does not literally make that resource available to users. It is merely a way of signifying your intent to make the resource available. You actually make the resource available by using mechanisms like metadata access controls and operating system permissions.

The Java Application Server

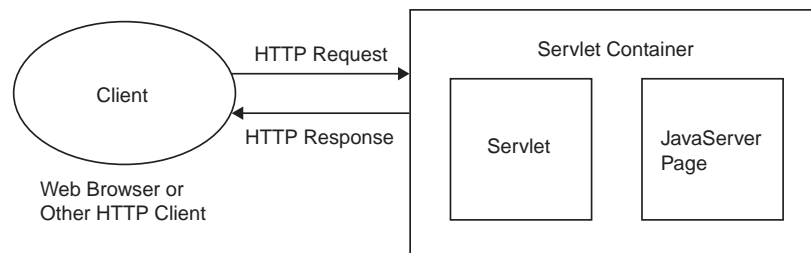
The Java application server executes the Java code in SAS applications that are implemented as Java 2 Enterprise Edition (J2EE) Web applications or J2EE enterprise applications. For example, SAS Web Report Studio and SAS Information Delivery Portal are J2EE Web applications, and the SAS Solutions are J2EE enterprise applications. What is the difference between these two types of applications? A J2EE Web application is built by using Java servlet technology and is delivered as a WAR

(Web archive) file (a ZIP file that contains all of the files that make up the application and whose name ends with the extension `.war`). A J2EE enterprise application, on the other hand, uses not only Java servlet technology but other technologies—in particular, Enterprise JavaBeans—and is delivered as an EAR (enterprise archive) file. This is an important distinction because the two types of Java Web applications have resulted in there being two types of Java application servers: one type for executing J2EE Web applications and one type for executing both J2EE Web applications and J2EE enterprise applications. A Java application server that executes only J2EE Web applications is called a *servlet container*, or a *Web container*, and an application server that can execute enterprise applications is called a *J2EE server*.

Servlet Container

As its name implies, a servlet container provides the execution environment for servlets. It also provides the execution environment for JavaServer Pages (JSPs) because JSPs are translated to servlets. See the following figure.

Figure 2.2 Servlet Container



A Java Virtual Machine in the servlet container executes the Web application's Java code. It also provides additional services. For example:

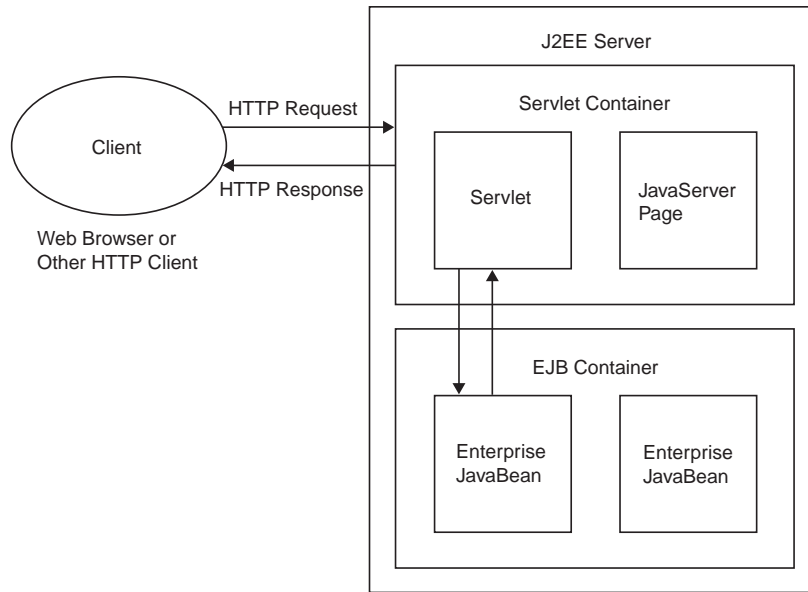
- 1 When a client sends an HTTP request to the application, the servlet container packages the contents of that request as a Java object and passes the object to the method of the servlet that will process the request.
- 2 The container provides for session management. That is, the container recognizes that a sequence of HTTP requests are coming from a single client and provides for the storage of data between requests.

The only servlet container that can participate in the SAS Intelligence Platform is Apache Tomcat, which is available free of charge. This is an excellent product that provides the reference implementation for Sun Microsystem's servlet and JSP specifications. However, it does have a few limitations. As mentioned earlier, you cannot run enterprise applications in this container. Also, the product might not meet the scalability and security requirements of a large enterprise. As a result, we recommend that you use Apache Tomcat only in small systems and in system prototypes.

J2EE Server

A J2EE server includes a servlet container, but also contains an Enterprise JavaBean container. See the following figure.

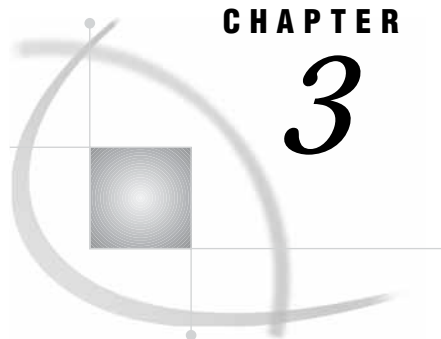
Figure 2.3 J2EE Server



Just as the servlet container provides an execution environment for servlets, the Enterprise JavaBean container provides an execution environment for Enterprise JavaBeans (which often contain the business logic of an application). The Enterprise JavaBean container also provides a set of services for the enterprise beans, including security and transaction management.

Most SAS business-intelligence systems include such a J2EE server. This type of server is required if you will be running any SAS solutions, and the enterprise features of the J2EE server make it a good investment. The currently supported J2EE servers are the BEA WebLogic Server and the IBM WebSphere Application Server.

For information about the currently supported versions of these products, see the SAS Third Party Software Downloads page at support.sas.com/thirdpartysupport.



CHAPTER

3

Clients in the SAS Intelligence Platform

<i>Overview of Clients</i>	21
<i>SAS Add-In for Microsoft Office</i>	21
<i>SAS Enterprise Guide</i>	22
<i>SAS Enterprise Miner</i>	23
<i>SAS ETL Studio</i>	23
<i>SAS Information Delivery Portal</i>	24
<i>SAS Information Map Studio</i>	25
<i>SAS OLAP Cube Studio</i>	25
<i>SAS Web Report Studio</i>	26

Overview of Clients

This chapter introduces the SAS Intelligence Platform clients and explains how each client interacts with the SAS servers. The following table lists the clients by type.

Table 3.1 SAS Intelligence Clients

Java Applications ¹	Web Applications ²	Windows Applications
SAS ETL Studio	SAS Web Report Studio	SAS Enterprise Guide
SAS OLAP Cube Studio	SAS Information Delivery Portal	SAS Add-In for Microsoft Office
SAS Information Map Studio		
SAS Enterprise Miner		

1 Require a JRE on each client machine.

2 Require a Web browser on each client machine and a J2EE server on the middle-tier machine where the application will run.

For information about how each type of client participates in the SAS Intelligence Platform, see “Architecture of the SAS Intelligence Platform” on page 9 and “Deployed Architecture” on page 27.

SAS Add-In for Microsoft Office

SAS Add-In for Microsoft Office is a Component Object Model (COM) add-in that enables you to run stored processes and SAS tasks from Microsoft Office. With the SAS

Add-In for Microsoft Office, you can embed SAS reports and analyses in Microsoft Word documents and Microsoft Excel spreadsheets. The following table describes how SAS Add-In for Microsoft Office interacts with each SAS server.

Table 3.2 How SAS Add-In for Microsoft Office Interacts with SAS Application Servers

Server	Interaction
SAS Metadata Server	Read metadata for repository objects such as stored processes, libraries, tables, SAS Workspace Servers, and SAS Stored Process Servers.
SAS Workspace Server	Execute SAS code and queries through execution of SAS tasks. Access resources such as tables.
SAS Stored Process Server	Execute stored processes and collect resulting output.

SAS Enterprise Guide

SAS Enterprise Guide is a SAS application that runs on the Microsoft Windows platform. SAS Enterprise Guide 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 SAS server.

Table 3.3 How SAS Enterprise Guide Interacts with SAS Servers

Server	Interaction
SAS Metadata Server	Read and write metadata for stored process repository objects.
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	Create new stored processes and edit existing stored processes.**

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

** Stored process execution is not supported in SAS Enterprise Guide 2.1.

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

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 SAS server.

Table 3.4 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 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 packages.

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

SAS ETL Studio

SAS ETL Studio is a Java application that enables you to create and manage 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 SAS server.

Table 3.5 How SAS ETL Studio Interacts with SAS 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.

Server	Interaction
SAS/CONNECT Server	Submit generated SAS code to machines that are remote from the default SAS application server. Gain interactive access to remote libraries.
SAS Stored Process Server	Create and execute stored processes and collect resulting output.

For more information about SAS ETL Studio, see the SAS ETL Studio Help, which is available from within the product and the *SAS ETL Studio: User's Guide*.

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 SAS server.

Table 3.6 How SAS Information Delivery Portal Interacts with SAS 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. View information maps that are based on relational data.
SAS Stored Process Server	Execute stored processes with streaming results.
SAS OLAP Server*	View information maps that are based on cubes.

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

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: Server Administrator's Guide*, which are part of the SAS Integration Technologies documentation available at support.sas.com/rnd/itech/library/library9.html.

SAS Information Map Studio

SAS Information Map Studio is a Java application that enables you to create and manage SAS Information Maps, which are business metadata about your physical data. Information maps enable you to surface your data warehouse data in business terms that typical business users understand, while storing key information that is needed to build appropriate queries. The following table describes how SAS Information Map Studio interacts with each SAS server.

Table 3.7 How SAS Information Map Studio Interacts with SAS Servers

Server	Interaction
SAS Metadata Server	<p>Read and write metadata for repository objects such as information maps.</p> <p>Read metadata for objects such as libraries, tables, and SAS Workspace Servers, SAS Stored Process Servers, and SAS OLAP Servers.</p>
SAS Workspace Server	<p>Generate and submit SAS code or access data for a query.</p> <p>Access resources such as tables.</p> <p>Identify and run stored processes when information maps that are associated with stored processes are used in a query.</p>
SAS OLAP Server*	<p>Generate cube-based information maps or run cube-based queries.</p>

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

For more information about SAS Information Map Studio, see the SAS Information Map Studio Help, which is available from within the SAS Information Map Studio 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 a specified location. The following table describes how SAS OLAP Cube Studio interacts with each SAS server.

Table 3.8 How SAS OLAP Cube Studio Interacts with SAS 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 the *SAS OLAP Server Administrator's Guide*.

SAS Web Report Studio

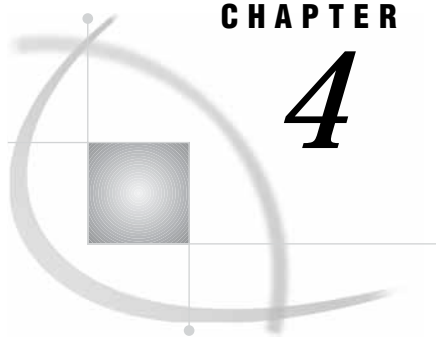
SAS Web Report Studio is a Web application that enables you to issue queries and view reports from a Web browser. The input to the application is an information map that is created using SAS Information Map Studio. The following table describes how SAS Web Report Studio interacts with each SAS server.

Table 3.9 How SAS Web Report Studio Interacts with SAS Servers

Server	Interaction
SAS Metadata Server	Read and write metadata for repository objects such as reports. Read metadata for objects such as information maps, stored processes, and images.
SAS Workspace Server	Access resources such as tables.
SAS OLAP Server*	Access cube data and process MDX queries.
SAS Stored Process Server	Execute stored processes and collect resulting output.

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

For more information about using SAS Web Report Studio, see the SAS Web Report Studio Help, which is available from within the product. For information about administrative tasks associated with SAS Web Report Studio, see Chapter 17, “Managing the Reporting Environment,” on page 305.



CHAPTER

4

Deployment Process

<i>Overview of the Deployment Process</i>	27
<i>Deployed Architecture</i>	27
<i>Client Tier</i>	28
<i>Middle Tier</i>	29
<i>Server Tier</i>	30
<i>Data Tier</i>	30
<i>Types of Deployment</i>	30
<i>Deployment Process</i>	31
<i>Planning</i>	33
<i>Installation</i>	33
<i>Configuration</i>	33

Overview of the Deployment Process

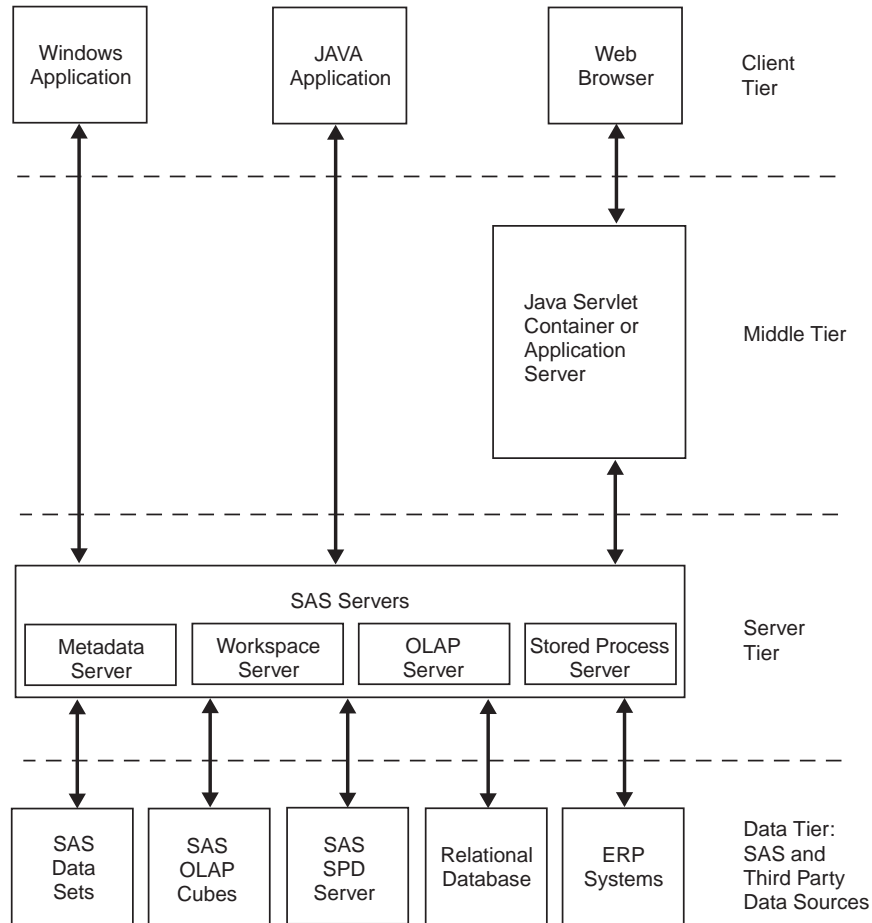
This chapter provides an overview of the process of deploying your business intelligence platform. You will find detailed information on deploying your software in Part 2 “Installation and Configuration.”

The chapter contains three main sections:

- “Deployed Architecture” on page 27 explains what a deployed system looks like if you view your SAS applications as n-tiered applications. It discusses which products and components are installed in each tier.
 - “Types of Deployment” on page 30 introduces the three types of installations that the SAS software installation tool enables you to perform.
 - “Deployment Process” on page 31 provides an overview of the procedure that you will use to install and configure your software. As you will see, the main topics in this section cover planning your system, installing your software, and configuring your software.
-

Deployed Architecture

From your perspective as an administrator, it is 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 the following figure.

Figure 4.1 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 clients are Windows applications and run only on Microsoft Windows systems:

- SAS Enterprise Guide
- SAS Add-In for Microsoft Office.

The Java applications include the following:

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

You can run SAS Management Console on Windows systems, Solaris, HP-UX Itanium, and AIX. The remaining applications are supported only 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 Foundation on a host, the JRE will already be present on that machine. Otherwise, you can install the JRE from a CD that is supplied by SAS before you install the first Java client.

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

- SAS Web Report Studio
- SAS Information Delivery Portal.

These products run in a servlet container or J2EE server on the middle 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.

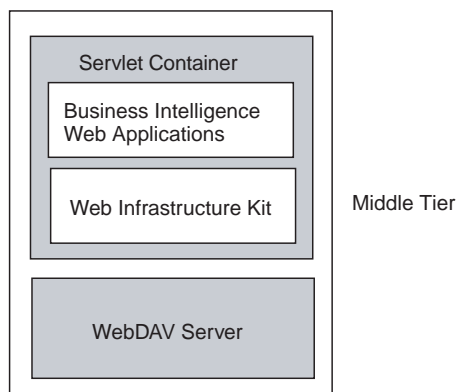
Middle Tier

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

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

See the following figure.

Figure 4.2 Middle 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 content that users might want to access through SAS Web Report Studio or the SAS Information Delivery Portal. The server acts like a network accessible file system and can store such content as documents, report definitions, and images.

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 uses a different set of Integrated Object Model (IOM) interfaces and has a different purpose.

- 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 SAS code submitted by a client application.
- The SAS Stored Process Server executes stored processes, which are SAS programs stored on a server where they can be called by clients.
- The SAS OLAP Server handles MDX (multidimensional expression language) queries.

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 Data Surveyor products provide direct access to ERP systems such as the following:

- SAP
- Oracle Applications
- Siebel
- PeopleSoft.

Types of Deployment

When you deploy your business intelligence system, you install your software using a tool called the SAS Software Navigator. This tool enables you to perform three types of installations:

- Advanced installations

- Personal installations
- Software Index installations.

When you initially set up your system, you generally perform either an Advanced installation or a Personal installation. Both types are considered *planned installations* because they depend on the existence of a plan—an XML file—that describes the machines in your system, the software to be installed on each machine, and the components that need to be configured on each machine. Plans come in two types: standard plans and customized plans. Standard plans are plans that SAS has developed to cover the most common configurations. A customized plan is a plan that a SAS representative and a customer develop to handle that customer's special requirements.

An Advanced installation is the most common type of installation. In an Advanced installation, you can install software on a single machine or on multiple machines, and you can use a standard plan or a customized plan. A Personal installation is more restrictive: it enables you to install software of a single machine using a standard plan. For more information about the steps in the deployment process for these types of installations, see “Deployment Process.”

The Software Index installation is designed primarily for the installation of a single product, or of a few products. It is used most often to add software to an existing SAS system. When you perform a Software Index installation, you do not follow a plan; rather, you can choose to install any product from any CD that you have received from SAS, as long as your license entitles you to install that product. For more information about Software Index installations, see Appendix 2, “Software Index Installations,” on page 443.

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.

There are three general types of activities that you must perform:

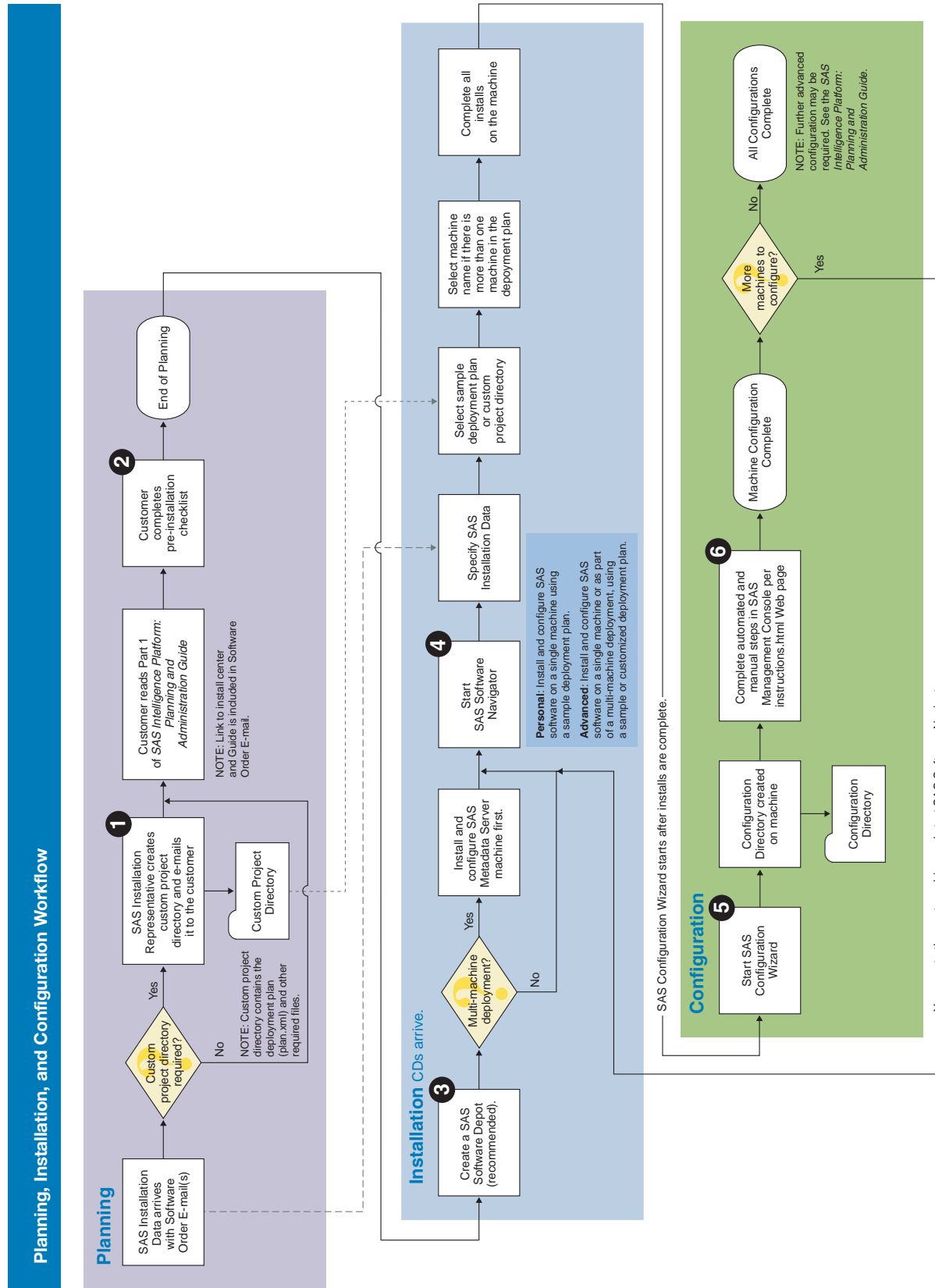
- Planning your installation. We recommend that you and your SAS representative choose or create a deployment plan for your site. This plan contains information about
 - the different machines in your system
 - the software to be installed on each machine
 - which machine you should install software on first, second, and so on
 - 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 starts, or enables you to start, an installation program for each product to be installed. (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 after you have installed all of the software on a machine. There will be some manual configuration that you need to perform as well.

The following figure shows the procedure that you will use to build your system. The sections that follow the flowchart provide annotations to the flowchart and direct you to additional information about important topics. (You can find an easier-to-read version of this flowchart in your Installation Kit, near the end of the “Getting Started” section.)

Figure 4.3 Planning, Installation, and Configuration Workflow



Note: You follow a different procedure to install the SAS 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 9.1.3 Foundation for z/OS*. For configuration information, see Chapter 6, “Pre-Installation Tasks,” on page 47 and Chapter 7, “Installing and Configuring Your Software,” on page 79. △

Planning

For a planned installation (an Advanced or Personal installation), you need to perform the following tasks:

- If you and your SAS representative created a customized plan for your deployment, place your planning file and related files in a shared directory, which is 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 an unplanned installation. △

❶ If you are using a customized plan, before you begin your installation, you must create a project directory and place your planning file in it. Both the SAS Software Navigator and the SAS Configuration Wizard read files from this directory. For more information about setting up a project directory, see “Setting Up Your Project Directory” on page 48.

❷ You must also fill out a pre-installation checklist (or checklists). This checklist explains the set of tasks that you must perform before you begin installing your SAS software. One such task is setting up operating system user accounts that can be used to authenticate special users. Other tasks might include installing the Java 2 SDK, a J2EE server, and a WebDAV server. For more information on pre-installation tasks, see the following sections:

- “Pre-Installation Checklists” on page 49
- “Setting Up Required User Accounts” on page 68
- “Servers Required to Run SAS Web Applications” on page 71.

Installation

❸ We recommend that before you begin installing your software, you create a SAS Software Depot by copying the contents of your SAS CDs to a network drive. 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 84.

❹ 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 start an installation program. For further information about installing software, see “Installing Software on a Machine” on page 87.

Configuration

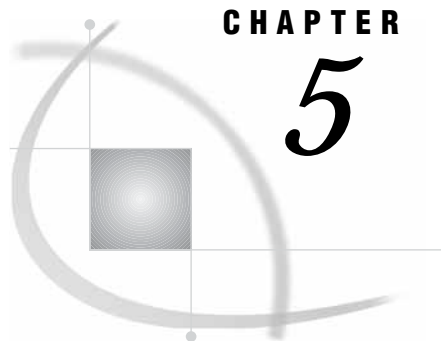
After you have installed all of the software on a machine, you run the SAS Configuration Wizard to configure the software on that machine.

❺ On each machine in your deployment where you installed a configurable component, 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 100
- “Configuring SAS Servers on z/OS Systems” on page 106
- “Configuration Directory: Server-Tier Machines” on page 130
- “Configuration Directory: Middle-Tier Machines” on page 132.

⑥ The SAS Configuration Wizard also produces an HTML document that explains which configuration you need to perform manually after the wizard exits. Often, this configuration involves adding objects to your metadata repository using SAS Management Console or a script. For further information on this subject, see “Performing the Steps Listed in instructions.html” on page 108.



CHAPTER

5

Security Overview

<i>Introduction to the Security Overview</i>	35
<i>Authentication in the SAS Intelligence Platform</i>	35
<i>Why and When Identities Are Verified</i>	35
<i>How Identities Are Verified</i>	36
<i>How User IDs and Passwords Are Acquired</i>	36
<i>Automated Support for Identity Management</i>	37
<i>Authorization in the SAS Intelligence Platform</i>	38
<i>Metadata-Based Authorization</i>	38
<i>Multiple Authorization Layers</i>	38
<i>Guide to Security Administration Activities</i>	39
<i>Security Tasks During Deployment Planning</i>	39
<i>Security Tasks During Pre-Installation and Configuration</i>	39
<i>Security Tasks During Post-Configuration</i>	41
<i>Security Tasks for Establishing Basic Protections</i>	43
<i>Security Tasks for Expanding Your Deployment</i>	44

Introduction to the Security Overview

This chapter introduces the SAS 9 security model by describing how authentication and authorization work in the SAS Intelligence Platform. This chapter also outlines the security administration activities for a deployment of the SAS Intelligence Platform.

Authentication in the SAS Intelligence Platform

Authentication is an identity verification process that attempts to determine whether users (and other entities) are who they say they are. This section provides an overview of how the SAS Intelligence Platform manages authentication. For a comprehensive discussion of this subject, see Chapter 10, “Understanding Authentication,” on page 147.

Why and When Identities Are Verified

Your identity is the basis for decisions about which actions you are permitted to perform with which resources. For this reason, your identity is verified first when you log on to a SAS application and again each time you make a request that requires access to an additional server such as a SAS OLAP Server, a third-party database server, a SAS Stored Process Server, or a SAS Workspace Server.

For example, when you log on to SAS ETL Studio, the metadata server wants to know who you are so it can determine which libraries, stored processes, and jobs should be displayed in your desktop client. If you then make a request in SAS ETL Studio to run a job against an Oracle table, the Oracle server will want to know who you are so it can determine whether you have access to the data in that table.

How Identities Are Verified

In the SAS Intelligence Platform, authentication to a SAS server usually consists of direct verification against user accounts that have been established in the host operating system. In this process, each server that you attempt to access asks its host operating system whether your credentials (your user ID and password) correspond to a valid user account for that computer. For example, before allowing you to run a stored process, a stored process server will check with its host computer in order to determine whether your credentials correspond to a valid operating system account on that computer.

Note: Identity verification is not always performed on an individual basis. For example, access to a pooled server is determined by group memberships rather than by individual user accounts on the pooled server.* You can also choose to use shared accounts to provide access to a server. \triangle

As an alternative to direct verification, authentication can be based on a trust relationship. In a trust relationship, a server trusts verification that has already been performed by another component in the architecture. By default, the SAS Intelligence Platform supports these trust relationships:

- The metadata server trusts the identity verification that the SAS OLAP Server performs.
- The metadata server trusts the identity verification that a connecting SAS process has performed (this is called a trusted peer session connection).
- The metadata server trusts the identity verification that the Web server performs (if you configure your SAS Web applications to use Web server authentication).

As an alternative to using the host operating system accounts for validation, authentication can be performed against Lightweight Directory Access Protocol (LDAP) accounts or Microsoft Active Directory accounts. However, the following limitations apply:

- The only SAS servers that support LDAP and Active Directory are the SAS OLAP Server and the SAS Metadata Server. The SAS Stored Process Server and the SAS Workspace Server must use the host operating system to verify user identities.
- LDAP and Active Directory are used only to verify identities — these technologies are not an alternative to storing user information in a SAS metadata repository. Regardless of whether you authenticate users against the host operating system or LDAP or Active Directory, you must also store user information in a SAS metadata repository to support the security model's credential management and authorization features.

How User IDs and Passwords Are Acquired

As the previous section suggests, a server must somehow acquire your credentials before those credentials can be verified against a list of valid accounts. This section

* For more information, see "Accessing a Pooled SAS Workspace Server" on page 167.

describes the mechanisms by which servers in the SAS Intelligence Platform obtain your credentials.

When you log on to a SAS application, you interactively submit your credentials. After that, SAS applications use credential management features to provide your user ID and password to each server that must verify your identity. The primary credential management feature is the ability to store your account information in the metadata repository. An application can retrieve your credentials from the metadata repository and provide those credentials to a target server. In the metadata repository, passwords are stored in an encrypted format. For more information, see “Using Credentials That Are Stored in the Metadata” on page 157.

The following additional credential management features might enable you to reduce the amount of user account information that you store in the metadata repository.

- Many SAS applications can cache the credentials that you submit when you log on and then provide those credentials to a target server. For more information, see “Using Cached Credentials” on page 158.
- SAS Web applications can share user and session contexts. This enables a client that you launch from within another client to reuse your existing context rather than prompting you to log on. For more information, see “Sharing User Context” on page 160.
- Some SAS applications can prompt you for your credentials for accessing SAS servers if those credentials are not stored in the metadata repository.

To learn which credential management features work with each client, see Table 10.3 on page 161.

Automated Support for Identity Management

The SAS 9 security model requires you to create and maintain user information in the metadata repository. This metadata identity information is used by the security model’s credential management and authorization features. As an alternative to performing identity management tasks manually using the SAS Management Console, you can use the following batch processes:

- To load user information into the metadata repository, you first extract user and group information from one or more enterprise identity sources. Then you use SAS bulk-load macros to create identity metadata from the extracted information. SAS provides sample applications that extract user and group information and logins from an Active Directory domain controller and from UNIX `/etc/passwd` and `/etc/group` files.
- To periodically update user information in the metadata repository, you first use SAS macros to compare that information to the information in your external enterprise identity source. You then use SAS macros to update the information in the metadata repository to reflect any changes that are found.

Note: You cannot use these batch processes to manage passwords. Users can manage their own passwords with the SAS Personal Login Manager. △

The macros and sample applications for these batch processes are documented in “Creating and Maintaining User and Group Definitions” in the *SAS Metadata Server: Setup and Administration Guide* at support.sas.com/rnd/eai/openmeta/v9/setup/authmacros.html.

Authorization in the SAS Intelligence Platform

Authorization is the process of determining which users have which permissions for which resources. This section provides an overview of how authorization works in the SAS Intelligence Platform. For a comprehensive discussion of this subject, see Chapter 11, “Understanding Authorization,” on page 175.

Metadata-Based Authorization

The SAS Intelligence Platform includes an authorization mechanism that consists of access controls that you define and store in a metadata repository. These metadata-based controls enable you to manage access to metadata and, in some cases, to the computing resources that the metadata represents.

The available metadata-based permissions are summarized in the following table.

Table 5.1 Metadata-Based Permissions

Permissions	Use
ReadMetadata, WriteMetadata, CheckInMetadata	Use to control user interactions with a metadata object.
Read, Write, Create, or Delete	Use to control user interactions with the underlying computing resource that is represented by a metadata object.
Administer	Use to control administrative interactions (such as starting or stopping) with the SAS server that is represented by a metadata object.

The consequences of granting or denying a metadata-based permission vary depending on factors such as these:

- Whether you assign the permission to an individual user or to a user group. The metadata-based authorization mechanism includes an identity precedence ranking that is based on your group membership hierarchy.
- Whether you assign the permission by applying a pre-existing permission pattern. The metadata-based authorization mechanism enables you to set controls by using either ad-hoc assignments or reusable patterns.
- Whether you set the permission on an object from which other objects can inherit access controls. The metadata-based authorization mechanism includes a set of rules that determine which objects can be parents to which other objects.
- Whether the application that the requesting user is using enforces the permission. In the current release, not all applications enforce the metadata-based Read, Write, Create, Delete, and Administer permissions.

Multiple Authorization Layers

Your ability to perform a particular action is determined not only by these metadata-based access controls but also by external authorization mechanisms such as operating system permissions and database controls. In order to perform a particular action, you must have the necessary permissions in *all* of the applicable authorization layers.

For example, regardless of the access controls that have been defined for you in the metadata repository, you cannot access a particular file if the operating system permissions do not permit the action.

Guide to Security Administration Activities

This section outlines the security administration activities for the SAS Intelligence Platform and can be used as a roadmap for the security aspects of a deployment. The following topics list the security related tasks for each phase of the deployment process — from planning through establishing protections after installation is complete. Each topic explains why the tasks are necessary and indicates where you can find documentation that will help you understand and complete each task.

Security Tasks During Deployment Planning

Planning a deployment of the SAS Intelligence Platform includes the security related tasks that are described in the following table.

Table 5.2 Security Tasks During Deployment Planning

Task Description and Purpose	Information Sources
Understand the security model. In order to plan a successful deployment, you must have at least a basic understanding of the security model's capabilities and requirements.	"Authentication in the SAS Intelligence Platform" on page 35 and "Authorization in the SAS Intelligence Platform" on page 38 provide a basic description of the security model.
Make some preliminary decisions about your security architecture. This will help you incorporate authentication considerations into your plan for what software components will run on which computers under which operating systems.	"Making Preliminary Decisions about Your Security Architecture" on page 194 includes a list of the issues that you should address. Chapter 10, "Understanding Authentication," on page 147 contains background information for this task.

After you complete these early stage security planning tasks, you can either begin the installation process or you can do some further security planning to prepare for the security activities that occur in later phases of the deployment.

Security Tasks During Pre-Installation and Configuration

The pre-installation and configuration processes includes the security related tasks that are described in the following table.

Table 5.3 Security Tasks During Pre-Installation and Configuration

Task Description and Purpose	Information Sources
<p>Create the required user accounts in the operating system. Each of the accounts serves a specific purpose. You create the accounts before you begin installing software because information about these accounts is incorporated into the customized configuration files and instructions that are generated during the installation process.</p>	<p>“Setting Up Required User Accounts” on page 68 explains how each account is used.</p> <p>“Pre-Installation Checklists” on page 49 provides instructions for creating the accounts.</p>
<p>Create the required identities in the metadata repository. Most of these metadata identities correspond to the required user accounts that you created in the operating system. Creating these identities in the metadata makes the necessary credentials available to each of the required user accounts. Creating these identities in the metadata also enables you to give the necessary metadata layer access controls to each of the required user accounts.</p>	<p>Instructions for creating these identities (either manually or by running a script) are in the customized instructions.html file that is generated for you during the installation process.</p> <p>“Checking Your Metadata for Required Objects” on page 111 summarizes the metadata identities that are created.</p> <p>“Authentication Concepts and Terminology” on page 148 contains background information about metadata identities, authentication domains, and logins.</p>
<p>Create an initial authentication domain in the metadata repository.¹ By assigning server definitions and certain user credentials to this authentication domain, you will create associations in the metadata between servers and the user credentials that are valid for those servers.</p>	<p>Instructions for creating this authentication domain are included in the instructions for creating the required metadata identities. The instructions.html file also specifies which logins and servers should be associated with this authentication domain.</p> <p>“Authentication Concepts and Terminology” on page 148 contains background information about metadata identities, authentication domain, and logins.</p>

Task Description and Purpose	Information Sources
Expand the default access that certain administrative and service metadata identities have to the metadata repository. This helps to ensure that these identities will retain the access that they need, even after you restrict access to the repository at a later stage in the deployment.	<p data-bbox="919 212 1458 365">Instructions for adding these default authorization settings (either manually or by running a script) are in the customized instructions.html file that is generated for you during the installation process.</p> <p data-bbox="919 394 1458 485">“Repository Level Access Controls” on page 185 explains how default access to the repository is managed.</p> <p data-bbox="919 514 1458 604">“Managing Access to Server Definitions” on page 229 provides an example of why this access is needed.</p>
Verify that your servers and Web applications are working. If you encounter problems at this stage, those problems are often the result of an error or omission in your user and group set up, either in the operating system or in the metadata.	<p data-bbox="919 621 1458 711">Verification instructions are in the customized instructions.html file that is generated for you during the installation process.</p> <p data-bbox="919 741 1458 831">“Overview of Troubleshooting Your Initial Setup” on page 114 explains how to diagnose and fix common problems.</p> <p data-bbox="919 861 1458 1014">“The Authentication Process” on page 152 contains conceptual information that will help you understand the authentication requirements for logging on to a particular application or accessing a particular server.</p>
1 In the next stage, you will be instructed to create additional authentication domains as appropriate for your environment.	

After you complete the installation and configuration process, you will be instructed to complete any additional configuration steps that are necessary to further customize your deployment for your environment. In most cases, you will need to perform one or more post-configuration tasks before your deployment is fully functional. The next section outlines the security aspects of those tasks.

Security Tasks During Post-Configuration

After you complete the installation and configuration process, you will probably need to make some changes to further customize your security configuration for your environment. The following table describes the security tasks that are involved with the more common customizations.

Table 5.4 Security Tasks During Post-Configuration

Task Description and Purpose	Information Sources
<p>If your deployment includes multiple operating systems, alternative authentication providers, or third-party database systems, then you might need to create additional authentication domains.¹ You must also create associations in the metadata that use authentication domains to tie each server to the user credentials that are valid for that server.</p>	<p>“Managing Authentication Domains” on page 230 explains how to create authentication domains and change the authentication domain assignment for a login or server.</p> <p>“Authentication Concepts and Terminology” on page 148 explains the relationships between authentication domains, servers, and logins.</p> <p>“Examples: Using Authentication Domains” on page 161 illustrates authentication domains for several different environments.</p> <p>“Examples: Accessing Third-Party Servers” on page 170 demonstrates how authentication domains are used to support access to third party servers.</p>
<p>If you want to use Web server authentication to verify the identity of users who log on with SAS Web applications, you must make changes to the property files for those applications.²</p>	<p>“Initial Authentication on a Middle Tier Server” on page 155 explains how this authentication process works.</p> <p>For instructions, see “Defining Users (Web Server Authentication (Trusted Realm))” in the <i>SAS Web Infrastructure Kit: Administrator’s Guide</i> at support.sas.com/rnd/itech/doc9/portal_admin/security/ag_user_trust.html.</p>
<p>If your deployment includes OLAP data, then you must grant additional metadata layer permissions to the users who interact with that data, and to the service account that supports this access.</p>	<p>“Managing Access to OLAP Data” on page 229 explains how you can provide the necessary access to OLAP data.</p>
<p>If your deployment uses the SAS metadata LIBNAME engine, then you must give users who will interact with data using this engine additional metadata layer permissions to that data. You must also give addition permissions to any service accounts that support this access.</p>	<p>“Defining Data Sources with the Metadata LIBNAME Engine” on page 245 explains how this engine is used and provides a reference to the user’s guide that documents the necessary metadata layer permissions for interacting with data through the metadata LIBNAME engine.</p>

Task Description and Purpose	Information Sources
If your deployment includes a Xythos server, then you must understand the security aspects of integrating that server with your SAS applications.	Integration with the SAS Information Delivery Portal is described in "Implementing Authentication and Authorization for the Xythos WFS WebDAV Server" in the <i>SAS Integration Technologies: Server Administrator's Guide</i> at support.sas.com/rnd/itech/doc9/admin_oma/security/security_accessdav.html .
If you want to set up server pooling, you must understand the authentication aspects of that configuration.	<p>"Accessing a Pooled SAS Workspace Server" on page 167 explains how users access a pooled server.</p> <p>"Workspace Server Pooling for SAS Web Report Studio and SAS Information Delivery Portal" on page 366 provides background information on server pooling.</p>
<ol style="list-style-type: none"> 1 During deployment planning, you determined how many authentication domains you need in your environment. 2 This is another one of the preliminary security architecture decisions that you made during deployment planning. 	

After you complete any necessary customizations, you will be instructed to set up some basic protections for your deployment. These activities are outlined in the next section.

Security Tasks for Establishing Basic Protections

It is recommended that you set up some basic protections immediately after you complete your other configuration tasks. You will be instructed to perform the tasks that are described in the following table.

Table 5.5 Security Tasks for Establishing Basic Protections

Task Description and Purpose	Information Sources
Select and implement an encryption technology and level to control how data is protected while in transit across the network.	"Setting an Encryption Method" on page 141 contains background information and instructions for this task.
Define appropriate operating system protections for your configuration directories.	"Protecting the Metadata Repository and Configuration Directories" on page 140 contains information to help you with this task.
Define some initial protections in the metadata authorization layer.	"Protecting the Foundation Repository" on page 214 explains how to set some initial repository level controls.
Add some administrative users to the deployment. This enables each administrator to use his or her own account to perform most tasks, rather than continuing to share the highly privileged unrestricted user account.	"Setting Up Security for Administrators" on page 215 provides a process and instructions for adding these users.

At this point your deployment includes only a few users and some basic security protections. The remaining tasks are to define any additional access controls that you

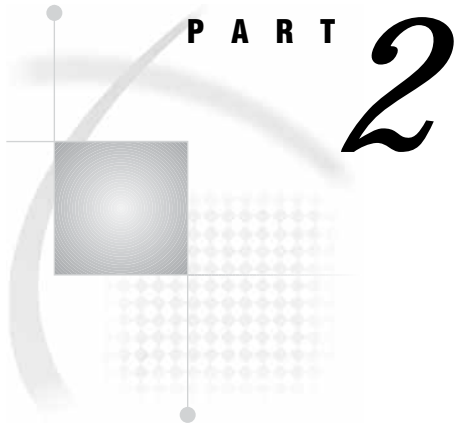
need in order to meet your security goals and to populate your deployment with regular (non-administrative) users. These tasks are outlined in the next section.

Security Tasks for Expanding Your Deployment

Adding users and access controls to your deployment involves these activities:

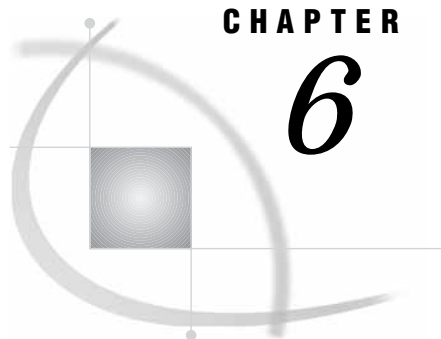
- identifying your security goals
- planning the user groups and access controls you will use to achieve those goals
- setting up user groups to support granular access controls
- defining those controls in the metadata layer
- adding users to the deployment
- performing ongoing maintenance tasks.

Information to help you understand and perform these tasks is provided in Chapter 12, “Developing Your Security Plan,” on page 193 and Chapter 13, “Implementing Security,” on page 213.



Installation and Configuration

- Chapter 6* **Pre-Installation Tasks** 47
- Chapter 7* **Installing and Configuring Your Software** 79
- Chapter 8* **Troubleshooting Your Initial Setup** 113
- Chapter 9* **Post-Configuration Tasks** 129



CHAPTER

6

Pre-Installation Tasks

<i>Overview of Pre-Installation Tasks</i>	47
<i>Setting Up Your Project Directory</i>	48
<i>Pre-Installation Checklists</i>	49
<i>Pre-Installation Checklist for Windows</i>	50
<i>Pre-Installation Checklist for UNIX</i>	56
<i>Pre-Installation Checklist for z/OS</i>	61
<i>Setting Up Required User Accounts</i>	68
<i>Users Authenticated Only on the Metadata Server Host</i>	68
<i>SAS Administrator</i>	68
<i>SAS Trusted User</i>	68
<i>SAS Web Administrator</i>	68
<i>Users That Might Require Network Accounts</i>	69
<i>SAS Guest User</i>	69
<i>SAS Demo User</i>	69
<i>SAS General Server User</i>	69
<i>SAS Installer</i>	69
<i>How the Accounts Are Used by the SAS Servers</i>	70
<i>Setting User Rights (Windows Systems Only)</i>	71
<i>Servers Required to Run SAS Web Applications</i>	71
<i>Which Software Is Required?</i>	71
<i>Requirements for the Java 2 SDK</i>	72
<i>Requirements for a Servlet Container or J2EE Server</i>	72
<i>Requirements for a WebDAV Server</i>	73
<i>Default Ports</i>	74
<i>What's Next?</i>	77

Overview of Pre-Installation Tasks

There is 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, if necessary.
You *must* perform this step if you will be using a customized deployment plan. You will know that your site is using such a plan if you receive an e-mail from SAS that contains a deployment plan (called `plan.xml`), a set of pre-installation checklists, and related files. If you receive such an e-mail, you must set up a network-accessible shared directory and place your deployment plan in that directory. It is also a good idea to put into that directory any other files that you received with your deployment plan. The SAS Software Navigator and the SAS Configuration Wizard both use the `plan.xml` file as input. For further information, see “Setting Up Your Project Directory” on page 48.

- 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 49. If you will be using a customized deployment plan, you should also have copies of these checklists in your project directory. Most of the tasks on these checklists involve setting up required operating system user accounts and optionally setting up an infrastructure for running Web applications such as SAS Web Report Studio and SAS Information Delivery Portal. The checklists are largely self documenting; however, you should also read the following sections before filling out the checklists and performing the tasks that the checklists tell you to perform:
 - “Setting Up Required User Accounts” on page 68
 - “Servers Required to Run SAS Web Applications” on page 71.
- 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 74.

Note: This chapter is intended primarily for those of you who are preparing for an Advanced or Personal installation. However, if you are preparing for a Software Index installation (which is covered in Appendix 2, “Software Index Installations,” on page 443), 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 to review the list of default ports. △

Setting Up Your Project Directory

When your company and your SAS representative discuss the requirements for your SAS Business Intelligence Platform, they must choose a standard deployment plan or create a customized deployment plan for your site. This plan contains information about the hosts that will participate in the system, the products that will be installed on each machine, and the configuration required for each host. This plan can be a standard plan, one that might be useful at any number of sites. Or it can be a customized plan, one that is designed specifically for your site.

If your company decides to use a customized plan, your SAS representative will use a planning tool (a Web application) to help you develop the plan. This planning tool creates a number of files that will be e-mailed to you. The most important of these is the deployment plan itself, an XML file called `plan.xml`. The e-mail that you receive will also contain

- pre-installation checklists
- a list of the machines in your system
- a list of the components to be deployed and configured on your machines
- documentation on installation and configuration.

If you receive such an e-mail, you should copy all of the attached files—or the files in the attached ZIP file—to a project directory. This project directory is simply a directory that

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

When you perform your Advanced installation, both the SAS Software Navigator and the SAS Configuration Wizard will read your deployment plan from this directory. The plan will tell the navigator which components to install on each host and will tell the configuration wizard which components to configure on each machine.

Note: You might also want to copy the SAS Installation Data (SID) files that you receive from SAS to this project directory so that the file(s) will be easy to find. However, this is not a requirement. △

Pre-Installation Checklists

This section contains three pre-installation checklists: one for Windows systems, one for UNIX systems, and one for z/OS systems. If you are installing your SAS servers and your Web application infrastructure on the same platform, you will need only one checklist. If the machines in your configuration are running different operating systems, you will use multiple checklists.

For example, if you are using two UNIX machines in your configuration (one for the SAS servers and one for your Web application infrastructure), you will use the Pre-Installation Checklist for UNIX for both machines. On the other hand, if you have a UNIX machine for your SAS servers and a Windows machine for your Web application infrastructure, you will need to use the Pre-Installation Checklist for UNIX for the UNIX machine on which you will be installing your SAS servers and the Pre-Installation Checklist for Windows for the Windows machine on which you will be building your Web application infrastructure.

Before you begin to install any software (that 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 that is described in those checklists. To prepare for that process, 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 68 for additional information about setting up the user accounts that are required by the SAS Configuration Wizard
- read “Servers Required to Run SAS Web Applications” on page 71 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 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 and groups, see “Setting Up Required User Accounts” on page 68.

When creating these users, it is recommended that you:

- 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. Δ

If you are installing Platform JobScheduler: The user account used to install Platform JobScheduler is called the installation account. It does not need to be an LSF user. It must be a local administrator on all the hosts you are installing (a Windows domain administrator account is normally a local administrator on every host in the domain) and it must be granted the “act as part of the operating system” user right on Windows NT and Windows 2000.

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 this checklist.

Passwords for these user IDs cannot contain more than one “\$” character. Δ

- 1 Create a *SAS Administrator* that can be authenticated on the 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:	_____ <p>e.g., <domain>\sasadm, where <domain> is the Windows domain qualifier</p>	[]
Full Name:	_____ <p>e.g., SAS Administrator</p>	[]
Password:	_____	[]

- 2 Create a *SAS General Server* user that can be authenticated 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:	_____ <p>e.g., <domain>\sassrv, where <domain> is the Windows domain qualifier</p>	[]
Full Name:	_____ <p>e.g., SAS General Server</p>	[]
Password:	_____	[]

- 3 Create a *SAS Guest* that can be authenticated that can be authenticated 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:	_____ <p>e.g., <domain>\sasguest, where <domain> is the Windows domain qualifier</p>	[]
Full Name:	_____ <p>e.g., SAS Guest User</p>	[]
Password:	_____	[]

- 4 Create a *SAS Trusted User* that can be authenticated on your metadata server machine. Because the user ID is a trusted ID, SAS servers such as the OLAP

server, object spawner 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.

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* that can be authenticated 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 <i>SAS Server Users</i> .	[]
Add all the user IDs listed above (steps 1 to 5) to the <i>SAS Servers Users</i> group.	[]
Grant the "Log on as a batch job" permission/policy to the <i>SAS Server Users</i> group.	[]
<i>Windows NT and Windows 2000 only:</i> Grant the <i>Act as part of the operating system</i> permission/policy to <i>SAS General Server</i> . You might have to log off or reboot after granting this permission.	[]

Note: The remaining steps are required only if you will be using any of the SAS Web software (SAS Web Report Studio, SAS Information Delivery Portal, SAS Web Infrastructure Kit, SAS Solutions) in your deployment. Δ

- 7 Create a *SAS Web Administrator* that can be authenticated 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 Web Report Studio, SAS Web Report Viewer, or 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:	_____ <p>e.g., <domain>\saswbadm, where <domain> is the Windows domain qualifier</p>	[]
Full Name:	_____ <p>e.g., SAS Web Administrator</p>	[]
Password:	_____	[]

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

Table 6.8 Adding the SAS Web Users to a Group

Task	Done
Add <i>SAS Web Administrator</i> user ID to the <i>SAS Server Users</i> group.	[]

- 9 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 a servlet container (WebLogic, WebSphere, or Tomcat) from the SAS Software Navigator.

Note: When you install your servlet container, make sure that the full path to the installation directory contains NO SPACES. △

Note: You need to make sure you have the correct Java Development Kit (JDK) installed for the servlet container you are using. Review the Java environment requirements for your servlet container. Then download a JDK from the SAS support site for third-party software: support.sas.com/thirdpartysupport. Select “SAS 9.1.3. Java Development Kits (JDK).”

Installation programs attempt to determine available JDK versions by checking operating system environment variables and executable paths. When installing software that requires a JDK, check to see if the detection process has obtained the correct location. If not, correct the location during the installation. You can determine the default Java version in a shell script by typing the command: `java -version`

In some operating system environments, you might need to temporarily set the environment variable for the Java install directory just for the duration of the installation program or script.

Some Java versions might require patches to the operating system. Consult the Java installation documentation. △

Table 6.9 Servlet Container Information

Servlet Container Information		Done
Server Container Provider:	_____ <p style="text-align: center;">e.g., Tomcat</p>	[]
Servlet Container Version:	_____ <p style="text-align: center;">e.g., 4.1.18</p>	[]
Server Container Machine:	_____ <p style="text-align: center;">e.g., myServer.myCompany.com</p>	[]
Servlet Container Location:	_____ <p style="text-align: center;">e.g., C:\Tomcat4.1</p> <p style="text-align: center;"><i>Note: You must change the installation location to contain NO SPACES.</i></p>	[]

10 If you have already installed a WebDAV (Web Based Distributed Authoring and Versioning) Server, fill in the details in the table below. Note that if you plan to use the Xythos Webfile Server (WFS) as your WebDAV server, you should install the version supplied with your SAS software (4.0.48), and you must install the SAS User Management Customization software as well.

If you have not already installed a WebDAV Server, skip this step now and install an Apache Server or a Xythos Server from the SAS Software Navigator. A WebDAV server is required for some components of the SAS Business Intelligence middle tier. WebDAV-based functionality in the SAS Information Delivery Portal requires the Xythos WFS. SAS Web Report Studio supports both the Apache DAV Server and the Xythos WFS. The Apache DAV Server is the recommended WebDAV server for SAS Web Report Studio installations that do not intend to use SAS Information Delivery Portal or upgrade to the portal in the future.

The version of Xythos WFS included with SAS 9.1.3 requires a JDK version 1.4 or higher.

Xythos WFS requires a database and supports PostgreSQL, Microsoft SQL Server, Oracle, and IBM DB2. The Xythos installation documentation, which includes an administration guide, lists supported database vendors and releases. By default, the Xythos WFS is deployed in a Tomcat servlet container, which is provided in the Xythos installation.

Table 6.10 WebDAV Server Information

WebDAV Server Information		Done
WebDAV Server Name:	_____ <p style="text-align: center;">e.g., My WebDAV Server</p>	[]
WebDAV Server Provider:	_____ <p style="text-align: center;">e.g., Apache</p>	[]
WebDAV Server Version:	_____ <p style="text-align: center;">e.g., 2.0.45</p>	[]

WebDAV Server Information		Done
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 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 68.

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.

Passwords for these user IDs cannot contain more than one “\$” character. △

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

Table 6.11 SAS Group

Task	Done
Create a new group called <i>SAS</i> .	[]

- 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 6.12 SAS User Information

SAS User Information		Done
User ID:	_____ e.g., <i>sas</i>	[]
Full Name:	_____ e.g., <i>SAS</i>	[]

SAS User Information		Done
Password:	_____	[]
Task:	Assign the <i>SAS group</i> as the primary group for this user.	[]

- 3 Create a *SAS Administrator* that can be authenticated 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 6.13 SAS Administrator Information

SAS Administrator Information		Done
User ID:	_____ e.g., <i>sasadm</i>	[]
Full Name:	_____ e.g., SAS Administrator	[]
Password:	_____	[]

- 4 Create a *SAS General Server* user that can be authenticated 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 6.14 SAS General Server User Information

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

- 5 Create a *SAS Guest* that can be authenticated 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 Information Delivery Portal to log users into the public kiosk area).

Table 6.15 SAS Guest Information

SAS Guest Information		Done
User ID:	_____ <p style="text-align: center;"><i>e.g., sasguest</i></p>	[]
Full Name:	_____ <p style="text-align: center;"><i>e.g., SAS Guest User</i></p>	[]
Password:	_____	[]

- 6 Create a *SAS Trusted User* that can be authenticated on your metadata server machine. Because the user ID is a trusted ID, SAS servers such as the OLAP server, object spawner 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.

Table 6.16 SAS Trusted User Information

SAS Trusted User Information		Done
User ID:	_____ <p style="text-align: center;"><i>e.g., sastrust</i></p>	[]
Full Name:	_____ <p style="text-align: center;"><i>e.g., SAS Trusted User</i></p>	[]
Password:	_____	[]

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

Table 6.17 SAS Demo User Information

SAS Demo User Information		Done
User ID:	_____ <p style="text-align: center;"><i>e.g., sasdemo</i></p>	[]
Full Name:	_____ <p style="text-align: center;"><i>e.g., SAS Demo User</i></p>	[]
Password:	_____	[]

Note: The remaining steps are required only if you will be using any of the SAS Web software (SAS Web Report Studio, SAS Information Delivery Portal, SAS Web Infrastructure Kit, Solutions) in your deployment. △

- 8 Create a *SAS Web Administrator* that can be authenticated 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 Web Report Studio, SAS Web Report Viewer, or SAS Information Delivery Portal. This user has permission to administer the SAS Web infrastructure.

Table 6.18 SAS Web Administrator Information

SAS Web Administrator Information		Done
User ID:	_____ <p style="text-align: center;">e.g., <i>saswbadm</i></p>	[]
Full Name:	_____ <p style="text-align: center;">e.g., SAS Web Administrator</p>	[]
Password:	_____	[]

- 9 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 a servlet container (WebLogic, WebSphere, or Tomcat) from the SAS Software Navigator.

Note: When you install your servlet container, make sure that the full path to the installation directory contains NO SPACES. △

Note: You need to make sure you have the correct Java Development Kit (JDK) installed for the servlet container you are using. Review the Java environment requirements for your servlet container. Then download a JDK from the SAS support site for third-party software: support.sas.com/thirdpartysupport. Select “SAS 9.1.3. Java Development Kits (JDK).”

Installation programs attempt to determine available JDK versions by checking operating system environment variables and executable paths. When installing software that requires a JDK, check to see if the detection process has obtained the correct location. If not, correct the location during the installation. You can determine the default Java version in a shell script by typing the command: `java -version`

In some operating system environments, you might need to temporarily set the environment variable for the Java install directory just for the duration of the installation program or script.

Some Java versions might require patches to the operating system. Consult the Java installation documentation. △

Table 6.19 Servlet Container Information

Servlet Container Information		Done
Server Container Provider:	_____ <p style="text-align: center;">e.g., Tomcat</p>	[]
Servlet Container Version:	_____ <p style="text-align: center;">e.g., 4.1.18</p>	[]

Servlet Container Information		Done
Server Container Machine:	<hr/> e.g., myServer.myCompany.com	[]
Servlet Container Location:	<hr/> e.g., /usr/local/Tomcat4.1 <i>Note: You must change the installation location to contain NO SPACES.</i>	[]

10 If you have already installed a WebDAV (Web Based Distributed Authoring and Versioning) Server, fill in the details in the table below. Note that if you plan to use the Xythos Webfile Server (WFS) as your WebDAV server, you should install the version supplied with your SAS software (4.0.48), and you must install the SAS User Management Customization software as well.

If you have not already installed a WebDAV Server, skip this step now and install an Apache Server or a Xythos Server from the SAS Software Navigator. A WebDAV server is required for some components of the SAS Business Intelligence middle tier. WebDAV-based functionality in the SAS Information Delivery Portal requires the Xythos WFS. SAS Web Report Studio supports both the Apache DAV Server and the Xythos WFS. The Apache DAV Server is the recommended WebDAV server for SAS Web Report Studio installations that do not intend to use SAS Information Delivery Portal or upgrade to the portal in the future.

The version of Xythos WFS included with SAS 9.1.3 requires a JDK version 1.4 or higher.

Xythos WFS requires a database and supports PostgreSQL, Microsoft SQL Server, Oracle, and IBM DB2. The Xythos installation documentation, which includes an administration guide, lists supported database vendors and releases. By default, the Xythos WFS is deployed in a Tomcat servlet container, which is provided in the Xythos installation.

Table 6.20 WebDAV Server Information

WebDAV Server Information		Done
WebDAV Server Name:	<hr/> e.g., My WebDAV Server	[]
WebDAV Server Provider:	<hr/> e.g., Apache	[]
WebDAV Server Version:	<hr/> e.g., 2.0.45	[]
WebDAV Server Machine:	<hr/> e.g., myServer.myCompany.com	[]
WebDAV Server Location:	<hr/> 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 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 and groups, see the section “Setting Up Required User Accounts” on page 68.

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 *SASGRP*. 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 6.21 RACF Group

Task	Group Name	Installer Done	Systems Programmer Done
Create a RACF group	_____ e.g., <i>SASGRP</i>	[]	[]

- 2 Create a *SAS* user. Make the *SASGRP* 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 6.22 SAS User Information

SAS User Information		Installer Done	Systems Programmer Done
User ID:	_____ e.g., <i>sas</i>	[]	[]
Full Name:	_____ e.g., SAS	[]	[]
Password:	_____	[]	[]
Task:	Assign the <i>SASGRP</i> 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 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. Δ

Table 6.23 SAS Administrator Information

SAS Administrator Information		Installer Done	Systems Programmer Done
User ID:	_____ e.g., <i>sasadm</i>	[]	[]
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. Δ

Table 6.24 SAS General Server User Information

SAS General Server User Information		Installer Done	Systems Programmer Done
User ID:	_____ e.g., <i>sassrv</i>	[]	[]
Full Name:	_____ e.g., SAS General Server	[]	[]
Password:	_____	[]	[]
Task:	Assign the <i>SAS group</i> 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 Information Delivery Portal to log users into the public kiosk area).

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

Table 6.25 SAS Guest Information

SAS Guest Information		Installer Done	Systems Programmer Done
User ID:	_____ e.g., <i>sasguest</i>	[]	[]
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, object spawner 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 6.26 SAS Trusted User Information

SAS Trusted User Information		Installer Done	Systems Programmer Done
User ID:	_____ e.g., <i>sastrust</i>	[]	[]
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. Δ

Table 6.27 SAS Demo User Information

SAS Demo User Information		Installer Done	Systems Programmer Done
User ID:	_____ e.g., <i>sasdemo</i>	[]	[]
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 Web Report Studio, SAS Web Report Viewer, or 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 (SAS Web Report Studio, SAS Information Delivery Portal, SAS Web Infrastructure Kit) in your deployment.

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

Table 6.28 SAS Web Administrator Information

SAS Web Administrator Information		Installer Done	Systems Programmer Done
User ID:	_____ e.g., <i>saswbadm</i>	[]	[]
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 6.29 Names and Port Numbers for Started Tasks

Server Name	Started Task Name	Num of Ports Required	Service Name (Optional, for <code>/etc/services</code>)
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 6.30 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. Δ

Table 6.31 Port Numbers for Spawned Servers

Spawned 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 config directory where these servers will run and where the directory substructure can be defined.

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

Table 6.32 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 data sets to the program control list prior to data set creation.

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

Table 6.33 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 completed prior to configuring 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 might 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.

Users Authenticated Only on the Metadata Server Host

The following users are authenticated only on your metadata server host. Therefore, these can be local accounts on the metadata server host on Windows systems. They can also be domain accounts. The first two accounts are always required, and the third account is required if you will be running any of the SAS Web applications:

- SAS Administrator (**sasadm**)
- SAS Trusted User (**sastrust**)
- 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. Also, you should use it to run SAS Management Console only when you are performing tasks that require special privileges. In particular, do *not* use this account in cases where passwords need to be acquired, for example, when you are defining a database library.

SAS Trusted User

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 multiple back-end servers. The **sastrust** account is also used by the object spawner. When the spawner receives a request to start a workspace or stored process server, it uses this account to connect to the metadata server in order to read the appropriate server definition.

SAS 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 to set up a `sasoladm` account for an OLAP content administrator.)

Users That Might Require Network Accounts

You also need to create accounts for a

- SAS Guest User (`sasguest`)
- SAS Demo User (`sasdemo`)
- SAS General Server User (`sassrv`)
- SAS User (`sas`).

These accounts are typically network accounts. That is, on Windows systems, they are domain accounts. On a single-machine Windows system, they can be local accounts, but in a distributed system, local accounts are usually not feasible for reasons such as the following:

- If your workspace server will run on a different host than the metadata server, `sasguest` and `sasdemo` should be network accounts. The metadata server needs to be able to authenticate these users during initial authentication, and the workspace server needs to be able to perform additional authentication on these users. (For information on additional authentication, see “Additional Authentication” on page 156.)
- If you might run load balanced stored process servers on multiple hosts, the `sassrv` account should be a network account.
- If your SAS and middle-tier servers will reside on more than one host, the `sas` account should be a network account.

SAS Guest User

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

SAS Demo User

The SAS Demo User account is also used by the SAS Information Delivery Portal. This account allows users to test the portal Web application implementation and to learn about its features.

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.

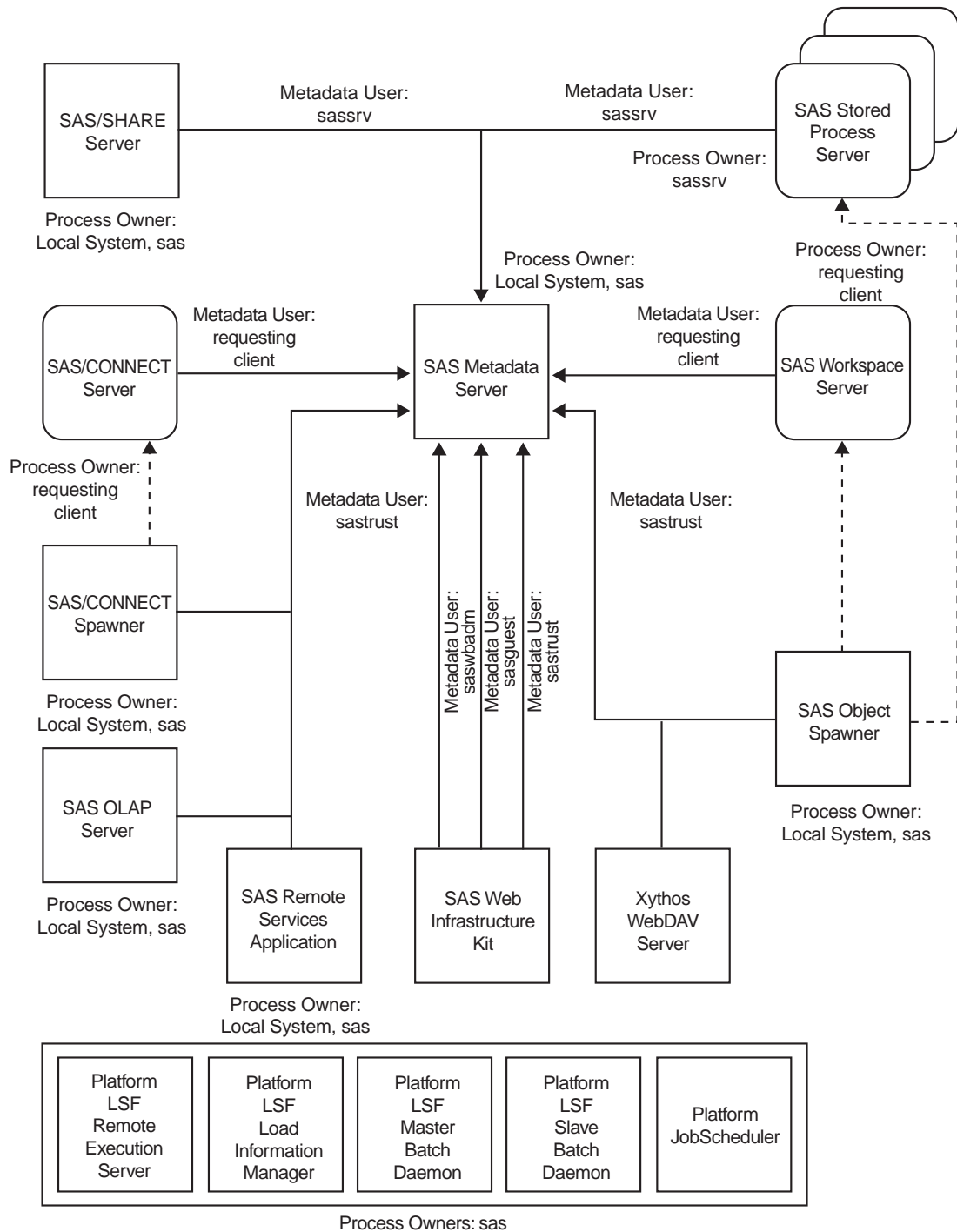
SAS Installer

You should be logged on 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 that you create during pre-installation are used by the SAS servers in a business intelligence system. The figure shows who owns each server process and which account each server uses to communicate with the metadata server.

Figure 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 the SAS Server Users group. 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.

Servers Required to Run SAS Web Applications

If you will be running SAS Web applications—such as SAS Web Report Studio and SAS Information Delivery Portal—at your site, you must install the third-party servers that are required by these applications on your middle-tier host. Ordinarily, you install these servers and related software at the same time that you install your SAS software. (The procedure for installing the business intelligence platform is explained in detail in Chapter 7, “Installing and Configuring Your Software,” on page 79.) One advantage of installing these third-party products at this time is that you will be sure to get the supported version of each product.

However, you might already have some of the necessary products installed, or you might want to install these products before your SAS software arrives. Make sure that you install the correct version of each product and that you record information about the products that you install on the appropriate pre-installation checklist.

Which Software Is Required?

In general, you will need to install three products in support of your Web applications:

- the Java 2 SDK
- a servlet container or J2EE server
- a WebDAV (Web-Based Distributed Authoring and Versioning) server.

The Java 2 SDK is a software development kit that includes a Java Runtime Environment and a Java compiler. This SDK is a prerequisite for the servlet container or J2EE server that is discussed in the next paragraph. These servers require the SDK's Java compiler and class libraries to compile servlets that are derived from JavaServer Pages. For information about which version of the Java 2 SDK you need, see “Requirements for the Java 2 SDK” on page 72.

A servlet container or J2EE server provides the execution environment for Web applications. A servlet container provides a subset of the functionality of a J2EE server. Such a container provides an execution environment for servlets and JavaServer Pages, which are translated to servlets. A J2EE server includes a servlet container, but also includes an Enterprise JavaBean container (for applications that use distributed objects) and a message server, and supports a host of other technologies. A widely used servlet container is Apache Tomcat, and popular J2EE servers include the BEA WebLogic Server and the IBM WebSphere Application Server. For information about how to choose the correct product, see “Requirements for a Servlet Container or J2EE Server” on page 72.

A WebDAV server is an HTTP server that supports the WebDAV extensions to the HTTP protocol. These extensions enable multiple authors to collaborate on documents that are located on an HTTP server. Whereas HTTP without the WebDAV extensions enables you to read a document that is identified by a particular Uniform Resource Locator (URL) or write a document to a server, the WebDAV extensions enable you to

edit a document that is located at a URL. Many times, these are HTML or XML documents, but a WebDAV server can also host word-processing documents, images, or other types of content.

WebDAV provides support for

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

The ability of an 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.

The SAS business intelligence Web applications use a WebDAV server as a content repository. For information about which WebDAV server you should install, and which version of the server you need, see “Requirements for a WebDAV Server” on page 73.

Note: Your WebDAV server and J2EE server do not have to reside on the same machine. \triangle

Requirements for the Java 2 SDK

The only requirement for the Java 2 SDK is that you install the correct version of the SDK for your operating system. The following table shows the appropriate version for the platforms on which the SAS Business Intelligence Platform is supported.

Table 6.34 Supported Versions of the Java 2 SDK

Platform	Supported Version of Java 2 SDK
Windows (NT, 2000, XP)	1.4.2_04
Solaris	1.4.2_04
HP-UX IPF	1.4.1.05
AIX	1.4.1.3 (also called 1.4.1 SR1)

Note: To download or check the currently supported versions of this product, go to the SAS support site for third-party software: support.sas.com/thirdpartysupport. \triangle

Requirements for a Servlet Container or J2EE Server

When you select a servlet container or a J2EE server, there are at least two variables to consider.

First, you need to decide whether to install a servlet container (such as Apache Tomcat) or a J2EE server (such as the BEA WebLogic Server or the IBM WebSphere Application Server). In some cases, your decision will be dictated by the Web applications that you plan to use at your site. For example, the Web applications that are part of the business intelligence platform—including SAS Web Report Studio and SAS Information Delivery Portal—can run in a servlet container because these products do not use Enterprise JavaBeans. However, if you also will be running SAS Marketing Automation or a SAS solution, you will need a J2EE server because these applications do use Enterprise JavaBeans.

Another factor to consider—if your applications require only a servlet container—is whether Apache Tomcat provides the performance and features that you need. Apache Tomcat has these advantages:

- It is free.
- It is the reference implementation for the Java servlet and JSP application programming interfaces (APIs). Therefore, Web applications that run in this environment are guaranteed to run correctly.

On the other hand, products such as the BEA WebLogic Server and the IBM WebSphere Application Server are more “industrial strength” products than Apache Tomcat. These products provide a number of features that you might require, for example,

- scalability
- load balancing
- security
- persistent sessions.

Ultimately, you need to base your decision on the number of users that you need to support and the number of features that your system requires. Your SAS representative can help you select the appropriate product.

After you have selected a servlet container or J2EE server, you must ensure that you get the correct version of the product. The following table indicates which products are supported on each platform.

Table 6.35 Supported Servlet Containers and J2EE Servers

Platform	Product and Version
Windows (NT, 2000, XP)	Apache Tomcat 4.1.18
	BEA WebLogic 8.1 (SP2)
	WebSphere 5.1
Solaris	Apache Tomcat 4.1.18
	BEA WebLogic 8.1 (SP2)
HP-UX IPF	Apache Tomcat 4.1.18
	BEA WebLogic 8.1 (SP1)
AIX	Apache Tomcat 4.1.18
	BEA WebLogic 8.1 (SP2)
	WebSphere 5.1

Note: To download or check the currently supported versions of these products, go to the SAS support site for third-party software: support.sas.com/thirdpartysupport. If possible, you should visit the vendor’s Web site before you begin an installation and see if you need to register before you can download a product or a patch. Sometimes it might take a day or so to register. △

Requirements for a WebDAV Server

Two WebDAV servers are supported for use by the business intelligence Web applications: the Apache HTTP Server (with its WebDAV module enabled) and Xythos Software’s WebFile Server (WFS). As with your servlet container or J2EE server, you need to first select the correct product and then install the correct version of the product.

Use the following guidelines to decide which product to select:

- If you plan to use SAS Web Report Studio at your site and plan not to use SAS Information Delivery Portal now or in the future, we recommend that you use the Apache HTTP Server as your WebDAV server.

- If you plan to use SAS Information Delivery Portal at your site, we recommend that you use the Xythos WebFile Server. Some of the WebDAV-based functionality in the portal product requires the use of WebFile Server, and SAS Web Report Studio also works with this server.

Note: Xythos WFS depends on a variety of other components, such as a DBMS (PostgreSQL, SQL Server, DB2, or Oracle), an associated JDBC driver, and a servlet container (Tomcat, WebLogic, or WebSphere). Tomcat 4.0.6 is included with the Xythos distribution as a default servlet container. Windows installations that use PostgreSQL require the Cygwin UNIX on Windows emulation environment. Windows PostgreSQL is a component of Cygwin. Δ

After you have selected a WebDAV server, use the following table to identify the required version of that product.

Table 6.36 Supported WebDAV Servers

Product	Version
Apache HTTP Server	2.0.45
Xythos WebFile Server	4.0.48

Both products are available for all of the platforms on which the SAS business intelligence servers are supported.

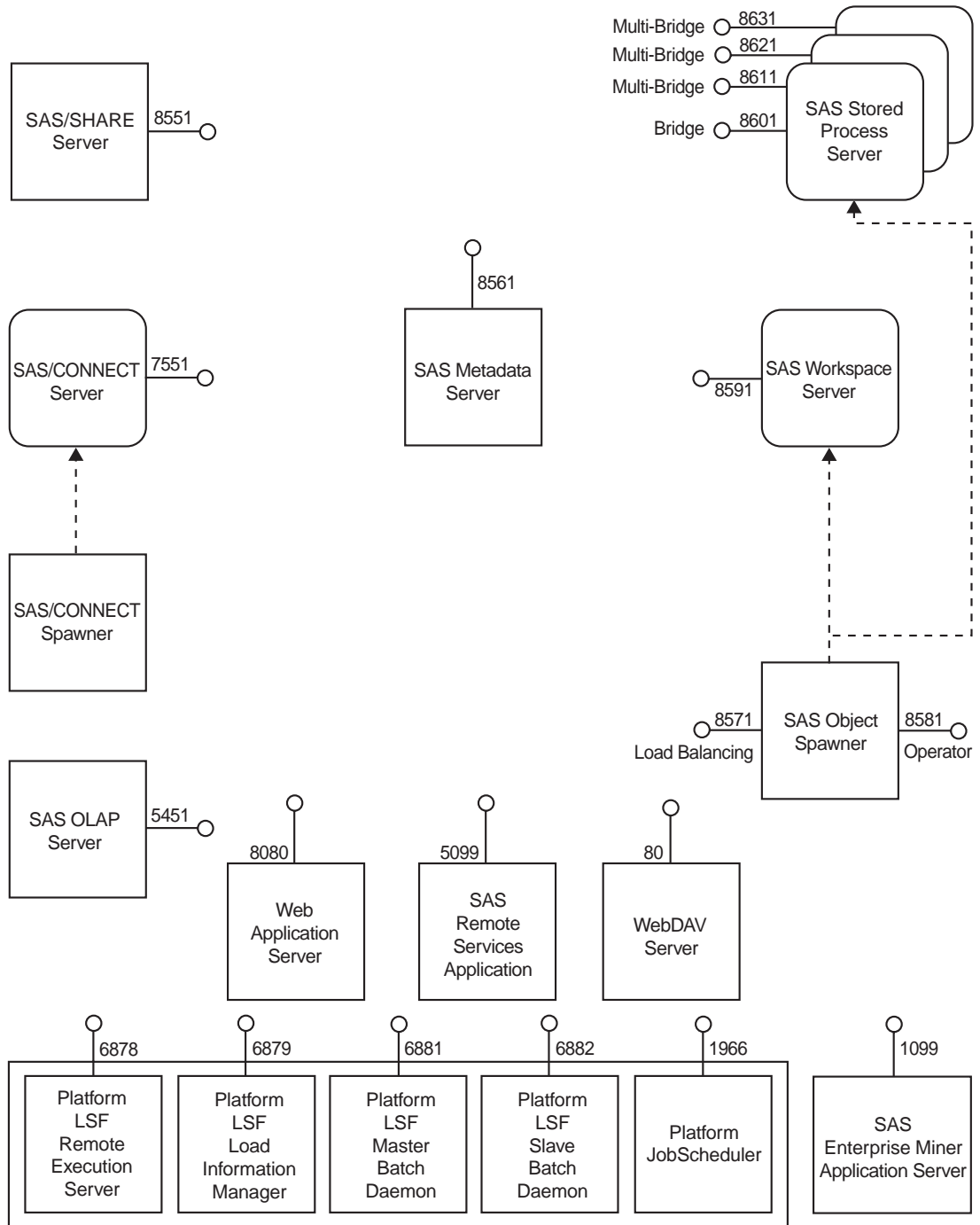
Note: To download or check the currently supported version of the Apache HTTP Server, or to check the currently supported version of Xythos WFS, go to the SAS support site for third-party software: support.sas.com/thirdpartysupport. (Xythos WFS is distributed on a SAS CD.) Δ

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 79, 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 standard ports are in use. If any 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 100.)

Table 6.37 Default Port Numbers, Property Names, and Descriptions

Property Name	Default Port Value	Description
OMAPORT	8561	SAS Metadata Server port. The SAS Configuration Wizard explicitly asks for this value.
CONNECT_PORT	7551	SAS/CONNECT Server port
SHAREPORT	8551	SAS/SHARE Server port
OLAP_PORT	5451	SAS OLAP Server port
SERVICES_RMI_PORT	5099	SAS Remote Services Application port
SPAWNER_OPERATOR_PORT	8581	SAS Object Spawner operator port
SPAWNER_LOADBALANCING_PORT	8571	SAS Object Spawner load balancing port
STP_PORT	8601	SAS Stored Process Server port
STP_PORT1	8611	SAS Stored Process Server port 1
STP_PORT2	8621	SAS Stored Process Server port 2
STP_PORT3	8631	SAS Stored Process Server port 3
IOM_PORT	8591	SAS Workspace Server port
DAV_PORT	80	WebDAV Server port (Apache HTTP Server)
DAV_PORT	83000	WebDAV Server port (Xythos WFS)
WEBSRV_PORT	8080	Servlet container port (Apache Tomcat)
WEBSRV_PORT	9080	J2EE Server port (IBM WebSphere Application Server port)
APP_SERVER_WEBLOGIC_ADMIN_PORT	7501	WebLogic Administration Server port
APP_SERVER_WEBLOGIC_MANAGED_PORT	7001	WebLogic Managed Server port
LSF_RES_PORT	6878	LSF Scheduler Remote Execution Service (RES) port*
LSF_LIM_PORT	6879	LSF Load Information Manager Service (LIM) port*
LSB_MDB_PORT	6881	LSF Master Batch Daemon (MBD) port*
LSB_SBD_PORT	6882	LSF Slave Batch Daemon (SBD) port*

Property Name	Default Port Value	Description
JS_PORT	1966	Job Scheduler port**
EM_APPSrv_PORT	1099	SAS Enterprise Miner application server port***

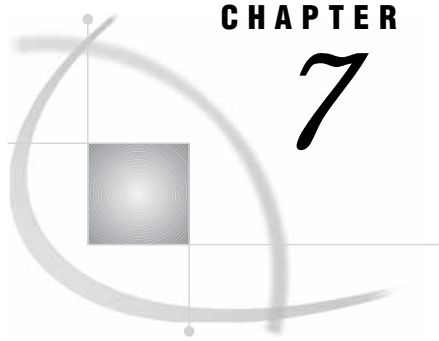
* For information about how to change this value, see the LSF documentation.

** 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 that are described on those checklists. Then, proceed to Chapter 7, "Installing and Configuring Your Software," on page 79.



CHAPTER

7

Installing and Configuring Your Software

<i>Overview of Installing and Configuring Your Software</i>	79
<i>Starting the SAS Software Navigator</i>	81
<i>Creating a SAS Software Depot</i>	84
<i>Installing Software on a Machine</i>	87
<i>Performing an Advanced Installation</i>	87
<i>Supplying the SAS Software Navigator with the Information That It Needs</i>	88
<i>Running the System Requirements Wizard (Windows Only)</i>	94
<i>Performing Interactive Installations</i>	97
<i>Performing Nonstandard Installations</i>	98
<i>Performing a Personal Installation</i>	99
<i>Configuring a Machine</i>	100
<i>Running the Configuration Wizard on Windows and UNIX Systems</i>	100
<i>Configuring SAS Servers on z/OS Systems</i>	106
<i>Edit and Submit &prefix.W0.SRVCNTL(COPYIA) Job</i>	106
<i>Log On to the USS Shell</i>	107
<i>Edit the configuration.properties File</i>	107
<i>Run the deploy_IA.sh Script</i>	107
<i>Verify the Results of Running the Script</i>	107
<i>Follow the Instructions in instructions.html</i>	107
<i>Performing the Steps Listed in instructions.html</i>	108
<i>Checking Your Metadata for Required Objects</i>	111

Overview of Installing and Configuring Your Software

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

Note: This chapter explains how to perform Advanced and Personal installations, which are geared toward the installation of the entire platform. For information on Software Index installations—which are intended for adding products to an existing installation—see Appendix 2, “Software Index Installations,” on page 443. Δ

This chapter covers the following topics:

- Starting the SAS Software Navigator.
This is the tool that you use to install your software.
- Creating 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

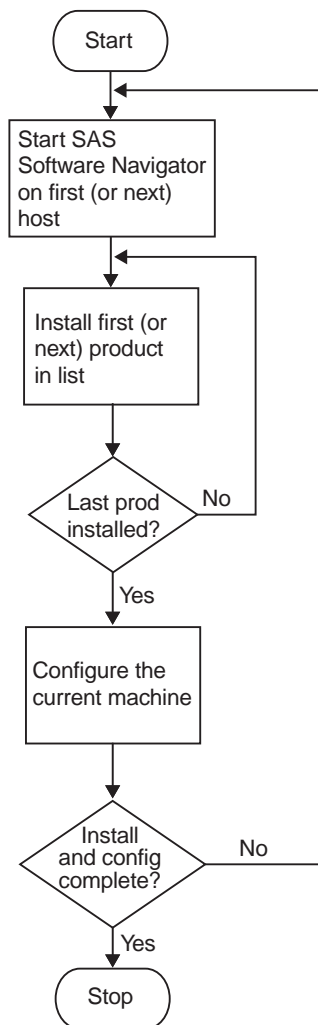
task is highly recommended. If you copy your CDs to the network, you will not have to insert a new CD each time that you want to install a new product.

- Installing 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.
- Configuring 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 by following instructions that are generated by the wizard. You can perform these tasks manually, often using SAS Management Console, or you can perform the tasks by running a set of scripts.

Note: When you install the SAS 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 9.1.3 Foundation for z/OS*, which is available at support.sas.com/documentation/installcenter/. And you configure the software on the machine using a script rather than a configuration wizard. The latter task is covered in “Configuring SAS Servers on z/OS Systems” on page 106. Δ

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, in general you should install server-tier components before middle-tier components and middle-tier components before clients. Δ

Starting 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 from a CD (in your Installation Kit).

Use the following steps to start the SAS Software Navigator.

- 1 Log on to the machine on which you will be installing software or creating a SAS Software Depot. On Windows systems, you should log on as a member of the **Administrators** group, and on UNIX systems, you should log on as the SAS user (recommended name **sas**).
- 2 The SAS Software Navigator loads and runs in temporary space. Prior to loading, the SAS Software Navigator searches your system for a SAS Private Java

Runtime Environment (JRE), which it needs in order to execute. If the SAS Software Navigator finds such a JRE, it will use that runtime environment, which reduces the amount of temporary space required. If it cannot find this JRE, the SAS Software Navigator will load a copy of the JRE into temporary space. In addition, each installation program that you run requires temporary space in which to execute (the amount of space varies). All of the installation programs and the SAS Software Navigator will free this space after exiting normally.

Check the following table for the system on which you are running the SAS Software Navigator to determine whether you have adequate free temporary space to execute the program. The space estimates include space for loading a JRE to temporary space. You can reduce the estimated temporary space requirements if you install the SAS Private JRE before performing your software installations.

Table 7.1 Temporary Space Requirements

Operating System	Temporary Space Required
Windows	400 MB free in temp
Solaris	675 MB free in /tmp*
HP-UX IPF	500 MB free in /tmp
AIX	400 MB free in /tmp

* On some Solaris systems, **/tmp** and swap might be mapped to the same device, which means that the amount of space in **/tmp** might vary widely based on the number and size of the currently running programs. As a result, the SAS Software Navigator might indicate that you have inadequate space in **/tmp**. If this occurs, when the software navigator prompts you, specify an alternate location with adequate space on a device free of swap.

On UNIX, you can specify an alternate directory where temporary files are loaded. This is useful when the default directory for temporary files (typically **/tmp** or **/usr/tmp**) does not have adequate free space.

You can specify an alternate temporary directory in one of the following ways.

- Set the **SSNTMPDIR** environment variable to the path of the alternate directory:

```
$ export SSNTMPDIR=path
```

- Specify the **-ssntmpdir** command line flag to set the alternate directory when launching the SAS Software Navigator:

```
$ setup.sh -ssntmpdir path
```

- 3 Remove the SAS Software Navigator CD from your Installation Kit, place it in your CD drive, and mount it using the procedure that is appropriate for your operating system. (All CDs should be run on a native platform. If you attempt to use a terminal emulator, the software might not function properly.)

Windows

After you have inserted the SAS Software Navigator CD, the navigator should start automatically. If it does not start automatically, find the executable file for the wizard (**setup.exe**) and open it.

UNIX

On UNIX systems, the SAS Software Navigator CD will be mounted automatically

if you are running an automount program, such as `vold` on Solaris. Otherwise, you must mount the CD manually. Manually mounting a CD on a UNIX system requires `root` privileges. So you must switch users to `root` before mounting the CD. Use the following command:

```
$ su root
```

On single-drive systems, you must mount and unmount CDs often. We recommend that you maintain a separate window, where you are logged on as `root` , for mounting and unmounting CDs, while you are running the SAS Software Navigator.

The `mount` command on UNIX systems follows this format:

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

In this command, *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 UNIX system on which the SAS Business Intelligence Platform is supported. The device names in the following list are used only for example. Substitute your actual device names as necessary. Also, these instructions assume that your mount point is `/cdrom` ; however, you can choose another location.

Solaris

```
# mount -r -F hsfs /dev/cd0 /mnt/cdrom
```

HP-UX for the Itanium Processor Family

```
# mount -F cdfs -o rr,ro /dev/dsk/c0t0d0 /mnt/cdrom
```

AIX

```
# mount -r -v cdrfs /dev/cd0 /mnt/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 remote_host:/cd_rom_dir /mnt/cdrom
```

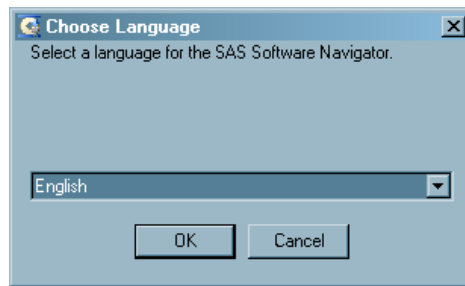
In this command, *remote_host* refers to the machine that owns the CD-ROM drive, and *cd_rom_dir* is the actual mount point for the CD-ROM drive on the server.

After you have mounted the CD, launch the SAS Software Navigator with this command:

```
$ /mnt/cdrom/setup.sh
```

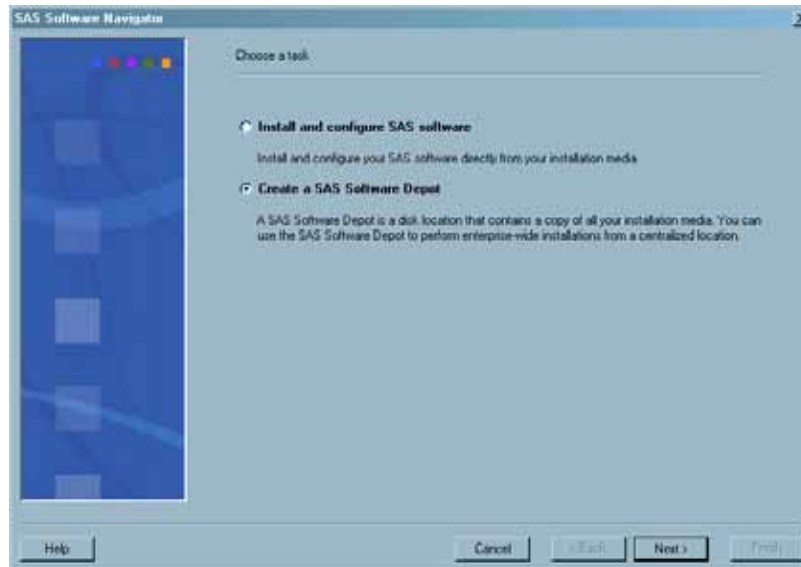
Note: When you are using a system with an automounter, it is common for a File Manager window to display when you insert media into the CD-ROM drive. Do not invoke the SAS Software Navigator with the File Manager because doing so will cause a “Device busy” condition and prevent disk swapping later in the installation process. △

- 4 At this point, the SAS Software Navigator starts and prompts you for a language.



From the drop-down list, select the language in which you want the SAS Software Navigator to present text. Then, click **OK**.

- 5 After you have selected a language, you see a splash screen and then a **Choose a task** window.



This is the point from which you can create a SAS Software Depot *or* begin installing software on a machine. For information on how to perform these tasks, see the following sections:

- “Creating a SAS Software Depot” on page 84
- “Performing an Advanced Installation” on page 87
- “Performing a Personal Installation” on page 99.

Creating a SAS Software Depot

If you are preparing to perform an Advanced or a Personal installation, we recommend that you create a SAS Software Depot before you begin the installation. This depot is simply a network copy of some or all of the CDs in your Installation Kits. If you want to lessen the time that it takes to create the depot and to conserve disk space, you can copy to the depot only the CDs for the products that are listed in your standard or customized planning file. If you do not have a planning file, are not sure which CDs you need to add to the depot, or 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 that 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 need to do is supply the physical path of the directory that holds the software depot. We highly recommend that you create this depot.

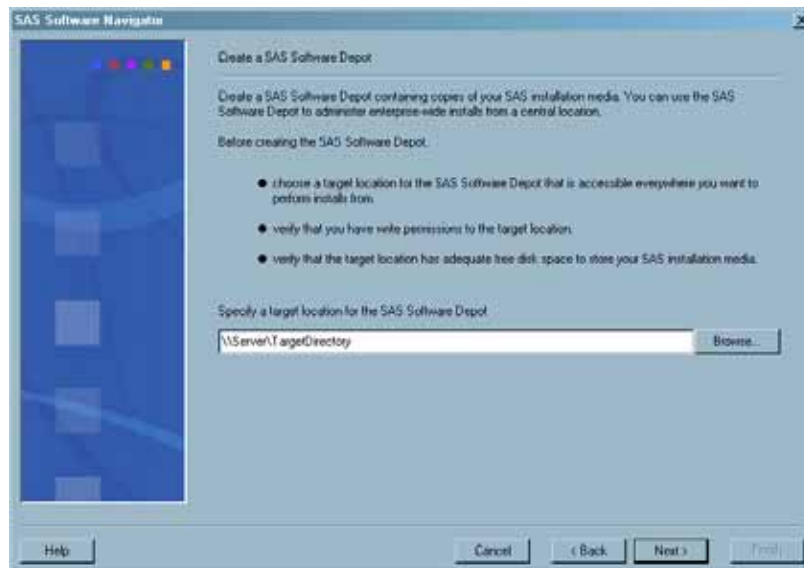
Note: Because z/OS software is distributed on cartridges, SAS Foundation for z/OS cannot be a part of your SAS Software Depot. △

To create your depot, follow these instructions:

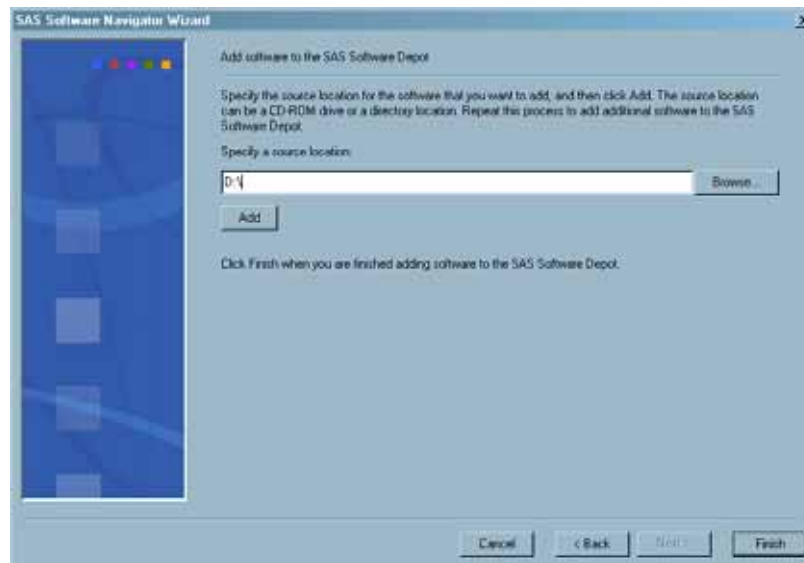
- 1 Determine the amount of disk space that is required for the depot. To do this, count the number of CDs that you intend to use to create your SAS Software Depot and multiply the number of CDs by 600 MB. The resulting product is the approximate space required. For example, if you intend to copy 12 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 possible), and find the SAS Software Navigator CD in the Installation Kit for that machine. Insert this CD into the machine's CD drive. Then, follow the instructions in "Starting the SAS Software Navigator" on page 81 to start the navigator. At the end of this step, you will be looking at the navigator's **Choose a task** window.
- 4 Select the **Create a SAS Software Depot** radio button, and click **Next**. The following window appears.



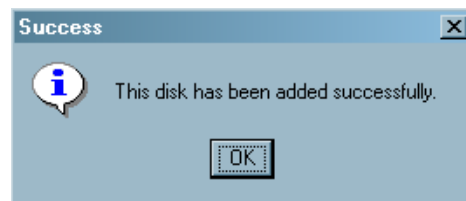
- 5 In the **Specify a target location for the SAS Software Depot** text box, enter the network location where you want to create the SAS Software Depot. You can also browse to the location by using the **Browse** button. After entering a value in the text box, click **Next**. The wizard adds the contents of the SAS Software Navigator CD to the depot. Then, the wizard's next screen appears.



- 6 In the **Specify a source location** text box, enter the location from which the application will be copying data. Most often, this will be your CD drive. Then insert the next CD to be copied into your drive, and click **Add**.

Note: For certain media, the SAS Software Navigator might respond with a message that indicates that it does not recognize the media you inserted, followed by a prompt for the name of the directory in which you will store this media. You should enter a name for the directory that is indicative of the contents of the CD. 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 that 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. Δ

- 7 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 added successfully.



Click **OK** in this dialog box. Then, continue building your SAS Software Depot by repeating steps 6 and 7 for each piece of media that 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. Δ

- 8 After you have copied the last piece of media, click **Finish**. The SAS Software Navigator closes. You can restart it from the top-level directory in your new SAS Software Depot location. On Windows systems, run the file `setup.exe`, and on UNIX systems, run the script `setup.sh`.

Ensure that all interested parties are aware of the network location of your SAS Software Depot. If you have created a project directory, let these parties know about the location of that directory as well.

Note: If you would like to add content to an existing SAS Software Depot or to 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, the opening dialog box, which contains the option to **Create a SAS Software Depot**, will not be displayed. △

Installing Software on a Machine

As discussed in “Types of Deployment” on page 30, the SAS Software Navigator enables you to perform three types of installations: Advanced installations, Personal installations, and Software Index installations.

- The most common type of installation—at least for setting up a new system—is the Advanced installation. Using this method of installation, you can deploy the SAS Intelligence Platform on a single machine or distribute it across a number of machines. You tell the SAS Software Navigator which components to deploy on each machine by supplying it with a deployment plan. This can be a standard plan or a customized plan (a plan that has been customized for your site). For information about how to perform an Advanced installation, see “Performing an Advanced Installation” on page 87.
- A Personal installation, which is useful for prototyping a system or building a small system, is very similar to an Advanced installation. However, there are two principal differences. In a Personal installation
 - you must install the platform on a single machine
 - you must use a standard deployment plan.

For information about the effects that these restrictions have on the SAS Software Navigator’s GUI, see “Performing a Personal Installation” on page 99.

- A Software Index installation is useful for adding one or more components to an existing installation. It does not rely on a deployment plan; rather, you select components to install from a list (an index) of licensed products. For more information about Software Index installations, see Appendix 3, “Software Index Installations.”

Performing an Advanced Installation

In an Advanced installation, you perform the same set of steps on each machine in your system—beginning with your metadata server host.

- 1 You start the SAS Software Navigator and supply it with the information that it needs in order to install software on the current machine. For example, you specify the location of a SID file (which indicates that you have licensed the products that you want to install), and if you are using a customized deployment plan, you specify the location of your project directory.
- 2 After you have entered this information, the navigator enters its software installation mode. While it is in this mode, you use the program to perform these tasks:
 - a Make sure that the machine meets the system requirements for the software that you are about to install (Windows systems only).

- b Install the appropriate products on the machine.
- c Start the SAS Configuration Wizard, which enables you to configure all of the software that you have installed.

The following subsections explain in more detail how to perform these three tasks.

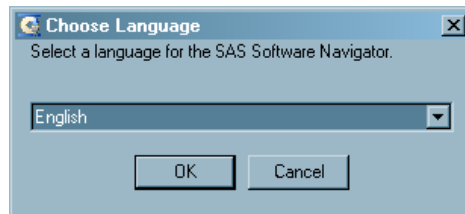
Supplying the SAS Software Navigator with the Information That It Needs

This section explains how to provide the SAS Software Navigator with the information that it needs to collect before it begins installing software on a machine. Use the following steps. At the end of these directions, you will be told to click an **Install** button that initiates the actual installation of the software.

- 1 Log on to the machine on which you will be installing software. On Windows systems, you can log on as any user who belongs to the **Administrators** group. On UNIX systems, log on as the SAS user (recommended name **sas**), which you created as one of your pre-installation tasks.

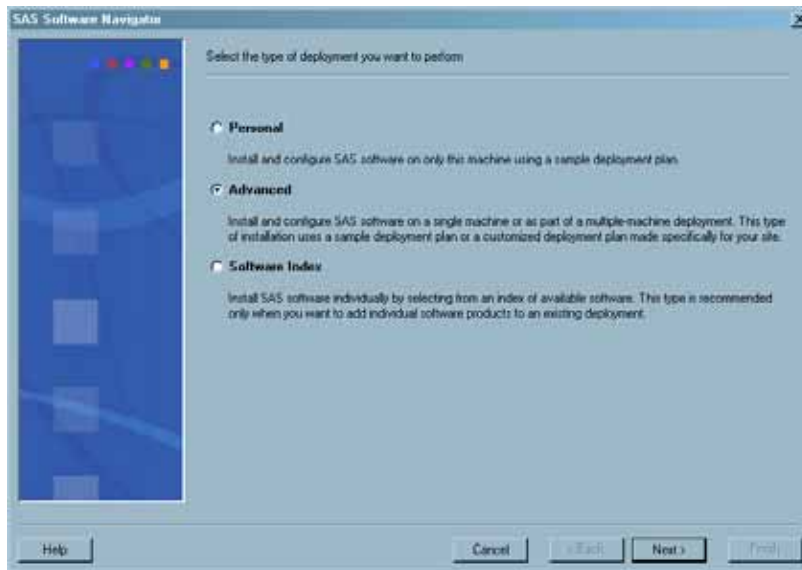
Note: We recommend that you do not log on as **root** to install software on a UNIX system. \triangle

- 2 Start the SAS Software Navigator. If you have created a SAS Software Depot (recommended), you can run the navigator from its network location in this depot. The script that starts the navigator is located in the depot's top-level directory. Otherwise, you can use the CD in your Installation Kit to run the program, as explained in "Starting the SAS Software Navigator" on page 81. A Choose Language dialog box appears.



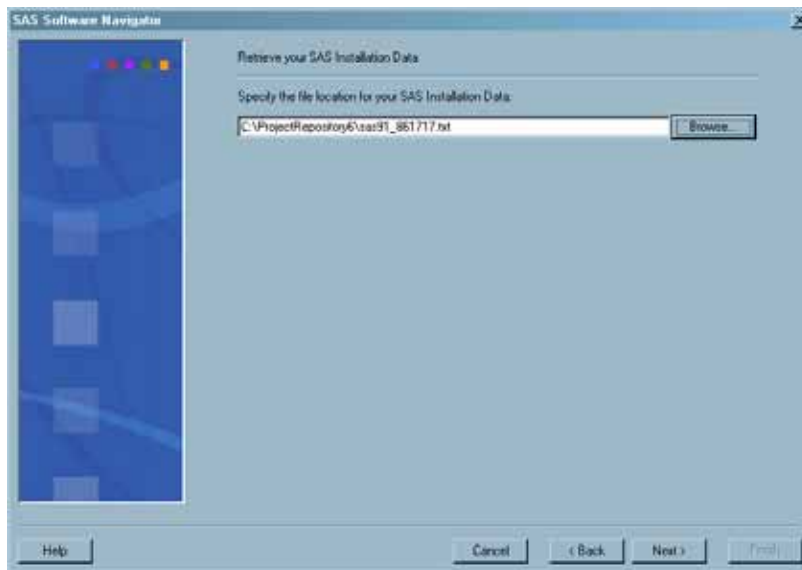
From the language drop-down list, select the language that you want the SAS Software Navigator to use when it displays text. Then click **OK**. The SAS Software Navigator starts.

If you started the navigator from a software depot, you will see a screen that you use to choose a type of deployment.

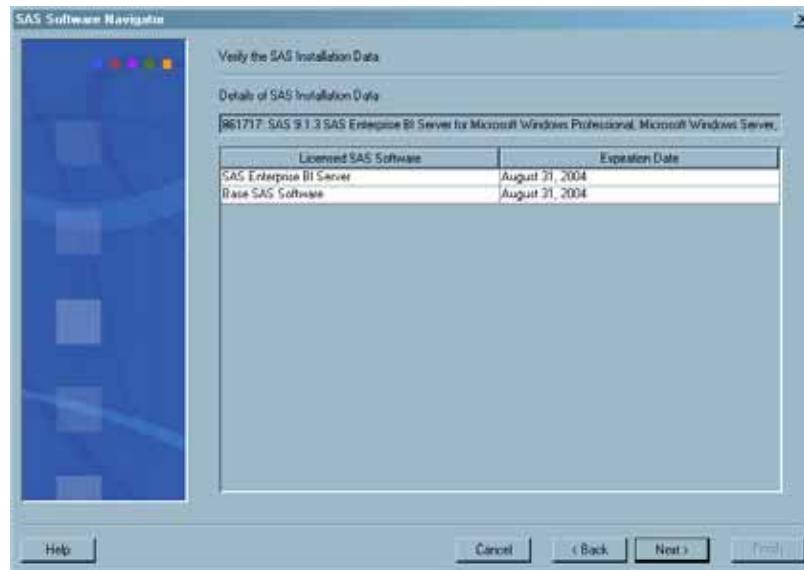


If you started the SAS Software Navigator from a CD, you will instead see a screen that asks whether you want to create a SAS Software Depot or to install software. In this case, select the **Install and configure SAS software** radio button, and click **Next**. You will then see the screen shown above.

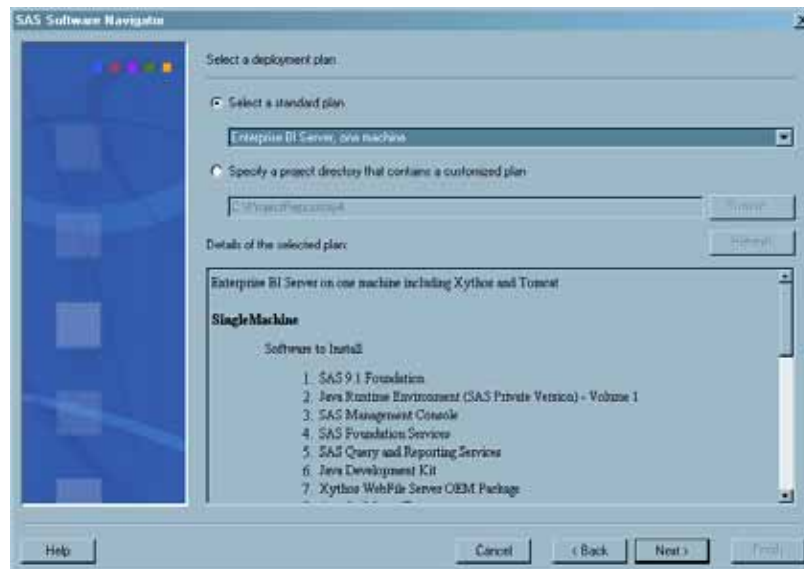
- 3 To perform an Advanced installation, select the **Advanced** radio button and click **Next**. You will be prompted for a SAS Installation Data file.



- 4 In the **Specify the file location for your SAS Installation Data** text box, enter the path to the SID file that you received with your Software Order e-mail. Or click **Browse** to open a file system browser that enables you to navigate to the SID file. (If you have already retrieved your SAS Installation Data while working on another machine, the path to that file will be the default value of the text box.) After you have entered the correct location, click **Next**. You will see a screen that asks you to verify that you have selected the proper SID file.



- 5 In the **Verify the SAS Installation Data** screen, review the list of licensed software and the expiration date for each item in the list to ensure that you selected the correct SID on the previous screen. If you selected the wrong SID, click **Back** and correct your error. If you have selected the correct SID, click **Next**. You will see the **Select a deployment plan** screen.



- 6 In the **Select a deployment plan** screen, specify which deployment plan the SAS Software Navigator should use for the current installation. This plan defines which software components should be installed and configured on each machine in the deployment.

There are two type of plans:

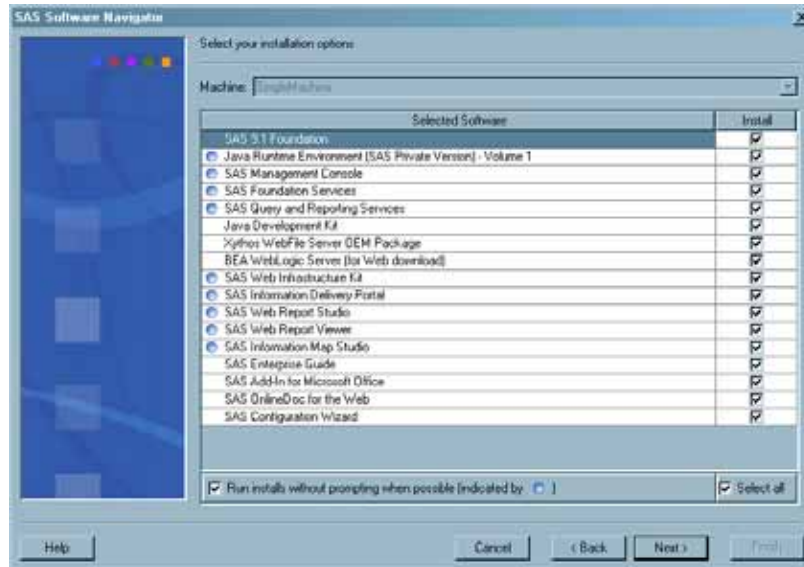
- standard plans
- customized plans.

Standard plans come with your SAS software and describe a set of commonly used deployments. A customized plan is a plan that was developed specifically for

your site by a SAS representative and representatives from your company using a SAS planning application. If a customized plan was created for your site, it will have been mailed to your company and should have been stored in a *project directory*, along with supporting files.

To choose a standard plan, select the **Select a standard plan** radio button; then, select the plan from the drop-down list. To select a customized plan, select the **Specify a project directory that contains a customized plan** radio button; then, enter the path to the project directory in the text box. In either case, you will see a description of the plan in the bottom half of the screen.

After you have chosen the correct plan, click **Next**. You will then see a **Select your installation options** screen.



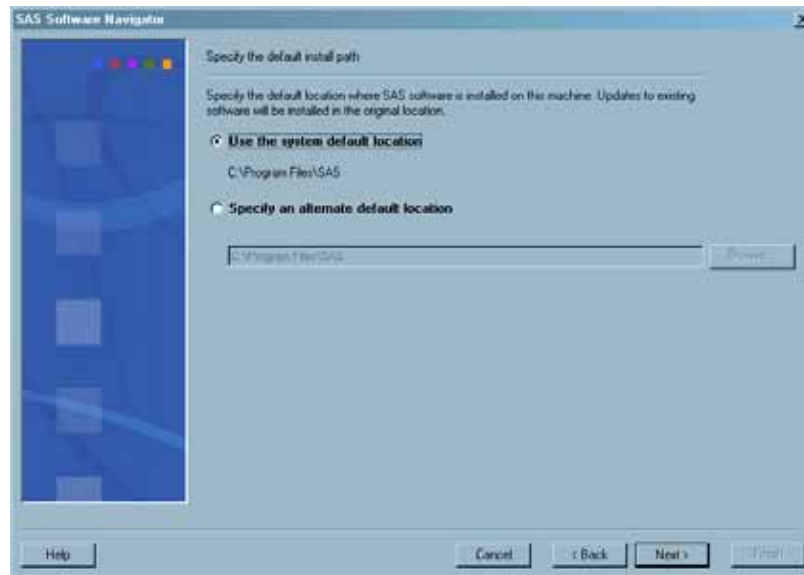
- 7 In the **Select your installation options** screen, you specify three things:
- the machine on which you will be installing software
 - which products you want to install on that machine
 - whether you want installation programs that can run silently to run in that mode.

To select the machine on which you want to install software, choose a value from the **Machine** drop-down list. If you are using a standard plan, choose a descriptive like **SingleMachine**, **ServerTier**, **MiddleTier**, or **ClientTier**. If you are using a customized plan, you might be asked to choose a descriptive name or an actual host name, such as **D1234.na.sas.com**.

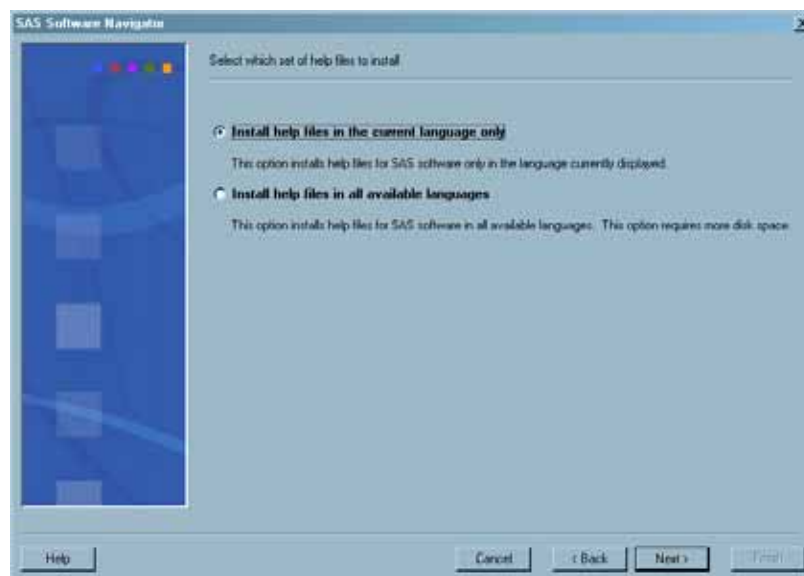
When you choose a machine—if not before—you will see a list of the products that you can install on that machine. By default, all of the products in the list are selected for installation. (See the check marks in the **Install** column.) You can deselect a product by deselecting the check box that is associated with the product. You can also select or deselect all products by selecting or deselecting the **Select all** check box.

Finally, you can indicate whether you want all of the installation programs that can run silently to do so. A blue sphere to the left of the product name indicates that a silent installation is possible. If you want these installation programs to run silently (recommended), leave the **Run installs without prompting when possible** check box selected.

After you have completed your work on this screen, click **Next**. The **Specify the default install path** screen appears.



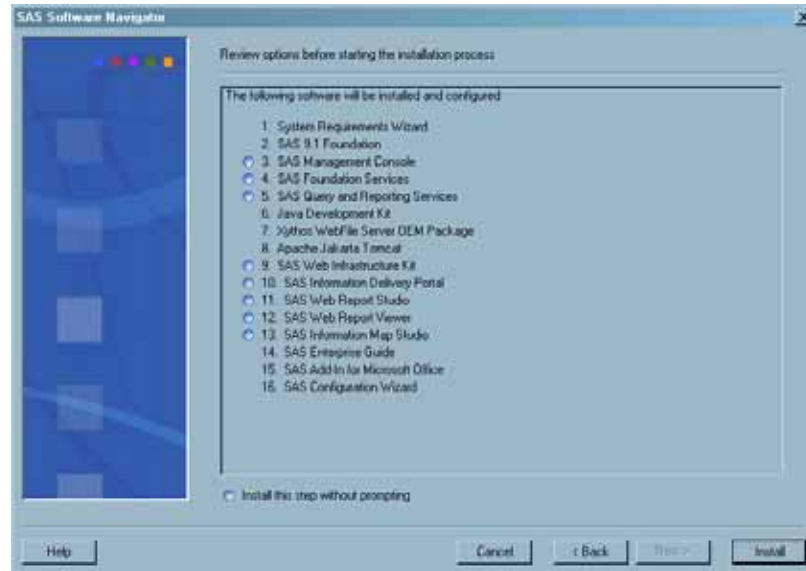
- 8 The **Specify the default install path** screen enables you to specify the directory in which your SAS software will be installed. You can select the **Use the system default location** radio button to choose the default location for your platform. For example, on a Windows system, this default location will be *drive:\Program Files\SAS*. Or you can select the second radio button to indicate that you want to specify an installation directory to be used in place of the default. If you select this radio button, you must enter the complete path to the installation directory in the **Specify an alternate default location** text box in the middle of the screen. Then click **Next**. The **Select which set of help files to install** screen appears.



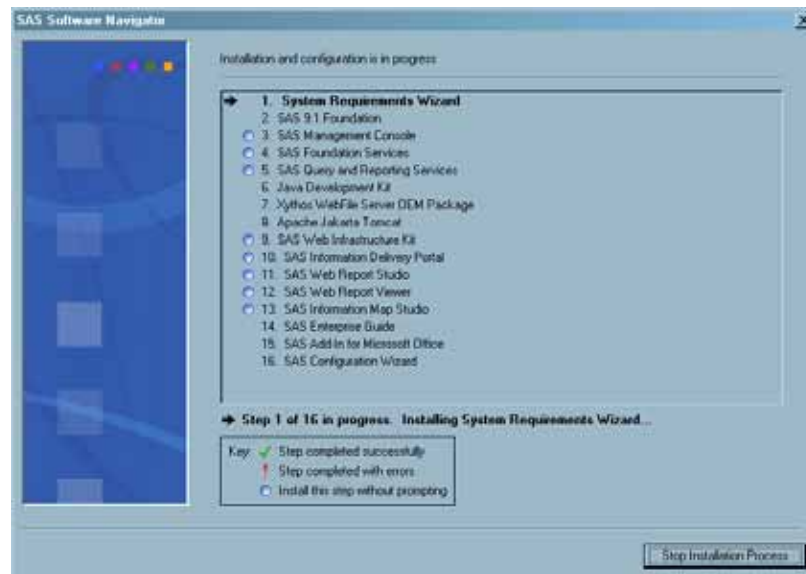
- 9 In the **Select which set of help files to install** screen, select the help files that you want to install. Select the **Install help files in the current**

language only radio button to indicate that you want to install only the help files for the language in which you are running the SAS Software Navigator. Select the **Install help files in all available languages** radio button to indicate that you want to install the help for all languages in which help is available.

After you select your help files, click **Next**. The **Review options before starting the installation process** window appears.



10 Review the list of products that you are about to install; then, click **Install**. At this point, the SAS Software Navigator switches from its information gathering mode to an installation mode.



In its installation mode, the SAS Software Navigator leads you through the following tasks:

- The checking of system requirements on Windows systems. The System Requirements Wizard determines whether your machine meets the prerequisites for all of the software that you are about to install. If certain components are

missing, you can use the requirements wizard to install them. For more information about running this wizard, see “Running the System Requirements Wizard (Windows Only)” on page 94.

- The installation of your software. The navigator installs each product that is shown in the list of products in the order shown. In addition, the installations are chained. That is, you do not need to initiate the installation programs; after installing product 1, the navigator automatically proceeds with the installation of product 2.

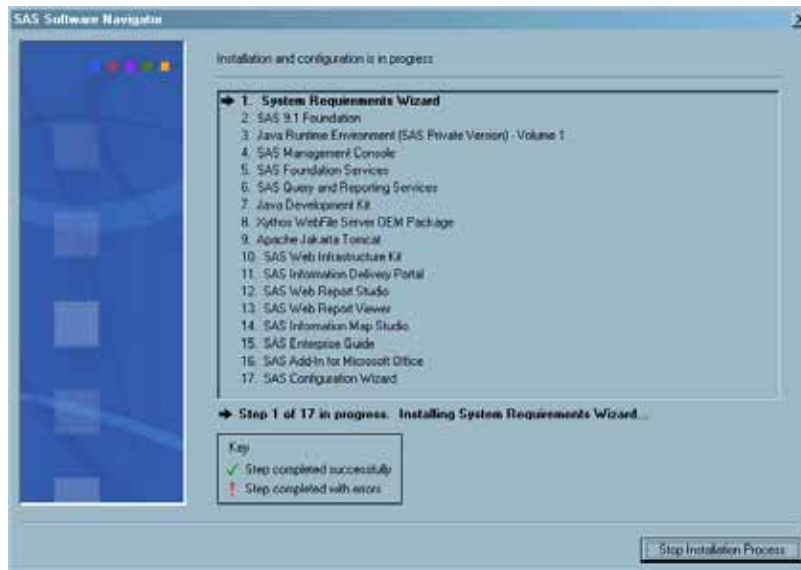
The installation of a particular product can be one of three types: a quiet installation, an interactive installation, or a nonstandard installation.

- Products by which you see a blue sphere can be installed quietly. No input is required from you in a quiet installation. You might see an occasional message, but basically the installation is automatic.
- A few SAS products require an interactive installation program. When the SAS Software Navigator needs to install such a product, it starts this installation program automatically, and you then use the installation program to install the software. For more information about interactive installations, see “Performing Interactive Installations” on page 97.
- A few products—primarily third-party products—require a nonstandard installation. When the SAS Software Navigator needs to install this type of product, it displays an HTML page that directs you to installation instructions and an installation program. You use the instructions and the program to install the product; then, control returns to the navigator. For more information about nonstandard installations, see “Performing Nonstandard Installations” on page 98.
- The configuration of your software. The last product in the SAS Software Navigator’s list of products is the SAS Configuration Wizard. When the navigator reaches this point in its list, it starts the configuration wizard, which configures the software that you have installed on the current machine. Like an interactive installation program, the SAS Configuration Wizard relies on you to supply certain information. For instructions on how to run the configuration wizard, see “Running the Configuration Wizard on Windows and UNIX Systems” on page 100.

Note: Machines on which you install only client-tier software might not require configuration. \triangle

Running the System Requirements Wizard (Windows Only)

When you click the [Install](#) button in the SAS Software Navigator, the navigator switches to a mode in which you can monitor the progress of the installation.



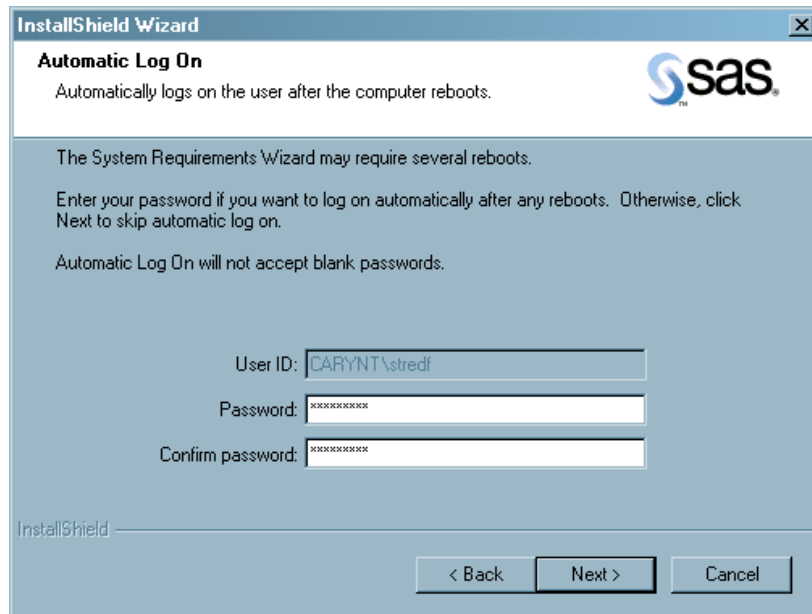
As you can see, the first step in the process is to run the System Requirements Wizard. This wizard starts automatically and enables you to detect missing system requirements and to install the missing components.

Of course, there are many paths through the wizard. However, here is what you would see if you were installing the SAS Foundation on a Windows machine that did not contain the SAS Private JRE.

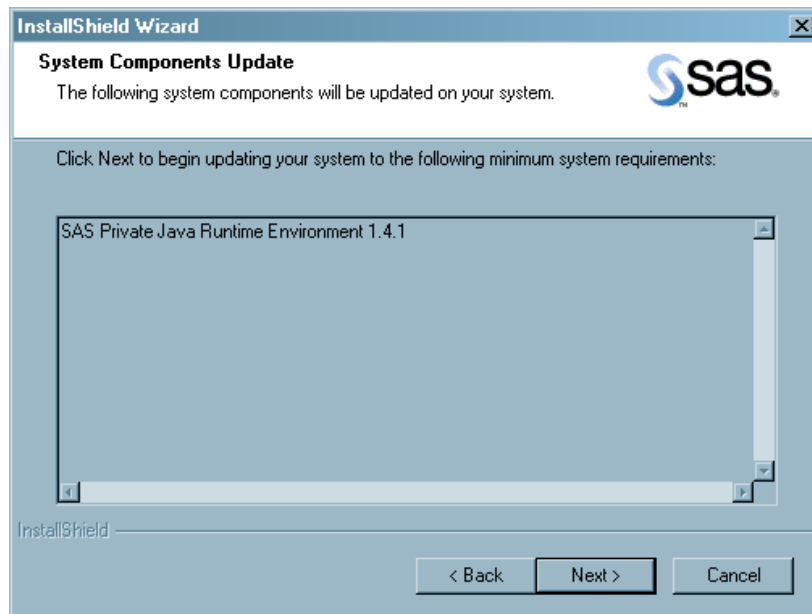
- 1 The wizard starts up automatically.



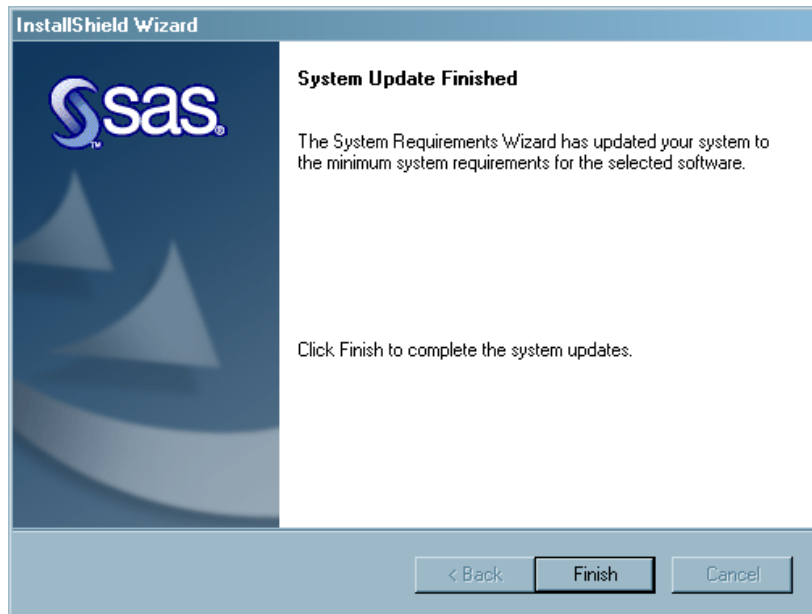
- 2 The wizard prompts you for a user ID and a password to use to log on to the computer after a reboot. (Some of the changes that the System Requirements Wizard makes might cause your system to reboot.)



3 The wizard installs the SAS Private JRE.



4 The wizard finishes.



If the system is rebooted, you must restart the SAS Software Navigator and go back through the set of screens that were discussed in “Supplying the SAS Software Navigator with the Information That It Needs” on page 88. The information that you entered earlier should have been saved, so the screens should contain the information that you entered earlier. (If, for any reason, the screens do not contain the correct information, you should reenter the information that you entered the first time you ran the navigator.) After you click the **Install** button, the installation will resume at the point where you left off.

Performing Interactive Installations

There are some products, for example the SAS Foundation software, that the SAS Software Navigator does not install silently. For these products, the navigator starts an installation wizard like the one shown in the following display.



To install a product, run the installation wizard to completion. The wizards are written to be self documenting; that is, if a screen prompts you for information, text on that screen will explain how you should respond to the prompt.

Note: A few SAS products, such as the SAS Foundation application, have accompanying installation guides. If such a guide is available, you will find it in the “Installation” section of your Installation Kit. \triangle

Performing Nonstandard Installations

For a few products—particularly third-party products such as J2EE servers and WebDAV servers—the SAS Software Navigator asks you to perform a *nonstandard installation*. This means that you use a nonstandard installation program to install the product. For example, to install the Java 2 Software Developer’s Kit, you use an installation program that is provided by Sun Microsystems Inc.

You will know that you have reached a nonstandard installation if the navigator presents you with an HTML page inside a window like the one shown in the following display.



If you encounter such a page, you should perform these steps.

- Read the installation instructions on the page. Or select the **Installation Instructions** link on the page (you might have to scroll down to see it) and then read the installation instructions that are presented.
- On the original HTML page, locate the **Install** link for your operating system, and click that link. This link will be presented in a table, as shown in the following display.



This link will start, or guide you to, the installation program for the appropriate product.

- Use this installation program to install the product, following the installation instructions as necessary.
- After the installation is complete, click **Close** in the window that contains the HTML instructions. A dialog box asks whether you installed the product successfully. Click **Yes**.

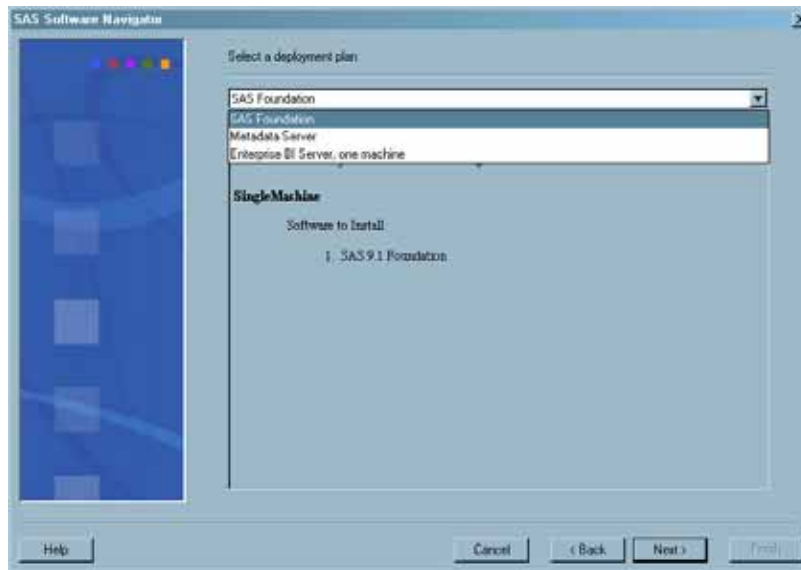
At this point, the SAS Software Navigator will begin installing the next product in its list.

Performing a Personal Installation

A Personal installation is a slight variation on an Advanced installation. It imposes two restrictions.

- A Personal installation does not enable you to use a customized deployment plan.
- A Personal installation does not let you use a multi-machine standard plan.

In other words, the Personal installation allows only for single-machine standard deployment plans. This difference can be seen on the SAS Software Navigator's **Select a deployment** screen.



Note that there is no option to specify a customized plan and that there are no multiple machine plans in the list.

In all other regards, the Personal installation and the Advanced installation are identical. Therefore, if you remember this one difference, you can follow the directions in “Performing an Advanced Installation” on page 87 to perform this type of installation as well.

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 what additional configuration steps you need to perform. You can perform these additional steps manually—generally using SAS Management Console—or you can run a set of scripts to accomplish the same thing. On z/OS systems, you run a script that performs the tasks that the SAS Configuration Wizard performs automatically on the other platforms, and you perform any additional configuration manually. For more information on these subjects, see the following sections:

- “Running the Configuration Wizard on Windows and UNIX Systems” on page 100
- “Configuring SAS Servers on z/OS Systems” on page 106.

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 Foundation software and were configuring the following servers:

- SAS Metadata Server
- SAS Workspace Server
- SAS Stored Process Server

□ SAS Object Spawner.

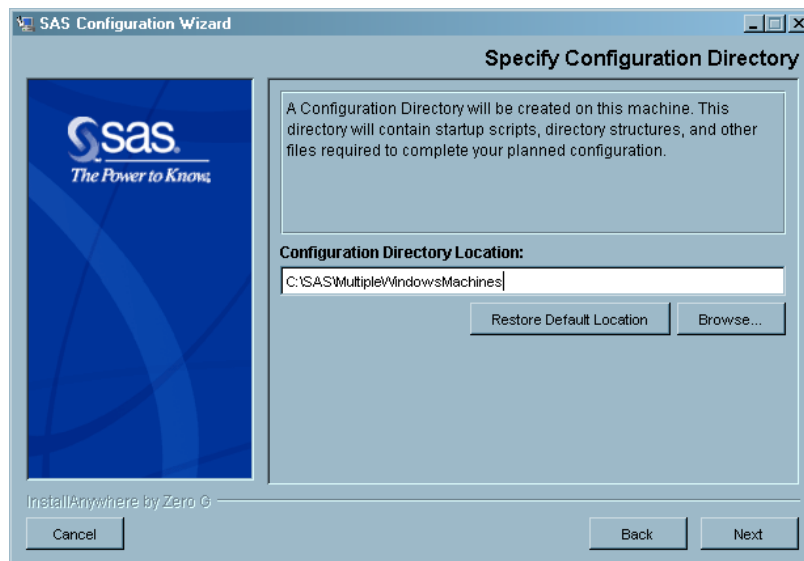
Running the SAS Configuration Wizard on a different machine will be very similar. Keep your pre-installation checklist nearby.

The configuration wizard starts automatically after the SAS Software Navigator installs the last product on a machine, and you will see the following splash screen.



From that point, you perform the following tasks:

- 1 In the wizard's splash-screen window, select a language from the text box at the bottom of the window, and click **OK**.
- 2 In the Introduction window, read the text in the window, and click **Next**.
- 3 You will see the 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, which is described in the following sections:

- “Configuration Directory: Server-Tier Machines” on page 130
- “Configuration Directory: Middle-Tier Machines” on page 132
- Appendix 1, “Understanding the SAS Configuration Environment,” on page 431.

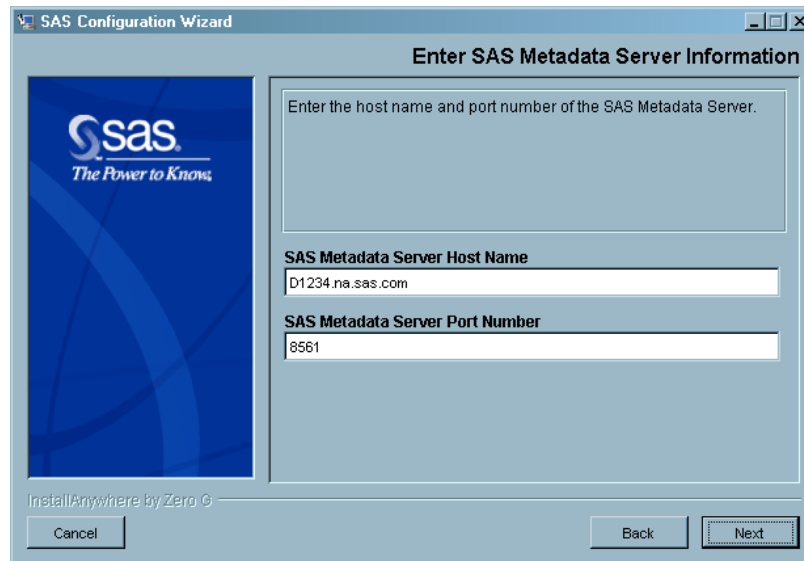
Click **Next**.

- 4 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.

- 5 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 as well.

(For a list of the ports that are used in the default configuration, see “Default Ports” on page 74.)

Click **Next**.

- 6 The SAS Administrator Information window appears.



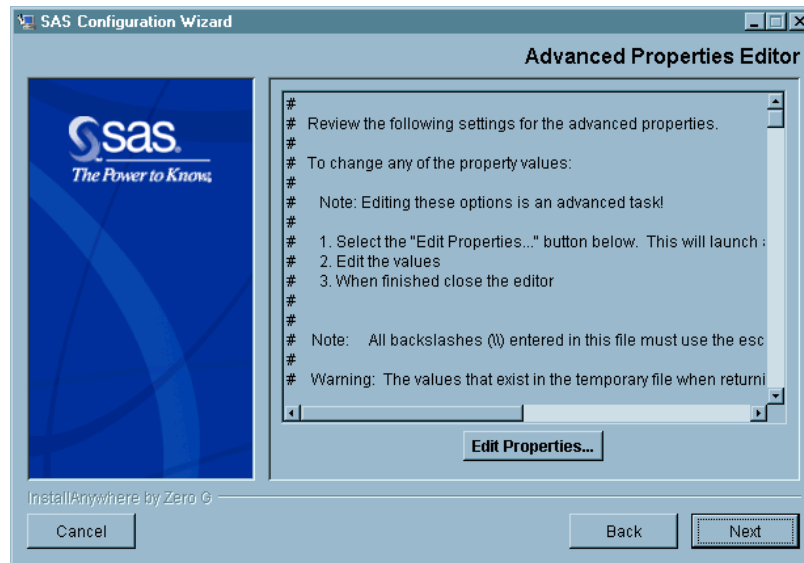
Enter the following information:

- 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**.

Note: The configuration wizard does not check at this point to ensure that this user ID and password can be authenticated by your authentication provider—usually the operating system. Therefore, it is important that you not make a typographical error. This applies to the following steps as well. △

- 7 In the SAS General Server Information window, enter information about the SAS General Server User (**sassrv**). Click **Next**.
- 8 In the SAS Guest Information window, enter information about the SAS Guest account (**sasguest**). Click **Next**.
- 9 In the SAS Trusted User Information window, enter information about the SAS Trusted User (**sastrust**). Click **Next**.
- 10 In the SAS Demo User Information window, enter information about the SAS Demo User (**sasdemo**). Click **Next**.
- 11 The Advanced Properties Editor window appears.



This window enables you to edit a set of properties that the configuration wizard will use while configuring the current machine. These properties may contain a variety of information, including

- the names of the users and groups that you defined during pre-installation
- the location of your configuration directory
- the name of your SAS application server
- the location of your SAS installation directory.

Some of the property values (such as user IDs) are values that you entered while running the wizard, while others (such as most port numbers) are default values. To get a complete idea of which properties you can edit, click the **Edit Properties** button. The wizard opens the properties file in an editor. The comments in the file provide a description for each property.

For most of the properties, you will be able to use the default values. However, there are a couple of types of information that you should verify:

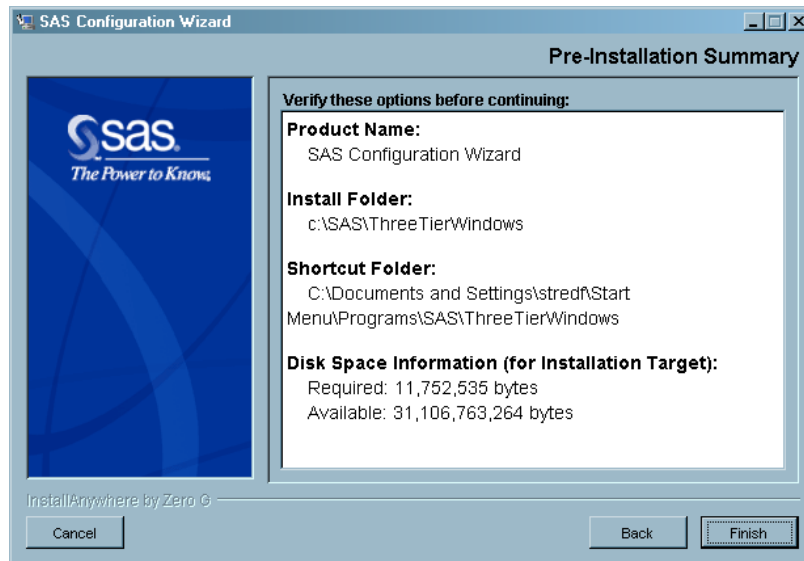
- Verify the port numbers in the file. Most of the port numbers are default values. If any of these ports is being used by another application, you must supply a new port number in the properties file.
- Verify the values that are associated with your servlet container or J2EE server. For example, if you are using the BEA WebLogic Platform, the properties file will contain values for the name and port number of an Administration Server and the name and port number of a Managed Server. The values of these properties must match the values that you specified when you installed and configured these servers.

CAUTION:

It is important that all of the properties that are defined in this file have the correct values before you proceed. The SAS Configuration Wizard uses these properties while configuring your machine and could possibly write an incorrect value to multiple locations. Δ

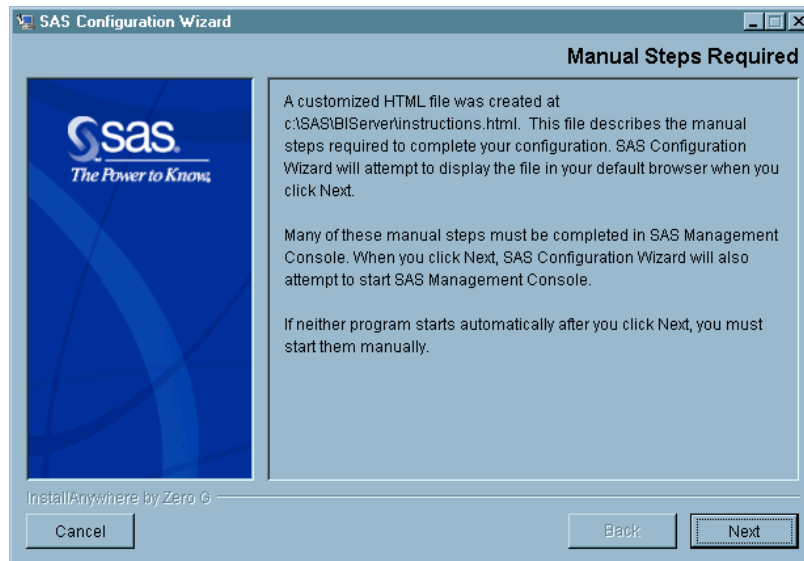
After setting the advanced properties, click **Next**.

12 The Pre-Installation Summary window appears.



Click **Finish**.

- 13 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 130.
- 14 When the automated configuration is complete, the Manual Steps Required window appears.



Read the instructions about manual configuration. Then, click **Next**.

- 15 The Configuration Complete window appears. Click **Done**.
- 16 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 them manually. (Performing the first task in the following list 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 what additional 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 the Steps Listed in `instructions.html`” on page 108.
- Create a metadata profile. This profile enables the user of SAS Management Console to connect to the metadata server.

See “Performing the Steps Listed in `instructions.html`” on page 108 for information on how to perform the configuration steps listed in `instructions.html`.

Configuring SAS Servers on z/OS Systems

Before you configure the SAS software on a z/OS system, you must complete these steps:

- 1 Fill out the pre-installation checklist shown in “Pre-Installation Checklist for z/OS” on page 61.
- 2 Submit the pre-installation checklist to the data center staff so that they can perform the tasks that are described in the checklist.
- 3 Install SAS Foundation for z/OS by following the instructions in the *Installation Instructions for SAS 9.1.3 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 on 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 that are needed for the SAS Intelligence Platform 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` data set 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:
 - a Verify that the COPYIA job's return code was 0.
 - b View the output from the COPYIA1, UNTAR, and SHELLOUT steps for possible problems.

Log On to the USS Shell

Invoke the UNIX System Services shell (or `rlogin` to your z/OS host).

Note: You must be logged on as the SAS user (`sas`), which you specified on your pre-installation checklist. Δ

Edit the `configuration.properties` File

Edit the file `configuration.properties` to add the values that are 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 in “Edit and Submit &prefix.W0.SRVCNTL(COPYIA) Job.”)
- 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 in “Edit and Submit &prefix.W0.SRVCNTL(COPYIA) Job.”)
- 2 Run the script using the following command:

```
./deploy_IA.sh -properties configuration.properties
```

- 3 Review the output for potential errors.

Verify the Results of Running the Script

Check the `&prefix.W0.SRV*` data sets 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 the Steps Listed in `instructions.html`” on page 108.

Performing the Steps Listed in instructions.html

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 additional tasks that you must perform to complete the configuration of the current machine.

On Windows and UNIX systems, the SAS Configuration Wizard attempts to start SAS Management Console, to connect the SAS Management Console to the metadata server, and to display `instructions.html` in a browser before the wizard exits. If the SAS Configuration Wizard is unsuccessful, you must start these applications yourself.

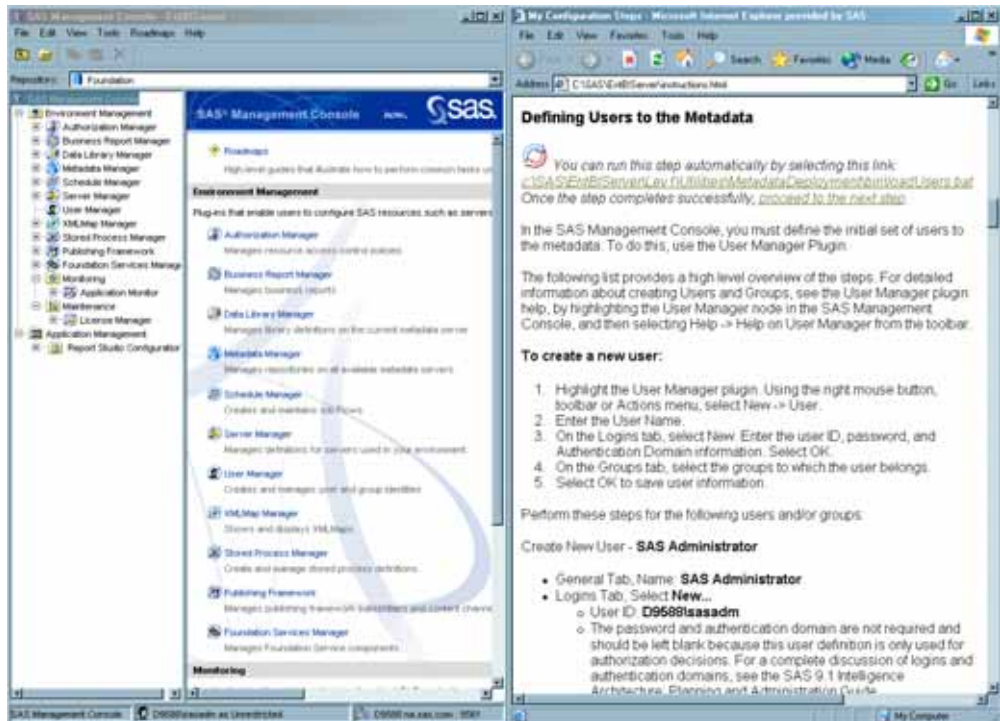
If you are configuring SAS servers on a z/OS system, the situation is a little different. SAS Management Console does not run on z/OS, so you must move your `instructions.html` file to a Windows or UNIX system from which you can communicate with your z/OS system. On that system, perform these steps:

- 1 Install SAS Management Console. (For information on how to install a product without having a planning file, see Appendix 2, “Software Index Installations,” on page 443.)
- 2 Start SAS Management Console.
- 3 Connect to the metadata server by creating a metadata profile and opening it.
- 4 Use a Web browser to display `instructions.html`.

From this point on—for all platforms—follow the directions in the file `instructions.html` *precisely*. On Windows and UNIX systems, you have two options for carrying out the instructions. You can either perform each step manually or run a series of scripts that perform the steps listed in the file. For z/OS systems, you must perform the steps manually.

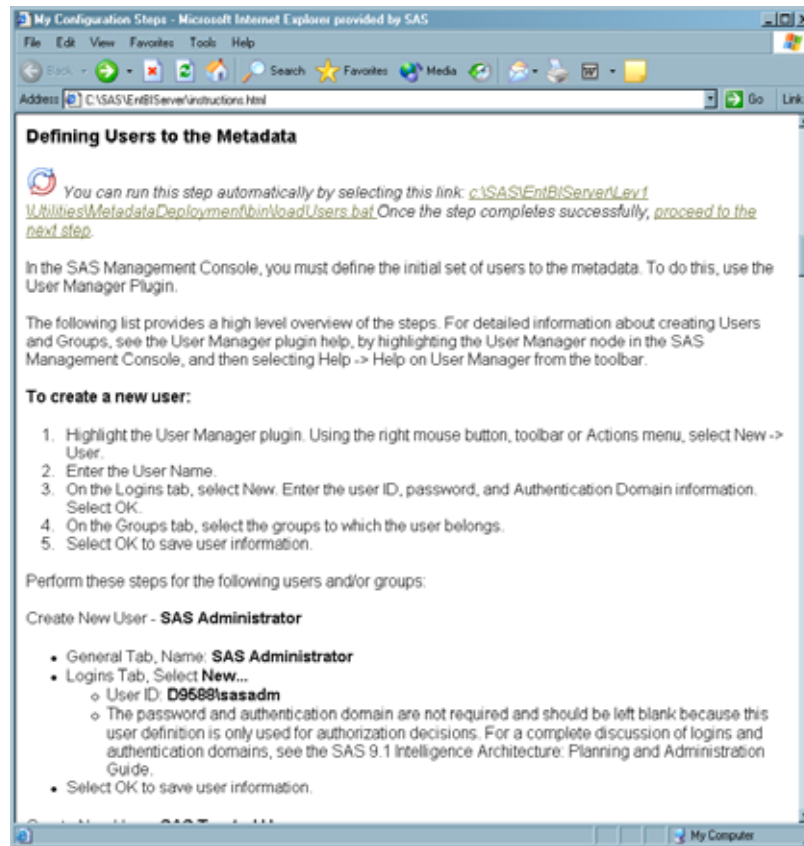
- Although performing the tasks listed in `instructions.html` manually takes a little longer than running the scripts, there are advantages to taking the former approach: you will see exactly what metadata is being written to your foundation metadata repository, and you will gain some valuable experience using SAS Management Console.

If you decide to follow the instructions manually, we recommend that you position the SAS Management Console and the instructions side by side, as shown in the following display.



In many places, you are asked to enter text in SAS Management Console. In the instructions, such text is formatted in a bold font. Positioning these windows side by side makes it easy to cut text from the instructions and to paste that text into SAS Management Console.

- As an alternative to performing these instructions manually, you can perform most of the tasks by running scripts. The following display shows part of an **instructions.html** file on a Windows platform.



The symbol that is near the upper left-hand corner of the window indicates that a script is available to perform the instructions in a particular section. In this case, you can run the script by clicking on the link `c:\SAS\EntBIServer\Level1\Utilities\MetadataDeployment\bin\loadUsers.bat`. (On a UNIX system, you would have to run the script by calling it from a terminal window.) After you have run such a script, you can move to the next section of the instructions by clicking the link **proceed to the next step**.

CAUTION:

You cannot perform all of the instructions by running a script. You must perform some steps manually. Be careful not to skip these steps. △

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 “Establishing Basic Protections” on page 140 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 7.2 Summary of Metadata Identities

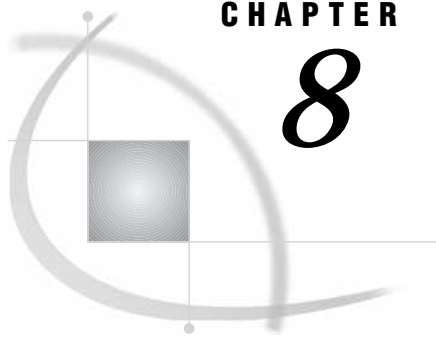
Metadata Identities	Logins			Group Membership Information
	User ID*	Password**	Authentication Domain	
User: SAS Administrator	sasadm			
User: SAS Trusted User	sastrust			member of: SAS System Services group member of: SAS General Servers group
User: SAS Guest User	sasguest	*****	DefaultAuth	
User: SAS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

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

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

***You only need this metadata identity if you have a middle 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 383.



CHAPTER

8

Troubleshooting Your Initial Setup

<i>Overview of Troubleshooting Your Initial Setup</i>	114
<i>Troubleshooting SAS Servers</i>	114
<i>Metadata Server</i>	114
<i>Metadata Profile Contains an Incorrect Host Name or Port Number</i>	114
<i>SAS Administrator Is Not Listed in adminUsers.txt</i>	115
<i>Object Spawner</i>	116
<i>Object Spawner Is Not Running</i>	117
<i>Metadata Server Cannot Authenticate the Object Spawner</i>	117
<i>SAS Trusted User Is Not Authorized to Read the Server Definition</i>	118
<i>The Object Spawner Is Not Configured to Start a Workspace Server or Stored Process Server</i>	118
<i>Stored Process Server</i>	119
<i>Object Spawner Does Not Have the Credentials to Start the Server</i>	119
<i>Object Spawner Cannot Read the Password for the SAS General Server User</i>	120
<i>Object Spawner Does Not Have the Correct Command to Start the Stored Process Server</i>	121
<i>Object Spawner Does Not Have the Current Metadata for the Stored Process Server</i>	121
<i>Workspace Server</i>	122
<i>Object Spawner Does Not Have the Correct Command to Start the Workspace Server</i>	122
<i>Object Spawner Does Not Have the Current Metadata for the Workspace Server</i>	122
<i>Troubleshooting Web Servers and Web Applications</i>	122
<i>SAS Services Application</i>	122
<i>RMI Port Is in Use</i>	122
<i>Apache Tomcat</i>	123
<i>Another Application Is Using Port 8080</i>	123
<i>Insufficient Memory on Host System</i>	123
<i>Start-Up Script Cannot Find the Java 2 SDK</i>	124
<i>Web Applications</i>	124
<i>Initial Page Cannot Be Loaded</i>	124
<i>Pages Take a Long Time to Load</i>	125
<i>SAS Web Report Studio</i>	126
<i>Apache HTTP Server Is Not Running</i>	126
<i>Apache HTTP Server Configuration File Is Set Up Incorrectly</i>	126
<i>You Did Not Create the Directory That Serves as the Content Base Path</i>	126
<i>Your WebDAV Server Is Configured Incorrectly</i>	126
<i>You Did Not Set the Properties of the BIP Tree Correctly</i>	127
<i>SAS Information Delivery Portal</i>	127
<i>User Is Not Registered in the Metadata Repository</i>	127
<i>User Does Not Have the Correct Permissions</i>	127
<i>User's Metadata Identity Does Not Contain a Domain (Windows Only)</i>	127

Overview of Troubleshooting Your Initial Setup

Chapter 7, “Installing and Configuring Your Software,” on page 79 led you through the installation and initial configuration of your system. If all went well, you have been able to successfully test the connections to your SAS servers, and if you are using any SAS Web applications, you have been able to start the SAS Services Application and your J2EE server and to start and log on to your SAS Web applications.

If you encountered problems in any of these areas, look in one of the following sections for information on how to troubleshoot your problem:

- “Troubleshooting SAS Servers” on page 114
- “Troubleshooting Web Servers and Web Applications” on page 122

Troubleshooting SAS Servers

During the initial configuration of your system, you were asked to connect to the metadata server, create a metadata repository, define your SAS application server, and test the connections to your SAS servers. If you were unable to perform any of these tasks, see the appropriate section below:

- If you cannot connect to the metadata server or cannot create a foundation repository, see “Metadata Server” on page 114.
- If you cannot connect to either a stored process server or a workspace server, see “Object Spawner” on page 116.
- If you are able to connect to a workspace server, but not a stored process server, see “Stored Process Server” on page 119.
- If you are able to connect to a stored process server, but not a workspace server, see “Workspace Server” on page 122. If you cannot start a workspace server and do not have a stored process server, see “Object Spawner” on page 116 as well.

Metadata Server

When you run the SAS Configuration Wizard on your metadata-server host, you are prompted for several pieces of information that the wizard will use to help you establish your initial connection to the metadata server and to create your initial metadata repository. These bits of information include

- the name of the host on which the metadata server will run
- the port on which the server will listen for requests
- the name of the SAS Administrator (`sasadm`).

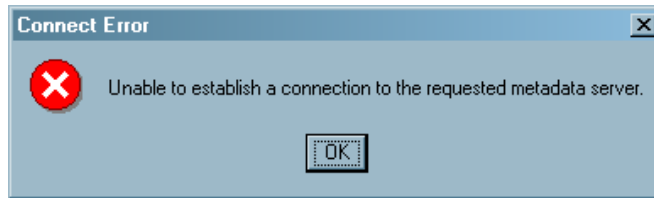
If you mis-type any of this information, you are likely to experience one of the problems that are discussed below.

Metadata Profile Contains an Incorrect Host Name or Port Number

Explanation: The SAS Configuration Wizard prompts you for the name of the host on which the metadata server will run and the port number on which the server will listen. The wizard then uses this information to create a metadata profile that it will use to help you connect to the metadata server for the first time. (This metadata profile also contains the user ID for the SAS Administrator, but not a password.) After the wizard has performed all of the configuration tasks that it can perform without your help, it starts SAS Management Console and attempts to open the metadata profile. In

addition to the information in the profile, SAS Management Console needs a password to establish a connection to the metadata server, so it displays an **Enter your user information** dialog box. The **Username** text field will contain the user ID of the SAS Administrator; you need to supply a password and click **OK**. If the host name and port number in the metadata profile (as well as the user name and password) are correct, the connection will be established.

Confirmation: If the host name or port number in the profile is incorrect, you will see the following error message:



In addition, if you check the metadata-server log file (*path-to-config-dir\Lev1\SASMain\MetadataServer\logs\MetadataServerdate.log*), you should see no error message about the failed connection.

Fix: To fix this problem with the metadata profile, follow these steps:

- 1 Click **OK** in the **Connect Error** dialog box. The Open a Metadata Profile dialog box appears.
- 2 Click **Edit**. You will see a Metadata Profile window.
- 3 Click **Next**. You will see a Connection Information window.
- 4 Enter the correct information in the **Machine** and **Port** text boxes. Then click **Finish**. You will be returned to the Open a Metadata Profile dialog box.
- 5 Click **OK**. The **Enter your user information** dialog box appears.
- 6 Supply a user name and password, and click **OK**.

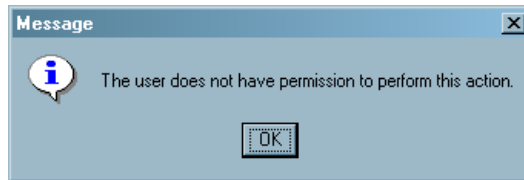
SAS Administrator Is Not Listed in adminUsers.txt

Explanation: If you supplied an incorrect user ID for the SAS Administrator while running the SAS Configuration Wizard, the user ID that the SAS Configuration Wizard places in the **Enter your user information** dialog box will be incorrect and probably will not match an account in your authentication provider. If you enter a password and click **OK**, you will see the Connect Error dialog box that is shown above. In addition, you will see the following error message in the metadata server log file:

```
ERROR: Error authenticating user incorrect-ID in function LogonUser.
        Error 1326 (Logon failure: unknown user name or bad password.)
ERROR: Access denied.
```

You can connect to the metadata server by returning to the **Enter your user information** dialog box, supplying the correct user ID and password for the SAS Administrator, and clicking **OK**. However, when you try to create your first metadata repository, you will be unable to create it.

Confirmation: When you attempt to follow the instructions that are generated by the SAS Configuration Wizard to define your foundation metadata repository, you get the following error:



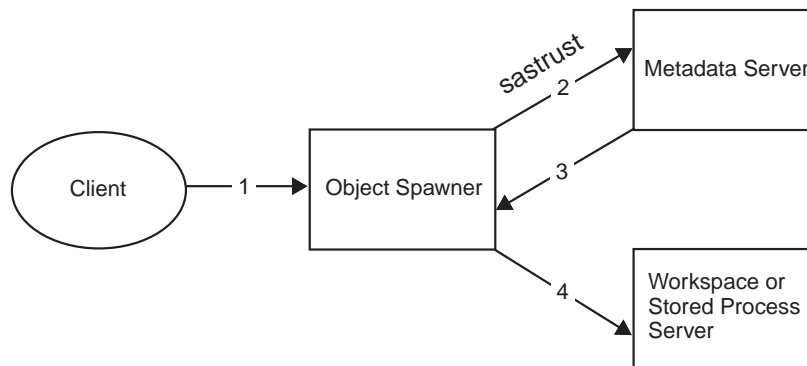
You get this error because the SAS Administrator is not an unrestricted user.

Fix: Unrestricted users must be listed in the file `path-to-config-dir\Lev1\SASMain\MetadataServer\adminUsers.txt`. Because you entered an incorrect ID while running the SAS Configuration Wizard, the wizard wrote that incorrect value to `adminUsers.txt`. To fix the problem, edit the `adminUsers.txt` file to correct the user ID (of the SAS Administrator). Then restart the metadata server.

Object Spawner

An object spawner is used to start the workspace servers and the stored process servers. The figure below illustrates how the object spawner interacts with a client and the metadata server in the course of starting a workspace server or stored process server.

Figure 8.1 How the Object Spawner Starts a Workspace Server or Stored Process Server



The following list explains the steps in the diagram:

- 1 A client sends a request to the object spawner that the spawner create a workspace server or stored process server. (You can use SAS Management Console to test the object spawner. The Server Manager plug-in enables you to test a connection to a workspace server or stored process server.)
- 2 To get the server definition that it needs to start the workspace server or stored process server, the object spawner connects to the metadata server as the SAS Trusted User (`sastrust` by default).
- 3 Assuming that the SAS Trusted User has permission to read the metadata for the workspace server or stored process server, the object spawner reads that metadata. This metadata includes the `sas` command that the spawner will use to start the new server.
- 4 The object spawner executes the `sas` command.

If you are unable to start a workspace server *and* you are unable to start a stored process server, something is probably wrong with your object spawner setup. See the following sections to troubleshoot this problem.

Object Spawner Is Not Running

Explanation: No one has started the object spawner.

Confirmation: Under Windows, use the Task Manager to determine whether an object spawner is running on a particular machine. Go to the **Processes** tab and look for a process named `objspawn.exe`. If no such process exists, the object spawner is not running. On UNIX systems, use the `ps` command to determine whether an `objspawn` process exists. Again, if this process does not exist, the object spawner is not running.

You can also look in the object spawner's log file to diagnose this problem. This log file is located at `path-to-config-dir\Lev1\SASMain\ObjectSpawner\logs\objspawn.log`. Open this file in a text editor. If you do not see the message "Objspawn has completed initialization," the object spawner is not running.

Fix: Start the object spawner by using the instructions in "Starting and Stopping Your SAS Servers" on page 134.

Metadata Server Cannot Authenticate the Object Spawner

Explanation: The object spawner must read a server definition from the metadata server before it can start a workspace server or stored process server. And before it can read this definition, the object spawner must be authenticated by the metadata server. The spawner attempts to connect to the metadata server by using the user name and password that are stored in the file

`path-to-config-dir\Lev1\SASMain\ObjectSpawner\OMRConfig.xml` (`sastrust` by default). If this connection fails—because the metadata server cannot authenticate the user—the spawner will not be able to start the workspace server or stored process server.

Note: The SAS Configuration Wizard creates the `OMRConfig.xml` file. The values of the attributes `Userld` and `Password` are values that you entered when you were prompted for the user ID and password of the SAS Trusted User. If you made a typographical error at that point, you will run into the problem explained above. △

Confirmation: Look in the object spawner's log file: `path-to-config-dir\Lev1\SASMain\ObjectSpawner\logs\objspawn.log`. If the metadata server was unable to authenticate the object spawner, you will see an error message that is similar to this error message:

```
ERROR: An attempt to communicate with the SAS Metadata Server failed.
ERROR: Error authenticating user sastrust in function LogonUser.
       Error 1326 (Logon failure: unknown user name or bad password.)
ERROR: Access denied.
```

On Windows systems, you can also use the Event Viewer to diagnose this problem. Go to the **Security** section, and look for a Failure Audit event that occurred at the time that you tried to start the workspace server or stored process server. If you look at the properties of this event, you will see a description that is similar to this description:

```
Logon Failure:
Reason:          Unknown user name or bad password
User Name:      sastrust
Domain:         D1234
Logon Type:     4
Logon Process:  Advapi
Authentication Package: Negotiate
Workstation Name: D1234
```

Fix: Because this error is usually the result of a typographical error in the SAS Configuration Wizard, the standard fix is to edit the file `OMRConfig.xml` so that the

values of the `UserId` and `Password` attributes (of the `Login` element) contain valid credentials for the SAS Trusted User. The password can be in clear text, but it should be encoded. You can encode the password using `PROC PWENCODE`:

```
PROC PWENCODE IN='password';
RUN;
```

After you have entered the correct credentials in `OMRConfig.xml`, you must restart the object spawner.

SAS Trusted User Is Not Authorized to Read the Server Definition

Explanation: Even if the object spawner is able to connect to the metadata server, it is possible that the spawner will not be able to read the necessary metadata for the workspace server or stored process server. By default, the `sastrust` account has permission to read server definitions, both as a member of the `PUBLIC` group and as a member of the SAS System Services group. However, someone might have changed the metadata access controls so that `sastrust` no longer has that permission. For example, someone might have explicitly denied the `PUBLIC` group `readMetadata` access to the `SASMain` logical workspace server. Because `sastrust` is a member of the `PUBLIC` group, it will no longer be able to read metadata for the workspace server.

Note: In this case, the explicit denial of the `readMetadata` permission to `PUBLIC` overrides the inherited grant of this permission to the SAS System Services group. Δ

Confirmation: Look at the object spawner log. If the log indicates that the object spawner initialized successfully, but there is no error message stamped with the time at which you tried to start the workspace server, the problem could be that `sastrust` does not have access to read the server definition. The fact that there is no error indicates that the SAS Trusted User was authenticated by the metadata server.

Fix: Using SAS Management Console, grant the SAS Trusted User permission to read the metadata for the workspace server or stored process server.

The Object Spawner Is Not Configured to Start a Workspace Server or Stored Process Server

Explanation: As part of the definition of an object spawner, you specify what types of servers the object spawner can start. If you do not specify that the object spawner can start a workspace server, the spawner will not be able to start such a server.

Confirmation: The symptoms of this problem are similar to those you see when the spawner does not have access to a server definition. In the object spawner log, you should see a message that says that the object spawner initialized successfully and that no error message was written at the time the spawner attempted to start the server. To determine the cause of the problem for certain, look at the object spawner's properties:

- 1 In SAS Management Console, right-click the object spawner icon (in the Server Manager), and select **Properties** from the pop-up menu. A Spawner Properties dialog box appears.
- 2 Select the **Servers** tab in this dialog box.

If the type of server that you are trying to start does not appear in the **Selected servers** list, you have identified the problem.

Fix: Change the object spawner's properties to indicate that the spawner *can* start the type of server that you want it to start. Then restart the object spawner.

Stored Process Server

If you have confirmed that the object spawner can connect to the metadata server and read the necessary server definition—as explained in “Object Spawner” on page 116—and the spawner still cannot start a stored process server, the problem probably lies in one of the following areas:

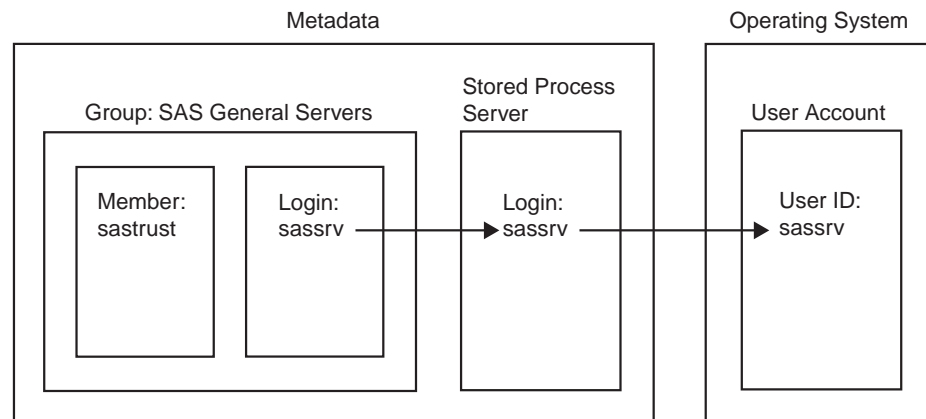
- The object spawner cannot get the user ID and password that it needs to start a stored process server.
- The object spawner can read the user ID, but it cannot read the password. The object spawner must start a stored process server.
- The object spawner is trying to use an invalid command to start the stored process server.
- The object spawner does not have an up-to-date definition of the stored process server.

For more information about how to detect and correct these problems, see the sections below.

Object Spawner Does Not Have the Credentials to Start the Server

Explanation: By default, the business intelligence platform is set up so that the object spawner starts stored process server processes using the SAS General Server User (**sassrv**) account. For this to work, several metadata objects and an operating system user account need to be set up correctly. See the following figure.

Figure 8.2 Setup Required for Starting Stored Processes



The metadata for the stored process server should specify that the object spawner will need to have the credentials for the SAS General Server User (**sassrv**) to start a stored process server. The SAS Trusted User account, which the object spawner uses to connect to the metadata server, then needs a way to read the password for the **sassrv** account. This is where the SAS General Servers group comes in. Because the SAS Trusted User is a member of this group, it can read the logins that are associated with the group. One of these logins must contain the user ID and password for the SAS General Server User.

Confirmation: If the metadata that was discussed previously is not set up correctly, you will see a message that is similar to the following one in the object spawner log:

```
ERROR: This server (A5U46TRS.AT000002) cannot be spawned without credentials
       which specify the server process username. You can specify these credentials
```

using SAS Management Console in the Advanced Options of the server definition (found under the Options tab in the server's properties dialog).

This log file is located at

path-to-config-dir\Lev1\SASMain\ObjectSpawner\logs\objspawn.log.

Fix: Perform the following steps:

- 1 Use the User Manager plug-in to SAS Management Console to make sure that a metadata group named SAS General Servers has been defined. The SAS Trusted User (**sastrust**) must be a member of this group, and the group must contain a login for the SAS General Server User (**sassrv**). This login must contain the user ID **sassrv** and the password that this user needs to be authenticated by the operating system.
- 2 Use the Server Manager plug-in to SAS Management Console to examine the properties of your stored process server. The login for this server must be set to **sassrv**. To view the current login, perform these steps:
 - a Right-click the icon that represents the stored process server, and select **Properties**. A Stored Process Server Properties dialog box appears.
 - b Select the **Options** tab.
 - c Click **Advanced Options**. The Advanced Options dialog box appears. The **Credentials** tab is displayed by default. The **Login** list box should be showing the SAS General Server User's user ID.
- 3 Make sure that the operating system user account for **sassrv** contains the same password as the metadata object for this user.

Object Spawner Cannot Read the Password for the SAS General Server User

Explanation: Even if you have set up the metadata and user account described in "Object Spawner Does Not Have the Credentials to Start the Server" on page 119, the object spawner will not be able to read the password it needs to start a stored process server if the SAS Trusted User is an unrestricted user of the metadata. This is true because an unrestricted user cannot read passwords.

Confirmation: If the SAS Trusted User is an unrestricted user, you will see an error message that is similar to this one in the object spawner log:

```
ERROR:   Error authenticating user sassrv in function LogonUser.
         Error 1326 (Logon failure: unknown user name or bad password.)
ERROR:   Access denied.
```

On Windows systems, you can also use the Event Viewer to diagnose this problem. Go to the **Security** section, and look for a Failure Audit event that occurred at the time that you tried to start the stored process server. If you look at the properties of this event, you will see a description that is similar to this one:

```
Logon Failure:
Reason:           Unknown user name or bad password
User Name:       sassrv
Domain:          D1234
Logon Type:      4
Logon Process:   Advapi
Authentication Package: Negotiate
Workstation Name: D1234
```

Fix: Make sure that the SAS Trusted User is not an unrestricted user. Use a text editor to remove the asterisk that precedes this user's ID from the file **adminUsers.txt** (which is located in the directory *path-to-config-dir*\Lev1\SASMain\MetadataServer). Then restart the metadata server. The SAS Trusted User will now be an administrative user, but not an unrestricted user.

Object Spawner Does Not Have the Correct Command to Start the Stored Process Server

Explanation: When you first configure a machine on which you will run a stored process server, the SAS Configuration Wizard instructs you to enter the `sas` command that the object spawner will use to start the server. This command is stored in the metadata repository as part of the definition of the stored process server. If you make a cut-and-paste error, a typographical error, or any other error while you type this command, the object spawner will probably not be able to start a stored process server.

Confirmation: This problem can be difficult to diagnose by looking at the object spawner log file, because the errors that you see in the log will depend on what the error is in the command. For example, suppose that the instructions that were generated by the SAS Configuration Wizard tell you to enter the following command in the metadata:

```
sas -config "C:\SAS\BIEntServerMin\Levl\SASMain\StoredProcessServer\
sasv9_StorProcSrv.cfg"
```

If you cut this command from `instructions.html` and paste it into SAS Management Console—and omit the initial 's'—the log will contain a message from which you could infer the problem:

```
ERROR: Unable to launch the process; CreateProcessAsUser returned rc 2
(The system cannot find the file specified.)
```

However, if you omitted the closing quotation mark instead, the log file messages would not be as helpful.

Probably the easiest way to check for this error is to compare the command that is in `instructions.html` with the command that is stored in the stored process server definition. You can find `instructions.html` in your configuration directory. You can see the command that is stored in the metadata by using SAS Management Console to look at the properties of the stored process server:

- 1 In the Server Manager, right-click the icon that represents the stored process server, and select **Properties** from the pop-up menu. The Stored Process Server Properties dialog box appears.
- 2 Click the **Options** tab. The current command is shown in the **Command** text field.

You can also find the command that the object spawner tried to use to start the stored process server by examining the object spawner log file. Search for the string "Command being used is".

Fix: Edit the `sas` command in the metadata so that it matches the command in `instructions.html`.

Object Spawner Does Not Have the Current Metadata for the Stored Process Server

Explanation: The object spawner reads the metadata for a stored process server when the object spawner starts. Thus, if you have to correct the `sas` command that is used to start a stored process server or some other piece of metadata, the object spawner will not read this updated metadata until you restart the object spawner.

Confirmation: You change the definition for your stored process server, but the changes do not have any effect.

Fix: Restart the object spawner. Then test the connection to your stored process server again.

Workspace Server

If you have set up your object spawner correctly, and you still cannot connect to a workspace server, the problem probably lies in one of the following areas.

Object Spawner Does Not Have the Correct Command to Start the Workspace Server

This case is analogous to the case in which the object spawner does not have the correct command to start a stored process server. See “Object Spawner Does Not Have the Correct Command to Start the Stored Process Server” on page 121 for information on how to detect and resolve this problem.

Object Spawner Does Not Have the Current Metadata for the Workspace Server

This case is analogous to the case in which the object spawner does not have the current metadata that is needed to start a stored process server. See “Object Spawner Does Not Have the Current Metadata for the Stored Process Server” on page 121 for information on how to detect and resolve this problem.

Troubleshooting Web Servers and Web Applications

This section discusses some of the common problems that can occur in systems where you have installed Web applications. This list summarizes what topics are covered:

- If you cannot start the SAS Services Application, see “SAS Services Application” on page 122.
- If you can start the SAS Services Application but cannot start Apache Tomcat, see “Apache Tomcat” on page 123.
- If you can start your J2EE server but cannot start any of your Web applications, or if your Web applications perform poorly, see “Web Applications” on page 124.
- If you are able to start SAS Web Report Studio but cannot log on, see “SAS Web Report Studio” on page 126.
- If you are able to start SAS Information Delivery Portal but cannot log on, see “SAS Information Delivery Portal” on page 127.

SAS Services Application

The SAS Services Application provides a set of reusable services to Web applications such as SAS Web Report Studio and SAS Information Delivery Portal. These services run outside your Web server and are accessed by using Java Remote Method Invocation (RMI). If you are unable to start the SAS Services Applications, see the following section.

RMI Port Is in Use

Explanation: By default, the SAS Services Application uses port 5099 to communicate with the Web applications that use its services. If this port is being used by another application, you will not be able to start the SAS Services Application.

Confirmation: If the RMI port is in use by another application, you will see a message similar to this message in a command prompt or shell:

```
[WARN] com.sas.services.deployment.RMIConfiguration ---
Unable to locate RMI registry
java.rmi.ConnectIOException: non-JRMP server at remote endpoint
```

You can also look for this error message in the log file that is located in *path-to-config-dir\Lev1\web\Deployments\RemoteServices\logs*.

Fix: You can configure the SAS Services Application to listen on a different port. You do this by editing the file *sas_services_idp_remote_omr.xml*, which is located in the directory *path-to-config-dir\Lev1\web\Deployments\RemoteServices*. Search for the XML element `TCPIPConnection`, and change the value of the port attribute to something other than 5099.

Apache Tomcat

If you have installed Apache Tomcat as your servlet container and have run the SAS Configuration Wizard, the configuration wizard will have created a script called `startServletContainer.extension` that you call, either directly or indirectly, to start Tomcat. When you call this script, Tomcat can fail to start for any one of several reasons. For further information, see the following sections.

Another Application Is Using Port 8080

Explanation: By default, Tomcat listens for HTTP requests on port 8080. If another application is already using this port, Tomcat will not be able to start.

Confirmation: If Tomcat is unable to start because port 8080 is already in use, you will see an error message similar to the following one in a command prompt or shell:

```
SEVERE: Error initializing endpoint
java.net.BindException: Address already in use: JVM_Bind:8080
```

Fix: Configure Tomcat to listen on a port that is not being used. You can do this by editing the file *Tomcat-install-dir\conf\server.xml*. Search for the XML element that begins with this string:

```
<Connector classname="org.apache.coyote.tomcat4.CoyoteConnector" port="8080"
```

Then, change the value of the port attribute.

Note: If you make this change, you will need to make a corresponding change in the URLs that you use to start your Web applications. That is, instead of starting SAS Web Report Studio with the URL `http://host-name:8080/SASWebReportStudio`, you would use the URL `http://host-name:new-port-number/SASWebReportStudio`. △

Insufficient Memory on Host System

Explanation: The script `startServletContainer.extension` sets some Catalina options—Catalina is another name for Tomcat 4.x—and then calls the Catalina start-up script. One of these options specifies the minimum amount of memory that must be available to Tomcat in order for it to run. For example, the option `-Xms512m` indicates that 512 MBs of memory must be available. If the minimum amount of memory is not available, the servlet container will not start.

Confirmation: If Tomcat is unable to start because of insufficient memory, you will see an error message that is similar to the following message in a command prompt or shell:

```
Error occurred during initialization of VM
Could not reserve enough memory for object heap
```

Fix: Add more memory to the machine on which your servlet container will run. A less desirable solution is to edit the `startServletContainer` script so that it attempts to reserve less memory for Tomcat's use.

Start-Up Script Cannot Find the Java 2 SDK

Explanation: The script `startServletContainer.extension` also sets the value of the environment variable `JAVA_HOME`. The value of this environment variable must be the full path to the installation directory for the Java 2 SDK, for example, `C:\j2sdk1.4.2_02`. If this directory does not exist, Tomcat will not start. (If you do not have the correct version of the Java 2 SDK installed, the directory probably will not exist.)

Confirmation: If Tomcat is unable to start because the value of `JAVA_HOME` is set incorrectly, you will see an error message that is similar to the following message in a command prompt or shell:

```
'-Xms512m' is not recognized as an internal or external command,
operable program or batch file
```

Fix: Make sure that you have the correct version of the Java 2 SDK installed. Then, edit the `startServletContainer` script, if necessary, so that `JAVA_HOME` contains the fully qualified path to the installation directory for the Java 2 SDK.

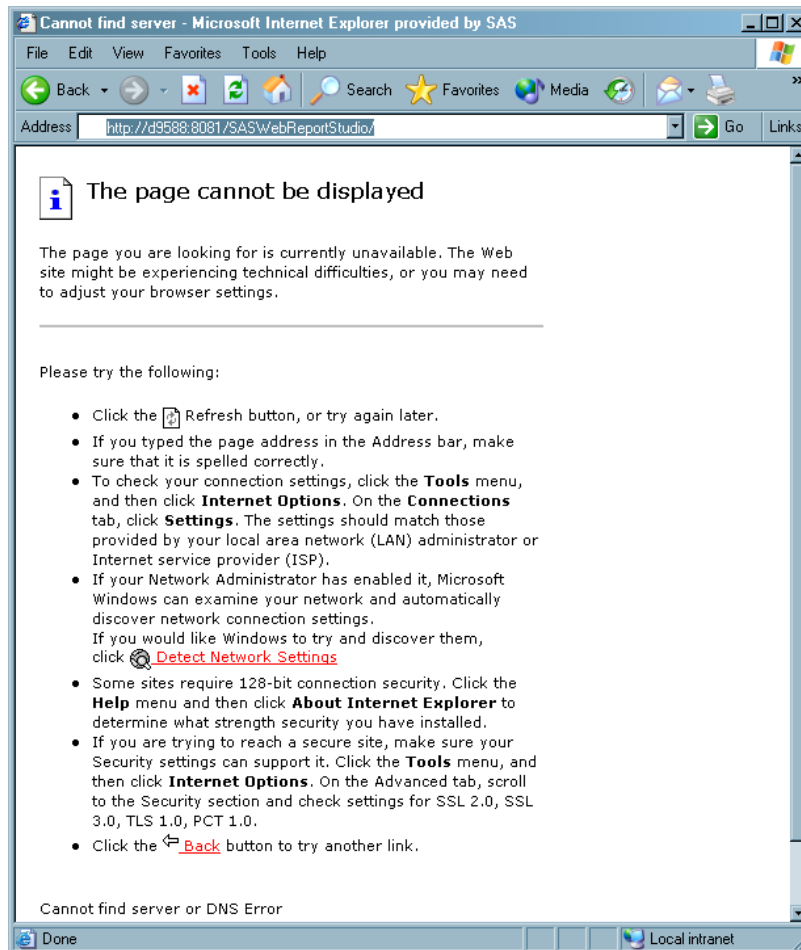
Web Applications

The troubleshooting tips in this section apply to all SAS Web applications including SAS Web Report Studio and SAS Information Delivery. If you are unable to start any of these applications, or if you see poor performance from all of these applications, see the following sections.

Initial Page Cannot Be Loaded

Explanation: If you try to start a Web application, such as SAS Web Report Studio or SAS Information Delivery Portal, and receive a "Cannot find server" error, one of two things is wrong. Either your Web container—or J2EE server—is not running, or there is problem with the URL that you are using to start the application.

Confirmation: When you submit a request to start the Web application, your browser displays a server-not-found error. If you are using Internet Explorer, you will see a page that is similar to this page.



Fix: Make sure that your Web container or J2EE server is running. If it is not running, start it. This will probably solve the problem. If the server is already running, the problem could be with the URL that you are using to start the Web application. This URL has the form `http://host-name:port-number/application-name`. Make sure that the host name is the fully qualified name of the host on which your servlet container or J2EE server is running. Also, make sure that the port number identifies the port on which the server is listening. Normally, this will be port 8080. Finally, make sure that the application name in the URL matches the actual name of the Web application—including case. Correcting the URL should fix the problem.

Pages Take a Long Time to Load

Explanation: Your Web application works, but each time that you request a new page there is a long delay.

Confirmation: Not applicable.

Fix: Web applications such as SAS Web Report Studio and SAS Information Delivery Portal use JavaServer Pages. When a JSP is requested for the first time, your Web container or J2EE server must translate the JSP to a servlet and then compile the servlet to create a Java class file. It is this class file that is loaded and run by the Java Virtual Machine. So each time that you request a JSP for the first time, you can expect a significant delay. When all of your JSPs have been converted to compiled servlets, the problem will be resolved.

If performance remains poor, you might need to tune your Web container or J2EE server. For example, if you are using the BEA WebLogic Server, the server might be

checking (on a per application basis) for updated JSPs and updated compiled servlets. You can suppress these checks by setting an application's `JSPPageCheck` and `ServletReloadCheck` properties to -1. See your server vendor's documentation for similar tips.

SAS Web Report Studio

This section explains how to troubleshoot the following situation:

- You are using the Apache HTTP server as your content repository.
- You can get to the login page of SAS Web Report Studio, but cannot log on. When you attempt to log on, you get the error "The user name or password is incorrect. Please re-enter," or the error "Access to Repository Failed."

This problem can occur for a number of reasons. See the following sections.

Apache HTTP Server Is Not Running

Explanation: No one has started the Apache HTTP server.

Confirmation: Not applicable.

Fix: Start the Apache HTTP Server. On Windows systems, you can start the server using the Apache Service Monitor or by selecting **Start ► Programs ► Apache HTTP Server 2.0.45 ► Control Apache Server ► StartOn** UNIX systems, you can start the server by running the script `path-to-config-dir/Lev1/web/startServletContainer.sh`.

Apache HTTP Server Configuration File Is Set Up Incorrectly

Explanation: If you are using the Apache HTTP Server as a WebDAV server, the SAS Configuration Wizard will have instructed you (in the `instructions.html` file) to make some changes to the `httpd.conf` configuration file. These changes enable the server's WebDAV capabilities. If you made a mistake while you were editing this file, users might not be able to log on to SAS Web Report Studio.

Confirmation: Not applicable.

Fix: A copy of the original configuration file is saved in the `httpd.default.conf` file. If you think that you might have edited your configuration file incorrectly, you should follow these steps:

- 1 Delete the `httpd.conf` file.
- 2 Rename the `httpd.default.conf` file to `httpd.conf`.
- 3 Edit `httpd.conf` by following the instructions in the `path-to-config-dir\instructions.html` file. See "Define Your HTTP Server."

You Did Not Create the Directory That Serves as the Content Base Path

Explanation: You must create a directory on the file system that will serve as the root directory for your content repository. Your `instructions.html` file will tell you exactly what directory to create and where to create it.

Confirmation: Not applicable.

Fix: Create the root directory for your content repository. Make sure that you have spelled the name of the directory correctly and that you use the appropriate case.

Your WebDAV Server Is Configured Incorrectly

Explanation: As part of the configuration of your system, you should have created a metadata object that represents your WebDAV server. If this object is not set up correctly, users will not be able to log on to SAS Web Report Studio.

Confirmation: Not applicable.

Fix: Set up this metadata object according to the instructions in `instructions.html` (see “Define an HTTP Server to the metadata”). In particular, make sure that the server’s **Base Path** is set to the directory discussed in “You Did Not Create the Directory That Serves as the Content Base Path” on page 126 and that you specified that the server “Supports WebDAV.”

You Did Not Set the Properties of the BIP Tree Correctly

Explanation: Your `instructions.html` file also explains that you must set some properties for the Business Report Manager’s BIP Tree. If these properties are not set correctly, users will not be able to log on to SAS Web Report Studio.

Confirmation: Not applicable.

Fix: Follow the directions in the section “Attach the HTTP Server as the content manager for the SAS Business Intelligence Platform (BIP) metadata tree” in `instructions.html`. In particular, you must specify the location of your WebDAV server and a **Content Base Path**.

SAS Information Delivery Portal

If users can get to the SAS Information Delivery Portal login page, but cannot log on to the application, see the following sections. The problem probably has to do with the metadata objects that represent these users.

User Is Not Registered in the Metadata Repository

Explanation: Before a user can successfully log on to SAS Information Delivery Portal, that user must be registered in the metadata repository. (That is, a User metadata object must have been created for the user.) If the user is not registered in the metadata, the user will see the message “Could not authenticate user” on the login page.

Confirmation: To determine why the user cannot log on, look in the portal log file, `path-to-config-dir\Lev1\web\Deployments\Portal\portal.log`. If the user is not registered in the metadata, you will see the following message:

```
[WARN] com.sas.services.information.OMIRepository -- The Authenticated user
D1234\TestUser is not represented in the repository by a metadata object.
```

Fix: Use the User Manager plug-in to SAS Management Console to create a User object for the portal user. Make sure that you add a login to the User object that contains at least a user name.

User Does Not Have the Correct Permissions

Explanation: If the SAS Guest or the SAS Web Administrator cannot log on, they may not have the correct permissions to access the repository. Both need ReadMetadata and WriteMetadata access to the repository.

Fix: Do either of the following:

- On the repository ACT, grant ReadMetadata and WriteMetadata permissions to the SASUSERS group.
- Add the SAS Guest and the SAS Web Administrator to the user group that you are using to manage your portal users.

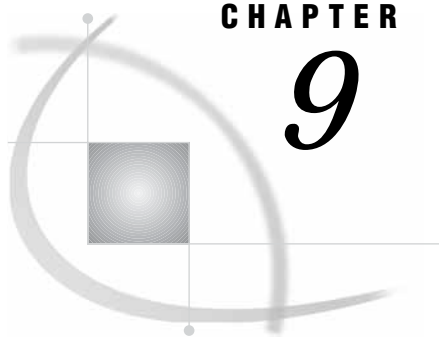
User’s Metadata Identity Does Not Contain a Domain (Windows Only)

Explanation: On Windows systems, you can experience a slight variation of the problem discussed in “User Is Not Registered in the Metadata Repository” on page 127.

If the user that is trying to log on is registered in the metadata, but the user name is not domain qualified, the user will not be able to log on.

Confirmation: The portal log will contain the message that was shown in the preceding section.

Fix: Use SAS Management Console to modify the user name in the appropriate Login in the User object. Change the name so that it has the form *host\user-ID* or *domain\user-ID*.



CHAPTER

9

Post-Configuration Tasks

<i>Overview of Post-Configuration Tasks</i>	129
<i>Understanding the State of Your System</i>	130
<i>Configuration Directory: Server-Tier Machines</i>	130
<i>Security-Related Files</i>	131
<i>Server Start-Up Scripts and Logs</i>	132
<i>Metadata Repository</i>	132
<i>Configuration Directory: Middle-Tier Machines</i>	132
<i>Servlet Container Start-Up Script</i>	133
<i>The Deployments Directory</i>	133
<i>The webapps Directory</i>	133
<i>SAS Application Servers</i>	133
<i>Tasks That You Might (or Will) Need to Perform</i>	134
<i>Starting and Stopping Your SAS Servers</i>	134
<i>Windows</i>	134
<i>UNIX</i>	135
<i>z/OS</i>	135
<i>Modifying the Default Security Configuration</i>	135
<i>Pre-assigning Libraries</i>	136
<i>For SAS 9.1 and higher</i>	138
<i>Registering Data Sources</i>	139
<i>Working with User-Defined Formats</i>	139
<i>Configuring SAS/ACCESS Products</i>	140
<i>Establishing Basic Protections</i>	140
<i>Protecting the Metadata Repository and Configuration Directories</i>	140
<i>Setting an Encryption Method</i>	141
<i>Ongoing Administration and Maintenance</i>	143

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.

- 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. “Understanding the State of Your System” on page 130 explains those actions so that you understand the current state of your system.
- “Tasks That You Might (or Will) Need to Perform” on page 134 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 registering data sources and configuring SAS/ACCESS products. Before you perform these tasks, workers such as ETL specialists will not have access to the data sources that they need.

- “Establishing Basic Protections” on page 140 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 143 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.

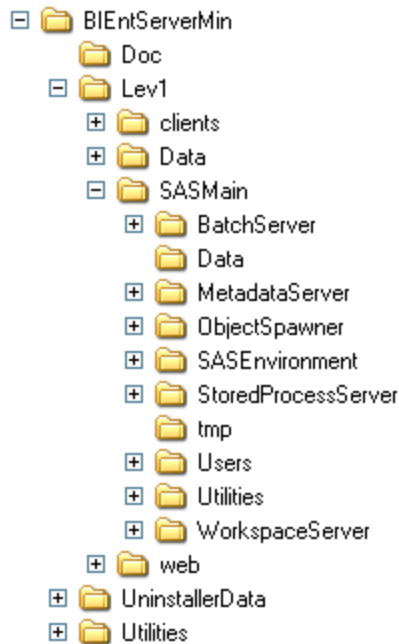
Understanding the State of Your System

As mentioned previously, when you ran the SAS Configuration Wizard, that wizard did some behind-the-scenes work. Most important, on server-tier machines and middle-tier machines, the wizard created a configuration directory. This directory and its subdirectories contain major parts 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. See the following display.

Display 9.1 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 431. 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. Δ

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 191.

The configuration wizard creates one unrestricted user: the SAS Administrator (**sasadm**). This user is listed in **adminUsers.txt**, and the ID is preceded by an asterisk. The configuration wizard also creates one trusted user: the SAS Trusted User (**sastrust**). You create this type of user by adding a user ID to the file **trustedUsers.txt**. (For more detailed information about how to create these special users, see the section “Configuring Special Users” in the *SAS Metadata Server: Setup Guide* at support.sas.com/rnd/eai/openmeta/v9/setup/.)

You do not normally need to edit the **trustedPeers.xml** file. By default, the metadata server trusts connecting workspace and stored process servers as peers. That is, these clients do not have to supply credentials when they connect to the metadata server. For information about the role of the **trustedPeers.xml** file and how to edit it,

see the *SAS Integration Technologies: Server Administrator's Guide* at support.sas.com/rnd/itech/doc9/admin_oma/security/auth/security_imptrust.html.

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 and an **ObjectSpawner** folder. On UNIX and Windows systems, each such directory for a server that you can start directly contains a script called *server-type.extension* that takes a parameter **start**. On UNIX systems, you call these scripts directly to start servers and spawners. On Windows systems, you can call the scripts directly, or you can use the Start menu, for example, **Start** \blacktriangleright **Programs** \blacktriangleright **SAS** \blacktriangleright **configuration-directory** \blacktriangleright **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
```

Δ

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.

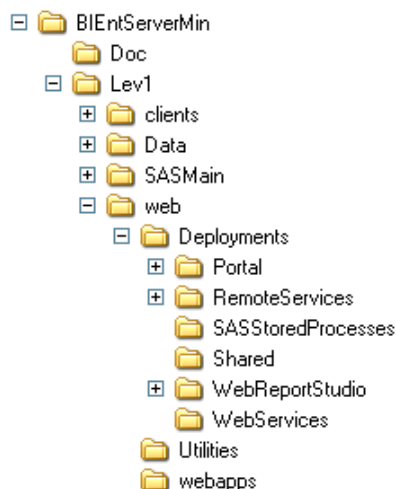
Metadata Repository

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 287.

Configuration Directory: Middle-Tier Machines

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

Display 9.2 Configuration Directory (Middle Tier)



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

Servlet Container Start-Up Script

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

If you are using a J2EE server such as the BEA WebLogic Platform or the IBM WebSphere Application Server, the configuration wizard will create a **startServletContainer** script, just as it does for Apache Tomcat. However, administrators generally start these servers from an administrative console or by using a script supplied with the server.

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

The Deployments Directory

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

The webapps Directory

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

If you are using the Tomcat servlet container to execute your Web applications, the SAS Configuration Wizard has already copied these WAR files to Tomcat's **webapps** directory. If you are using the BEA WebLogic Platform or the IBM WebSphere Application Server for this purpose, you deploy your Web applications using your server's administration console.

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. You can customize a workspace server (or servers) for better performance or a specific use using

- pooling
- load balancing.

When you configure a pooling workspace server, you enable clients to use a connection from a pool of previously created connections to workspace server processes. Using a pooled server is a good idea in cases where clients need to use a connection for

a brief period of time, because it enables clients to avoid the overhead of opening connections and starting server processes. For information on how to convert a standard workspace server to a pooled server for use with SAS Web Report Studio or SAS Information Delivery Portal, see “Workspace Server Pooling for SAS Web Report Studio and SAS Information Delivery Portal” on page 366.

If you have created workspace servers on more than one host, you can balance a load across these servers by defining a load balanced logical workspace server. When you load balance a set of workspace servers, you create a cluster. The object spawners responsible for starting the workspace servers in a cluster take care of the load balancing and direct new traffic to the most available server. This type of configuration is most useful when you have a large number of workers (such as 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 379.

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 385.

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

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 and Stopping Your SAS Servers

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

Windows

If your metadata server, object spawner, or OLAP server is running on a Windows machine and you have chosen to run servers as services (highly recommended), your

servers will start automatically when you restart your machine. You can then stop, start, or restart your servers in one of two ways.

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

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

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

UNIX

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

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

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

z/OS

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

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

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

or

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

Each server is associated with a different started task.

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

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 151. 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 191. △

- change the way that users of Web applications 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's or application server's authentication mechanisms.

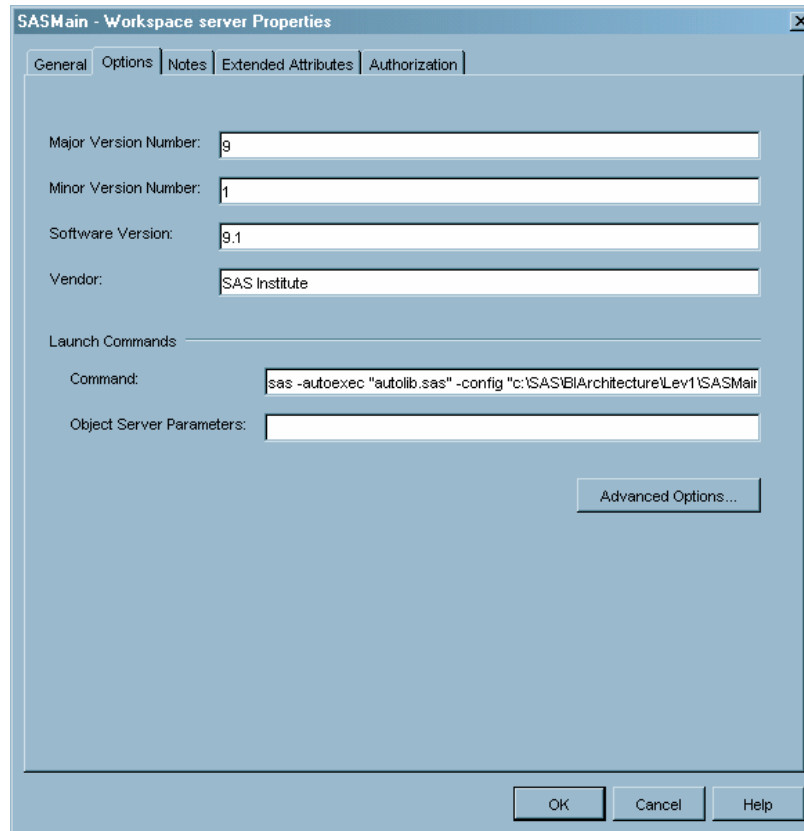
If you want users of the SAS Information Delivery Portal to be authenticated by your servlet container or J2EE server, see “Setting Up Web Server Authentication” in the *SAS Web Infrastructure Kit: Administrator's Guide* at support.sas.com/rnd/itech/doc9/portal_admin/.
- restrict access to the SAS Workspace 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.

Pre-assigning Libraries

As part of the configuration process for your servers, you might choose to assign libraries automatically. This ensures that the libraries will be available and assigned in the same way for all users and applications, which can be very useful for servers that support numerous uses.

The most common way to pre-assign a library is by manually configuring the server using one of the following techniques. For more information on these techniques, see “Setting Up Libraries” in the *SAS Integration Technologies: Server Administrator's Guide* at support.sas.com/rnd/itech/doc9/admin_oma/getstart/gs_setres.html.

- Create a SAS autoexec file that includes one or more LIBNAME statements. A SAS autoexec file contains SAS statements that are executed immediately after SAS initializes and before any user input is accepted. You can add a reference to the autoexec file to the server's command line in SAS Management Console, as shown in the following display.

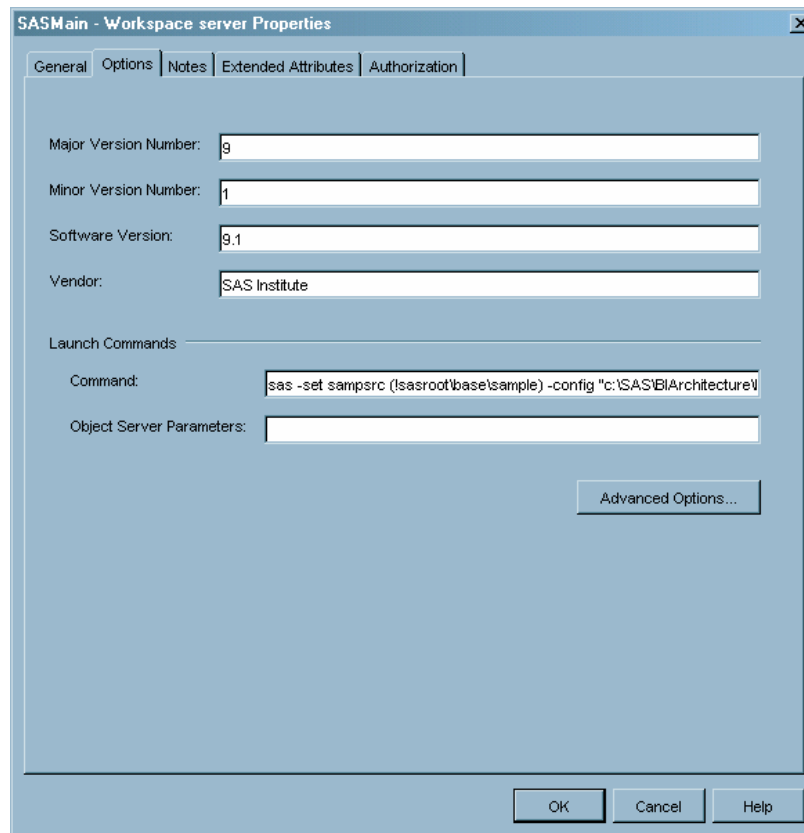


Note: For more information about how to create autoexec files, see the companion documentation for your operating system. △

- Use the SET system option to define an environment variable that is valid within the SAS session. For example, the following code defines an environment variable for the sample base library:

```
-set sampsrc (!sasroot\base\sample)
```

When you refer to SAMPSRC as a library name during your SAS session, SAS automatically assigns the library with the path that is listed. Add the `-set` option to the server's command line in SAS Management Console, as shown in the following display.



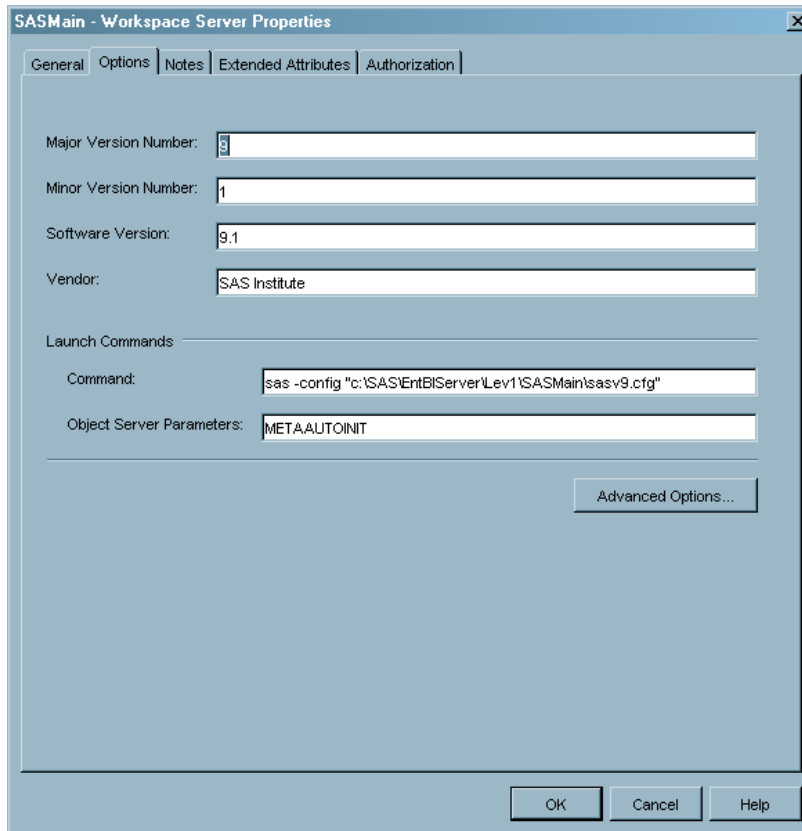
Note: For more information about the SET system option, see the companion documentation for your operating system. This option is supported only by the Base SAS engine.

After you properly configure your server to pre-assign a library, it is recommended that you use SAS Management Console to define a library metadata object that is designated as pre-assigned that references the library. This will enable applications to use the library from the metadata.

For SAS 9.1 and higher

For SAS 9.1 and higher, an additional technique for pre-assigning libraries is available. This technique allows you to define the library with SAS Management Console and then ask the server to read the metadata and assign the library. This technique consists of the following steps:

- 1 Fully define your library in SAS Management Console. You need to be sure that you associate the library with the appropriate server, using the **Assign** tab. To register a library in the SAS Metadata Server, use the New Library wizard, which is available with the Data Library Manager plug-in to SAS Management Console. To identify the new library as pre-assigned, select the **Library is preassigned** check box in the Advanced Options dialog box on the library properties wizard window.
- 2 For the SAS Workspace Server and SAS Stored Process Server, add the METAAUTOINIT object server parameter to the launch command.



(The METAUTOINIT object server parameter is automatically turned on for SAS OLAP Servers.) For additional information about pre-assigning libraries for these servers, see the *SAS Integration Technologies: Server Administrator's Guide* at support.sas.com/rnd/itech/doc9/admin_oma.

- For other SAS servers (such as SAS/CONNECT, SAS Batch, and SAS/SHARE), set the METAUTORESOURCES option in their configuration files. Note that using METAUTORESOURCES to assign a library will work with only those tables that are defined in the metadata, not for tables that are in the physical library.

Registering Data Sources

Before your coworkers can use products such 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 239.

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.sas7bcats` and to place the catalog in the directory `path-to-config-dir\Lev1\SASMain\SASEnvironment\SASFormats`.

Note: This approach does not work on z/OS systems. If you are working on that platform, you should use the following alternative approach. △

- 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 9.1.3 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.

Establishing Basic Protections

This section outlines some basic security tasks that you should perform after running the SAS Configuration Wizard. These tasks include

- protecting the metadata repository and the configuration directories
- encrypting data sent over the network

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 213.

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 214 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 440.

On Windows systems, you must set these permissions manually. 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. Δ

- Grant Full Control to **SYSTEM** and Read to **Everyone** for the following folders:
 - BatchServer**
 - SASEnvironment**
 - Users**
 - Utilities**
 - WorkspaceServer**

In addition, for the subfolder **SASEnvironment\SASCode\Jobs**, grant Modify rights to **Everyone** or to the **SAS Server Users** group. This will enable users SAS ETL Studio users to write SAS programs to this directory.

- For the **StoredProcessServer** folder and the folder **StoredProcessServer\logs**, grant Full Control to **SYSTEM** and the SAS General System User (**sassrv**).

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. Δ

Setting an Encryption Method

By default, only user credentials that are 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 that are 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 Web-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** - Login 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 middle-tier machine.

Install the client-side version of the product. As part of that installation, two Java Archive (JAR) files, `sas.core.jar` and `sas.rutil.jar` will be written to your machine. Copy the JAR files that were 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 middle-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. \triangle

- 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 on the particular machine.

After 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\SASManagementConsole\9.1` (Windows) or `install-location/SASManagementConsole/9.1` (UNIX).
- For SAS ETL Studio, copy the file to `drive:\Program Files\SAS\SASETLStudio\9.1`.
- For SAS OLAP Cube Studio, copy the file to `drive:\Program Files\SAS\SASOLAPCubeStudio\9.1`.
- For SAS Information Map Studio, copy the file to `drive:\Program Files\SAS\SASInformationMapStudio\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\leve\SAS-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 132.

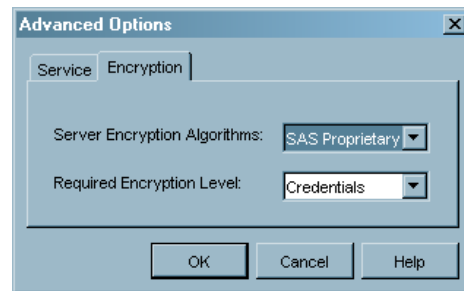
- 5 Edit the metadata objects that represent your SAS servers, such as your workspace, stored process, and OLAP servers.

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 tasks for each physical server:

- a Select the server. This causes the server's connection(s) to appear on 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.

Ongoing Administration and Maintenance

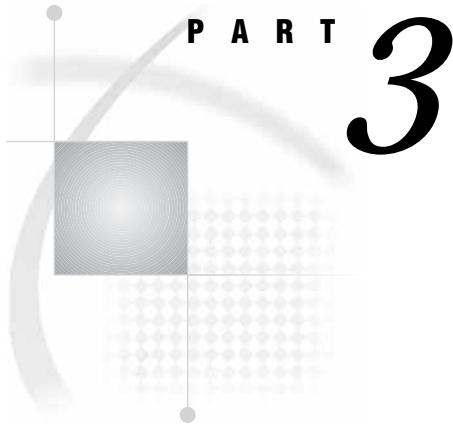
After you have completed the initial configuration of your system, you might need to perform the following tasks:

- 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/.

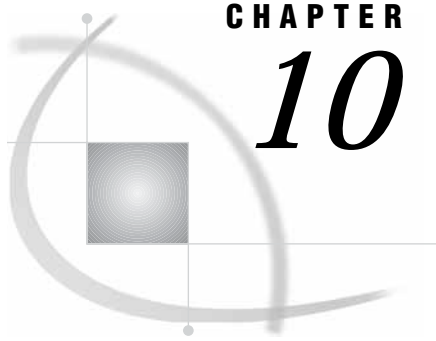
- You will need to optimize your data storage to facilitate the querying and analysis of that data. This subject is discussed in Chapter 15, "Optimizing Data Storage," on page 257.
- You will need to administer the applications that are being used at your site, such as SAS ETL Studio, SAS Information Map Studio, SAS Web Report Studio, and SAS Enterprise Miner. See the following chapters:
 - Chapter 16, "Administering SAS ETL Studio," on page 281
 - Chapter 17, "Managing the Reporting Environment," on page 305
 - Chapter 18, "Preparing SAS Enterprise Miner for Use," on page 341

- 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 19, “Configuring Your Servers for Better Performance,” on page 361.
- You might want to perform the metadata-related tasks described in Chapter 20, “Promoting and Replicating Metadata,” on page 393.



Security Administration

- Chapter 10* **Understanding Authentication** 147
- Chapter 11* **Understanding Authorization** 175
- Chapter 12* **Developing Your Security Plan** 193
- Chapter 13* **Implementing Security** 213



CHAPTER

10

Understanding Authentication

<i>Introduction to Understanding Authentication</i>	147
<i>Authentication Concepts and Terminology</i>	148
<i>Authentication Providers</i>	148
<i>Metadata Identities</i>	148
<i>Logins</i>	150
<i>Authentication Domains</i>	151
<i>The Authentication Process</i>	152
<i>Initial Authentication</i>	153
<i>Initial Authentication on a Metadata Server</i>	154
<i>Initial Authentication on a Middle Tier Server</i>	155
<i>Initial Authentication on a SAS OLAP Server</i>	156
<i>Trusted Peer Session Connections</i>	156
<i>Additional Authentication</i>	156
<i>Using Credentials That Are Stored in the Metadata</i>	157
<i>Using Cached Credentials</i>	158
<i>Sharing User Context</i>	160
<i>Summary: Credential Management Features by Client</i>	160
<i>Examples: Using Authentication Domains</i>	161
<i>Single Platform Environments</i>	161
<i>Mixed Platform Environments</i>	162
<i>Diverse Environments</i>	164
<i>Examples: Accessing SAS Servers</i>	165
<i>Accessing a SAS OLAP Server</i>	165
<i>Accessing a SAS Workspace Server</i>	166
<i>Accessing a Pooled SAS Workspace Server</i>	167
<i>Accessing a SAS Stored Process Server</i>	169
<i>Examples: Accessing Third-Party Servers</i>	170
<i>Accessing a DB2 Database</i>	170
<i>Accessing an SAP System</i>	171
<i>Accessing a Xythos WebFile Server</i>	171

Introduction to Understanding Authentication

This chapter explains in detail how the authentication process works in the SAS Intelligence Platform. For a brief overview of this subject, see “Authentication in the SAS Intelligence Platform” on page 35. Systems architects and administrators who perform security-related tasks will benefit from understanding the information 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 Platform 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 (user IDs and passwords) you must store in the metadata.

Authentication Concepts and Terminology

This section introduces four terms that are essential for understanding how authentication works in the SAS Intelligence Platform. These terms are

<i>authentication provider</i>	a technology that servers or applications can use to verify that users are who they say they are.
<i>metadata identity</i>	a metadata object that represents an individual user or a group of users on a SAS Metadata Server.
<i>login</i>	a metadata object that is owned by a metadata identity. Each login stores the user ID and password for a user account that has been established with an authentication provider.
<i>authentication domain</i>	a metadata object that links logins to the servers for which the logins are valid.

The following topics explain each of these concepts in detail.

Authentication Providers

An *authentication provider* is a technology that servers or applications can use to verify that users are who they say they are. By default, the authentication provider for a SAS server is the host operating system of the machine on which the server is running. When you request access to a SAS server that is using the default authentication process, the server asks its host environment to verify that your user ID and password correspond to a valid user account in the operating system. This method of verifying identities is called host authentication.

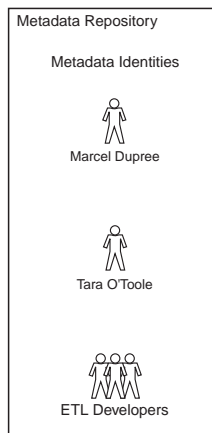
SAS Metadata Servers and SAS OLAP Servers can also use Lightweight Directory Access Protocol (LDAP) server or Microsoft Active Directory server as alternative authentication providers. For more information, see "Implementing Alternative Authentication Providers" in the *SAS Integration Technologies: Server Administrator's Guide* at support.sas.com/rnd/itech/doc9/admin_oma/security/auth/security_impauthalt.html.

SAS Web applications run under third-party servers that can use a variety of authentication providers. For more information, see the documentation for the third-party server under which your SAS Web applications run.

Metadata Identities

A *metadata identity* represents an individual user or a group of users in the metadata environment. Each metadata identity must be unique within a metadata server. The following figure depicts several metadata identities within a SAS Metadata Repository.

Figure 10.1 Metadata Identities



The metadata server uses your metadata identity to respond to requests for credentials and to make authorization decisions. Your access to resources is controlled by the logins and access controls that have been created for your metadata identity (or for a user group to which your metadata identity belongs).

The metadata server discovers your metadata identity by performing these steps:

- 1 The metadata server searches the metadata repository for a login that contains a user ID that matches the user ID with which you were authenticated.

In this process, the metadata server attempts to match your fully qualified user ID. For example, if you log on to a server that is using Windows host authentication, and your Windows user ID is `marcel` in a Windows domain named `winNT`, then the metadata server searches the repository for a login that includes a user ID of `winNT\marcel`. For this reason, you must carefully specify the user ID in each login that you create.

- When you create a login for a network Windows user account, specify the user ID in the form *Windows-domain-name\userID* or in the form *userID@Windows-domain-name*.
- When you create a login for a local Windows user account, specify the user ID in the form *machine-name\userID* or in the form *userID@machine-name*.
- When you create a login for an LDAP user account, specify the user ID in the form *userID@authentication-provider*.
- When you create a login for a Microsoft Active Directory user account, specify the user ID in the form *Windows-domain-name\userID* or in the form *userID@Windows-domain-name*.
- When you create a login for a UNIX or z/OS operating system user account, specify the user ID in the form *userID*.

Note: For more information, see "Defining Users, Groups, and Logins on the SAS Metadata Server" in the *SAS Integration Technologies: Server Administrator's Guide* at support.sas.com/rnd/itech/doc9/admin_oma/security/grpuserlog_defs.html. △

- 2 The metadata server determines which metadata identity owns the login (or logins) that contain the matching user ID. In the previous example, if the login that contains a user ID of `winNT\marcel` is owned by a metadata identity named Marcel Dupree, then the metadata server knows that Marcel Dupree is your metadata identity.

If the metadata server does not find a matching user ID, then you do not have an individual metadata identity. 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.

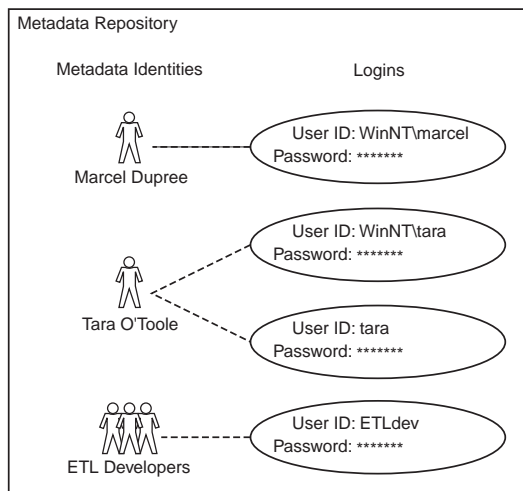
Note: If you are an *unrestricted user* or an *administrative user* of the metadata server, you can perform certain tasks even if you do not have an individual metadata identity. For more information, see “Special Users of the Metadata Server” on page 191. Δ

Logins

A *login* contains the user ID and (often) the password for a user account that has been established in the operating system or with an alternative authentication provider. Each login corresponds to a particular user account with a particular authentication provider. For example, if you have a UNIX account with a user ID of `tara` and a password of `tara1234`, then you can store that account information in the metadata as a login.

Each login is owned by only one metadata identity. Each metadata identity can own multiple logins. The following figure depicts the relationships between logins and metadata identities in a metadata repository.

Figure 10.2 Metadata Identities and Logins



Logins can be used in any or all of these ways:

- The metadata server uses logins to determine your metadata identity. When a login is used to determine your metadata identity, the login is functioning as an *inbound login* (the login is inbound to the metadata server). The metadata server does not examine passwords or consider authentication domains in this process. For more information, see “Initial Authentication on a Metadata Server” on page 154.
- Applications use logins to acquire your credentials as part of a single sign-on approach to authentication. An application can retrieve a login from the metadata server and send those credentials to another system that needs to verify your identity. When a login is used to provide access to a server other than the metadata server, the login is functioning as an *outbound login* (the login is

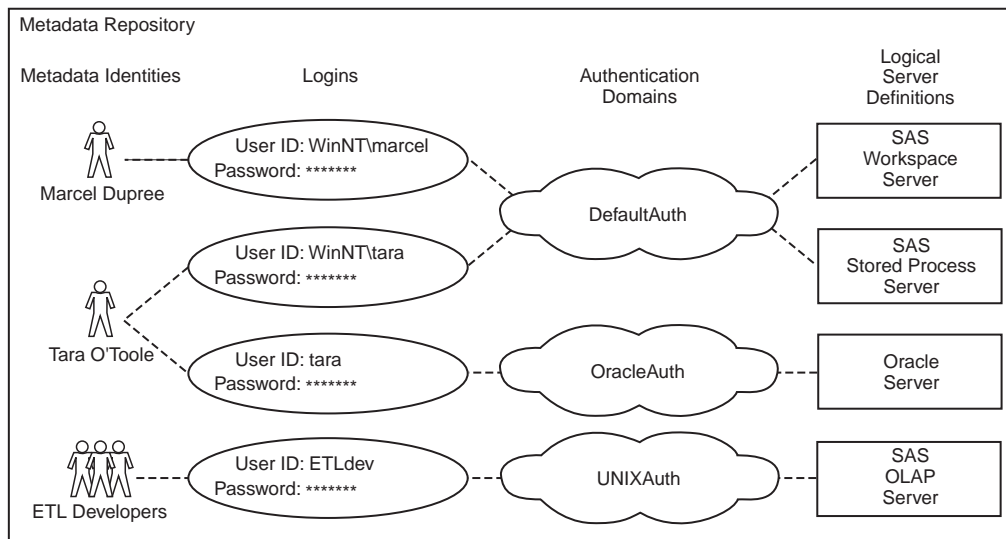
outbound from the metadata server to another system). An outbound login must include a user ID and password that are appropriate for the server or host to which the login provides access. An outbound login must be associated with an authentication domain. For more information, see “Using Credentials That Are Stored in the Metadata” on page 157.

- The Xythos WebFile Server uses logins to discover and set up users for access control in the WebDAV authorization layer. Xythos builds its list of users by retrieving from the metadata server all of the logins that are associated with the authentication domain of the Xythos WebFile Server. In the default configuration, that server is associated with the DefaultAuth authentication domain, so you must have a login that is associated with the DefaultAuth authentication domain in order to be a valid user of the Xythos WebFile Server. In alternate configurations, you must have a login for some other authentication domain in order to be a valid user of the Xythos WebFile Server. For more information, see “Accessing a Xythos WebFile Server” on page 171.
- The object spawner uses logins to obtain credentials for launching servers that run under designated accounts. When you configure a stored process server or a pooled workspace server, you specify an account under which the server will run. For example, during installation the stored process server is configured to run under the sassrv account. In order to launch that stored process server, the object spawner needs the credentials for the sassrv account. The object spawner obtains those credentials from a login that is owned by the SAS General Servers group. For more information, see “Troubleshooting SAS Servers” on page 114.

Authentication Domains

An *authentication domain* can be associated with one or more servers and with the logins that provide access to those servers. Authentication domains enable you to define logical groupings of servers and logins within a metadata repository. All of the computing resources within an authentication domain use the same authentication provider. 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.

In the metadata, every logical server (other than the metadata server) must be associated with an authentication domain. Every login that is used for outbound purposes must be associated with an authentication domain. The following figure depicts the relationships between servers, authentication domains, and logins.

Figure 10.3 Metadata Identities, Logins, and Authentication Domains

Authentication domains are used when an application searches the metadata for a login that provides access to a particular server. The application uses authentication domains to determine which logins contain credentials that are appropriate for accessing the target server. For example, if Tara makes a request that requires access to the Oracle server, then the Oracle server will have to verify Tara's identity. The application that Tara is using must provide Tara's Oracle user ID and password to the Oracle server. The application will complete these steps:

- 1 Determine that the Oracle server definition is associated with the OracleAuth authentication domain.
- 2 Ask the metadata server for a login that is both associated with the OracleAuth authentication domain and owned by Tara's metadata identity (or by a group to which Tara's identity belongs).

In the preceding figure, Tara's second login 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 Authentication Process

This section explains when and how identities are verified in the SAS Intelligence Platform. The discussion assumes that you are familiar with the terms that are explained in the previous section.

The authentication process occurs in two phases:

- 1 In the *initial authentication* phase, you log on with a SAS Intelligence client or open a metadata profile. The user ID and password that you submit are sent to an authentication provider to verify your identity. After your user ID and password are verified, the metadata server determines your metadata identity.
- 2 In the *additional authentication* phase, you make a request that requires access to an additional system such as a workspace server, stored process server, or database server. The application that you are using provides your credentials to

the additional server. This enables the additional server to verify your identity against its authentication provider.

These phases are described in detail in the following sections.

Initial Authentication

Initial authentication is the verification of your identity based on information that you provide when you log on 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. 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).

Note: Storing your user ID and password in the metadata does not eliminate the need for this account. △

The initial authentication process varies depending on the software component that you are using. The following table describes how each software component verifies identities.

Table 10.1 Initial Authentication

Software Component	Identity Verification
A desktop application (such as SAS ETL Studio, SAS Information Map Studio, or SAS Management Console) or a Web application (such as SAS Web Report Studio or 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. For details, see "Initial Authentication on a Metadata Server" on page 154.
A Web application (such as SAS Web Report Studio or 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 <i>trusted user*</i> 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. For details, see "Initial Authentication on a Middle Tier Server" on page 155.
A component that connects directly to a SAS OLAP Server (such as the SAS OLAP Data Provider).	The SAS OLAP Server's authentication provider verifies the user ID and password that you submit. As a <i>trusted user*</i> 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. For details, see "Initial Authentication on a SAS OLAP Server" on page 156.

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

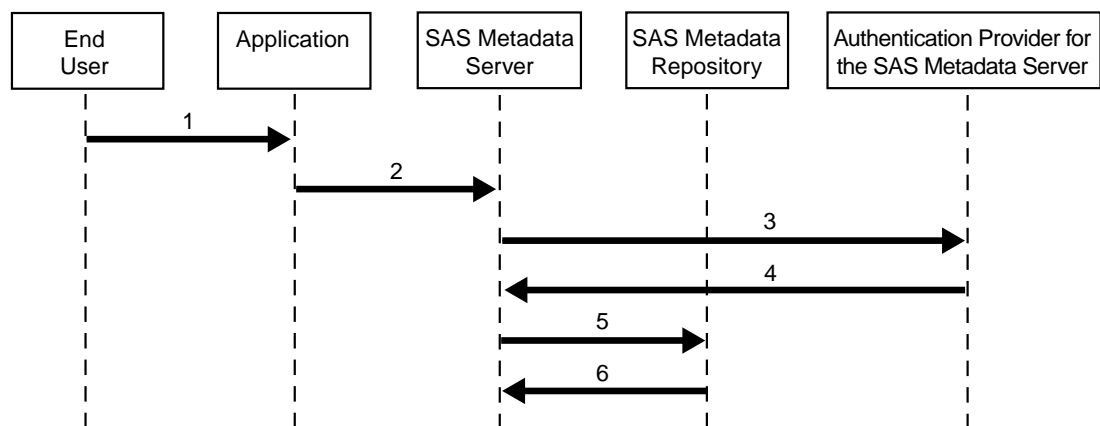
Initial Authentication on a Metadata Server

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

- A desktop application such as SAS Management Console, SAS ETL Studio, SAS OLAP Cube Studio, or SAS Information Map Studio.
- A Web application that is configured to authenticate users on the metadata server. The SAS Intelligence Web applications include SAS Web Report Studio, SAS Web Report Viewer, and SAS Information Delivery Portal.

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

Figure 10.4 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 (by logging on or by opening a metadata profile).
- 2 The application sends the user ID and password to the metadata server.
- 3 The metadata server passes the user ID and password to its authentication provider for verification. For example, if the authentication provider is the host operating system, then the metadata server passes the user ID and password to the operating system of the machine on which the metadata server is running.
- 4 The authentication provider verifies that the user ID and password combination corresponds to an existing user account. For example, if the authentication provider is the host operating system, then the user ID and password combination must correspond to a local or network user account that has been established in the operating system. After verification, the authentication provider tells the metadata server that the user ID and password are valid and sends the user ID back to the metadata server.
- 5 The metadata server looks for the user ID in the logins that are stored in the metadata repository.

Note: The metadata attempts to match the user ID in its fully qualified form, as described in “Metadata Identities” on page 148. Δ

- 6 The metadata server determines which metadata identity owns a login that contains the matching user ID.

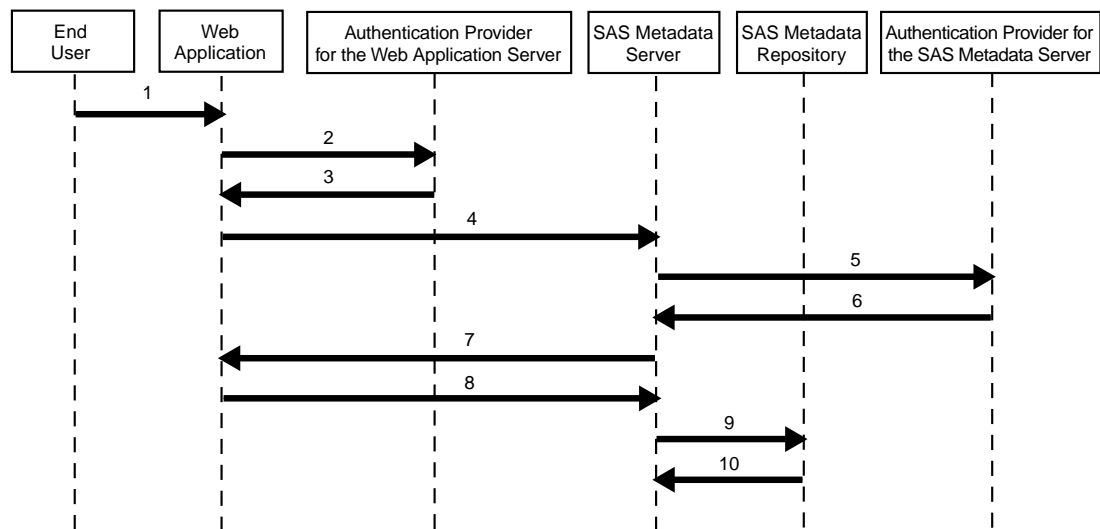
Initial Authentication on a Middle Tier Server

When you log on with a Web application that is configured to authenticate in the middle tier, a Web server handles the initial authentication. First, your identity is verified by the Web server's authentication provider. Then, the Web application uses a trusted user connection to access the metadata server.

Note: By default, SAS Web applications authenticate users on the metadata server. To learn how to configure a Web application to use middle tier authentication, see "Setting Up Web Server Authentication" in the *SAS Web Infrastructure Kit: Administrator's Guide* at support.sas.com/rnd/itech/doc9/portal_admin/installmig/ag_settrust.html. △

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

Figure 10.5 Initial Authentication on a Middle 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 ID and password to its authentication provider for verification.
- 3 The Web application's authentication provider verifies that the user ID and password corresponds to an existing account.
- 4 The Web application requests a trusted user connection to the metadata server. In the request, the Web application sends the user ID and password for the trusted user account (sastrust) to the metadata server.
- 5 The metadata server passes the trusted user's user ID and password to its authentication provider for verification.
- 6 The metadata server's authentication provider verifies that the trusted user's user ID and password corresponds to an existing account.
- 7 The metadata server tells the Web application that the trusted user connection has been accepted.
- 8 The Web application passes the requesting user's ID to the metadata server. The metadata server trusts that the Web application has already verified the user's ID.

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

Note: The metadata attempts to match the user ID in its fully qualified form, as described in “Metadata Identities” on page 148. Δ

- 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. Δ

Initial Authentication on a SAS OLAP Server

When you log on with a component that connects directly to a SAS OLAP Server, the SAS OLAP Server handles the initial authentication. First, your identity is verified by the SAS OLAP Server’s authentication provider. Then the SAS OLAP Server uses a trusted user connection to impersonate you while accessing the metadata server on your behalf.

For example, when you access SAS OLAP data from Microsoft Excel, the SAS OLAP Data Provider passes your credentials to the SAS OLAP Server for initial authentication. The process is similar to the process depicted in Figure 10.5 on page 155, if you substitute the SAS OLAP Server (and its authentication provider) in the place of the Web application (and its authentication provider) in that diagram.

Trusted Peer Session Connections

Trusted peer session connections enable a SAS process (a SAS session, SAS Workspace Server, or SAS Stored Process Server) to connect to a SAS Metadata Server without explicitly providing credentials. You can use trusted peer session connections to enable applications to generate code or run batch jobs without explicitly providing credentials to the metadata server.

When the trusted peer session mechanism is used, the connecting SAS process uses a proprietary protocol to request access to the metadata server. The proprietary protocol causes the metadata server to trust the authentication that the connecting SAS server has already performed.

Note: You must implement security that is appropriate for your environment in order to protect the metadata server. To learn how to set up and manage trusted peer session connections, see “Implementing Trusted Authentication Mechanisms” in the *SAS Integration Technologies: Server Administrator’s Guide* at support.sas.com/rnd/itech/doc9/admin_oma/security/auth/security_imptrust.html. Δ

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 Web Report Studio to view a report that contains live data, the application might have to provide your credentials to a SAS Stored Process Server or a SAS Workspace Server so that those servers 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 Platform uses a single sign-on model for additional 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 following sections describe the ways that applications can obtain your credentials and provide those credentials to the servers that need to verify your identity.

Note: In addition to the single sign-on features that are described in the following section, some SAS applications can prompt you for your credentials during additional authentication (if those credentials are not otherwise available). △

Using Credentials That Are Stored in the Metadata

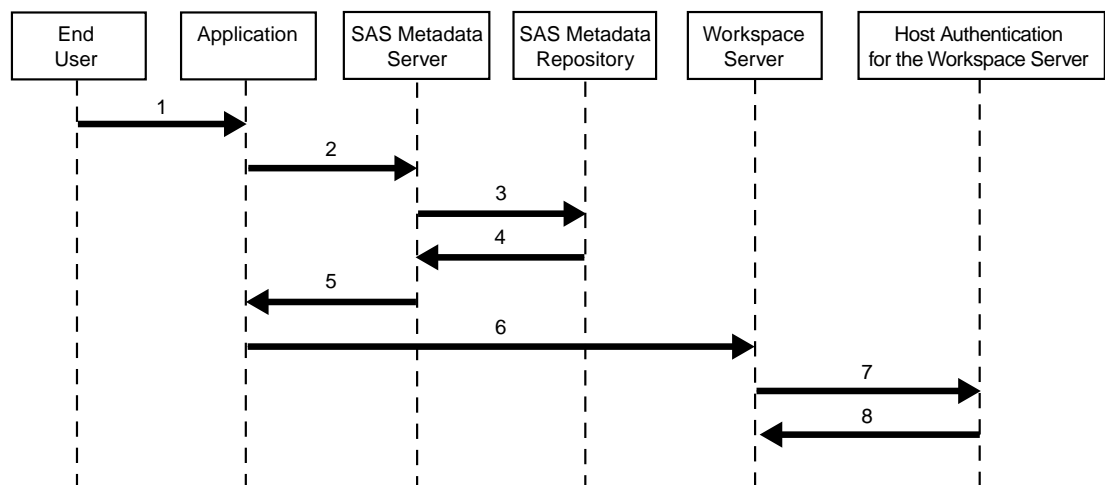
The SAS Metadata Server supports single sign-on by enabling you to store multiple sets of credentials in the metadata for each user and user group. Each set of credentials is stored in a login that is associated with a distinct set of computing resources.

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 passes that user ID and password to the server that you need to access. The target server then uses those credentials to verify your identity against its authentication provider.

The following figure depicts this process. The example assumes these things:

- The user is represented in the repository by a metadata identity that owns a login that contains credentials for accessing the SAS Workspace Server.
- The user has already completed initial authentication.
- The target server is a workspace server that is not configured for pooling.

Figure 10.6 Additional Authentication



Note: In order to provide a generalized depiction that is applicable to a wide variety of target servers, the figure omits the object spawner (which is used to launch workspace

servers and stored process servers). For the purposes of completeness, implementation details relating to the object spawner are noted in the following process description. \triangle

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

- 1 The user makes a request that requires access to a SAS Workspace Server.
- 2 The application recognizes that the request requires access to a workspace server, so the application goes to the metadata server to get credentials that will give the user access to a workspace server.
- 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.
- 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 target server.

Note: Because the target server is an unpooled workspace server, the application actually sends the credentials to the object spawner that will launch the workspace server (rather than to the workspace server itself). \triangle

- 7 The target server passes the credentials to its authentication provider for verification.

Note: In this example, it is actually the object spawner (rather than the workspace server) that passes the credentials to its authentication provider for verification. The authentication provider for a workspace server is always the host operating system. \triangle

- 8 The authentication provider tells the target server that the credentials are valid. The target server then accepts the connection.

Note: In this example, the host operating system tells the object spawner (rather than the workspace server) that the credentials are valid. The object spawner then launches a workspace server for the requesting user. \triangle

Additional examples and depictions of server-specific aspects of this process are provided in “Examples: Accessing SAS Servers” on page 165 and “Examples: Accessing Third-Party Servers” on page 170.

Using Cached Credentials

Many SAS applications also support single sign-on by caching the credentials that you provide when you log on. The cached credentials can be used during additional authentication. Cached credentials can enable you to access servers that use the same authentication provider as the server that initially authenticated you. For example, if you are initially authenticated on a metadata server that is using UNIX host authentication, SAS Information Delivery Portal caches the UNIX credentials that you submit and can provide those credentials to a SAS Workspace Server that is also running on UNIX. This enables the SAS Workspace Server to verify your identity—even if your UNIX password is not stored in the metadata.

The cached credentials are used to provide access to servers within only one specified authentication domain. When you request access to a server that is in that specified authentication domain, the application provides your cached credentials for authentication by the target server.

Note: For servers in that one specified authentication domain, your cached credentials are used even if those credentials are also stored in the metadata. △

The following table explains how some applications use your cached credentials.

Table 10.2 How Applications Use Your Cached Credentials

Application	How the Application Uses Your Cached Credentials
SAS Add-In for Microsoft Office	The authentication domain of the target server is the same as the authentication domain that is specified in the AuthenticationDomainName= field in the CSIDL_APPDATA\SAS\Metadata Server\oms_serverinfo.xml file. By default, the value in this field is DefaultAuth, so your cached credentials are used when you access servers that are associated with the DefaultAuth authentication domain.
SAS Information Delivery Portal	The authentication domain of the target server is the same as the authentication domain that is specified in the \$SERVICES_OMI_DOMAINS= field in the PortalConfigure\install.properties file. By default, the value in this field is DefaultAuth, so your cached credentials are used when you access servers that are associated with the DefaultAuth authentication domain. Note: The authentication domain that is specified in the install.properties file must match the authentication domain that is specified in the login.config file (or its equivalent) for the Web container that SAS Information Delivery Portal is using. If the two values do not match, the authentication fails.
SAS Information Map Studio	The authentication domain of the target server is the same as the authentication domain that you specify in your metadata profile (when you log on to SAS Information Map Studio). If you do not specify an authentication domain in your metadata profile, then SAS Information Map Studio attempts to use your cached credentials to access servers that are associated with the DefaultAuth authentication domain.
SAS Marketing Automation	The authentication domain of the target server is the same as the authentication domain that is specified in the login.config file (or its equivalent) for the Web container that SAS Marketing Automation is using. By default, this is the DefaultAuth authentication domain, so your cached credentials are used when you access servers that are associated with the DefaultAuth authentication domain.

Application	How the Application Uses Your Cached Credentials
SAS Web Report Studio	<p>The authentication domain of the target server is the same as the authentication domain that is specified in the \$LOGON_DOMAINS= field in the wrs.config file. By default, the value in this field is DefaultAuth, so your cached credentials are used when you access servers that are associated with the DefaultAuth authentication domain.</p> <p>Note: The authentication domain that is specified in the wrs.config file must match the authentication domain that is specified in the login.config file (or its equivalent) for the Web container that SAS Web Report Studio is using. If the two values do not match, the authentication fails.</p>
SAS Web Report Viewer	<p>The authentication domain of the target server is the same as the authentication domain that is specified in the \$LOGON_DOMAINS= field in the wrv.config file. By default, the value in this field is DefaultAuth, so your cached credentials are used when you access servers that are associated with the DefaultAuth authentication domain.</p> <p>Note: The authentication domain that is specified in the wrv.config file must match the authentication domain that is specified in the login.config file (or its equivalent) for the Web container that SAS Web Report Viewer is using. If the two values do not match, the authentication fails.</p>

Sharing User Context

SAS Web applications can also support single sign-on by sharing user context and session information. This enables you to launch one Web application from within another Web application without having to log on to the second application. For example, if SAS Web Report Viewer and SAS Information Delivery Portal use the same remotely-deployed session service, you can access SAS Web Report Viewer from the portal application without logging in again. In this example, SAS Web Report Viewer shares the session and user context that was initiated when you logged in to the SAS Information Delivery Portal. For more information, see "SAS Foundation Service Deployment and Use" in the *SAS Web Infrastructure Kit: Administrator's Guide* at support.sas.com/rnd/itech/doc9/portal_admin/deploy/ag_servdeployhow.html.

Summary: Credential Management Features by Client

The following table shows which credential management features work with each SAS client.

Table 10.3 Credential Management Features by Client

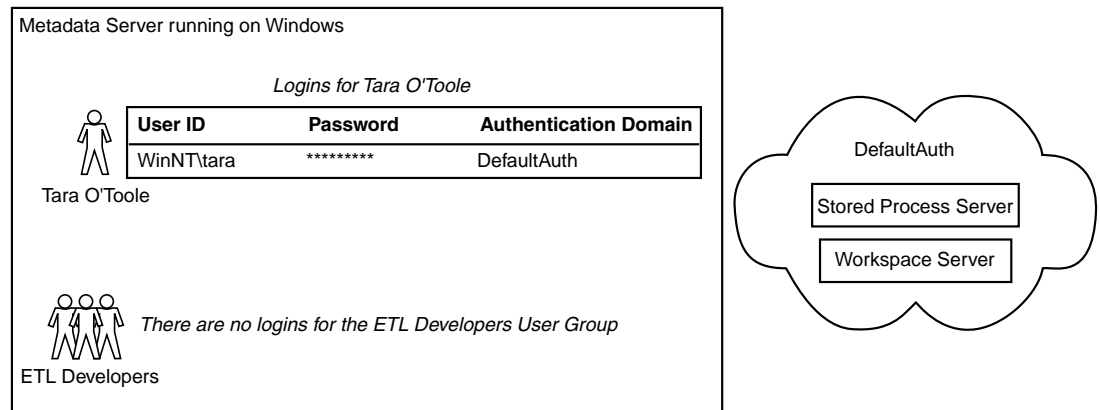
Client	Supported Credential Management Features			
	Repository Storage and Retrieval	Credential Caching	Context Sharing	Interactive Prompting
SAS Add-In for Microsoft Office	x	x		
SAS Enterprise Guide	x	x		x
SAS Enterprise Miner	x	x		
SAS ETL Studio	x			x
SAS Information Delivery Portal	x	x	x	
SAS Information Map Studio	x	x		
SAS OLAP Cube Studio	x			x
SAS Web Report Studio	x	x	x	
SAS Web Report Viewer	x	x	x	

Examples: Using Authentication Domains

This section explains 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.

Single Platform Environments

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 10.7 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.

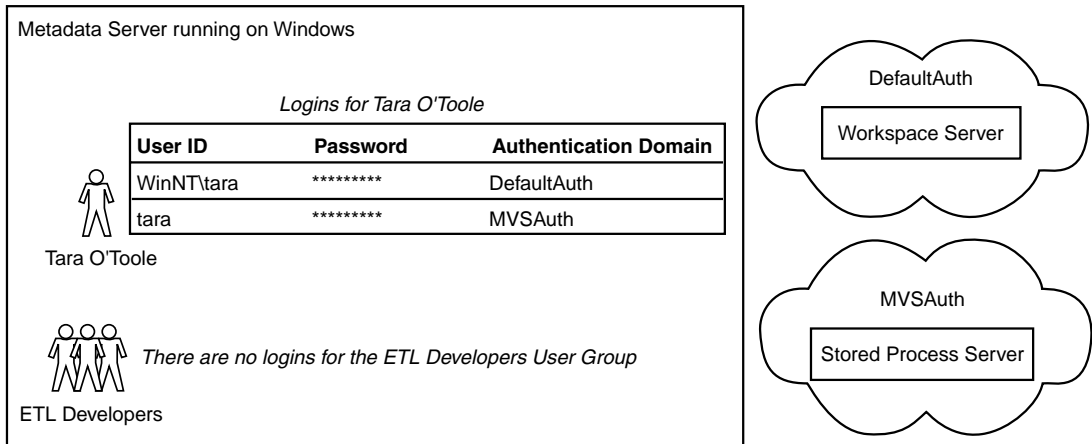
Note: If the application that Tara is using caches her credentials, then Tara can access the workspace and stored process servers using credentials that are cached from initial authentication. In this scenario, Tara's login would not have to include a password. △

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

Mixed Platform Environments

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 10.8 Mixed Environment, Two Authentication Domains



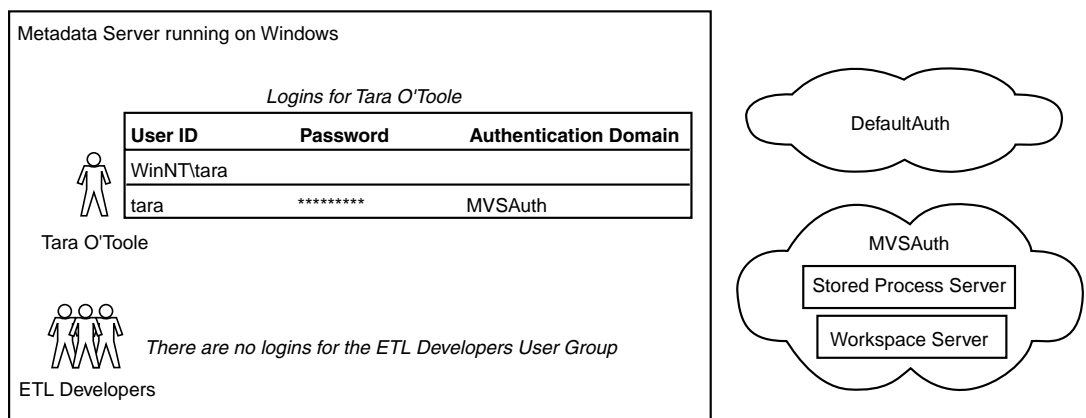
In this figure, a new authentication domain named MVSAuth has been defined, and the stored process server has been registered in that authentication domain. Two logins have been defined for Tara:

- 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 the application that Tara is using caches her credentials, then Tara 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. Δ

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 10.9 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

essential 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 for any other purposes.

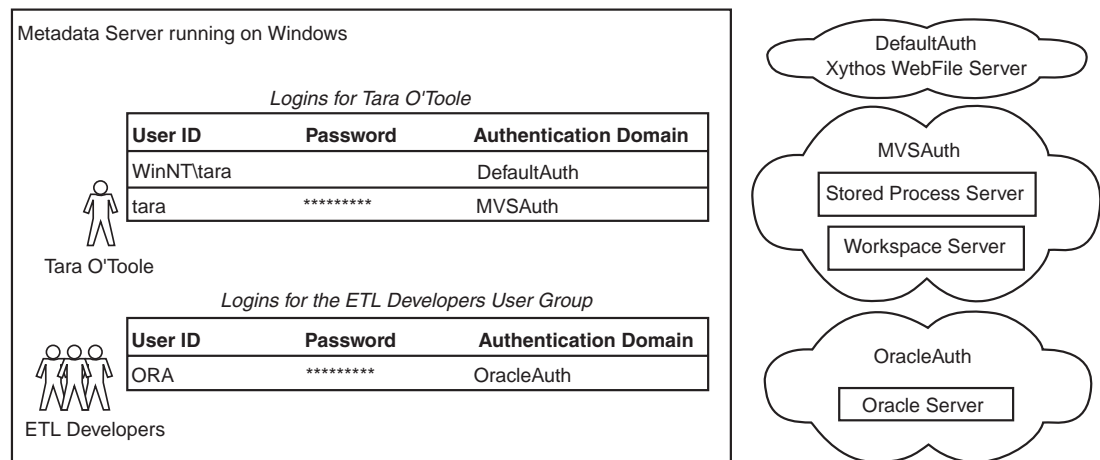
Diverse Environments

In a diverse environment, you might need more authentication domains. In this example, you add two servers to the previous deployment:

- an Oracle server that uses database authentication. When you add this server, you must 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 must 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.
- a Xythos WebFile Server that delegates authentication to the SAS Metadata Server.* When you add the Xythos server, you do not need to add a new authentication domain because your users will use their metadata server credentials to access the Xythos server. However, for each user who will access resources on the Xythos server, you must specify the DefaultAuth authentication domain on the login that the metadata server uses to determine that user's identity. This enables the user to authenticate to the Xythos server.

The following figure depicts the revised deployment.

Figure 10.10 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 Xythos WFS server has been added to the DefaultAuth authentication domain.

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. This login is now also used to enable the metadata server to authenticate Tara on behalf of the Xythos server. This use requires the first login to be assigned to the DefaultAuth authentication domain.

* This is the default configuration for authentication to a Xythos WebFile Server. More information and configuration details are provided in "Accessing a Xythos WebFile Server" on page 171.

- 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. △

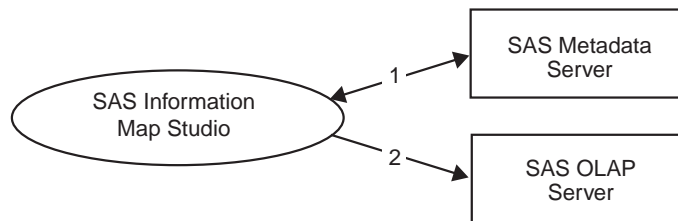
Examples: Accessing SAS Servers

This section contains specific examples of additional authentication from various applications to SAS OLAP Servers, SAS Workspace Servers, and SAS Stored Process Servers. The examples assume these things:

- your deployment uses the standard accounts that are described in your pre-installation checklist. For a summary of these accounts, see “Checking Your Metadata for Required Objects” on page 111.
- you have completed initial authentication
- you have a metadata identity
- the logins that you need for additional authentication are defined in the metadata repository
- the accounts that you need have been established with the appropriate authentication providers.
- each SAS OLAP Server, SAS Workspace Server, and SAS Stored Process Server is registered in the metadata and is associated with an appropriate authentication domain.

Accessing a SAS OLAP Server

This example describes the additional authentication process from SAS Information Map Studio to a SAS OLAP Server. The process is initiated when you make a request to access cubes from SAS Information Map Studio. The process is depicted in the following figure.

Figure 10.11 Additional Authentication to a SAS OLAP Server

The numbers in the diagram correspond to these activities:

- 1 SAS Information Map Studio goes to the metadata server to get your credentials for the SAS OLAP Server. As the requesting client, you must have ReadMetadata permission to the SAS OLAP server definition. You (or a group to which you belong) must have a login for the authentication domain that is associated with the SAS OLAP Server definition. The user ID and password in that login must correspond to an account that has been established with the SAS OLAP Server's authentication provider.

Note: If the application can use your cached credentials to access the SAS OLAP Server, then this step is omitted. Δ

- 2 SAS Information Map Studio provides your credentials to the SAS OLAP Server. The SAS OLAP Server then authenticates you against its authentication provider.

Accessing a SAS Workspace Server

This example describes the additional authentication process from SAS Web Report Studio to a SAS Workspace Server. The process is initiated when you make a request that requires access to a workspace server from SAS Web Report Studio.

Note: In this example, the SAS Workspace Server is not part of a pool. The next example describes the process for accessing a pooled workspace server. Δ

These are the prerequisites for accessing a workspace server:

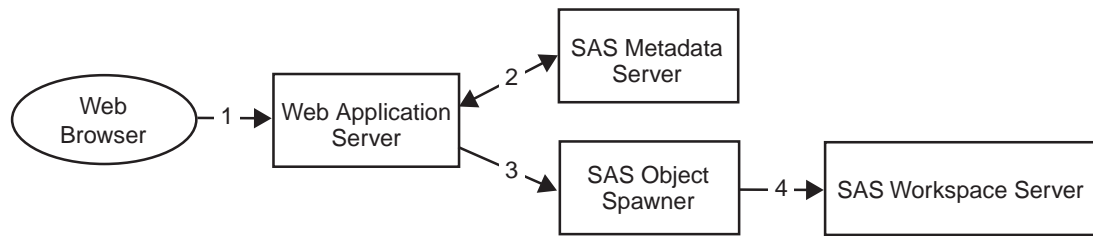
- The metadata server must be running.
- The object spawner must be running; and must have been started after the metadata server was started.
- When it initializes, the object spawner must be able to get information about the workspace server from the metadata server. To get this information, the object spawner connects to the metadata server as the SAS Trusted User (which corresponds to the sastrust account on the metadata server).*

By default, the SAS Trusted User can see the workspace server definition because sastrust is a member of the SAS System Services user group, which has ReadMetadata access to the repository. As you set access controls, you must ensure that the SAS System Services group does not lose its ReadMetadata access to the workspace server definition. To learn how to manage access to server definitions, see “Controlling Access to Resources” on page 228.

* Although the object spawner uses the sastrust account to connect to the metadata server, this connection does not make use of any trusted user functionality.

The following diagram depicts the process that is initiated by your request.

Figure 10.12 Additional Authentication to a SAS Workspace Server



The numbers in the figure correspond to these steps:

- 1 Your Web browser sends the request to the SAS Web Report Studio application.
- 2 The application goes to the metadata server to get your credentials for the workspace server. The application must find a login that is associated with the workspace server's authentication domain and is owned by you (or by a user group to which you belong).

Note: If the application can use your cached credentials to access the workspace server, then this step is omitted. Δ

- 3 The application asks the object spawner to launch a workspace server for you, using your credentials.
- 4 The object spawner uses the credentials that were obtained from the metadata server to authenticate you using host authentication. The object spawner then launches a workspace server for you.

Accessing a Pooled SAS Workspace Server

This example describes the additional authentication process from SAS Web Report Studio to a pooled SAS Workspace Server. For an overview of why and how to pool workspace servers, see “Workspace Server Pooling for SAS Web Report Studio and SAS Information Delivery Portal” on page 366.

When you set up pooling, you assign one login to each puddle within the pooled logical workspace server. Each puddle login corresponds to an account that has been established in the host environment of the workspace server. When an application asks the object spawner to launch an additional physical workspace server into the pool, the application must provide the user ID and password for one of the puddle logins. Before the object spawner launches the physical workspace server, the object spawner checks those credentials against the host operating system.

If you make a request that requires access to the pooled workspace server, your request does not trigger any further authentication. For this reason, you do not have to have a host account in order to access a pooled workspace server. You do, however, have to be a member of at least one user group that is associated with at least one puddle in the pool of workspace servers (as this example explains).

These are the prerequisites for accessing a pooled workspace server:

- The metadata server must be running.
- The object spawner must be running; and must have been started after the metadata server was started.
- When it initializes, the object spawner must be able to get information about the workspace server from the metadata server. To get this information, the object

spawner connects to the metadata server as the SAS Trusted User. As explained in the previous example, the SAS Trusted User can see the workspace server definition because `sastrust` is a member of the SAS System Services user group, which has `ReadMetadata` access to the repository.

- The requesting application must be able to obtain all of the puddle logins from the metadata repository. This enables the requesting application to provide the object spawner with the credentials that the object spawner will use to launch the workspace servers. The account that the requesting application uses to retrieve the puddle logins from the metadata repository is called the pool administrator.
- The logical workspace server must be configured for pooling.

The following diagrams depict the process that is initiated by your request.

Figure 10.13 Accessing an Existing Pooled SAS Workspace Server

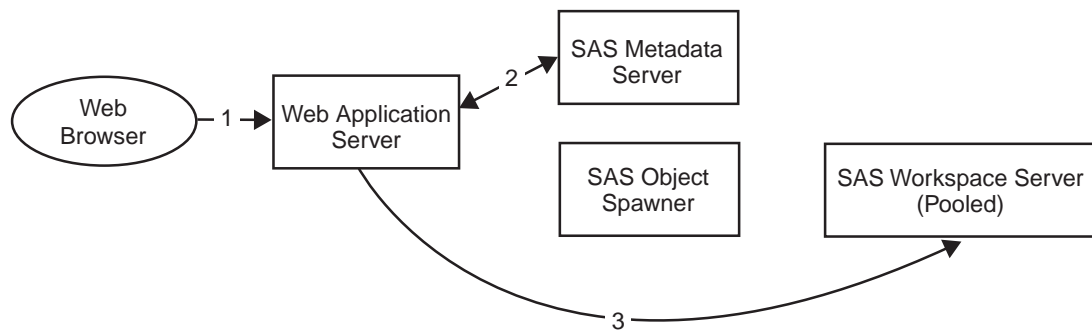
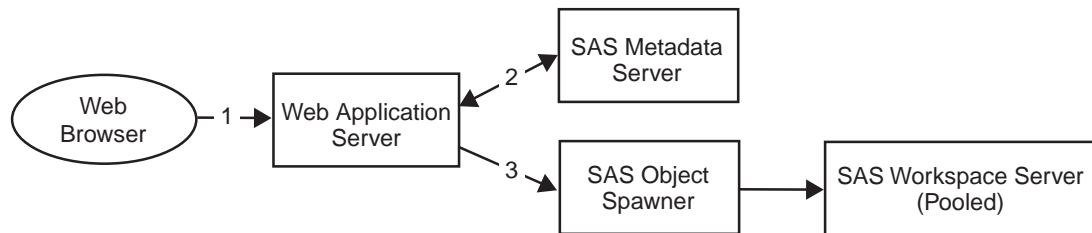


Figure 10.14 Accessing a Newly Launched Pooled SAS Workspace Server



The numbers in the figure correspond to these steps:

- 1 Your Web browser sends the request to the SAS Web Report Studio application.
- 2 SAS Web Report Studio checks your group membership information in the metadata repository in order to determine which puddles you are allowed to use.

Note: In the metadata, each puddle is assigned to one user group. If you do not belong to any user groups that are assigned to a puddle then you will not be able to connect to a workspace server. Δ

- 3 SAS Web Report Studio does one of these things:
 - If there is an available workspace server in a puddle that you are allowed to use, then SAS Web Report Studio sends your request to that workspace server.
 - If there are no available workspace servers in any of the puddles that you are allowed to use, then SAS Web Report Studio asks the object spawner to launch a new workspace server in an appropriate puddle.

Accessing a SAS Stored Process Server

This example describes the additional authentication process from SAS Add-In for Microsoft Office to a SAS Stored Process Server. The process is initiated when you make a request that requires access to a stored process server from SAS Add-In for Microsoft Office.

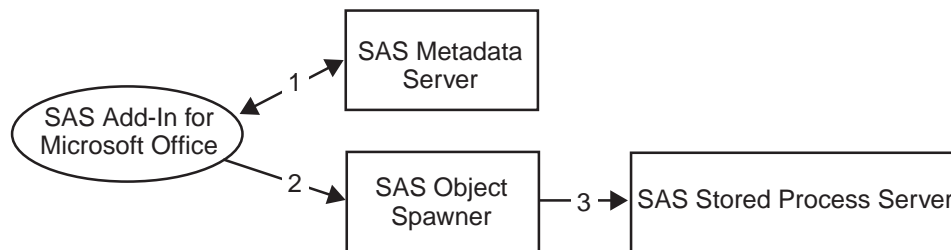
These are the prerequisites for accessing a stored process server:

- The metadata server must be running.
- The object spawner must be running; and must have been started after the metadata server was started.
- When it initializes, the object spawner must be able to get information about the stored process server from the metadata server. To get this information, the object spawner connects to the metadata server as the SAS Trusted User (sastrust). This user must be able to see the stored process server definition *and* to use the sassrv login (under which the stored process server runs).
 - 1 By default, the SAS Trusted User can see the stored process server definition because sastrust is a member of the SAS System Services user group, which has ReadMetadata permission for the repository. As you set access controls, you must ensure that the SAS System Services group does not lose its ReadMetadata access to the stored process server definition. To learn how to manage access to server definitions, see “Controlling Access to Resources” on page 228.
 - 2 The SAS Trusted User can use the sassrv login because sastrust is a member of the SAS General Servers user group, which owns the sassrv login.

Note: Only members of the SAS General Servers group can use the sassrv login. An unrestricted user such as the SAS Administrator (which corresponds to the sasadm account on the metadata server) cannot obtain any passwords, so you should not use the sasadm account in place of the sastrust account. △

The following diagram depicts the process that is initiated by your request.

Figure 10.15 Additional Authentication to a SAS Stored Process Server



The numbers in the figure correspond to these steps:

- 1 SAS Add-In for Microsoft Office goes to the metadata server to get your credentials for the stored process server. The application must find a login that is associated with the stored process server’s authentication domain and is owned by you (or by a user group to which you belong).

Note: If the application can use your cached credentials to access the stored process server, then this step is omitted. △

- 2 The application asks the object spawner for a stored process server.
- 3 The object spawner either uses an existing stored process server or launches a new one. The stored process server authenticates you using host authentication and then runs under the sassrv account.

Examples: Accessing Third-Party Servers

This section contains specific examples of additional authentication from various applications to third-party database servers and WebDAV servers. The examples assume these things:

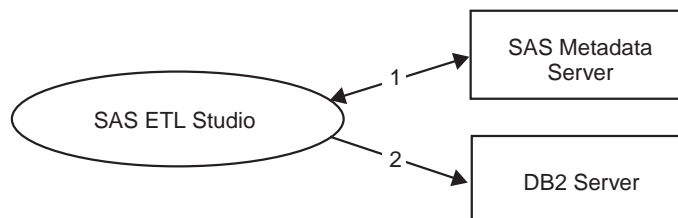
- your deployment uses the standard accounts that are described in your pre-installation checklist. For a summary of these accounts, see “Checking Your Metadata for Required Objects” on page 111.
- you have completed initial authentication
- you have a metadata identity
- the logins that you need for additional authentication are defined in the metadata repository
- the accounts that you need have been established with the appropriate authentication providers.
- each third party server is registered in the metadata and is associated with an appropriate authentication domain.

Accessing a DB2 Database

This example describes the additional authentication process from SAS ETL Studio to a DB2 database, using the SAS/ACCESS Interface to DB2. For information about this product, see *SAS/ACCESS for Relational Databases: Reference* in SAS OnlineDoc.

The process is initiated when you make a request to access DB2 data from SAS ETL Studio. The following figure depicts the process.

Figure 10.16 Additional Authentication to a DB2 Database



The numbers in the diagram correspond to the following activities:

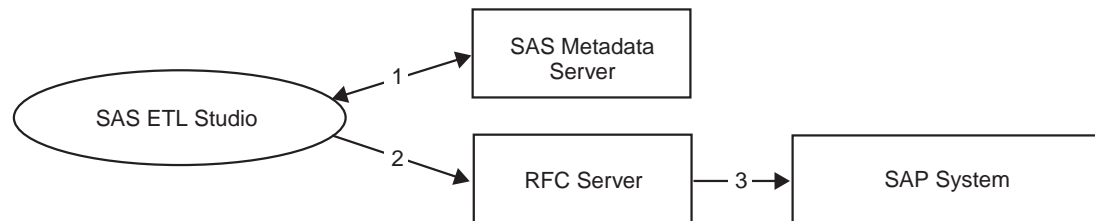
- 1 SAS ETL Studio goes to the metadata server to get your credentials for the DB2 system. As the requesting client, you must have ReadMetadata access to the DB2 server definition. You (or a group to which you belong) must have a login for the authentication domain that is associated with the DB2 server definition. The user ID and password in that login must correspond to an account that has been established with the DB2 server.
- 2 SAS ETL Studio provides your DB2 credentials to the DB2 server. The DB2 server verifies that those credentials correspond to an existing DB2 account.

Accessing an SAP System

This example describes the additional authentication process from SAS ETL Studio to an SAP system, using the SAS Data Surveyor for SAP. For information about the server communication for this product, see *SAS/ACCESS Interface to R/3: User's Guide* in SAS OnlineDoc.

The process is initiated when you make a request to access SAP data from SAS ETL Studio. The following figure depicts the process.

Figure 10.17 Additional Authentication to an SAP System



The numbers in the diagram correspond to the following activities:

- 1 SAS ETL Studio goes to the metadata server to get your credentials for the SAP system. As the requesting client, you must have ReadMetadata access to the SAP server definition. You (or a group to which you belong) must have a login for the authentication domain that is associated with the SAP server definition. The user ID and password in that login must correspond to an account that has been established with the SAP system.
- 2 SAS ETL Studio provides your SAP credentials to the Remote Function Call (RFC) server.
- 3 The RFC server passes your SAP credentials to the SAP system, which verifies that those credentials correspond to an existing account on the SAP system.

Accessing a Xythos WebFile Server

The process for accessing a Xythos WebFile Server differs from the process for accessing other servers in some important ways. For this reason, before presenting an example of how this process works, this topic explains how user credentials for Xythos are acquired and verified. For information about the role of Xythos WebFile Server and other WebDAV servers in the SAS Intelligence Platform, see “Requirements for a WebDAV Server” on page 73.

The first step in the authentication process is for the application (SAS Information Delivery Portal in this example) to acquire the requesting user’s credentials.

- In the default configuration, SAS Information Delivery Portal acquires the requesting user’s credentials for the Xythos server by caching the credentials that the user supplied when logging on.
- In an alternate configuration, SAS Information Delivery Portal retrieves the requesting user’s credentials for the Xythos server from the metadata repository. The SAS Information Delivery Portal determines the authentication domain for the Xythos server by checking a configuration file, rather than by examining metadata that describes that server.

The second step in the authentication process is for the target server (the Xythos WebFile Server) to verify the acquired credentials against its authentication provider.

- In the default configuration, the SAS User Customization for Xythos WFS uses the SAS Metadata Server as its authentication provider (this enables you to avoid maintaining an additional store of user information in the Xythos WebFile Server). In this process, the metadata server uses its authentication provider to verify the acquired credentials.

After the authentication provider of the metadata server verifies the requesting user's credentials, the SAS User Customization for Xythos WFS must locate a login that is owned by the requesting user and associated with the authentication domain of the Xythos server. If no such login exists, then the user cannot connect to the Xythos server. In the default configuration, the Xythos server is associated with the DefaultAuth authentication domain.

- In an alternate configuration, the SAS User Customization for Xythos WFS first retrieves (from the metadata server) the requesting user's login for the authentication domain with which the Xythos server is associated. The SAS User Customization for Xythos WFS then authenticates the requesting user by determining whether the password in the retrieved login is the same as the password that was provided by the connecting client (the SAS Information Delivery Portal in this example). This process requires that the requesting user's login for the authentication domain of the Xythos server includes a password.

The following example explains the configuration details and illustrates the additional authentication process from the SAS Information Delivery Portal to a Xythos WebFile Server. The example assumes that you are using the default configuration, which includes these things:

- In your SAS Information Delivery Portal configuration file, the authentication domain of the WebDAV server is the same as the cached credentials authentication domain. The `install.properties` file in the `PortalConfigure` directory includes these lines:

```
$DAV_DOMAIN$=DefaultAuth
$SERVICES_OMI_DOMAIN$=DefaultAuth
```

- In your SAS User Customization for Xythos WFS configuration file, the authentication domain of both the Xythos server and the metadata server is DefaultAuth. The `saswfs.properties` file in the `wfs-4.0.48` directory includes these lines:

```
com.sas.wfs.domain.dav=DefaultAuth
com.sas.wfs.domain.metadata=DefaultAuth
```

- The Xythos WebFile Server is *not* configured to force DIGEST HTTP authentication.

Note: By default, the SAS User Customization for Xythos WFS configures Xythos to use BASIC authentication. This is the preferred configuration. Δ

These configuration files are created for you based on values that you supply during installation of the SAS Information Delivery Portal and the SAS User Customization for Xythos WFS. The following tables document how the values that you supply during installation correspond to the variables in the configuration files.

Table 10.4 Installation of the SAS Information Delivery Portal

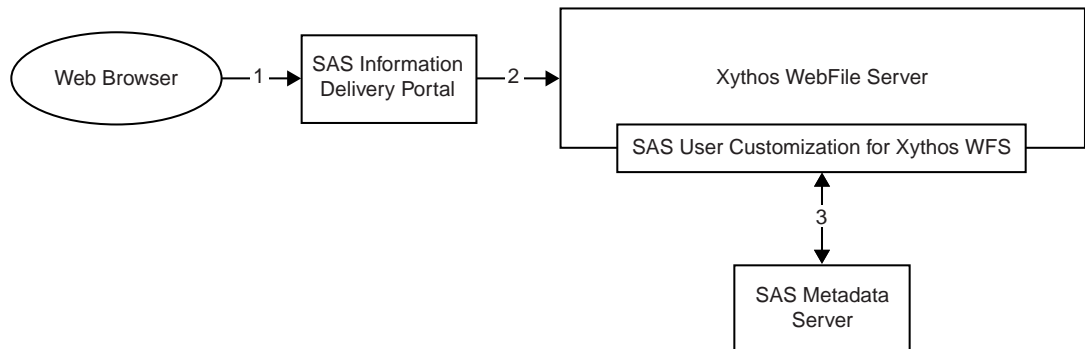
Value that you supply during installation	Line that is generated in the PortalConfigure\install.properties file
Enter the authentication domain for the SAS Metadata Server > <i>DefaultAuth</i>	\$SERVICES_OMI_DOMAINS=DefaultAuth
Enter the authentication domain for the WebDAV Server > <i>DefaultAuth</i>	\$DAV_DOMAINS=DefaultAuth

Table 10.5 Installation of the SAS User Customization for Xythos WFS

Value that you supply during installation	Line that is generated in the wfs-4.0.48\saswfs.properties file
Enter the authentication domain for the SAS Metadata Server > <i>DefaultAuth</i>	com.sas.wfs.domain.metadata=DefaultAuth
Enter the authentication domain for the Xythos WFS WebDAV Server > <i>DefaultAuth</i>	com.sas.wfs.domain.dav=DefaultAuth

The following figure depicts the process that is initiated when you make a request to access a resource that is stored on the Xythos WebFile Server.

Figure 10.18 Additional Authentication to a Xythos WebFile Server



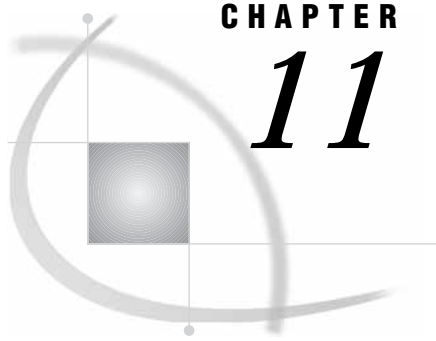
The numbers in the figure correspond to the following activities:

- 1 From your Web browser, you make a request to the SAS Information Delivery Portal for a resource that is stored in a WebDAV area on a Xythos WebFile Server.
- 2 The SAS Information Delivery Portal sends your credentials to Xythos for authentication. In this example the default configuration is used, so cached credentials are used.

Note: If the PortalConfigure\install.properties file does not assign both the Xythos server and the metadata server to the same authentication domain, then cached credentials are not used. Instead, the SAS Information Delivery Portal gets your credentials for the Xythos server from the metadata server. In this process, the SAS Information Delivery Portal searches the metadata repository for credentials that are associated with the authentication domain that is specified in the \$DAV_DOMAINS= setting in the SAS Information Delivery Portal's install.properties file. Δ

- 3 The SAS User Customization for Xythos WFS sends your credentials to the metadata server for authentication. In this example the default configuration is used, so the metadata server's authentication provider verifies your identity. After the credentials are verified, the SAS User Customization for Xythos WFS verifies that you have a login for the authentication domain of the Xythos server. In this example the default configuration is used, so the Xythos server is associated with the DefaultAuth authentication domain.

Note: If the `wfs-4.0.48\saswfs.properties` file does not assign the Xythos server and the metadata server to the same authentication domain, or if the Xythos server is configured to force DIGEST authentication, then your identity is not verified by the metadata server's authentication provider. Instead, your credentials are retrieved from the metadata repository and verified against the credentials that are provided by the SAS Information Delivery Portal. This requires that your login for the authentication domain of the Xythos server includes a password. \triangle



CHAPTER

11

Understanding Authorization

<i>Introduction to Understanding Authorization</i>	175
<i>Authorization Concepts and Terminology</i>	176
<i>Authorization Layers</i>	176
<i>Permissions in the Metadata Layer</i>	177
<i>ReadMetadata, WriteMetadata, and CheckInMetadata</i>	177
<i>Read, Write, Create, Delete, and Administer</i>	178
<i>Access Controls in the Metadata Layer</i>	179
<i>Direct Access Controls</i>	180
<i>Inherited Access Controls</i>	180
<i>Multiple Inheritance</i>	181
<i>Inheritance Rules</i>	182
<i>Inheritance in SAS Data</i>	182
<i>Inheritance in Relational Database Data</i>	182
<i>Inheritance in OLAP Data</i>	183
<i>Inheritance in Custom Trees</i>	184
<i>Repository Level Access Controls</i>	185
<i>The Repository ACT</i>	186
<i>Identity Hierarchy in the Metadata Layer</i>	186
<i>Principles of Access Control Precedence</i>	187
<i>Authorization Decision Process</i>	189
<i>Special Users of the Metadata Server</i>	191

Introduction to Understanding Authorization

This chapter explains in detail how the authorization process works in the SAS Intelligence Platform. For a brief overview of this subject, see “Authorization in the SAS Intelligence Platform” on page 38. Systems architects and administrators who perform security-related tasks will benefit from understanding the information that is presented in this chapter.

Authorization is the process of determining 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 Platform will help you do these things:

- manage access to resources across multiple authorization layers
- define an effective, manageable set of access controls in the metadata authorization layer.

Authorization Concepts and Terminology

This section introduces several terms that are important to understanding how authorization works in the SAS Intelligence Platform. These terms are

<i>access control</i>	a grant or denial of a particular permission to a particular resource for a particular user or group. For example, an access control can consist of a denial of the WriteMetadata permission for a particular information map to the PUBLIC user group.
<i>authorization layer</i>	a set of access controls that exists within a particular security framework, such as an operating system or a database management system. In a deployment of the SAS Intelligence Platform, there are multiple authorization layers.
<i>effective permissions</i>	a calculation of the access that you actually have to a particular object. The calculation determines the net effect of all applicable access controls. Your effective permissions to resources are limited to access that is permitted by all 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.
<i>identity hierarchy</i>	a ranking that is based on your metadata identity and group memberships. The ranking can have an impact on your effective permissions. For example, a permission that is assigned to you might override a permission that is assigned to a group to which you belong.

The next section outlines the authorization layers that can be involved in a SAS Intelligence Platform deployment and introduces the metadata authorization layer. The rest of this chapter describes the metadata authorization layer in detail, providing answers to these questions:

- What permissions are available, and which actions does each permission control?
- What types of access controls can be used to set permissions?
- How do a user's group memberships translate into an identity hierarchy?
- What principles govern the precedence interactions among access control types, object inheritance relationships, and user group memberships?
- When a user requests access to a resource, how is the authorization decision made?

Authorization Layers

Authorization layers that can affect your access to resources in a SAS Intelligence environment include:

- 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 SAS Intelligence Platform provides an additional layer, the *metadata authorization layer*. This layer enables you to control access to metadata objects that represent computing resources. In some cases, you can also use the metadata authorization layer to control access to the underlying computing resources. The metadata authorization layer has these 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 metadata identities (users or groups) 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 objects are initiated and enforced by the metadata server.
- Requests to create, view, update, or delete data can be initiated and enforced by other applications.

Permissions in the Metadata Layer

A metadata layer permission is a permission that is 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 components.

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 210 for best practice recommendations for using metadata layer permissions. Δ

ReadMetadata, WriteMetadata, and CheckInMetadata

The metadata server requests and enforces authorization decisions for the ReadMetadata, WriteMetadata, and CheckInMetadata permissions. Because all applications in the SAS Intelligence Platform 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 11.1 Permissions That the Metadata Server Enforces

Permission (Abbreviation)	Actions Controlled
ReadMetadata (RM)	Reading a metadata object. For example, you must have ReadMetadata permission to a SAS library definition in order to see that library in SAS Information Map Studio or SAS ETL Studio.
WriteMetadata (WM)	Creating, updating, or deleting a metadata object. For example, you must have WriteMetadata permission to a repository in order to add a new metadata object (such as an information map or a server definition) to that repository.
CheckInMetadata (CheckInM)	Checking in metadata from a project repository, and checking out metadata to a project repository. This permission is applicable only to SAS ETL Studio users who are working in a change-managed environment.

Read, Write, Create, Delete, and Administer

Some applications will 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.

Table 11.2 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.**

Permission (Abbreviation)	Actions Controlled
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.

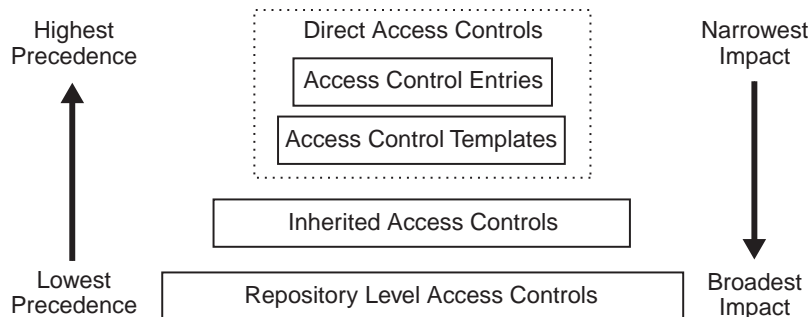
** In the current release, the Create, Write, and Delete permissions are enforced only if the SAS metadata LIBNAME engine is used to access the data.

Because not all applications in the current release enforce the Read, Write, Create, Delete, and Administer permissions, these permissions are not always sufficient to control access. 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 Controls in the Metadata Layer

The following figure provides an overview of the different types of access controls that you can use to set permissions in the metadata authorization layer. These access control types are described in the following sections.

Figure 11.1 Access Controls in the Metadata Authorization Layer



From top to bottom, the access control types in the figure are ordered as follows:

- from highest precedence (hardest to override) to lowest precedence (easiest to override). Access control precedence is explained in “Principles of Access Control Precedence” on page 187.
- from narrowest impact (most specific) to broadest impact (least specific). For example, a repository level access control can affect all of the resources in a repository, while an access control entry can be directly assigned to only one resource.

Direct Access Controls

An access control that is explicitly set on a resource is a direct access control for that resource. Direct access controls can be assigned at any level in the identity hierarchy. For example, all of the following controls are direct access controls for TableA:

- an access control that is set on TableA that grants ReadMetadata permission to you
- an access control that is set on TableA that grants ReadMetadata permission to a group to which you belong
- an access control that is set on TableA that grants ReadMetadata permission to the PUBLIC implicit group.

There are two types of direct access controls — access control entries (ACEs) and access control templates (ACTs).

- An ACE is an access control that is specifically tied to a particular resource. For example, an ACE can consist of a grant to you of ReadMetadata permission for a particular table.
- An ACT is a reusable, named pattern of identities and permissions. For example, you can create an ACT named AdminOnlyAccess that consists of a denial of WriteMetadata permission for the PUBLIC group and a grant of all permissions for an Administrators user group. You can then apply that ACT to all objects that should have those particular protections.

Note: Using ACTs rather than individual ACEs centralizes some of your access control management because an ACT can be updated independently of the resources to which it has been applied. So if you change your mind about how you want to handle this type of access, you can make changes on the AdminOnlyAccess ACT, without having to revisit every resource to which you applied that ACT. Δ

Direct access controls have precedence over all other controls in the metadata authorization layer. Conflicts among direct access controls are resolved first by applying the identity hierarchy and then by giving ACEs priority over ACTs. If neither of these factors resolve the conflict, then the permission is denied. For examples of how direct access controls are evaluated, see “Principles of Access Control Precedence” on page 187.

Note: To set direct access controls on a resource, you specify permissions on the resource’s Authorization tab in SAS Management Console. For information about using the Authorization tab, see “Controlling Access to Resources” on page 228. Δ

Inherited Access Controls

The inherited access controls for a resource consist of the effective permissions on each of the resource’s immediate parents. For example, a library is an immediate parent to all of the tables that are assigned to it. The server to which the library is assigned is an immediate parent to the library, but the server is not an immediate parent to the tables within the library. The process by which a resource’s parents are identified is described in “Inheritance Rules” on page 182.

The effective permissions that each parent conveys represent the net effect of all of the parent’s direct, inherited, and repository level controls in the metadata authorization layer. To the child resource, the only important information from the parent is the effective permission. It makes no difference to the child resource whether the access controls on the parent were direct controls or inherited controls, or whether the access controls on the parent were assigned to you or to a group to which you belong. The process by which effective permissions are calculated is described in “Authorization Decision Process” on page 189.

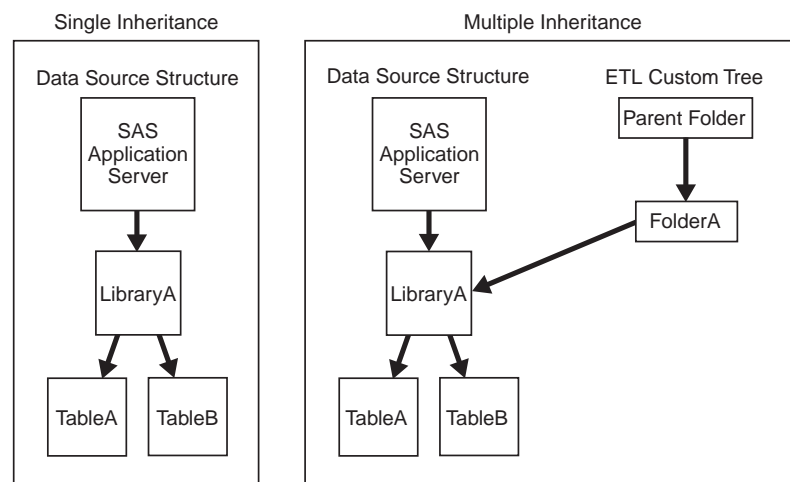
For example, if a calculation of the net effect of all of the access controls on LibraryA yields a grant of ReadMetadata permission for you, then that effective permission (that grant to you of the ReadMetadata permission) will be inherited by all of the tables that belong to LibraryA. In this example, if TableA belongs to LibraryA, then TableA will have an inherited grant of ReadMetadata permission for you.

Because the effective permissions on the parent encompass all access controls up to the level of the repository itself, every parent will convey either a grant or denial of every permission for every user. However, one resource can convey effective permissions to another resource only if both resources are registered in the same metadata repository. Even when there is a dependency relationship between two repositories, resources in one repository cannot convey effective permissions to resources in another repository.

Multiple Inheritance

The SAS metadata environment supports multiple inheritance, such that one resource can have more than one immediate parent. The following figure depicts examples of single inheritance and multiple inheritance in the SAS metadata environment.

Figure 11.2 Single and Multiple Inheritance



In the single inheritance figure, TableA and TableB share the same immediate parent, LibraryA. Both tables have inherited access controls that are conveyed from the effective permissions on LibraryA. LibraryA has inherited access controls that come from the effective permissions on the SAS Application Server. For example, if a calculation of the net effect of the applicable access controls on LibraryA yields a grant of ReadMetadata permission for you, then both TableA and TableB will have inherited access controls that grant ReadMetadata permission to you.

Note: You can override an inherited access control by adding a direct access control. For example, If you set a permission on TableA that denies you the ReadMetadata permission, then that direct access control on TableA will override the conflicting grant that LibraryA conveys to TableA. In this example, the authorization decision process will not consider the inherited access control because of the presence of the pertinent, direct access control. Δ

In the multiple inheritance figure, LibraryA is assigned to both a SAS Application Server and to a folder in the ETL custom tree. The inherited access controls for

LibraryA consist of the effective permissions on both of these immediate parents. For example, if the net effect of the access controls on the SAS Application Server is a grant of ReadMetadata permission to you, and the net effect of the access controls on FolderA is a denial of the same permission to you, then LibraryA has both an inherited grant and an inherited denial of this permission for you. In this circumstance, you can access LibraryA, because a grant that is conveyed by *any* parent object is sufficient.

Inheritance Rules

In a SAS metadata environment, inheritance rules determine which objects can be parents to which other objects. For example, in the previous section, the reason why TableA inherits the effective permissions on LibraryA is because there is an inheritance rule that specifies that a SAS library definition is a parent to the SAS data set definitions within that library.

The inheritance rules establish containment structures through which resources inherit access controls. In each structure, you can set access controls at varying levels of granularity. For example, you can deny someone ReadMetadata access to an entire library of SAS data sets, or to a particular data set, or a particular variable within a data set. The main resource containment structures that are established by inheritance rules are described in the following sections.

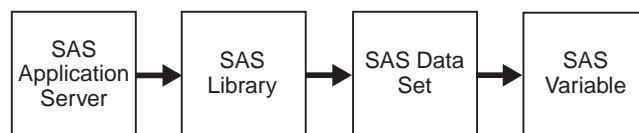
Inheritance in SAS Data

In a SAS metadata environment, effective permissions for SAS data flow 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 inheritance flow for SAS data in the metadata environment.

Figure 11.3 Inheritance Flow for 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 effective permissions from custom trees, as described in “Inheritance in Custom Trees” on page 184. Δ

Inheritance in Relational Database Data

In a SAS metadata environment, effective permissions for relational database data flow as follows:

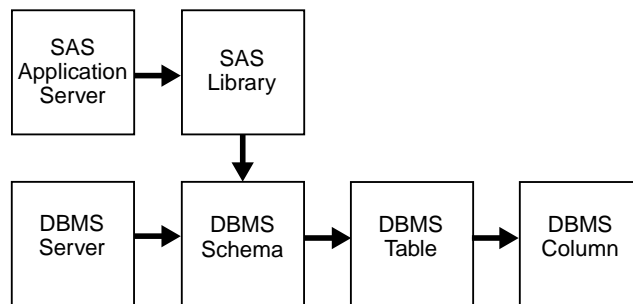
- from a database management system (DBMS) server definition to the DBMS schemas that are defined on that DBMS server

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

- 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 inheritance flow for database data in the metadata environment.

Figure 11.4 Inheritance Flow for 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 Trees” on page 184. Δ

Inheritance in OLAP Data

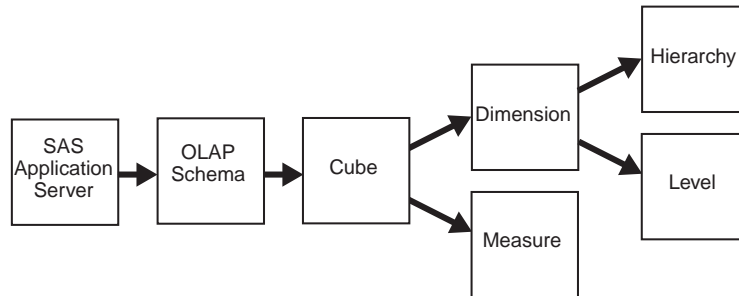
In a SAS metadata environment, effective permissions for OLAP data flow as follows:

- 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. Δ

- from a dimension to the hierarchies and levels within the dimension.

The following figure depicts the inheritance flow for OLAP data in the metadata environment.

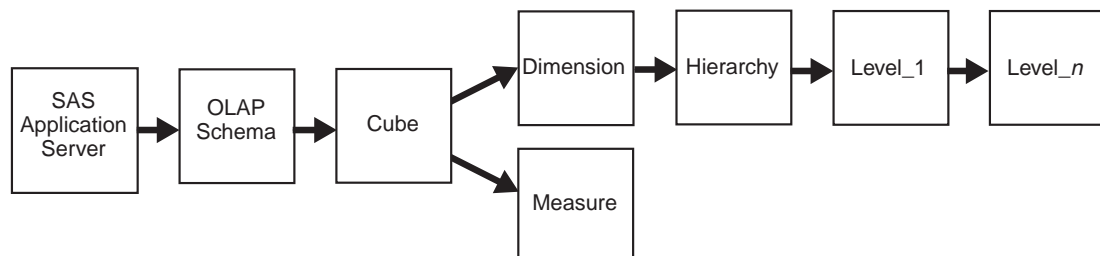
Figure 11.5 Inheritance Flow for OLAP Data

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 11.6 Access Requirements for Navigating through OLAP Data

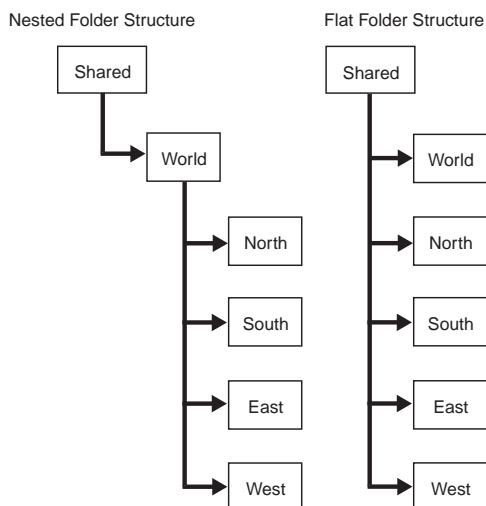
Inheritance in Custom Trees

A SAS Intelligence deployment can include one or more custom trees that you can use to organize and manage access for certain resources. For example, SAS ETL Studio

enables you to add folders and items to the ETL custom tree. Within a custom tree, each folder inherits the effective permissions of its parent folder. Most items in the ETL custom tree inherit the effective permissions of 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 292.

Selecting the optimal folder structure for your custom trees 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 11.7 Custom Folder Structures



The arrows show how the inheritance of effective permissions flows in each custom tree.

- In the nested structure, the regions each inherit the effective permissions of the World reports folder, which in turn inherits the effective permissions of the Shared folder.
- In the flat structure, the World folder and all of the regional folders inherit the effective permissions of the Shared folder.

Repository Level Access Controls

The permissions that you have been given to the entire repository are your repository level access controls. Repository level controls define your access to resources for which no specific controls have been set. For example, if a resource has no direct access controls and no parent objects, then your access to that resource is defined by the repository level access controls. When there are no other parent objects, the repository itself serves as a parent.

Repository level controls also define your ability to perform the actions that are listed in the following table.

Table 11.3 Actions That Are Controlled by Repository Level Access Controls

User Action	Required Access to the Repository
Access or view objects in the metadata repository.	In many cases, the requesting user must have ReadMetadata access to the repository.
Create a new object (such as an information map or report) anywhere in the metadata repository.	The requesting user must have WriteMetadata access to the repository.
Access the SAS Information Delivery Portal (Public Kiosk).	The SAS Guest and the SAS Web Administrator must have ReadMetadata and WriteMetadata access to the repository.
Log on to the SAS Information Delivery Portal.	The requesting user must have ReadMetadata and WriteMetadata access to the repository.

As the preceding table indicates, having both ReadMetadata and WriteMetadata access to the repository is a prerequisite to performing many tasks. For example, in order to add a new report to a particular folder, you must have both of these things:

- WriteMetadata access to that particular folder
- WriteMetadata access to the entire repository.

The Repository ACT

You can define access to most resources by locating the resource in SAS Management Console and then setting controls on the resource's Authorization tab. You do not use this method to define access to a repository. Instead, you specify repository level access controls on the Users and Permissions tab of an ACT that has been designated as the repository ACT. The repository ACT is represented in SAS Management Console by a blue cylinder icon and is named *Default ACT* by default. You can locate the repository ACT under **Environment Management** \blacktriangleright **Authorization Manager** \blacktriangleright **Access Control Templates**.

A repository ACT is created for you in each repository. The initial settings on a foundation repository ACT grant ReadMetadata and WriteMetadata permissions to the PUBLIC group. For information about narrowing this access, see "Protecting the Foundation Repository" on page 214.

Identity Hierarchy in the Metadata Layer

In the metadata authorization layer, the identity hierarchy is used to resolve conflicting permissions in these circumstances:

- When there are conflicting direct access controls on a resource.
- When there are conflicting settings on the Users and Permissions tab of an access control template.

Note: For examples that illustrate how the identity hierarchy is incorporated into the evaluation of access controls, see "Principles of Access Control Precedence" on page 187. \triangle

From highest precedence to lowest precedence, the ranking of metadata identities is as follows:

- 1 Your individual metadata identity. This is also called your primary metadata identity.

Note: The process that the metadata server uses to determine your metadata identity is described in “Metadata Identities” on page 148. △

- 2 A user-defined group that has your metadata identity as a member. This is a first-level group membership for you.
- 3 A user-defined group that has another user group as a member. For example, if you belong to a group named ETL_Advanced and that group is a member of another group called ETL_Basic, then the ETL_Basic group is a second-level group for you.

Note: You can create additional levels of nested user groups. Each successive group has less precedence than the group or groups that are its members. △

- 4 The SASUSERS implicit group, which includes everyone who has an individual metadata identity.
- 5 The PUBLIC implicit group, which includes everyone who can access the metadata server (regardless of whether they have an individual metadata identity or not).

Different users can have different identity hierarchies. The following table contains some examples of how the identity hierarchy can vary depending on user characteristics and group memberships.

Table 11.4 Examples of Identity Hierarchies

User Information	User's Identity Hierarchy
User has no metadata identity.	primary identity: PUBLIC
User has a metadata identity and no explicit group memberships.	primary identity: self first-level memberships: SASUSERS second-level memberships: PUBLIC
User is a direct member of two user defined groups (GroupA and GroupB).	primary identity: self first-level memberships: GroupA, GroupB second-level memberships: SASUSERS third-level memberships: PUBLIC
User is a direct member of two user-defined groups (GroupA and GroupB), and one of those groups is a member of a third group (GroupA is a member of the Portal Users group).	primary identity: self first-level memberships: GroupA, GroupB second-level memberships: Portal Users third-level memberships: SASUSERS fourth-level memberships: PUBLIC

Principles of Access Control Precedence

The following table summarizes the precedence principles for metadata layer access controls and presents an example of each principle.

Table 11.5 Principles of Metadata Layer Access Control Precedence

Principle	Example	
	Scenario	Outcome and Explanation
A direct access control has precedence over access controls that come from parent objects or from the repository ACT.	A direct access control on LibraryA denies ReadMetadata permission to PUBLIC. The repository ACT grants ReadMetadata permission to you.	You cannot see LibraryA. The deny to PUBLIC has precedence because it is assigned <i>directly</i> on the target resource (LibraryA). Direct access controls always have precedence over inherited controls <i>regardless of who the permissions are assigned to</i> .
If there are conflicting direct controls, then the identity hierarchy determines the outcome.	A direct access control on LibraryA denies ReadMetadata permission to PUBLIC. Another direct access control on LibraryA grants ReadMetadata permission to you.	You can see LibraryA. This is a conflict between two direct controls, so the identity hierarchy becomes relevant. Your primary identity has priority over the implicit group PUBLIC, so the direct grant to you overrides the direct denial to PUBLIC.
If there are conflicting direct controls at the same level in the identity hierarchy, then the type of access control (ACE or ACT) determines the outcome.	A direct ACT on LibraryA denies ReadMetadata permission to GroupA. A direct ACE on LibraryA grants ReadMetadata permission to GroupB. You are a member of both GroupA and GroupB.	You can see LibraryA. The conflict is between direct controls that both come from your first-level group memberships, so the type of access control becomes relevant. ACEs are given priority over ACTs, so the ACE grant overrides the ACT denial.
If there are conflicting direct controls at the same identity level <i>and</i> they are both ACEs (or they are both ACTs), then the outcome is a deny.	A direct ACE on LibraryA denies ReadMetadata permission to GroupA. Another direct ACE on LibraryA grants ReadMetadata permission to GroupB. You are a member of both GroupA and GroupB.	You cannot see LibraryA. The direct controls are both assigned at the same level in the identity hierarchy (to your first-level groups) and they are both of the same type (ACEs), so the outcome is a denial.
If there are no relevant direct controls and there is at least one parent object that conveys a grant, then the outcome is a grant.	There are no direct controls on LibraryA. LibraryA is assigned to ServerA, which makes ServerA a parent object to LibraryA. ServerA conveys a grant of ReadMetadata permission to you. ¹	You can see LibraryA. In the absence of relevant direct controls on LibraryA, a grant from <i>any</i> of LibraryA's parent objects is sufficient to get access. (Even if LibraryA had another parent object that conveyed a denial, the outcome would be a grant. For an illustration of this scenario, see Figure 11.2 on page 181.)

Principle	Example	
	Scenario	Outcome and Explanation
If there are no relevant direct controls and there are no parent objects, then the repository ACT determines the outcome.	There are no direct controls on LibraryA. LibraryA does not have any parent objects. The repository ACT denies ReadMetadata permission to you.	You cannot see LibraryA. Because there are no relevant direct access controls and no parent objects, the settings on the Users and Permissions tab of the repository ACT are determinative. ²
Conflicts within an ACT's Users and Permissions tab are resolved by the identity hierarchy rankings.	There are no direct controls on LibraryA. LibraryA does not have any parent objects. The repository ACT denies ReadMetadata permission to PUBLIC. The repository ACT grants ReadMetadata permission to SASUSERS. You have a metadata identity.	You can see LibraryA. You are a member of both PUBLIC and SASUSERS. In the identity hierarchy, SASUSERS has precedence over PUBLIC, so the net effect of the repository ACT settings is to grant ReadMetadata permission to all members of SASUSERS.

- 1 The conveyed permission comes from an *effective permission* on ServerA. For this reason, it makes no difference to LibraryA whether the source of the conveyed effective permission was a direct control on ServerA or a control that ServerA inherited from one of its parent objects. It also makes no difference to LibraryA whether the source of the conveyed permission was an access control that was assigned to you or an access control that was assigned to a group to which you belong.
- 2 If there is no repository ACT, then the outcome would be a grant. You should always have a designated repository ACT.

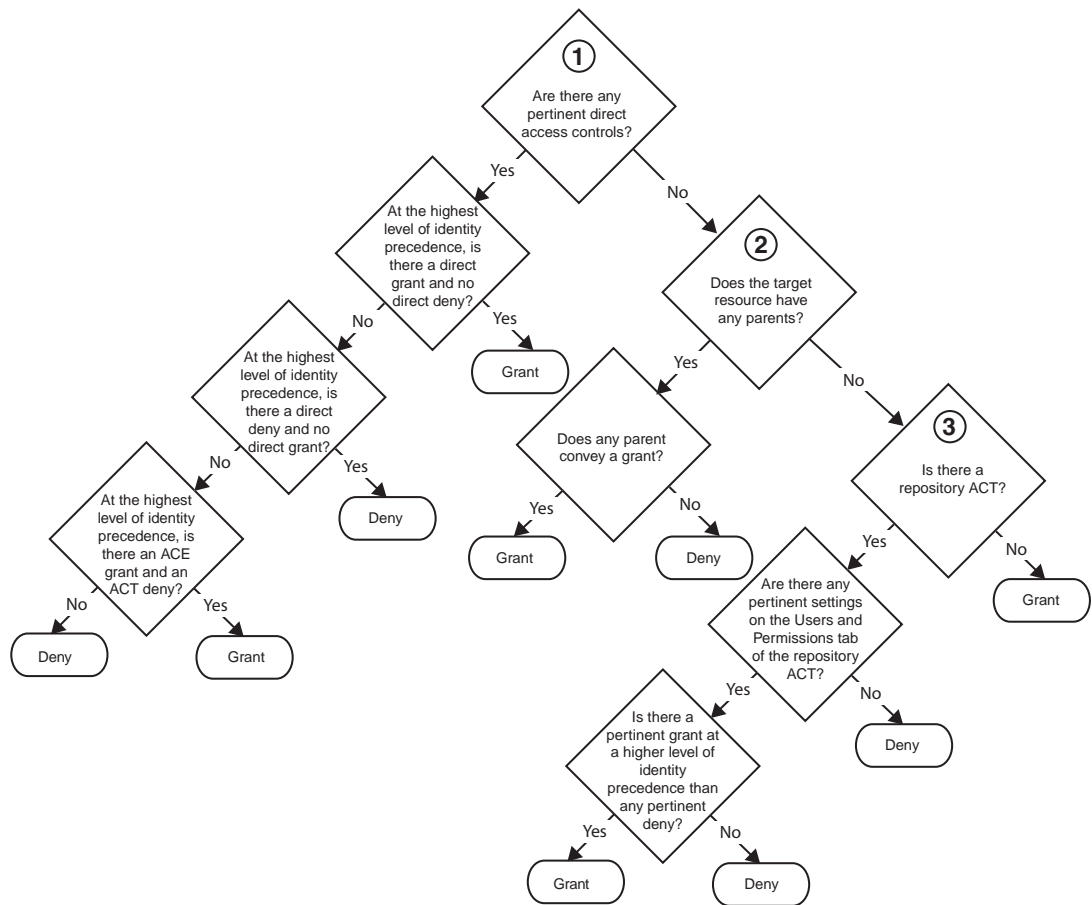
Authorization Decision Process

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 flowchart summarizes the authorization decision process that determines whether a particular identity can perform a particular action on a particular resource. In the figure, a pertinent access control is an access control that grants or denies the requested permission to the requesting identity (or to a group to which the requesting identity belongs).

Figure 11.8 How Permissions Are Evaluated



The numbers in the flowchart correspond to these activities:

- 1 ACEs and ACTs that are directly assigned to the target resource are examined.
 - Conflicting permissions that arise from group membership are resolved by the identity hierarchy. 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 identity 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 there are no pertinent direct ACEs or ACTs on the target resource, 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 (steps one through three) to each of the parent objects. The access controls on the parent object can be directly assigned permissions on the parent object, or inherited permissions from an object that is a parent to the parent object, or permissions from the repository ACT.
 - If *any* of the parent objects grants the requested permission to you (or to a group to which you belong), then that 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 denial is final.

- If the target resource does not have any parent objects, then 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 the repository ACT grants or denies the requested permission to you (or to a group to which you belong), then that grant or denial is determinative. If there are conflicting permissions within the repository ACT, those conflicts are resolved by the identity hierarchy.
 - If the repository ACT neither grants nor denies the permission to you (or to any group to which you belong), then the permission is denied.

Note: If there is no repository ACT, then the permission is granted. When there is no repository ACT and there are no relevant access controls, then any user who can access the metadata server can read and write metadata in the repository. You should always have a designated repository ACT. △

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

Special Users of the Metadata Server

In some cases, the ability to perform a particular task comes from having status as a special user of the metadata server, rather than from access controls that are defined in the repository.

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.

Note: An unrestricted user cannot access other servers by retrieving logins from the metadata server. For this reason, you should not use an *unrestricted user* account (such as sasadm) in the place of any of the other accounts that you create during installation. Also, do not use an *unrestricted user* account to log on to any application other than SAS Management Console. △

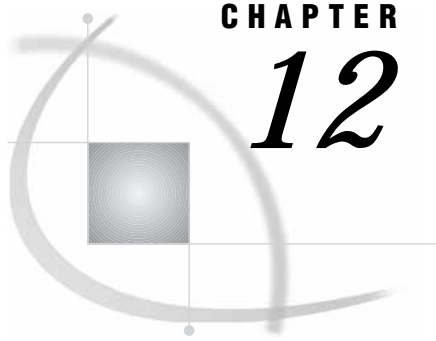
administrative users can perform the following tasks regardless of any access controls that are specified in the metadata:

- 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 middle tier applications and servers to impersonate authenticated clients.

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 131.



CHAPTER

12

Developing Your Security Plan

<i>Overview of Security Planning</i>	193
<i>Defining Your Security Goals</i>	194
<i>Making Preliminary Decisions about Your Security Architecture</i>	194
<i>Planning Your Users</i>	195
<i>Planning the User Accounts</i>	196
<i>User Accounts for Initial Authentication</i>	196
<i>User Accounts for Additional Authentication</i>	197
<i>Planning the User Metadata Identities</i>	197
<i>Planning the Logins</i>	198
<i>User Planning Summary</i>	199
<i>User Requirements in Environments That Have Homogeneous Authentication</i>	201
<i>Planning Your User Groups</i>	201
<i>Identifying Related Tasks</i>	202
<i>Defining the Group Structure</i>	203
<i>Implementation Strategy for Defining Groups</i>	204
<i>Planning Your Access Controls</i>	204
<i>Access Requirements for Common Tasks</i>	205
<i>Access Requirements for Working with Data</i>	205
<i>Access Requirements for Working with Stored Processes</i>	206
<i>Access Requirements for Working with Information Maps</i>	207
<i>Access Requirements for Working with Reports</i>	209
<i>Access Requirements for Working with Security-Related Objects</i>	210
<i>Implementation Strategy for Assigning Permissions</i>	210
<i>Considerations for Defining Effective, Efficient Access Controls</i>	210

Overview of Security Planning

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

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), organizing your users into groups, and planning for access control. A few of the tasks that are mentioned in this chapter, such as creating additional 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 Decide 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 Platform. 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:

- For servers that can use alternative authentication providers, select the technology that will be used to verify the identity of users. This choice determines whether the accounts that you define for users who are authenticated by the SAS Metadata Server, a SAS OLAP Server, or a SAS Web application 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. By default, all SAS Servers use host authentication.

You might choose to use an alternative authentication provider in any of these circumstances:

- you want to take advantage of user accounts that are already established with an alternative authentication provider
 - you want to minimize the number of host accounts that you have to create on the metadata server.
- For users who log on with Web applications, decide whether initial authentication will be on the metadata server or on a middle tier server. By default, SAS Web applications authenticate users on the metadata server.
- You might choose to modify the default configuration in any of these circumstances:
- you want to take advantage of user accounts that are already established with a middle tier authentication provider
 - you want to minimize the number of user accounts that you have to create on the metadata server.
- For each logical server, select the authentication domain to which the server will belong. By default, servers are associated with the DefaultAuth authentication domain. You might need to associate some servers with other authentication domains in any of these circumstances:
- your deployment includes multiple platforms
 - your deployment includes third-party database servers.
- For each set of computing resources, decide whether you will allow access 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 at the price of reduced individual accountability (because users access pooled workspace servers using shared accounts in the operating system).
- 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

The initial set of accounts that are required for installation are described in “Setting Up Required User Accounts” on page 68. 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.

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 50. \triangle

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. 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 Web Report Studio, SAS Management Console, and SAS Information Map Studio, you might have the following initial authentication processes:

- For SAS Information Delivery Portal and SAS Web Report Studio, you choose to have the Web application handle authentication using LDAP as the authentication provider.

- For SAS Management Console and SAS Information Map 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 or SAS Web Report Studio
- an operating system account on the computer on which the metadata server is running for each user who logs in to SAS Information Map Studio or SAS Management Console
- one operating system account on the computer on which the metadata server is running so the Web applications can access the metadata server as a *trusted user*.

Note: This is one of the required accounts that you create before installation. △

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 to identify 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 including 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 that is 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.

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 Information Map Studio and SAS Enterprise Miner require that every user has a unique metadata identity.
 - SAS Information Delivery Portal requires users to 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 "Metadata Identities" on page 148.

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 autocall macros. For more information, see "Creating and Maintaining User and Group Definitions" in the *SAS Metadata Server: Setup and Administration Guide* at support.sas.com/rnd/eai/openmeta/v9/setup/authmacros.html.
- Use the User Manager plug-in to SAS Management Console to create user and group definitions one at a time. For instructions, see *SAS Management Console: User's Guide* in SAS OnlineDoc.

Planning the Logins

In addition to the login that the metadata server uses to discover 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 either the SAS Personal Login Manager or SAS Management Console.

Note: In order to use the SAS Personal Login Manager, you must have an account with the metadata server's authentication provider. Δ

- Administrative users* and *unrestricted users* can use SAS Management Console to add logins to any user definition.
- 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 providers 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 a 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. Δ

- 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 12.1 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 is configured to authenticate users in the middle tier, then each user of that application must instead have an account with the Web server's authentication provider.	
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.

Requirement	Low Security Environment	High Security Environment
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 "Metadata Identities" on page 148.	
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 can access resources in one authentication domain using credentials that are cached from initial authentication. For details, see "Using Cached Credentials" on page 158.

* SAS Information Map Studio and SAS Enterprise Miner require 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 servers, stored process servers, 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 provider 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 support pooled workspace servers for SAS Web Report Studio. Or, 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 define a distinct level of access for each set of users. 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.

Note: 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. Δ

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 on with SAS Web Report Studio 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 on 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 on 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 on with an application that caches credentials will access some resources with their personal, cached credentials rather than with a login that is assigned to the PUBLIC group.

Planning Your User Groups

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

The types of user groups that are available in the metadata layer are listed in the following table.

Table 12.2 Metadata Layer User Groups

Type of User Group	How the Group Is Created	Group Membership
Implicit user groups	The two implicit groups, PUBLIC and SASUSERS, are created for you in every foundation repository.	Membership in the PUBLIC and SASUSERS groups is implicit. If you can access the metadata server, then you are automatically a member of the PUBLIC group. If you can access the metadata server <i>and</i> you have your own metadata identity, then you are automatically a member of both the PUBLIC group and the SASUSERS group.
Required user groups	Created during installation.	As specified in “Checking Your Metadata for Required Objects” on page 111.

Type of User Group	How the Group Is Created	Group Membership
Required roles*	Created during installation.	As specified in the documentation for your SAS solution.
Additional user-defined groups	Created by you at any time.	As needed for your environment.

* A role is a special type of user group that is used by SAS solutions applications. The solutions applications check a user's roles in order to determine whether to allow the user to perform actions such as launching the application or creating certain types of objects.

This section helps you plan the additional user groups that you will define in the metadata. 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 or 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 12.3 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 and maintain information maps.	Information Architects

Tasks	Group
Create and publish standard reports.	Report Creators
View all reports.	Power Report Consumers
Make ad hoc changes to reports.	
Save modifications to reports.	
View all reports.	Report Consumers

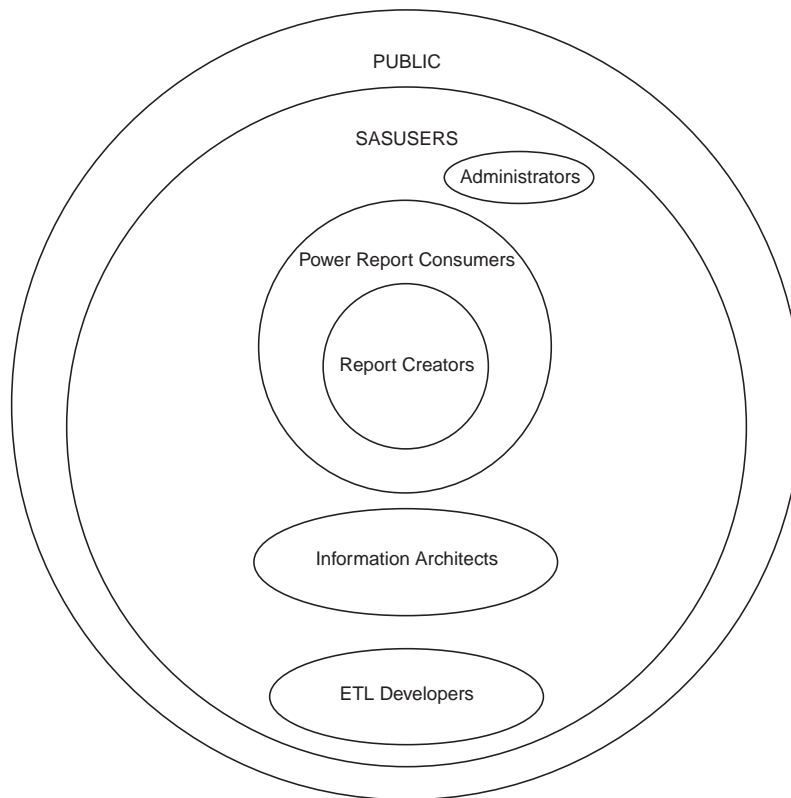
Note: Membership in a user group that is named Administrators does not enable you to perform tasks that require status as an *administrative user* of the metadata server. For details, see “Special Users of the Metadata Server” on page 191. Δ

Defining the Group Structure

Decide how the groups that you create 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.

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 12.1 Example of a Group Structure



This group structure simplifies the process of defining and maintaining access controls. For example,

- If you use the PUBLIC group to represent users who only view unsecured reports using SAS Web Report Studio, then you might not have to create a unique metadata identity for each of these users.
- If you make the Report Creators group a member of the Power Report Consumers group, you simplify the process of giving the report creators a superset of the access that you will give to the report consumers. You can assign permissions that you want both groups to have to the Power Report Consumers group, and the permissions that you want only the creators to have to the Report Creators 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. Δ

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. As explained in "Planning the User Metadata Identities" on page 197, you can import macros or SAS Management Console to create user and group definitions. To create a role for use with a SAS Solutions application, create a group definition and select the **Make this group available as a Role for applications** check box on the General tab of the group definition.

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 installation, you should secure the foundation metadata repository, the repository ACT, and the group definitions that you created. A recommended approach for performing these tasks is provided in "Overview of Implementing Security" on page 213. This section describes a planning process for defining 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 set specific 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, which is described in “Authorization Layers” on page 176. 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.

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 view a live report you must be able to access the generated report, the underlying information maps, the underlying data sources, the XML template that defines the report, and any auxiliary report files such as images. If stored processes are used to generate the report, you must also have the ability to run those stored processes.

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, this task involves reading some information from the source, so some permissions in the data source and operating system authorization layers are usually required. △

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 12.4 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

Task	Foundation Repository	Metadata Object That Describes the Data Source
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 287.

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. Δ

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. A stored process can be associated with an information map or with a report.

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 12.5 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

Task	Repository	Folder That Contains the Stored Process*	Server That Hosts the Stored Process	Stored Process	Data Sources
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 RM permissions 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: Server Administrator's Guide* at support.sas.com/rnd/itech/doc9/admin_oma/security/security_imperspws.html. △

Access Requirements for Working with Information Maps

Users who use SAS Information Map Studio to define and manage information maps must have permissions that enable them to view, create, modify, and delete those maps. An information map is a metadata object that contains a view of one or more data sources, with an added layer of business metadata. Information maps are used as the data sources for reports.

The column headings in the following table are the metadata objects that can be involved when you define and manage information maps. Each table cell contains the permissions to a particular metadata object that are required for a particular task.

Table 12.6 Metadata Authorization Layer Access Requirements for Common Information Map Tasks

Task	Foundation Repository	Folder That Contains the Information Map*	Information Map	Stored Processes	Data Sources
Create and save an information map	RM, WM	RM, WM	not applicable	RM	RM, R**
View an information map	RM	RM	RM	none	none
Edit and overwrite an existing information map	RM, WM	RM	RM, WM	none	R**
Move an information map to another folder	RM, WM	RM, WM	RM, WM	none	none
Rename or delete an information map	RM, WM	RM, WM	RM, WM	none	none
Run queries using an information map	RM	RM	RM	RM	RM, R**

* Users who navigate to an information map must have RM permissions to the folder that contains the information map, and to all of that folder's parent folders. If you cannot see a folder, you cannot browse the objects that the folder contains. Users who access an information map by searching do not have to have RM for the folder that contains the information map.

** You must have read access to the data in order to test a query or set a filter value from the data source. In the current release, the Read permission is not always required because not all applications enforce this permission.

Each information map should be created for a particular set of report creators and report consumers for the following reasons:

- If you attempt to create a report that includes any column to which you do not have access, then the entire report creation fails.
- If you attempt to view a report without having access to all of the underlying data sources, then only those report items (such as tables or graphs) that contain data to which you have access are displayed in the report.

A recommended approach to planning, organizing, and securing information maps is described in "Managing Information Maps" on page 308.

Access Requirements for Working with Reports

Users who view, create, modify, and delete reports must have permissions that enable them to perform those activities. The access requirements for working with reports that comply with the SAS Report Model* specification vary depending on the relationship the report has to its underlying data:

- *Automatically refreshed* reports run queries to get current data every time the report is accessed. Data in an automatically refreshed report is live data.
- *Manually refreshed* reports are generated and cached on a demand basis. Data in a manually refreshed report is static data that can be updated by a user action in the report viewer.
- *Batch* reports are generated and cached on a scheduled basis. Data in a generated batch report is static data that can be updated by a user action in the report viewer.

The column headings in the following table are the metadata objects that can be involved when you work with reports in a repository. Each table cell contains the permissions to a particular metadata object that are required for a particular task.

Table 12.7 Metadata Authorization Layer Access Requirements for Common Report Tasks

Task	Foundation Repository	Folder That Contains the Report*	Report	Stored Processes	Information Maps	Data Source
Create and save a new report	RM, WM	RM, WM	not applicable	RM	RM	RM, R**
View a report	RM	RM	RM	RM	RM	RM, R**
View a batch report	RM	RM	RM	none	none	none
Edit and overwrite an existing report	RM	RM	RM, WM	RM	RM	RM
Move a report to another folder	RM	RM, WM	RM, WM	none	none	none
Delete or rename a report	RM	RM, WM	RM, WM	none	none	none
Render a batch report	RM	RM	RM	RM	RM	RM

* Users who navigate to a report must have RM permissions to the folder that contains the report, and to all of that folder's parent folders. If you cannot see a folder, you cannot browse the objects

* For information about the SAS Report Model, see "The Parts of a Report" on page 311.

that the folder contains. Users who access a report by searching do not have to have permissions to the folder that contains the report.

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

CAUTION:

Even though you have already secured the underlying components (including the data sources, information maps, stored processes), you should also secure your report definitions and your rendered or cached reports. △

It is particularly important to secure batch reports because the viewing of these reports does not require access to the underlying stored processes, information maps, or data sources. Consider the other authorization layers (such as operating system permissions, data source controls, WebDAV controls) when planning security for reports.

Each report should be created for a particular audience of report consumers. If a report consumer attempts to view a report without having access to all of the underlying data items, only those objects (tables, graphs) that contain data to which the report consumer has access are displayed. You can verify the security that you define for a report by

- accessing the report while you are logged in as a member of the report consumers group for which you created the report
- attempting to access the report while you are logged in as a user who should not have access to the report.

A recommended approach to planning, organizing, and securing reports is described in “Overview of Reporting in the SAS Intelligence Platform” on page 306.

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.

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 12.8 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

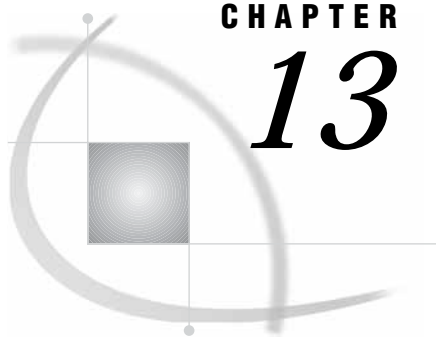
- Use other authorization layers, such as operating system permissions and relational database controls, to secure data.
- Understand how permissions are evaluated in the metadata authorization layer. Remember that this layer supports multiple inheritance, and that the inheritance rules make it much 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 report or information maps from one folder to another might change the effective access controls for that object.
- 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 reports, information maps, and stored processes.
- 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.

- Consider the balance between deployability, usability, maintainability, and security that you selected for your environment.



CHAPTER 13

Implementing Security

<i>Overview of Implementing Security</i>	213
<i>Protecting the Foundation Repository</i>	214
<i>Setting Up Security for Administrators</i>	215
<i>Securing ACTs and User-Defined Groups</i>	217
<i>Using an ACT to Secure ACTs and Group Definitions</i>	218
<i>Setting Up Security for Regular Users</i>	218
<i>SAS Information Delivery Portal Requirements</i>	220
<i>Example: Set Up Security for Regular Users</i>	221
<i>Security Goals and Configuration</i>	221
<i>Implementation Process</i>	222
<i>Security Maintenance Activities</i>	225
<i>Removing Users</i>	225
<i>Updating Passwords</i>	225
<i>Updating Passwords That Are Included in Configuration Files</i>	225
<i>Adding Users to a Deployment</i>	226
<i>Importing User Information</i>	227
<i>Controlling Access to Resources</i>	228
<i>Managing Access to Server Definitions</i>	229
<i>Managing Access to OLAP Data</i>	229
<i>Managing Authentication Domains</i>	230
<i>Managing Authentication for Added Servers</i>	231
<i>Example: Managing Authentication for an Oracle Server</i>	231

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 2 of this guide. For an overview of all of the security related aspects of a deployment, see “Guide to Security Administration Activities” on page 39.

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. Δ

- 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.

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.
- 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, create metadata that describes data, or log on to the SAS Information Delivery Portal. Typically, you give either the SASUSERS group or selected user-defined groups WriteMetadata access to the repository.

Note: For more information about the repository ACT and the actions that it controls, see "Repository Level Access Controls" on page 185. Δ

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: You can 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. Δ

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

- 1 Log on 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.

CAUTION:

Do not modify or remove the permissions that were granted to the SAS Administrator and the SAS System Services group as part of the initial installation. △

- 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. △

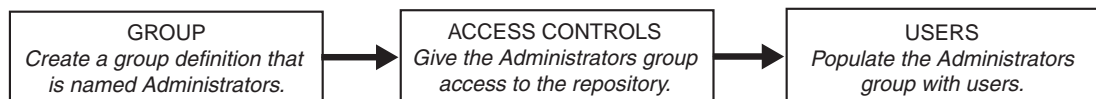
At this point, only the *unrestricted user* account (SAS Administrator) can access the repository. With the repository well-protected, you can begin setting up security for administrators first and then for regular users.

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 13.1 Sequence for Setting Up Security for Administrators



The process consists of the following steps:

- 1 Log on 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. Δ

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. Δ

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.) Δ

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

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

- a In the navigation panel, select **Environment Management** \blacktriangleright **Authorization Manager** \blacktriangleright **Access Control Templates** \blacktriangleright **Default ACT**.

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

- 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). Δ

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 217.

7 Secure the Administrators group definition with directly assigned permissions. The goal is to enable only members of the Administrators group to modify or 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 217.

* 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.

- 8 Add individual administrators to the deployment by following the instructions in “Adding Users to a Deployment” on page 226.
- 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 214. △

- 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. Δ

- 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. Δ

- 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:

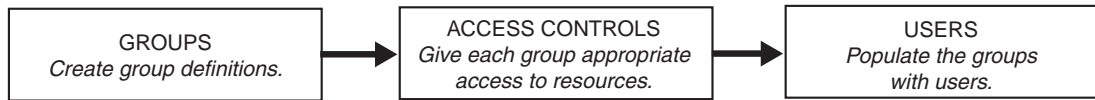
- 1 Create one ACT that has the desired identity/permission pattern on its Users and Permissions tab.
- 2 Apply that ACT on the Authorization tab of every group definition and ACT that you create.

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 *SAS Management Console: User's Guide* in SAS OnlineDoc.

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. Δ

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 13.2 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).

Most environments 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 201. 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 on 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: 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 194. Δ

- 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 203.
 - 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 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 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 217.
 - 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, on the Users and Permissions tab of the repository ACT, you might set these permissions:
 - grant ReadMetadata and WriteMetadata to SASUSERS
 - grant Read to a user group that you created to manage users who interact with OLAP data.
 - 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 228.
 - 8 Add individual users to the deployment by following the instructions in “Adding Users to a Deployment” on page 226.

SAS Information Delivery Portal Requirements

The SAS Guest and the SAS Web Administrator must have ReadMetadata and WriteMetadata access to the repository. You can meet this requirement by doing either of these things:

- On the repository ACT, grant ReadMetadata and WriteMetadata permissions to the SASUSERS group. This meets the requirement because the SAS Guest and the SAS Web Administrator are members of the SASUSERS group.
- Add the SAS Guest and the SAS Web Administrator to the user group that you are using to manage your portal users. This meets the requirement because any portal users group must have ReadMetadata and WriteMetadata access to the repository.

Example: Set Up Security for Regular Users

This example demonstrates how to set up regular users in a diverse 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 214.
- Set up security for your administrators as described in “Setting Up Security for Administrators” on page 215.

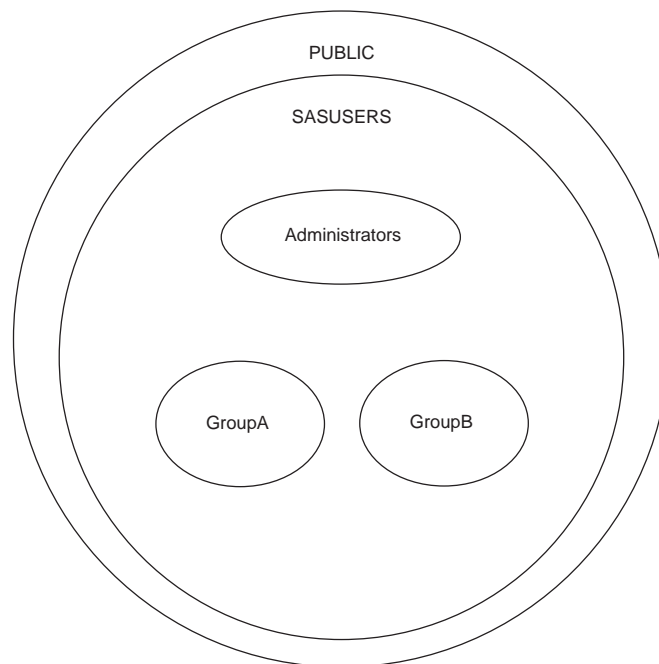
Security Goals and Configuration

These are the security goals for this example:

- 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:

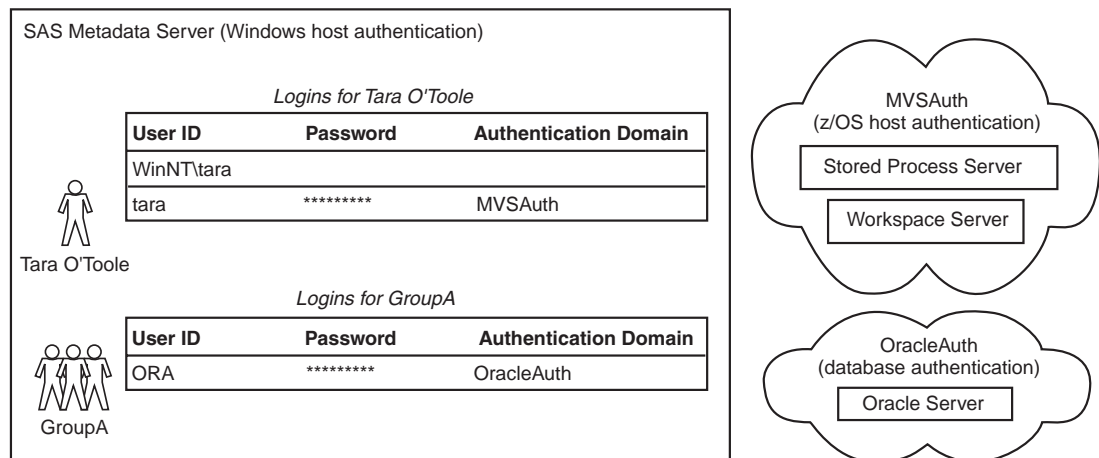
Figure 13.3 Group Structure



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 13.4 Servers, Authentication Domains, and Logins



Implementation Process

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

- 1 Log on 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 217.

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 "Defining Users for Host Authentication" in the *SAS Integration Technologies: Server Administrator's Guide* at support.sas.com/rnd/itech/doc9/admin_oma/security/auth/security_impauthhost.html. △

- 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 to enable the metadata server to determine the user's metadata identity. If this login is used only for the purpose of determining the user's metadata identity, then this login does not have to include a password or specify an authentication domain. In the figure, 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 ongoing security 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 on 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.

Updating 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 update his or her own passwords using either SAS Management Console or the SAS Personal Login Manager application.

Note: In order to use the SAS Personal Login Manager, you must have an account with the metadata server's authentication provider. △

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.

Note: You can use your configuration.properties file to determine which user accounts your deployment has been configured to use. For example, if your

* For information about configuration directories, see "Understanding the State of Your System" on page 130.

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. Δ

After you update 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.

Note: If no occurrences of the user ID are found, then the account that you updated is not referenced in your configuration files, so no password updates to those files are necessary. Δ

- 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 on to SAS Management Console as an *unrestricted user* (such as the SAS Administrator) in order to reset a password for another user. When you are logged in to SAS Management Console as an *unrestricted user*, all rows on every Logins tab contain asterisks in the Password column—even if no password has been specified. Δ

- 5 If your deployment includes any SAS Web applications, you must perform some additional tasks. Detailed instructions are provided in "Changing Passwords for User or Group Credentials" in the *SAS Web Infrastructure Kit: Administrator's Guide* at support.sas.com/rnd/itech/doc9/portal_admin/security/ag_changepass.html.

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 153.

- 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 156.

Note: On Windows platforms, assign user rights to these accounts as described in “Defining Users for Host Authentication” in the *SAS Integration Technologies: Server Administrator’s Guide* at support.sas.com/rnd/itech/doc9/admin_oma/security/auth/security_impauthhost.html. △

- 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 and Administration Guide* at support.sas.com/rnd/eai/openmeta/v9/setup/txtfiles.html. △

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

- Log on 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 199.
- 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 "Creating and Maintaining User and Group Definitions" in the *SAS Metadata Server: Setup and Administration Guide* at support.sas.com/rnd/eai/openmeta/v9/setup/authmacros.html.

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** list box.

When you select an identity in the **Names** list 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 Access to Server Definitions

The members of the SAS System Services group are used to connect to various servers, so these identities must be able to access the configuration information that is stored in server definitions. During installation, the SAS System Services group is granted ReadMetadata permission to the repository. This gives the SAS System Services group ReadMetadata access to all objects in the repository, including the server definitions.

Do not block the SAS System Services group's access to any server definition. If you choose to limit access to a logical server definition, you might need to set an additional access control to preserve the SAS System Services group's access. For example, you might set these direct access controls on the Authorization tab of an OLAP server definition:

- deny the PUBLIC group ReadMetadata access to the server definition
- grant ReadMetadata permission back to a user group that accesses data on that server.

The denial of ReadMetadata permission to PUBLIC that you set directly on the server definition overrides the grant to SAS System Services that comes from the repository ACT, so the SAS System Services group will not be able to obtain configuration information about the SAS OLAP Server from the metadata server.

You can remedy this situation by adding a direct grant of ReadMetadata permission to the permissions list for SAS System Services on the Authorization tab for the server definition. In the **Names** list box, select SAS System Services. In the permissions list, the group's repository ACT grant of ReadMetadata permission is indicated by a checked box with a gray background. To add a direct grant on top of the repository ACT grant, select the check box. The gray background is removed and the check box is still selected. This indicates that the SAS System Services group now has a direct grant of ReadMetadata permission to the server definition.

Note: Because the SAS Object Spawner attempts to read server definitions only during initialization, you must stop and restart the spawner after making these changes. △

Managing Access to OLAP Data

Only those schemas and cubes that the SAS OLAP Server can see can be made available to requesting users. The connection from the SAS OLAP Server to the metadata server is owned by the SAS Trusted User, so that user must have ReadMetadata access to the schema and cube definitions that are stored in the metadata repository. By default, the SAS Trusted User has the necessary access as a member of the SAS System Services group.

Do not block the SAS System Services group's access to any OLAP schema or cube definitions. If you choose to limit access to an OLAP cube or schema, you must ensure that the SAS System Services group retains its ReadMetadata access to these objects. The process is identical to that described in the preceding section.

In order to access OLAP data, a user must have both ReadMetadata and Read permission to each cube that the user will access. Typically, the user has ReadMetadata

through membership in an implicit group (PUBLIC or SASUSERS) that has ReadMetadata permission on the repository ACT. To give the user the necessary read access, you can do either of these things:

- On the Users and Permissions tab of the repository ACT, grant Read permission to a user group to which the user belongs.

Note: This is a liberal approach because it creates a default grant of Read permission for all SAS OLAP data and for any data that is accessed by the SAS Metadata LIBNAME Engine. △

- On the Authorization tab of each cube (or schema) that the user will access, grant Read permission to a user group to which the user belongs.

The SAS OLAP Server enforces the Read permission for cubes, dimensions, hierarchies, and levels. The SAS OLAP Server also enforces direct grants of the Read permission for a subset of the members within a dimension. For more information, see ***SAS OLAP Server Administrator's Guide*** ► **Installing and Configuring SAS OLAP Server** ► **Securing Cubes** ► **Permission Condition for Dimensions** in SAS OnlineDoc.

Managing Authentication Domains

During installation, all servers are assigned to a single authentication domain. If your deployment includes multiple operating systems, alternative authentication providers, or third-party database systems, then you might need to create additional authentication domains. For each authentication domain, you must define associations to the appropriate servers and logins.

You can create a new authentication domain while you are defining a login or a server in SAS Management Console (instead of selecting an existing authentication domain, click [New](#) to access the New Authentication Domain window). Before you create a new authentication domain, you should understand how authentication domains are used. You should also review your existing authentication domains to verify that you do need to add one. The name of an authentication domain should be meaningful to the people who create the logins and server definitions that will be associated with that authentication domain.

There is no direct method for deleting an authentication domain from SAS Management Console.

In order to modify existing authentication domain assignments, you need to know how to locate those assignments in SAS Management Console.

- The authentication domain for a logical server is specified on the Options tab of each of the server's connection definitions. For example, to access the authentication domain assignment for a SAS OLAP Server, you will select a path such as: **Server Manager** ► **SASMain** ► **SASMain Logical OLAP Server** ► **SASMain OLAP Server** ► **Connection: SASMain OLAP Server** ► **Properties** ► **Options**.
- The authentication domain for a login is specified on the Logins tab of the user or group definition to which the login is assigned. You can see a login that is assigned to another user only when you log on to SAS Management Console as an *unrestricted user*.

Note: "Authentication Concepts and Terminology" on page 148 explains the relationships between authentication domains, servers, and logins. "Examples: Using Authentication Domains" on page 161 illustrates authentication domains for several different environments. △

Managing Authentication for Added Servers

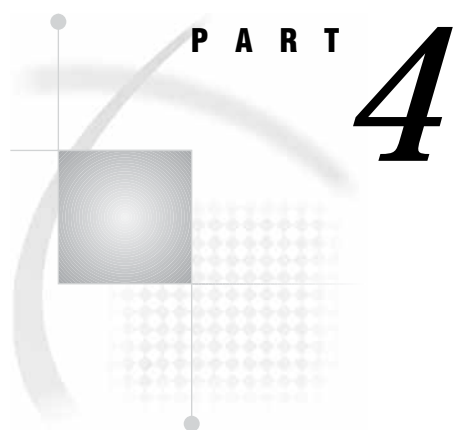
For each additional server that you register in the repository, complete the following steps to support authentication:

- 1 Associate the new server with an appropriate authentication domain, or create a new authentication domain for the server if necessary.
- 2 Set up metadata that enables your users to access the new server. In most cases, this involves making sure that exactly one login that contains credentials for accessing the new server is available to every user who will access that server.

Example: Managing Authentication for an Oracle Server

To manage authentication to an Oracle server that is using database authentication, you must perform these tasks:

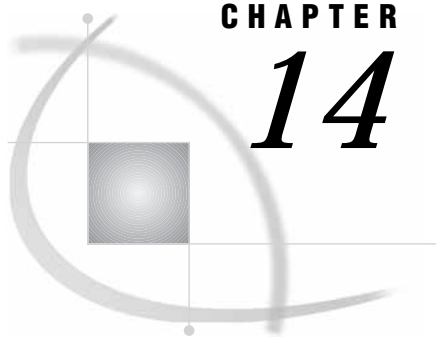
- 1 Create a new authentication domain for the Oracle server when you register that server in the metadata. The new authentication domain is necessary because users do not access the Oracle server with the same credentials they use for any other server in the deployment.
- 2 Define metadata that will enable your users to access the Oracle server. For example
 - If you use shared accounts on the Oracle server, then you must define a user group in the metadata for each shared account. You assign one login for the Oracle server to each of these user groups. You then assign all of the users who will share a particular account to the user group that owns the login for that account.
 - If you use individual accounts on the Oracle server, then you must add an Oracle login to each user definition.



Data Administration

Chapter 14 **Preparing Data for Use** 235

Chapter 15 **Optimizing Data Storage** 257



CHAPTER

14

Preparing Data for Use

<i>Overview of Preparing Data for Use</i>	235
<i>Understanding the Data Storage Options</i>	236
<i>Default SAS Storage</i>	236
<i>Third-Party Relational Data Storage</i>	236
<i>Parallel Storage</i>	237
<i>Symmetric Multiprocessing</i>	238
<i>Multidimensional Storage</i>	238
<i>Defining Metadata about the Data</i>	239
<i>Preliminary Tasks</i>	239
<i>Defining a Database Server</i>	240
<i>Defining a Database Schema</i>	241
<i>Defining a Library</i>	242
<i>Defining Data Sources with a Source Designer</i>	243
<i>Defining Data Sources with SAS Management Console</i>	244
<i>Defining Data Sources with the Metadata LIBNAME Engine</i>	245
<i>Creating Metadata for an Existing Data Source</i>	246
<i>Preparing for Cube Loading</i>	248
<i>Preliminary Tasks</i>	248
<i>Making Cubes Available to a SAS OLAP Server</i>	248
<i>Defining a New OLAP Schema with SAS OLAP Cube Studio</i>	249
<i>Editing a SAS OLAP Server Definition to Change the Schema Assignment</i>	252
<i>Ensuring That Tables Are Accessible at Query Time</i>	252
<i>Testing a SAS Workspace Server Connection with SAS OLAP Cube Studio</i>	253
<i>Securing Access to the Metadata That Defines the Data</i>	253

Overview of Preparing Data for Use

In a SAS Intelligence Platform 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) and the SAS Scalable Performance Data Server (SPD Server)

Note: The SAS SPD Engine is available with Base SAS software. The SAS SPD Server is an additional, individually licensed storage mechanism. The SAS SPD Server is a stand-alone client/server product that provides much of the functionality of the SAS SPD Engine plus additional features. For more

information, see the SAS SPD Server documentation at support.sas.com/rnd/scalability/spde/. △

- 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 that defines 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” on page 236
- “Third-Party Relational Data Storage” on page 236
- “Parallel Storage” on page 237
- “Multidimensional Storage” on page 238.

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 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 15, “Optimizing Data Storage,” on page 257. Δ

For more information about using the SAS/ACCESS interfaces, see *SAS/ACCESS for Relational Databases: Reference*.

Parallel Storage

Both the SAS Scalable Performance Data Engine (SPD Engine) and the SAS Scalable Performance Data Server (SPD Server) are designed for high-performance data delivery. They enable rapid access to SAS data for intensive processing by the application. The SAS SPD Engine and SAS SPD Server deliver data to applications rapidly by organizing the data into a streamlined file format that takes advantage of multiple CPUs and I/O channels to perform parallel input/output functions.

The SAS SPD Engine is included with Base SAS software. It is a single-user data storage solution that shares the high-performance parallel processing and parallel I/O capabilities of SAS SPD Server, but lacks the additional complexity of a full-blown server. The SAS SPD Server is available as a separate product or as part of the SAS Intelligence Storage bundle. It is a multi-user parallel-processing data server with a comprehensive security infrastructure, backup and restore utilities, and sophisticated administrative and tuning options. The SAS SPD Server libraries can now be defined using SAS Management Console. For more information, see the *SAS Management Console: User's Guide*.

The SAS SPD Engine and SAS SPD Server use threads to read blocks of data very rapidly and in parallel. The software tasks are performed in conjunction with an operating system that enables threads to execute on any of the machine's available CPUs.

Although threaded I/O is an important part of both product offerings' functionality, their real power comes from the way that the software structures SAS data. They can read and write partitioned files and, in addition, comprise a new file format. This data structure permits threads, running in parallel, to perform I/O tasks efficiently.

Although not intended to replace the default Base SAS engine for most tables that do not span volumes, SAS SPD Engine and SAS SPD Server are high-speed alternatives for processing very large tables. They read and write tables that contain 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 SAS SPD Engine and SAS SPD Server performance are 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 SAS SPD Engine runs on UNIX, Windows, z/OS (on HFS and zFS file systems only), and OpenVMS Alpha (on ODS-5 file systems only) platforms. The SAS SPD Server runs on Tru64 UNIX, Windows Server, HP-UX, and Sun Solaris platforms.

Symmetric Multiprocessing

The SAS SPD Engine exploits a hardware and software architecture known as symmetric multiprocessing (SMP). An SMP machine has multiple CPUs and an operating system that supports threads. An SMP machine is usually configured with multiple controllers and multiple disk drives per controller. When the SAS SPD Engine reads a data file, it launches one or more threads for each CPU; these threads then read data in parallel from multiple disk drives, driven by one or more controllers per CPU. The SAS 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 SAS 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. They are derived from source data such as SAS tables, SAS SPD Engine tables, and SAS/ACCESS database tables by using tools such as the Cube Designer wizard. The Cube Designer 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 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*.

A project 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 79.

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 Platform applications such as SAS ETL Studio to be able to use SAS tables, SAS 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 248. △

For information about the SAS Management Console tasks that are 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 project installation, a SAS Metadata Server, a SAS 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 project installation creates start-up scripts. The installation process also creates a foundation repository.

Note: For information about a project installation and configuration, see Chapter 7, “Installing and Configuring Your Software,” on page 79. △

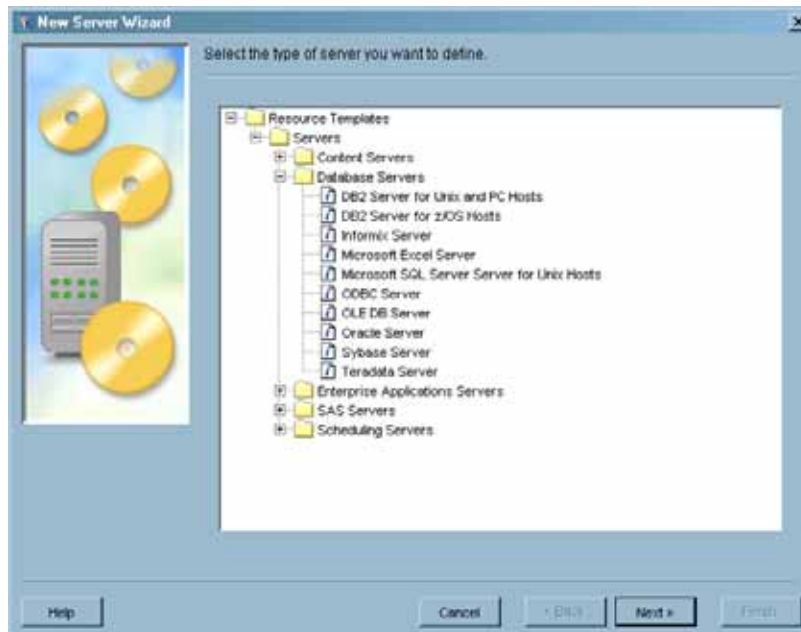
Defining a Database Server

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 on to 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 on to the SAS Metadata Server that contains the SAS Metadata Repository connection that you want to use.
- 2 In the SAS Management Console navigation tree, select **Environment Management ► Server Manager**
- 3 To display the first wizard window, select **Actions ► New Server**



- 4 Select 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

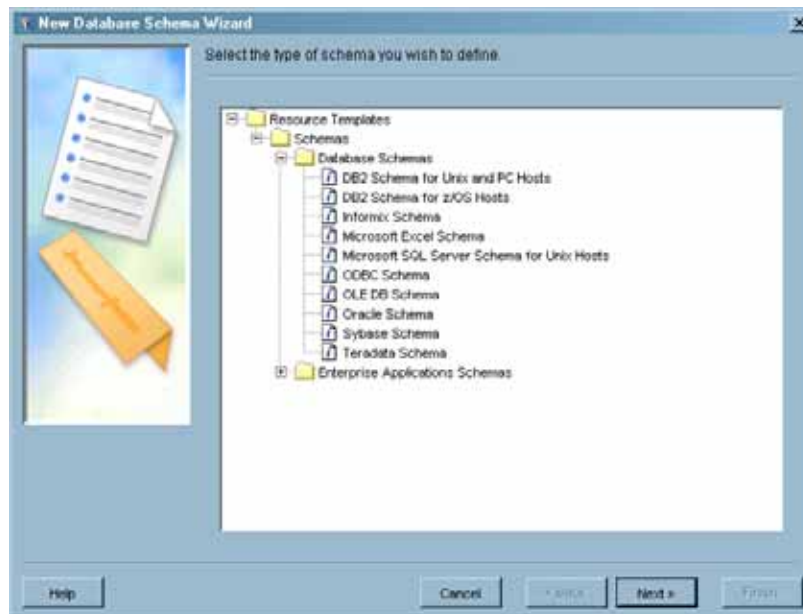
A database schema is a pointer to an existing schema, which is a map or model of the structure of a database. Before you can define a database schema, you must have a database server of the same type. In addition, both the database server and the database schema definitions are required in order to define 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 project 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 on to the SAS Metadata Server that contains the SAS Metadata Repository connection that you want to use.
- 2 In the SAS Management Console navigation tree, select **Environment Management ► 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 240 and “Defining a Database Schema” on page 241. Δ

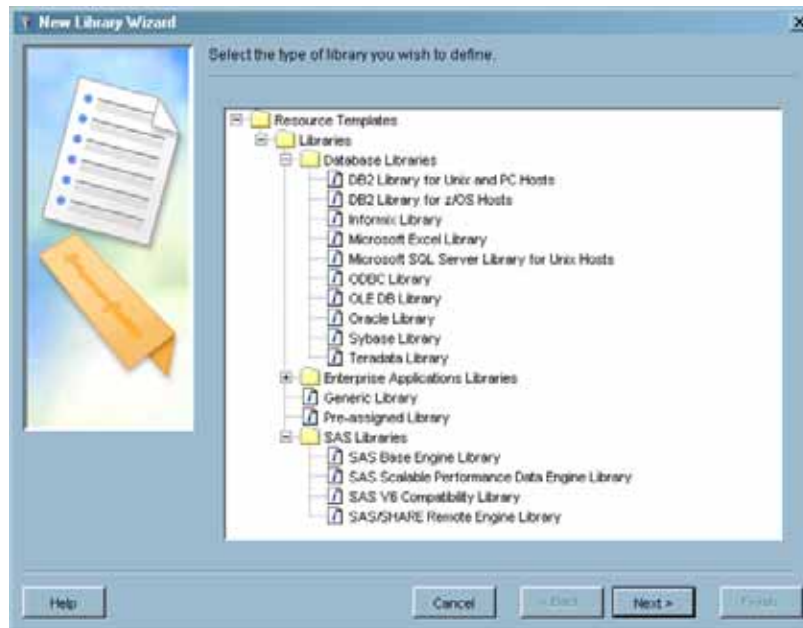
Note: In SAS ETL Studio and SAS OLAP Cube Studio, users can launch the New Library wizard from a source designer wizard or the Cube Designer wizard, so that they can define a new library when they are defining a data source or creating a cube. Δ

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 project installation, a foundation metadata repository is created. When the foundation repository is created, all available resource templates are also loaded into the repository. Δ

To launch the New Library wizard from SAS Management Console, complete these steps:

- 1 Use your metadata profile to log on to the SAS Metadata Server that contains the SAS Metadata Repository connection that you want to use.
- 2 In the SAS Management Console navigation tree, select **Environment Management** \blacktriangleright **Data Library Manager** \blacktriangleright **SAS Libraries**
- 3 To display the first wizard window, select **Actions** \blacktriangleright **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 15, “Optimizing Data Storage,” on page 257.

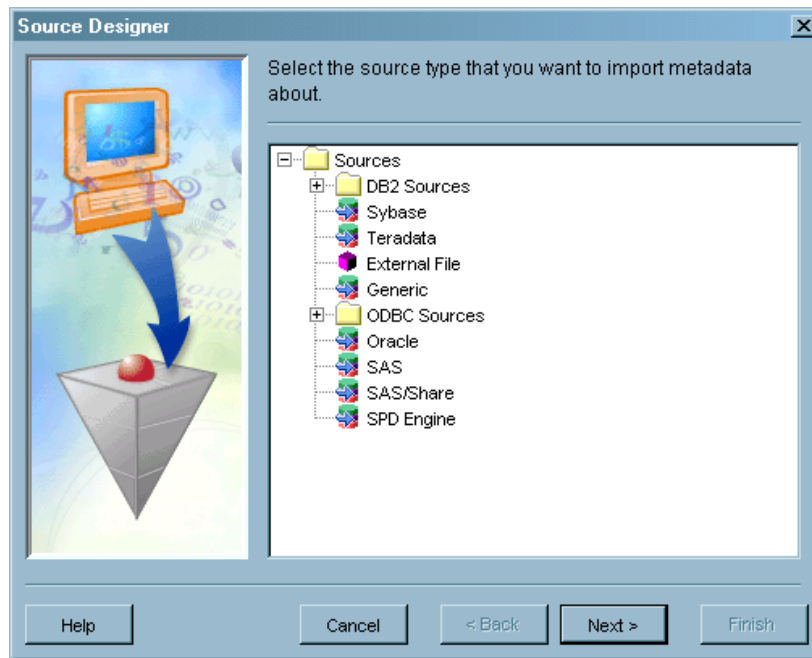
For more information about how to define a library, including how to pre-assign a library to a server, see the *SAS Management Console: User's Guide*.

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 on 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.

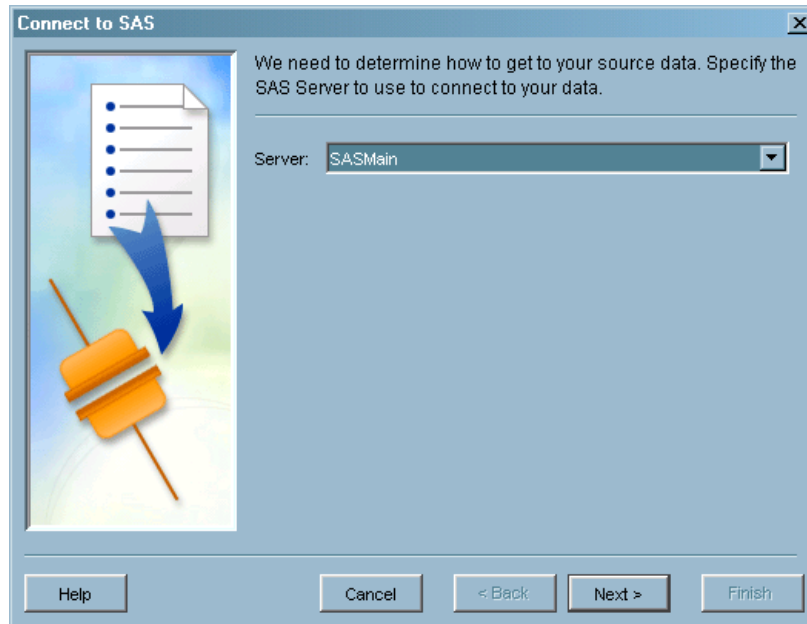
For more information about using source designers in SAS ETL Studio, see the SAS ETL Studio Help, which is available from within the product. For more information about using source designers in SAS OLAP Cube Studio, see the SAS OLAP Cube Studio Help, which is available from within the product.

Defining Data Sources with SAS Management Console

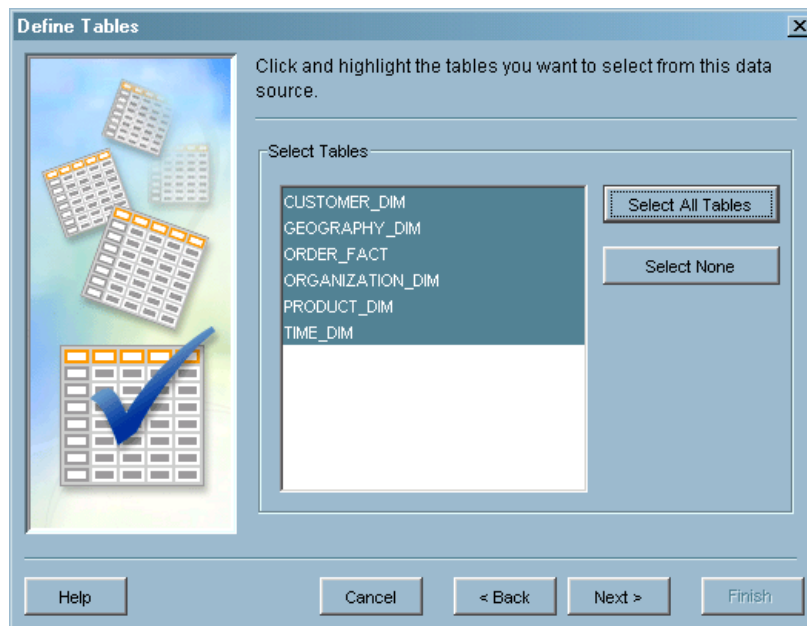
SAS Management Console has an Import Tables feature that allows you to import table definitions from external sources. It is included as part of the Data Library Manager plug-in to SAS Management Console. This feature uses the same code as the SAS ETL Studio Source Designer but has a different flow. The Import Tables feature is for users who do not have SAS ETL Studio. SAS ETL Studio users should continue to use the SAS ETL Studio Source Designer, which supports the grouping of tables and the use of Data Surveyors, neither of which is supported by SAS Management Console.

To define data sources with SAS Management Console, you will need to have defined a library (see “Defining a Library” on page 242). Then do the following.

- 1 In the SAS Management Console navigation tree, beneath **Data Library Manager ► SAS Libraries** select the library that you just created.
- 2 Right click on that library’s name and click **Import Tables**.
- 3 On the Connect to SAS window, select the correct SAS server. Click **Next**.



- 4 On the Select a SAS Library window, click **Next**.
- 5 On the Define Tables window, select those tables you want to define, then click **Next**.



- 6 On the Wizard Finish window, review the selections that you have made. If the values are correct, click **Finish**.

Defining Data Sources with the Metadata LIBNAME Engine

If you are a SAS ETL Studio or SAS OLAP Cube Studio user, you should use a source designer (which is described in “Defining Data Sources with a Source Designer” on page 243) to define metadata about your data sources. Otherwise, use SAS Management

Console (which is described in “Defining Data Sources with SAS Management Console” on page 244). Alternatively, you can use the metadata LIBNAME engine. You can do this by writing SAS code that reads the data source and writes the table metadata.

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
    metaserver=myip.us.org.com port=6401
    metaprotocol=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, applying rules and security to the data based on the metadata.

In order to interact with data that is accessed using the metadata LIBNAME engine, users must have all of the necessary metadata layer permissions, including the permissions that the metadata LIBNAME engine enforces (Read, Write, Create, and Delete). For example, in order to view data in a table that is accessed using the metadata LIBNAME engine, a user must have both the ReadMetadata permission and the Read permission to that table. For more information about the metadata LIBNAME engine and its enforcement of permissions, 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 204. \triangle

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
  metaserver=myip.us.org.com port=8561
  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 METAREPOSITORY="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. You also can 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.

Here is another example. It creates metadata about a SAS table, rather than an Oracle table, and it uses the LIBRARY= and METAREPOSITORY= options:

```

libname banking 'c:\FinancialData';

libname finance meta library=mlelib metarepository=foundation
  userid=metaid pw=metapw
  metaserver=myip.us.org.com port=8561
  metaprotocol=bridge (metaout=meta);

data finance.Sales ;
  set banking.Sales (obs=0);
run;

```

You also can specify metadata connection information with SAS system options.

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 248.
- 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 on 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. \triangle

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 253.

During a project installation, a SAS Metadata Server, a SAS 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 project installation creates start-up scripts. The installation process also creates a foundation metadata repository.

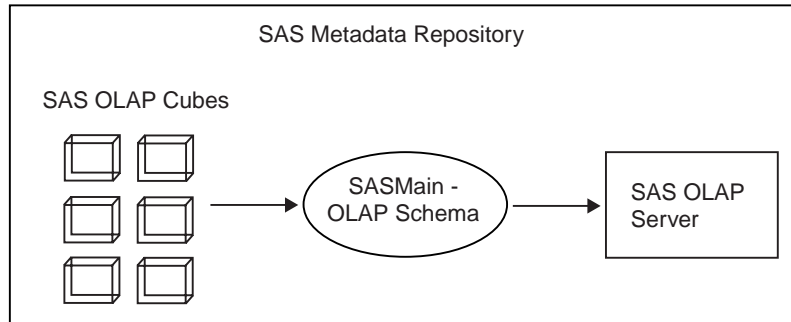
Note: For information about project installations, see Chapter 7, “Installing and Configuring Your Software,” on page 79. \triangle

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.

Display 14.1 Cubes Are Assigned to an OLAP Schema Which Is Associated with a SAS OLAP Server Metadata Definition



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.

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:

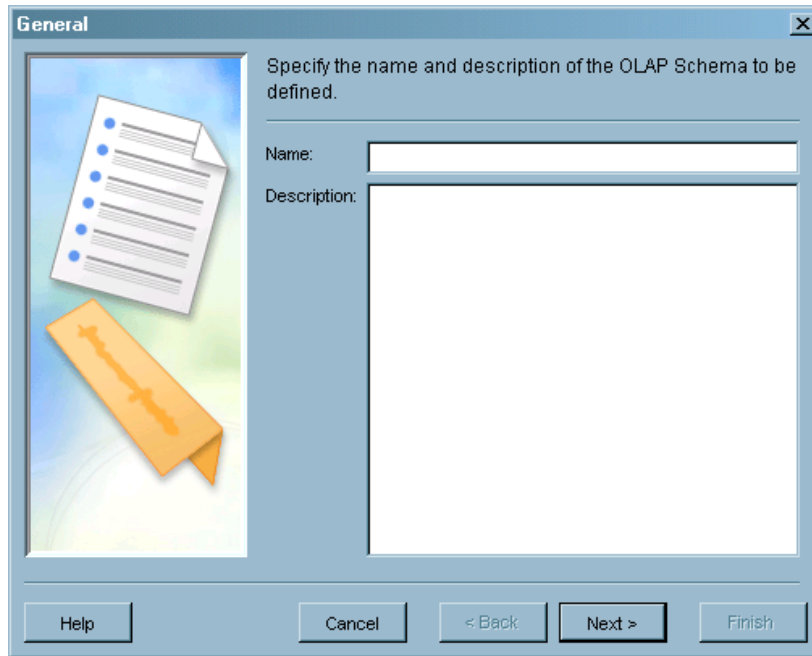
- In SAS OLAP Cube Studio, create a new OLAP schema with the OLAP Schema wizard. You can assign the new schema to one or more servers that are available in the current repository and in any repositories that depend on the current repository (see “Defining a New OLAP Schema with SAS OLAP Cube Studio” on page 249).
- 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: 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. Δ

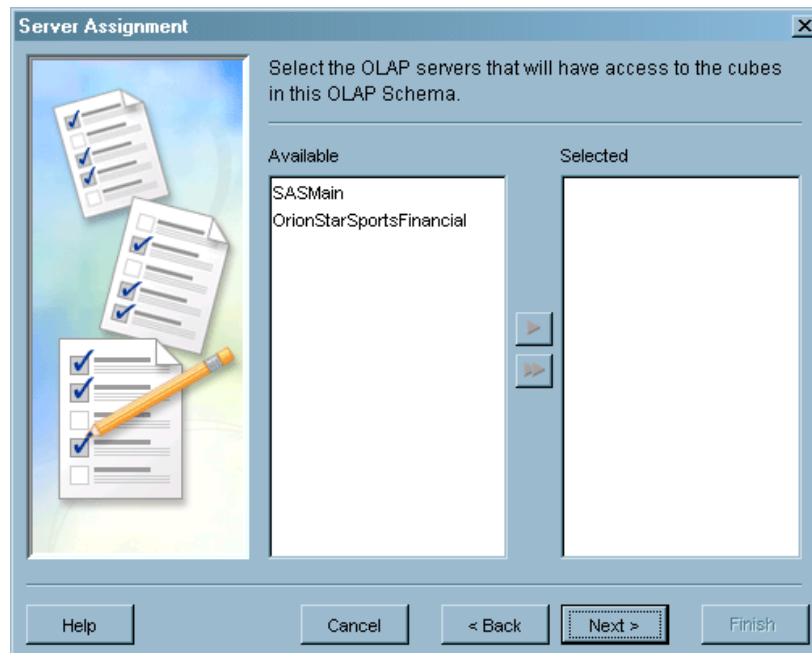
Defining a New OLAP Schema with SAS OLAP Cube Studio

To define a new schema with SAS OLAP Cube Studio, complete these steps:

- 1 Use your metadata profile to log on 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**.



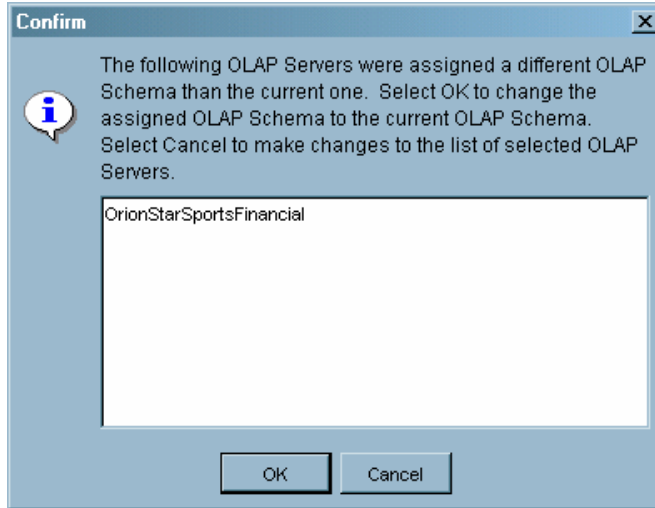
- 4 On the Server Assignment window, specify the OLAP servers that should have access to the group of cubes assigned to the OLAP schema. The **Available** list box lists the OLAP servers defined in the current metadata repository and in any repositories that are dependent on the current repository. To add a server to the schema, select server names in the **Available** list box and move them to the **Selected** list box.



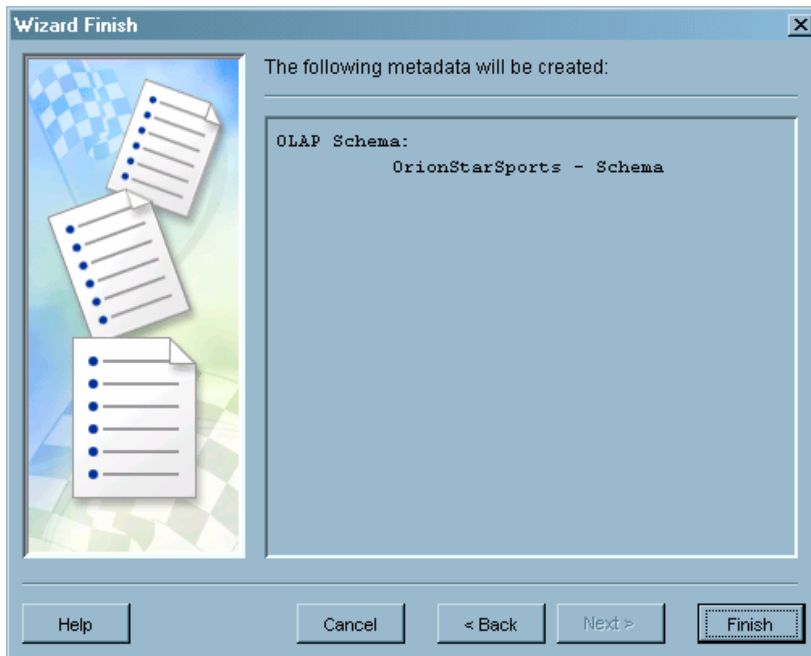
Click **Next** to display the Finish window.

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.

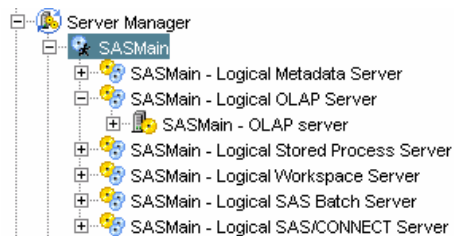


To change a selection on a previous window, click **Back**. Otherwise, click **Finish** to save the new schema and exit the wizard.

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 on 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** \blacktriangleright **Server Manager**
- 3 Select a SAS application server that contains the SAS OLAP Server that you want to edit.



- 4 Select **File** \blacktriangleright **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 249. From within SAS Management Console, the steps are the same, except that the Server Assignment window does not appear. \triangle

- 7 Click **OK** to save your changes and exit the dialog box.

Ensuring That Tables Are Accessible at Query Time

Data that is external to a cube must be available to the SAS OLAP Server under the following conditions:

- If the cube does not include an NWAY, then the SAS OLAP Server must have access to the input data source table (also called the detail data) and any specified dimension tables.
- If the cube is associated with a drill-through table, then the SAS OLAP Server must have access to the drill-through table.
- If the cube uses pre-summarized aggregation tables, then the SAS OLAP Server must have access to those tables.

To ensure that the necessary tables are accessible at query time, the applicable LIBNAMEs need to be allocated when the SAS OLAP Server that is associated with the OLAP schema that contains the cubes is invoked. For information about how to pre-assign libraries to servers, see the *SAS Management Console: User's Guide*.

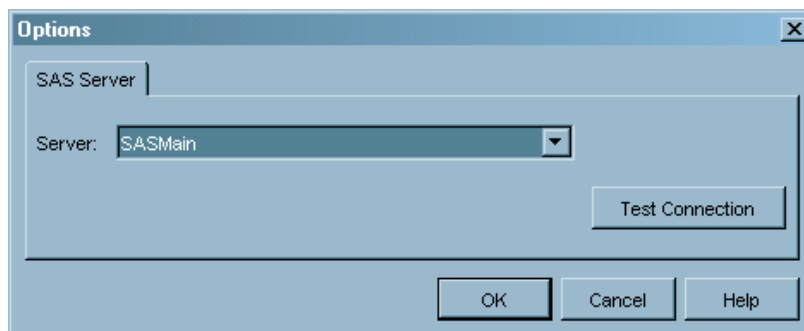
Note: If any of the tables contain user-defined formats, then the SAS OLAP Server also needs information about how to find those formats. For more information, see “Working with User-Defined Formats” on page 139. User-defined formats cannot be used with drill-through tables. \triangle

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 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 on 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**.

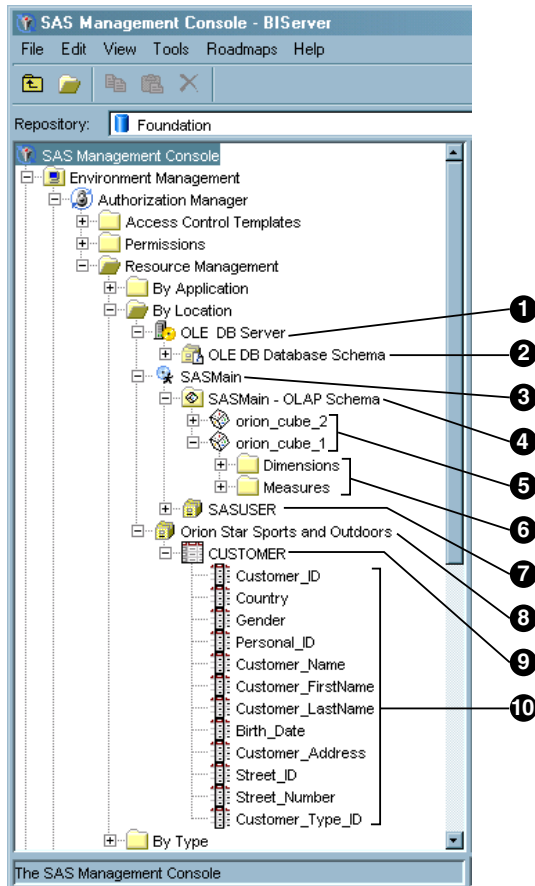
Note: If the Log On To dialog box appears, enter your **User Name** and **Password**, then click **OK**. △
- 5 If the SAS Workspace Server successfully connects to the SAS Metadata Server, then you will see the message “Connection to the server was successful!” Click **OK** in the message box.

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 on to the SAS Metadata Server that contains the SAS Metadata Repository connection that you want to use.
- 2 Select **Environment Management ► Authorization Manager ► Resource Management ► By Location**

Display 14.2 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 Platform 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. Δ

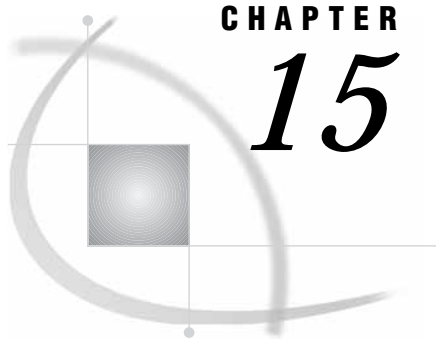
- 10 Columns in a data source

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 182
- “Inheritance in Relational Database Data” on page 182

- “Inheritance in OLAP Data” on page 183.

Note: Because all SAS Intelligence Platform applications use the metadata server when accessing resources, permissions that are enforced by the metadata server provide the strongest protections that are available in the metadata authorization layer. For more information, see “Planning Your Access Controls” on page 204. Δ



CHAPTER

15

Optimizing Data Storage

<i>Overview of Optimizing Data Storage</i>	257
<i>Compressing Data</i>	258
<i>Indexing Data</i>	259
<i>Sorting Data</i>	261
<i>Multi-Threaded Sorting</i>	262
<i>Sorting a Database Table</i>	262
<i>Buffering Data</i>	263
<i>Base SAS Tables</i>	263
<i>DB2 (UNIX and PC), ODBC, OLE DB, Oracle, SQL Server, and Sybase Tables</i>	263
<i>Using Threaded Reads</i>	264
<i>Building Cubes from Star Schemas</i>	265
<i>Validating SPD Engine Hardware Configuration</i>	265
<i>Building Optimized Cube Aggregations</i>	265
<i>Optimizing Performance of a SAS OLAP Server</i>	269
<i>Setting Caching Options for the SAS OLAP Server</i>	269
<i>Setting Server Options for the SAS OLAP Server</i>	270
<i>Setting Performance Options for the SAS OLAP Server</i>	270
<i>Capturing SAS OLAP Server Performance Information</i>	270
<i>Setting LIBNAME Options That Affect Performance</i>	271
<i>Setting LIBNAME Options That Affect Performance of SAS Tables</i>	271
<i>Setting LIBNAME Options That Affect Performance of SAS/ACCESS Databases</i>	272
<i>Setting LIBNAME Options That Affect Performance of SPD Engine Tables</i>	275

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 14, “Preparing Data for Use,” on page 235) 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 258
- “Indexing Data” on page 259
- “Sorting Data” on page 261
- “Buffering Data” on page 263
- “Using Threaded Reads” on page 264

- “Building Cubes from Star Schemas” on page 265
- “Validating SPD Engine Hardware Configuration” on page 265
- “ Building Optimized Cube Aggregations” on page 265
- “Optimizing Performance of a SAS OLAP Server” on page 269
- “Setting LIBNAME Options That Affect Performance” on page 271

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.

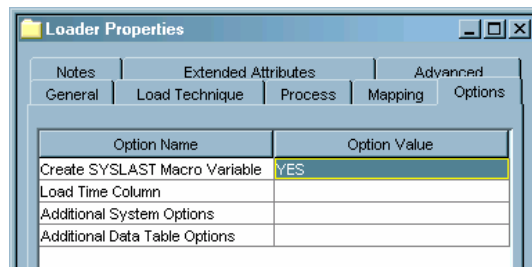
These are the types of compression that you can specify:

- 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 15.1 The Options Tab in a Loader Properties Dialog Box in SAS ETL Studio



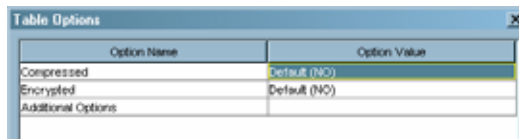
- all tables for a particular SAS data library. For example, when you define a Base SAS engine library in the metadata, you can specify the COMPRESS= option in

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 271). 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 272).

Note: You cannot specify compression for a SPD Engine data library. Δ

- 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 15.2 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). Δ

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:

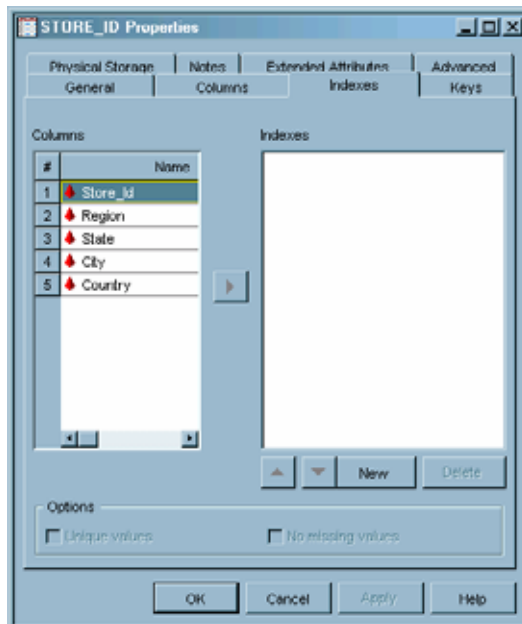
- a simple index, which consists of the values of one column.
- a composite index, which consists of the values of more than one column, with the values concatenated to form a single value.

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.

In addition to 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 15.3 The Indexes Tab in the Properties Dialog Box for a Table Named STORE_ID



In general, SAS can use an index to improve performance in these situations:

- For cube loading, a composite index on the columns that make up the cube's hierarchies might provide best results.
- For WHERE processing, an index can provide faster and more efficient access to a subset of data. Note that to process a WHERE expression, SAS decides whether to use an index or to read the table sequentially.

Note: For WHERE processing, the Base SAS engine uses a maximum of one index. The SPD Engine can use multiple indexes. Δ

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.

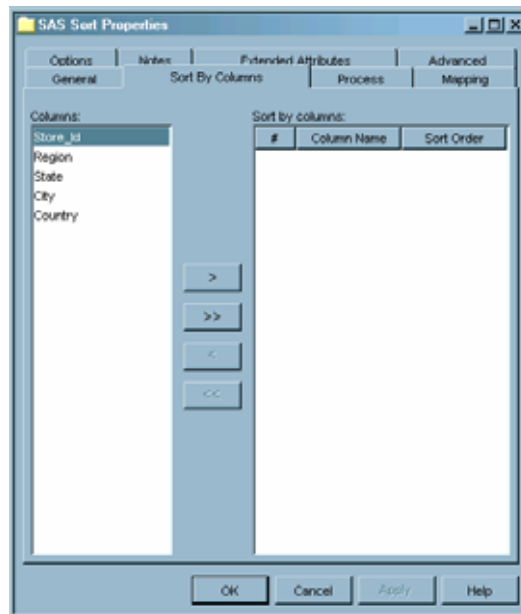
For more information about creating indexes, see *SAS Language Reference: Concepts*.

Sorting Data

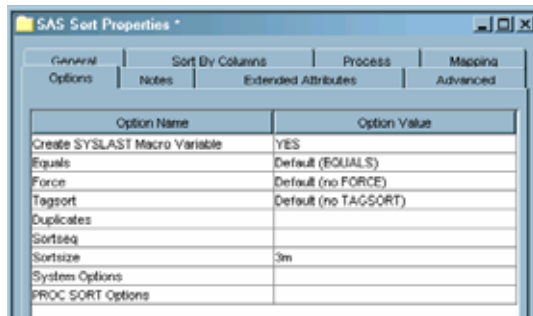
You can sort table rows by the values of one or more character or numeric columns. For Base SAS tables and third-party relational database tables, the process either replaces the original table or creates a new table. You can perform sorting in two ways:

- using the SAS SORT procedure.
- setting properties for a SAS sort template in SAS ETL Studio.

Display 15.4 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 15.5 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 275.

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. Δ

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

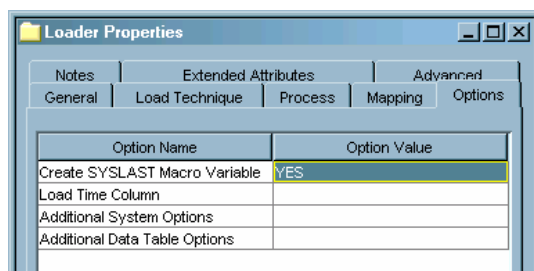
For Base SAS tables and some relational database tables, you can adjust page buffer settings to optimize CPU and I/O use. Different options are used for each type of table.

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 272) 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*. Δ

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 272.

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.

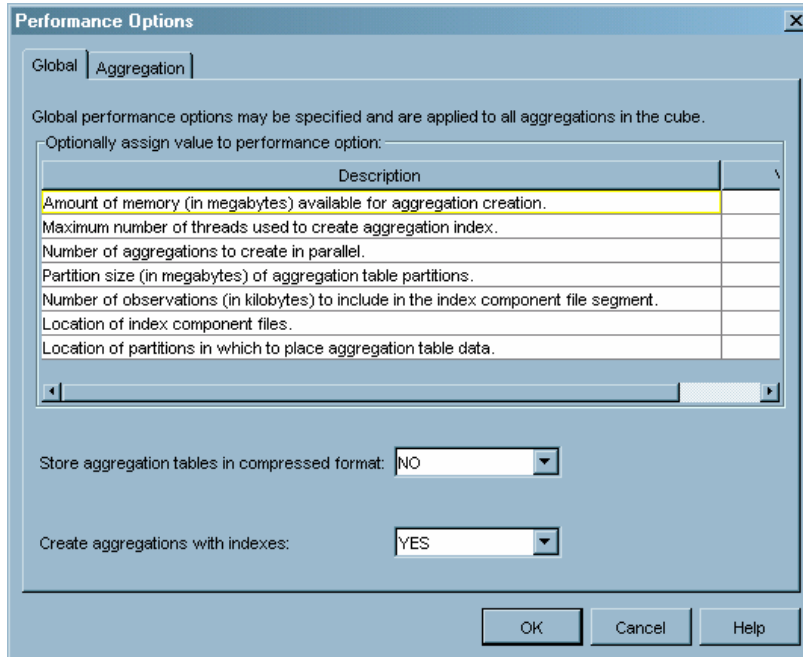
The resulting log file shows timing statistics for each cycle. You can examine this information to determine whether your system is configured correctly. The program is available at support.sas.com/rnd/scalability/spde/valid.html.

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.

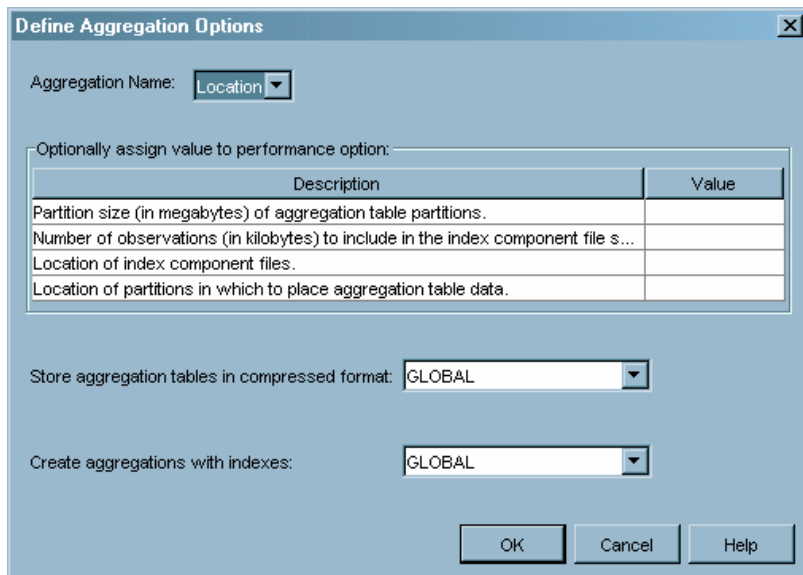
In the Performance Options dialog box, the **Global** tab enables you to set options that apply to all aggregations in the cube.

Display 15.6 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 15.7 The Define Aggregation Options Dialog Box Available from the Aggregation Tab in the Performance Options Dialog Box



Here is a description of each option:

<i>Amount of memory (in megabytes) available for aggregation creation</i>	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.
<i>Maximum number of threads used to create aggregation index</i>	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.
<i>Number of aggregations to create in parallel</i>	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.
<i>Partition size (in megabytes) of aggregation table partitions</i>	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.
<i>Number of observations (in kilobytes) to include in the index component file segment</i>	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. Δ

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.

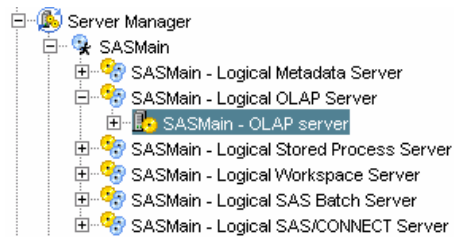
Create aggregations with indexes

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 project 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 result 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*. Δ

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*. Δ

Note: For instructions on how to access the Advanced Options dialog box, see “Optimizing Performance of a SAS OLAP Server” on page 269. Δ

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.

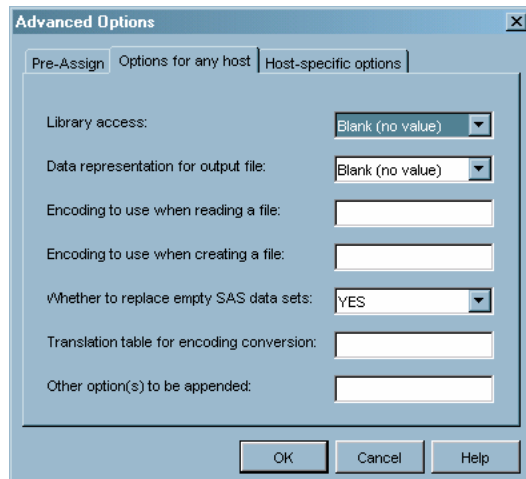
Setting LIBNAME Options That Affect Performance

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 15.8 The Options for Any Host Tab in the Advanced Options Dialog Box for a Base SAS Library



Here are some examples of options that might affect performance:

<i>Data representation for the output file</i> (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.
<i>Input/output block size</i> (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.
<i>Number of page caches to use for each open member</i> (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).

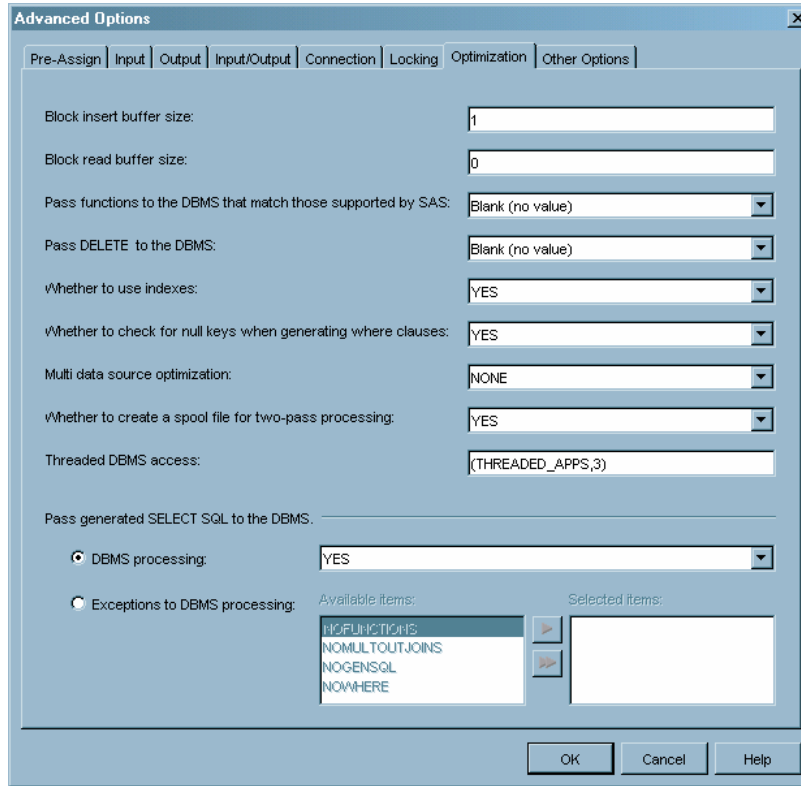
The **Other option(s) to be appended** field can be used to specify LIBNAME options such as COMPRESS= (see “Compressing Data” on page 258).

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 15.9 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 263.

Block read buffer size
(READBUFF=)

specifies the number of rows of DBMS data to read into the buffer. See “Buffering Data” on page 263.

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 259.

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 (DBSLICEPARAM=) 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**. Δ

For more information about threaded reads, see “Using Threaded Reads” on page 264.

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.

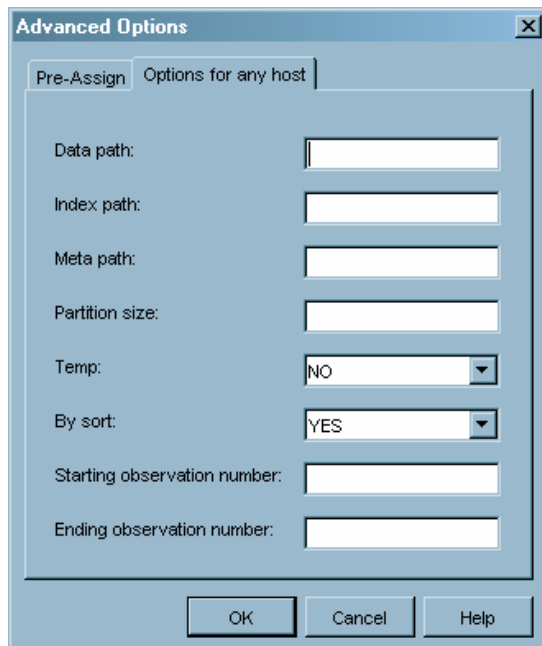
The **Other Options** tab, which is available for all database management systems, can be used to specify LIBNAME options such as COMPRESS= (see “Compressing Data” on page 258).

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 15.10 The Options for Any Host Tab in the Advanced Options Dialog Box for a SPD Engine Library



<i>Data path</i> (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).
<i>Index path</i> (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.
<i>Meta path</i> (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.
<i>Partition size</i> (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.
<i>Temp</i> (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.
<i>By sort</i> (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.
<i>Starting observation number</i> (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.
<i>Ending observation number</i> (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.

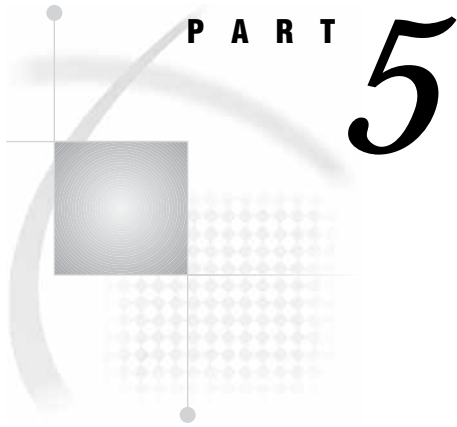
In addition to the LIBNAME options, there are also table and system options that can be used to tune SPD Engine performance. For example, the SPDEUTILLOC=

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

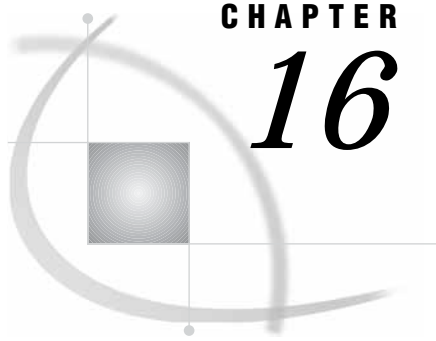
- determine the amount of space that needs to be allocated to the data, metadata, index, and work areas
- evaluate the advantages and disadvantages of different RAID groups for each of the different types of areas.

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.



Application Administration

- Chapter 16* **Administering SAS ETL Studio** 281
- Chapter 17* **Managing the Reporting Environment** 305
- Chapter 18* **Preparing SAS Enterprise Miner for Use** 341



CHAPTER

16

Administering SAS ETL Studio

<i>Overview of Administering SAS ETL Studio</i>	281
<i>Connecting to SAS Servers</i>	282
<i>Metadata Server</i>	282
<i>Workspace Server</i>	283
<i>Redirecting Output and Logging Information to a File</i>	284
<i>Connecting to Data Servers</i>	286
<i>Setting Up Change Management</i>	287
<i>Creating a Repository Directory</i>	287
<i>Creating a Project Repository</i>	288
<i>Setting Metadata Permissions for Your User</i>	289
<i>Creating a Metadata Profile</i>	290
<i>Using the Metadata Profile</i>	291
<i>If No Project Repositories Are Displayed</i>	291
<i>Using Custom-Tree Folders for Security</i>	292
<i>Setting Permissions on Custom-Tree Folders</i>	292
<i>Multiple Inheritance of Access Controls</i>	294
<i>Importing and Exporting SAS Code Transformations</i>	295
<i>Importing and Exporting Metadata</i>	295
<i>Supported Metadata Formats</i>	295
296	
<i>Exporting Metadata</i>	296
<i>Testing the Job Scheduler</i>	297
<i>Setting Up a SAS Data Quality Server</i>	298
<i>How the Data-Quality Software Has Been Configured</i>	299
<i>Testing the SAS Data Quality Server</i>	299
<i>Downloading Locales</i>	300
<i>Creating Schemes</i>	300
<i>Setting SAS ETL Studio's Data Quality Options</i>	301
<i>Enabling Status Code Handling</i>	302
<i>Supporting the E-mail Action</i>	302
<i>Supporting the Custom Action</i>	302

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** drop-down list on the Products & Solutions Documentation 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.

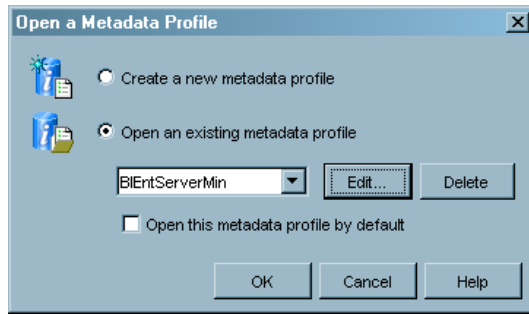
Metadata Server

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.



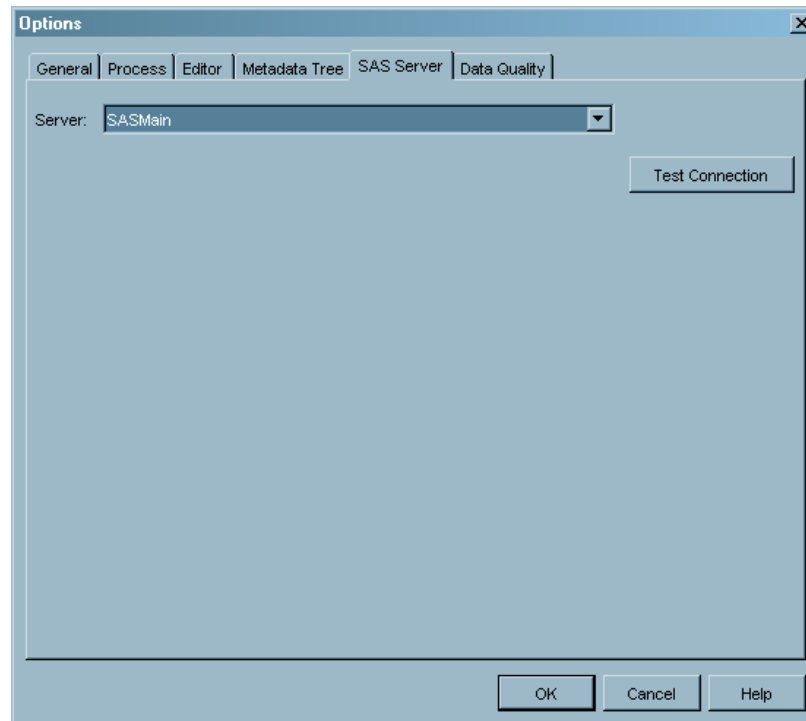
If a metadata profile has been created on this machine, you will see its name listed in the **Open an existing metadata profile** list box. The profile will have the same name as your configuration directory and will have been defined to connect to the metadata server using the SAS Administrator account. 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 on-line help, if necessary, to create a profile. (Do not save your password in the profile.) When you have finished defining the profile, SAS 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:

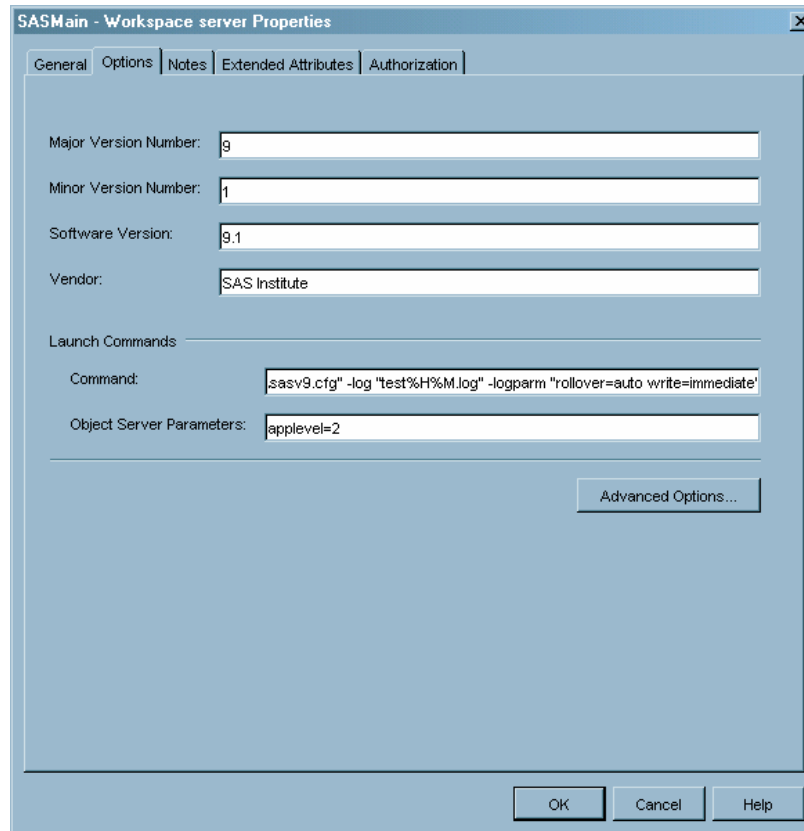


- 3 Select from the list the name of the application server to which the workspace server belongs.
- 4 Click **Test Connection**. You might be prompted for a user name and password. If you are, enter credentials that will allow you to be authenticated on the host where the workspace server is running. If all goes well, you will see an Information dialog box that says, "Connection to the server was successful."

Redirecting Output and Logging Information to a File

To specify alternative destinations for the SAS log and SAS output, add the following options to the `sas` command that starts the SAS Workspace Server.

- 1 In SAS Management Console, expand the Server Manager; then, expand the SASMain—Logical Workspace Server. You will see a tree node that represents the physical workspace server.
- 2 Right-click the icon for the physical workspace server, and select **Properties** from the pop-up menu. A Workspace Server Properties dialog box appears.
- 3 Click the **Options** tab. You will see the information that is shown in the following display.



- 4 Edit the text in the **Command** text box. By default, this text is set to

```
sas -config "path-to-config-dir\Lev1\SASMain\sasv9.cfg"
```

To route the SAS log to a file, edit the command to make it look something like this:

```
sas -config "path-to-config-dir\Lev1\SASMain\sasv9.cfg"
-log log-file-name%H%M.log -logparm "rollover=auto write=immediate"
```

For routing the SAS output to a file, the command should look like this:

```
sas -config "path-to-config-dir\Lev1\SASMain\sasv9.cfg"
-print print-file-name.lst
```

Also, for redirecting the log, in the **Object Server Parameters** text box, enter:

```
applevel=2
```

This will allow an appropriate level of log information to be routed to the file.

CAUTION:

Do not add the options to the configuration file specified in this command. Setting values in `Lev1\SASMain\sasv9.cfg` affects every server that is launched. This includes the metadata server, the OLAP server, and every workspace server and stored process server. △

- 5 Click **OK** in the Workspace Server Properties dialog box.

Connecting to Data Servers

After you have established that your users can connect to the metadata server and the system's workspace servers (as described in "Connecting to SAS Servers" on page 282), 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 239.
- 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.

#	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	Angoulême	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 box and selecting the **Physical Storage** tab. The **DBMS** list box will contain a value such as SAS, Oracle, or Sybase. △

Setting Up Change Management

SAS 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 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 that represents the user in the foundation repository. You should already have performed this task as well.
- 3 Create a new directory in the file system that will hold the contents of the new project repository. Make sure that only the user who 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 287.
- 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 288.
- 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 289.
- 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 290.
- 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 291.

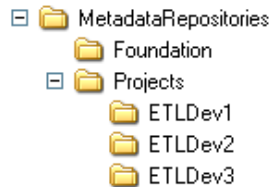
As previously mentioned, you should already have performed steps 1 and 2. If you have not performed those steps, return to "Setting Up Security for Regular Users" on page 218 and perform steps 1 and 2 now. The other steps are explained in the sections that follow.

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 box is optional. Click **Next**.
- 4 In the Definition of Data Source window, you are prompted for three pieces of information: an Engine, a Path, and Options.
 - a Engine—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 the user will check metadata objects out of the foundation repository into the user's project directory to work on the metadata objects. The user will then check these objects (and any new objects) back into the foundation repository.

To specify this relationship between repositories, select Foundation from the list of repositories on the left, and then click the right-arrow 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

After you have created a project repository for a user, you need to use SAS Management Console to set (in metadata) the permissions for that user to access the foundation repository and the permissions for that user to access his or her project repository. The user needs the ReadMetadata and CheckInMetadata permissions for the foundation repository and ReadMetadata and WriteMetadata permissions for the project repository. The following steps explain how to set up permissions for the owner of the project repository to access that repository. (You set the user's permissions to access the foundation repository in a similar way.)

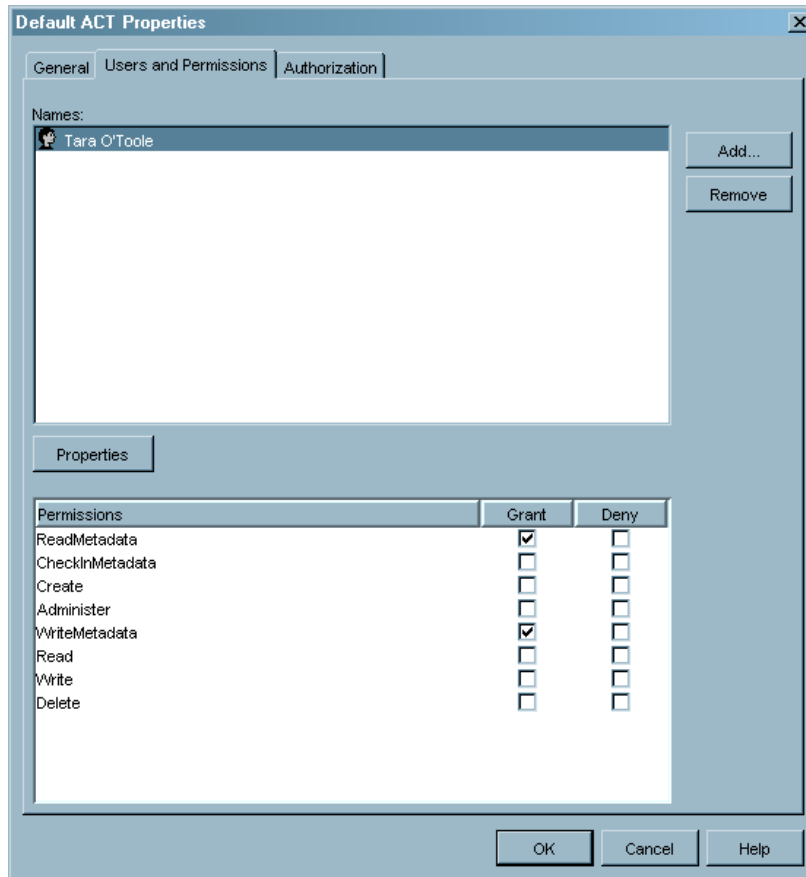
Note: When you perform this step, you must be logged on to SAS Management Console as an unrestricted user. For example, you could be logged on as the SAS Administrator (**sasadm**). To make another user an unrestricted user, you must add that user's ID to the file **adminUsers.txt** and prepend an asterisk to that ID. △

- 1 Select the user's project repository from the **Repository** list box, as shown in the following display.



- 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. After you have supplied this data, click **Next**.
- 5 In the Connection Information window, fill in the following text boxes:
 - Machine**—Enter the full name of the machine on which the metadata server is running. (This information is supplied by the administrator.)
 - Port**—Enter the number of the port on which the metadata server is listening. (This information is supplied by the administrator.)
 - Username**—Enter your user name. On Windows systems, this name should be of the form *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.) If no project repository is displayed, see below.
- 7 In the Finish window, click **Finish**.

Using the Metadata Profile

When the user finishes running the Metadata Profile wizard, SAS 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.

If No Project Repositories Are Displayed

The most likely reason why no **Project** tab or project repositories are displayed is that the login (username and password) that is specified in the user's metadata identity is different from the login that is used to connect to the metadata server. You might need to ask your administrator what the correct login is. Other troubleshooting steps you can take include

- Verify that the instructions in “Setting Up Change Management” on page 287 were followed.
- On a Windows platform, verify that the Windows network domain is specified for the login that is part of the user's metadata identity and for the login that is used to connect to the metadata server.
- Verify that operating system security enables the user to read the directory where the Project repository is located.

Note: For information about how to work in a change-managed environment, see the *SAS ETL Studio: User's Guide*. Δ

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.

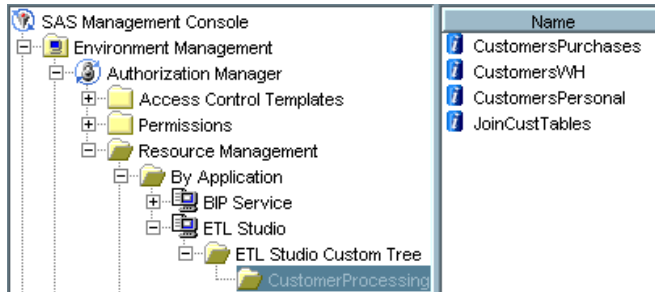
- The Custom tree enables developers to organize their metadata objects in any manner they see fit. For example, they might want to create a folder that contains all of the metadata objects that are related to a particular set of jobs.
- The Custom tree also displays in the SAS Management Console's Authorization 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:

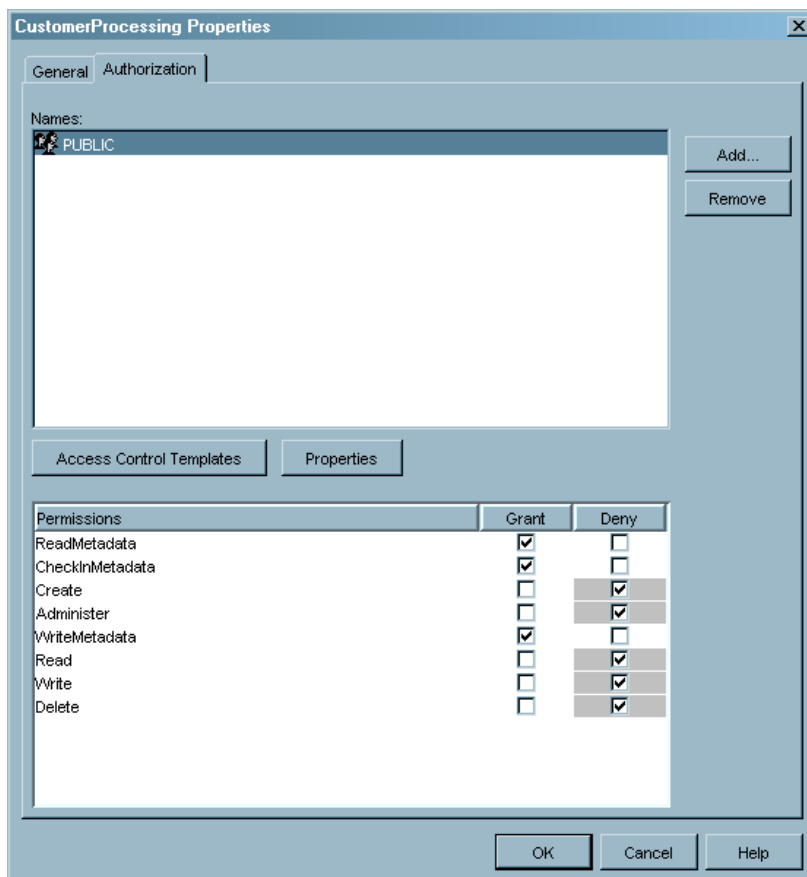
- 1 In SAS Management Console, expand the following sections of the SAS Management Console tree, in this order:
 - Authorization Manager
 - Resource Management
 - By Application
 - ETL Studio
 - ETL Studio Custom Tree.

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 16.1 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
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. Δ

Multiple Inheritance of Access Controls

As is explained in “Inherited Access Controls” on page 180, 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/XML) 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.

The SAS ETL Studio Metadata Import Wizard uses converters installed in the **plug-ins** 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 295.

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 by using a metadata profile that specifies the repository into which you want to import metadata, and select **Tools ► Metadata Importer**.
- In SAS Management Console, right-click the repository into which you want to import metadata (in the Metadata Manager portion on the tree in the left pane), and select **Import Metadata** from the pop-up menu.

Both procedures start the same wizard.

After you have started the wizard, consult the *SAS Management Console: User's Guide* for information about how to run the wizard. This document contains 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 **plug-ins** 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 “Supported Metadata Formats” on page 295 lists the types of objects that you can export.
- If you are exporting metadata from a dependent repository—such as a project directory—metadata is not retrieved from the parent(s) of the repository. For example, only tables that use library definitions in the exported repository are exported. Tables that use library definitions in a parent repository are not exported.

Do either of the following to start the export wizard:

- In SAS 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

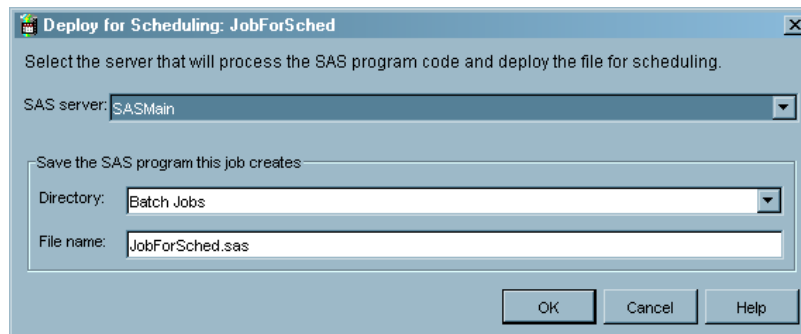
ETL specialists can create sets of SAS ETL Studio jobs, called flows, and execute each job within a flow

- 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.

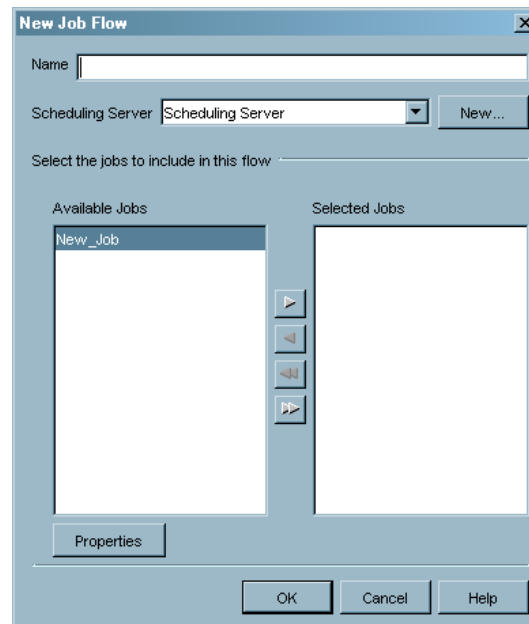
Here is how to quickly make sure that the scheduler is working correctly.

- 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 appears in the Jobs folder of the Inventory tree.
- 2 Deploy the new job for scheduling by right-clicking the icon for the new job and selecting **Deploy for Scheduling**. The Deploy for Scheduling dialog box appears.



From the **SAS server** list box, select the application server that contains your batch server, and from the **Directory** list box, select the name that you specified earlier (**SASEnvironment\SASCode\Jobs** by default) for your deployment directory. Use the default value for **File name**. Then click **OK**.

- 3 In SAS Management Console, use these steps to create a job flow (a set of related jobs):
 - a Right-click the Schedule Manager, and select **New Flow** from the pop-up menu. The New Job Flow dialog box appears.



- b Enter a unique name for the job flow in the **Name** text box.
- c From the **Scheduling Server** list box, select the name of the Platform Job 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 appears beneath the Schedule Manager.

- 4 In SAS Management Console, schedule the new job flow that you want to run:
 - a Right-click the Job Flow icon, and select **Schedule Flow** from the pop-up menu that appears. A Schedule Flow dialog box appears.
 - b In the Schedule Flow dialog box, leave the value of the **Trigger** list box set to **Run Once** and click **OK**.

You should see a message indicating that the flow has been scheduled to run.

- 5 To verify that the job ran successfully, use the Flow Manager application (part of the Platform 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 Foundation software, including the SAS Data Quality Server, on a machine and have run the SAS 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 `insas-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:

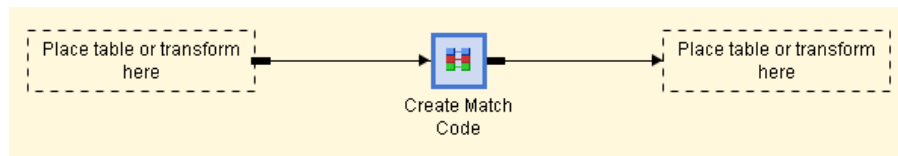
```
-dqlocale (ENUSA)
-dqsetuploc ''dqsetup.txt''
```

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

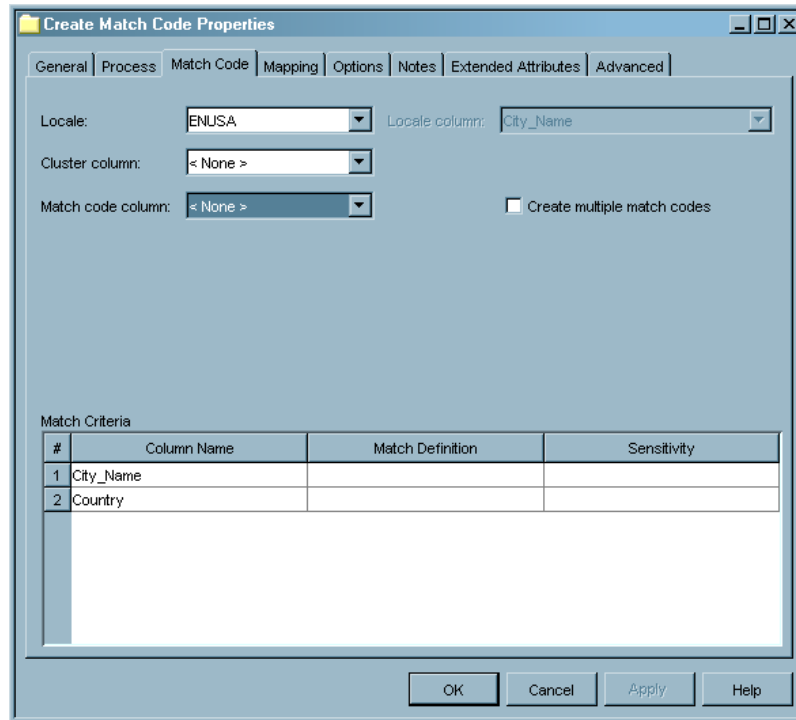
A simple procedure to verify that your SAS Data Quality Server is working is to create a job in SAS ETL Studio that contains a Create Match Code transformation. Follow these steps to create such a job.

- 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.

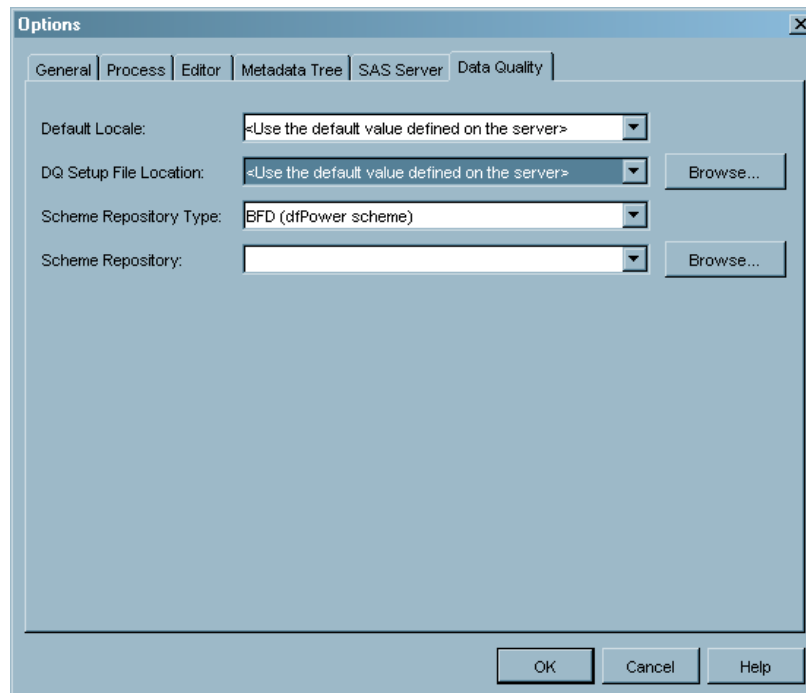
Typically, you should save these schemes to the directory `sas-root/dquality/sasmisc/content/scheme`. However, there is also a `scheme` directory in your configuration directory: `configuration-directory/Lev1/SASMain/SASEnvironment/`

QualityKB/scheme. Use the latter location for schemes that are specific to an application server. As explained in the section “Setting SAS ETL Studio's Data Quality Options” on page 301, 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 box 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 16.2 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.

Field	Explanation
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 <i>sas-root/dquality/sasmisc/content/scheme</i> . Use the list box or the Browse button to specify a different repository.

Enabling Status Code Handling

When an ETL developer executes a job in SAS ETL Studio, notification of the job's success or failure can be e-mailed to a person, can be written to a file, or can cause the execution of an autocall macro. A **Status Handling** tab is included in the property windows for jobs and for some transformations. Users can select options from a list of code conditions and actions on this tab. For example, a user can select a code condition such as successful and associate it with an action such as Send Email.

Before ETL developers can use some of the actions, you must set up the environment properly. Such setup is required for the following actions:

- Email actions - You must set SAS system options for e-mail for the SAS application server that is used to execute jobs.
- Custom actions - You must make a SAS macro autocall library accessible by the SAS application server that is used to execute jobs.
- Send Entry to a Data Set - You must preassign the library that contains the data set to an application server before the job or transformation executes.

For information on how to support the first two actions listed above, see the subsections below. For information on how to support the last action, see "Pre-assigning Libraries" on page 136.

Supporting the E-mail Action

Setting up the e-mail action is simple. Just add the appropriate SAS system options for e-mail to the configuration file *path-to-config-dir\Lev1\SASMain\sasv9.cfg*. For example, if you are using the SMTP e-mail interface, you would add to this file the following options:

```
-emailsys smtp
-emailhost email-server
```

In this case, the value of *email-server* specifies the SMTP server that supports e-mail access for your site.

Note: The e-mail system options are documented in SAS Help and Documentation. Δ

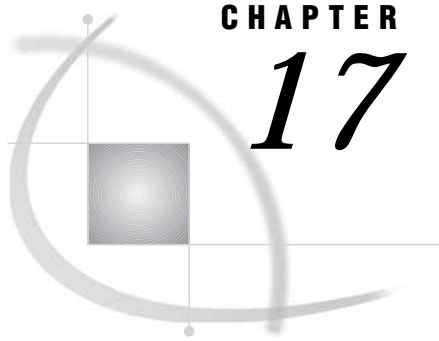
Supporting the Custom Action

Using the Custom action, an ETL developer can execute a macro in a macro autocall library in order to provide user-defined status handling for a job or transformation. On the administrative side, all you need to do is to make sure that the autocall library is known to the application server. You do this by editing the file

path-to-config-dir\Lev1\SASMain\sasv9.cfg. By default, the file contains a line similar to this:

```
-sasautos ("SASEnvironment/sasMacro" SASAUTOS)
```

Add the library to the list of libraries in parentheses. The path to the library can be a full path or a path relative to *path-to-config-dir*\Lev1\SASMain.



CHAPTER

17

Managing the Reporting Environment

<i>Overview of Reporting in the SAS Intelligence Platform</i>	306
<i>The Reporting Process</i>	307
<i>Managing Information Maps</i>	308
<i>Logging on to SAS Information Map Studio</i>	308
<i>Using Metadata Profiles with SAS Information Map Studio</i>	308
<i>Using Information Maps with SAS Web Report Studio</i>	309
<i>Exporting and Importing Information Maps</i>	309
<i>Exporting an Information Map</i>	309
<i>Importing an Information Map</i>	309
<i>Managing Reports</i>	310
<i>Types of Reports</i>	310
<i>The Parts of a Report</i>	311
<i>The Reporting Folder Structure</i>	311
<i>Overview of the Reporting Folder Structure</i>	311
<i>Understanding the ReportStudio Folder Structure</i>	312
<i>Creating the ReportStudio Folder Structure with SAS Web Report Studio</i>	313
<i>Manually Creating the ReportStudio Folder Structure</i>	314
<i>Customizing the Reporting Folder Structure</i>	314
<i>Working with Banner Images in SAS Management Console</i>	316
<i>Adding Images to the BannerImages Folder</i>	316
<i>Adding Descriptions to Images in the BannerImages Folder</i>	316
<i>Deleting Images from the BannerImages Folder in the Report Repository</i>	316
<i>Importing a Report</i>	317
<i>Administering Batch Reporting</i>	317
<i>Files Required for Batch Reporting</i>	317
<i>The Batch Reporting Process</i>	317
<i>The Batch Generation Tool</i>	318
<i>Configuring the Batch Generation Tool</i>	320
<i>Viewing and Editing Batched Reports</i>	322
<i>Example Batch Generation Tool Configuration File</i>	322
<i>Example 1: Extracting a Report</i>	322
<i>Example 2: Extracting a Directory of Reports</i>	323
<i>Example 3: Extracting While Excluding Prompts</i>	324
<i>Example 4: Extracting Recursively</i>	324
<i>Example 5: Using Edited Extract Data as Input for Run Mode</i>	325
<i>Example 6: Running a Single Report</i>	326
<i>Example 7: Running a Directory of Reports</i>	326
<i>Example 8: Running Without the -configFile Option</i>	326
<i>Registering Fonts for Use from SAS Web Report Studio</i>	327
<i>Printing Non-Latin1 Languages from SAS Web Report Studio</i>	327
<i>Using Xythos Software's WebFile Server with SAS Web Report Studio</i>	327

<i>Customizing the SAS Web Report Studio Banner and Title</i>	328
<i>Specifying an Image Reference</i>	328
<i>Example: Customizing the SAS Web Report Studio Banner</i>	329
<i>Printing Large Reports</i>	330
<i>Adjusting JVM Memory</i>	330
<i>Increasing the Memory Available to the Tomcat JVM</i>	330
<i>Writing ODS Output to a Report Repository</i>	330
<i>Example: Writing ODS Output to a Repository</i>	331
<i>Using a File-Based Content Server</i>	331
<i>Managing Stored Processes</i>	332
<i>Configuring Stored Processes to Work with SAS Web Report Studio</i>	332
<i>Converting an Existing SAS Program to a Stored Process</i>	332
<i>Example of Converting a SAS Program to a Stored Process</i>	333
<i>Requirements for Using Stored Processes in SAS Web Report Studio</i>	333
<i>Stored Process Output Style</i>	334
<i>Associating Stored Processes with Information Maps</i>	334
<i>Securing Your Reporting Environment</i>	335
<i>Securing Information Maps</i>	336
<i>Securing Batched Reports</i>	336
<i>Securing Temporary Files</i>	337
<i>Using SAS Web Report Studio with SSL</i>	337
<i>Preventing the DAV Navigator Portlet from Viewing Unauthorized Data</i>	338
<i>Delivering Reports</i>	338

Overview of Reporting in the SAS Intelligence Platform

A report is a visual representation of your data that is created from information maps and stored processes. Information maps are a layer of abstraction between a data source and the report, and describe data sources in a way that enables users to query that data in a user-friendly manner, without knowing the intricacies of the data. Stored processes are SAS programs that can be embedded in reports and information maps. Stored processes enable report consumers access to the results of potentially complex programs or queries without needing to know any programming or even how to execute a SAS program.

The SAS Intelligence Platform provides Web and desktop products that offer reporting environment configurations that enable you to address different skill levels and usage patterns. Whether your organization requires reports with interactive queries, delivery of content via a Web-based portal, or publish-and-subscribe channel distribution, the SAS Intelligence Platform can be tailored to meet your organization's reporting needs. The following list describes the applications that can be involved in the creation and viewing of reports:

<i>SAS Information Map Studio</i>	enables information architects to create and manage information maps. For more information about information maps, see "Managing Information Maps" on page 308 or the SAS Information Map Studio Help.
<i>SAS Web Report Studio</i>	enables information providers to build reports from information maps using a simple Web-based tool. Information consumers can then view and interact with the reports. For more information about SAS Web Report Studio, see the SAS Web Report Studio Help.
<i>SAS Web Report Viewer</i>	a report viewer without the report-building functionality of SAS Web Report Studio.

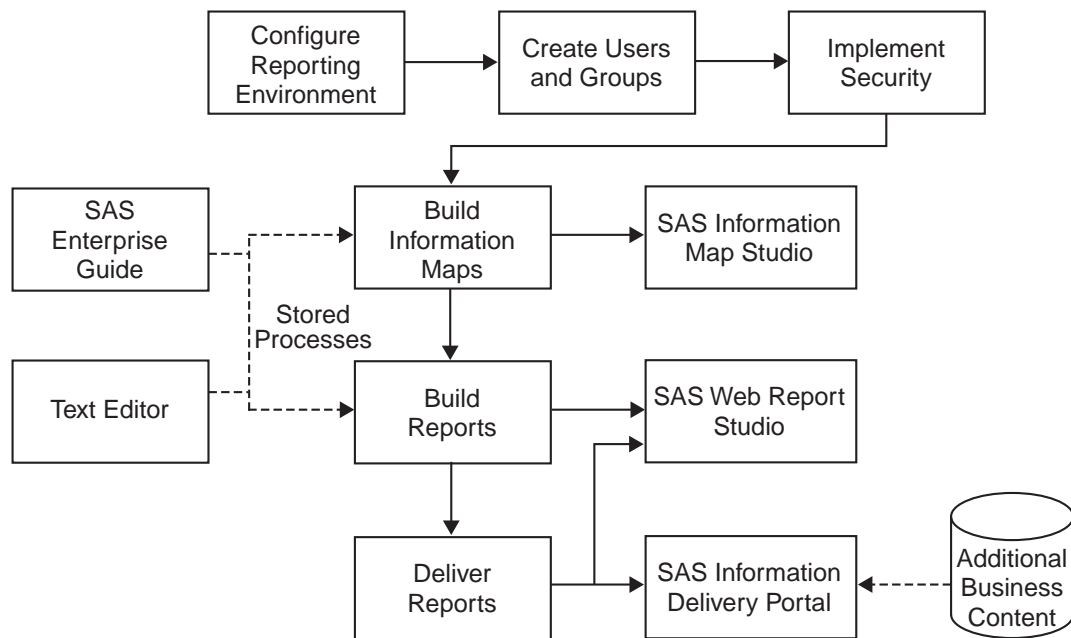
<i>SAS Information Delivery Portal</i>	enables report writers to share reports and information securely, and enables report consumers to view reports, access business documents, and launch Web-based applications. For more information about the SAS Information Delivery Portal, see “Delivering Reports” on page 338 or the SAS Integration Technologies Library, available at support.sas.com/rnd/itech/library/ .
<i>Business Report Manager</i>	enables information architects to define locations from and to which SAS applications can store and access report content such as report definitions and output from stored processes.
<i>SAS Enterprise Guide</i>	creates stored processes for use in reports and maps.
<i>SAS Add-In for Microsoft Office</i>	enables you to execute stored processes and embed the results in your Microsoft Word and Microsoft Excel documents and spreadsheets. Within Microsoft Excel, the add-in also enables you to access, view, and analyze data sources available from your SAS server.

SAS Web Report Studio and SAS Information Map Studio can only be deployed in, and can only retrieve content from, foundation repositories.

The Reporting Process

The first step in the reporting process is the installation and configuration of the reporting environment. In this phase administrators specify details such as the type of content server and the look of SAS Web Report Studio. Users and groups are then created, and the reporting environment is secured. Information architects then build information maps and, using SAS Enterprise Guide or any text editor, stored processes. These information maps and stored processes are used as information sources by which the report creators query data and build reports. After the reports are built, they can be viewed within SAS Web Report Studio, SAS Web Report Viewer, or the SAS Information Delivery Portal. The basic process of the reporting environment is illustrated in the following diagram.

Figure 17.1 The Reporting Process



Managing Information Maps

SAS Information Maps are metadata objects that translate complex data relationships into business terms that can easily be used by any non-technical information provider, giving that person immediate access to the information they need to make decisions. Information maps are created in SAS Information Map Studio, and provide business users with easy access to data by

- shielding users from the complexities of the data
- making data storage invisible (whether the data is relational or OLAP, in a SAS data set or a third-part database system)
- predefining business formulas and calculations to ensure consistency
- providing an easy way to query data for answers to business questions.

Logging on to SAS Information Map Studio

When using SAS Information Map Studio, you cannot log on to a metadata repository as PUBLIC. You must have at least one login definition in the metadata repository that contains the user ID that corresponds to the user name that you are logging on as. You can create a user via the User Manager plug-in to SAS Management Console.

Using Metadata Profiles with SAS Information Map Studio

If you have created metadata profiles in other SAS products such as SAS ETL Studio or SAS Management Console, you can use those profiles to log on to SAS Information Map Studio. However, there are certain optional profile settings that can only be modified by using the SAS Information Map Studio metadata profile wizard. For more

information, see the SAS Information Map Studio Help, which is available from within the product.

Using Information Maps with SAS Web Report Studio

When using information maps with SAS Web Report Studio, the information maps

- *must be stored in the Maps folder or its subfolders within the ReportStudio folder structure in your reporting environment.*

SAS Web Report Studio users can access only the information maps that are stored in the ReportStudio\Maps folder or its subfolders. For more information about the ReportStudio folder structure, see “The Reporting Folder Structure” on page 311.

- *must not be moved or have their paths altered.*

When a report is created using an information map, the report definition stores the URL of the information map. If the information map or the folder that contains the information map is moved or renamed, the report definition cannot reference the information map, and the report cannot be rendered (it does not report false data). This means that while reports can be freely copied, moved, and renamed within SAS Web Report Studio, indiscriminately altering the location or name of an information map can result in broken reports. You can, however, move an information map and a report to a different server if you maintain the same naming scheme between servers and between the repositories on those servers.

Exporting and Importing Information Maps

Exporting an Information Map

To export an existing information map, follow these steps:

- 1 Open SAS Information Map Studio by using the metadata profile for the repository where the information map is located.
- 2 Open the information map that you want to export by double-clicking it in the Repository pane or by selecting **File** ► **Open**.
- 3 Select **Tools** ► **XML**.
- 4 Click the **XML** tab.
- 5 Click the **EXPORT** button.
- 6 Save the XML representation of the information map by clicking the **SAVE** button.

Importing an Information Map

To minimize the amount of editing that an information map needs when you are importing it into a repository, make sure that you import the data sources and stored processes that the map uses before you import the map itself. Also be sure to use the same names, labels, and paths for those data sources in the new repository that you used in the source repository. For information on dealing with unresolved resource issues, see the SAS Information Map Studio Help, which is available from within the product.

To import an information map, follow these steps:

- 1 Open SAS Information Map Studio by using the metadata profile for the repository where you want to place the imported information map.
- 2 Select **Tools** ► **XML**.

- 3 Click the **XML** tab.
- 4 Click the **Import** button.
- 5 Select the previously exported information map XML file that you want to import into this repository.
- 6 Click the **Open** button.
- 7 Click **OK** in the Edit Information Map XML dialog box.

The XML file is read and interpreted by SAS Information Map Studio; however, the XML file has not yet been saved to the repository. Examine the imported information map for unresolved resources and other errors. Use the XML editing function to correct any unresolved resources or errors in the imported information map.

- 8 Save the information map to the new location.

To avoid problems in SAS Web Report Studio when importing reports, the information map should have the same path in the new repository as it did in the original repository. For example, an information map that was saved in Repository1 in **BIP Tree/ReportStudio/Maps/Sales** must be placed in Repository2 in **BIP Tree/ReportStudio/Maps/Sales**.

- 9 Copy any necessary formats to the new server.

Managing Reports

The configuration of the reporting environment

- allows information architects to have access to the data that they need in order to build information maps that are used to generate the reports
- allows report creators to access the information maps that they need in order to build reports
- ensures that the underlying report data is accessible and secure.

Types of Reports

Reports are named based on their relationship to the underlying data:

<i>automatically refreshed reports</i>	run queries each time that a report is accessed to get the latest data.
<i>manually refreshed reports</i>	run queries when they are requested. Although initially a static view of data, data in a manually refreshed report can be updated by a user. Additionally, when a report is saved from SAS Web Report Studio as a manually refreshed report, a batched report is automatically generated.
<i>batched reports</i>	a pre-rendered HTML and PDF version of a report that is generated by using the Batch Generation Tool or whenever you save a report from SAS Web Report Studio as a manually refreshed report. Batched reports can also be created by running the Batch Generation Tool on an automatically refreshed report. For more information, see “Administering Batch Reporting” on page 317.

The Parts of a Report

When a report is saved from SAS Web Report Studio, two objects are created:

report definition contains information about how the report is presented and what data is included in the report. The report definition is constructed according to the SAS Report Model, which is an XML specification for business reports. Reports that comply with the SAS Report Model can be created, viewed, and modified by a variety of SAS Business Intelligence applications.

Although the file structure that supports a report definition is accessible on the server, you must not delete or rename files on the server because doing so would change only the report definition, but would not change the underlying data that is stored in the metadata object in the repository. Manually deleting or renaming files in the report definition can leave the repository in an inconsistent state, which results in non-functioning reports.

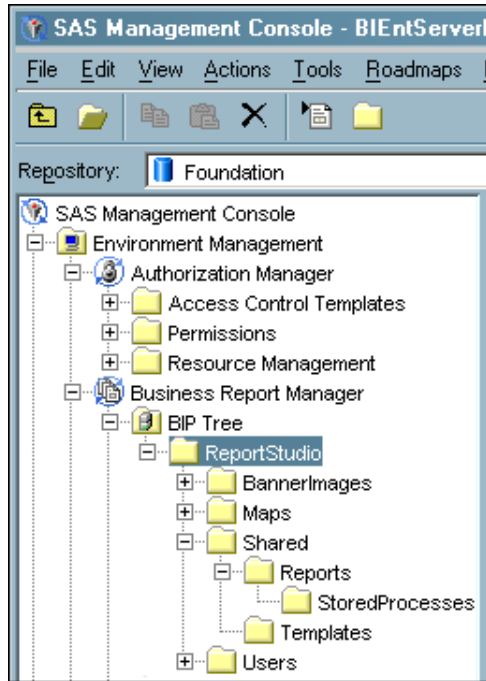
metadata object contains information such as time stamps, authorship, access controls that provide security for the report, and other report and application-specific properties. The metadata object that is associated with a report definition is stored on the metadata repository.

The Reporting Folder Structure

Overview of the Reporting Folder Structure

One of the primary tools for controlling how reports are secured, shared, and delivered is the reporting folder structure. The recommended reporting folder structure is the ReportStudio folder structure that is located in the BIP Tree folder in SAS Management Console (although the reporting environment can be configured to use different folder names). The following image shows the default ReportStudio folder structure.

Display 17.1 The ReportStudio Folder Structure



If your organization uses SAS Web Report Studio, then the ReportStudio folder structure is created automatically the first time that a user logs in to the SAS Web Report Studio application. If your organization does not use SAS Web Report Studio, then you must manually create the ReportStudio folder structure (see “Manually Creating the ReportStudio Folder Structure” on page 314).

Because repairing the associations between reports and the underlying information maps and stored processes can be difficult, the ReportStudio folder structure (or whatever folder structure your organization implements) should always be created before any information maps, stored processes, or report definitions are created.

Understanding the ReportStudio Folder Structure

Regardless of whether the ReportStudio folder structure is automatically created by SAS Web Report Studio or is manually created using the Business Report Manager plug-in to SAS Management Console, the ReportStudio folder structure is created in both the BIP Tree folder in the metadata server and in the content area that was defined for your report repository.

The ReportStudio folders in the content area represent the basic structure of your directory-based report repository. The ReportStudio folders in the BIP Tree folder are objects on the metadata server. The following list describes each of the default folders:

BannerImages folder

The location where SAS Web Report Studio looks for banner images when building a report.

Maps folder

The location for information maps. Because information maps are metadata objects, they are stored on the metadata server rather than in the corresponding folder in the report repository.

You can restrict access to the information maps by setting permissions on this folder and its subfolders. SAS Web Report Studio users can access only the information maps that are stored in this Maps folder or its subfolders.

Shared folder

Reports that are saved in the Shared folder in your report repository or in any subfolders within the Shared folder can be accessed by multiple users. You can restrict access to the reports by setting permissions at the folder level. For more information on setting access controls for reports, see “Access Requirements for Working with Reports” on page 209.

Shared/Reports folder

The location for shared reports.

Shared/Reports/StoredProcesses folder

The initial location that SAS Web Report Studio accesses when you attempt to insert a stored process section when you are building a report.

Shared/Templates folder

Stores templates used when creating reports with custom layouts in SAS Web Report Studio. This folder is created automatically when templates are used.

Users folder

A personal folder is created for each user account when that user logs in to SAS Web Report Studio. By default, only the user who is associated with a folder has the permissions to read and write metadata to that folder. Because the security for each user folder is automatically set when the user first logs in and the folder is created, you should not manually create user folders.

If the ReportStudio folder structure is created automatically via SAS Web Report Studio, the default folder names are defined when SAS Web Report Studio is installed. In order to maintain access to the information maps that are stored in your report repository, you should not change the folder names after the folders have been created and information maps have been stored in them and accessed by reports.

Creating the ReportStudio Folder Structure with SAS Web Report Studio

Logging on to the SAS Web Report Studio application generates the ReportStudio folder structure in the BIP Tree of SAS Management Console (the metadata server) and in the directory-based report repository. To create the ReportStudio folder structure using SAS Web Report Studio, follow these steps:

- 1 In your Web browser, enter the Web address for your SAS Web Report Studio application. This is usually `http://<host-name>:8080/SASWebReportStudio`.
- 2 Log on to the SAS Web Report Studio application using the SAS Demo User account that was created during the installation and configuration process.
- 3 Log off of SAS Web Report Studio.
- 4 Start SAS Management Console.
- 5 Navigate to **Business Report Manager/BIP Tree/ReportStudio**, and verify that the BannerImages, Maps, Shared, and Users folders were created. Also verify

that the subfolders inside the Shared folder were created. You should see the folder structure that is shown in Display 17.1 on page 312.

- 6 Navigate to the content path that was specified for your report repository.
- 7 In the ReportStudio folder, verify that the BannerImages, Maps, Shared, and Users folders were created.

Manually Creating the ReportStudio Folder Structure

If you do not install and log on to SAS Web Report Studio to generate the ReportStudio folder structure (the preferred method of its creation), you must manually create the ReportStudio folder structure with the Business Report Manager plug-in to SAS Management Console. Before you create the ReportStudio folder structure, you must have defined a content management server. Refer to the `instructions.html` file that was created during the installation process for more information on defining a content management server.

To manually create the ReportStudio folder structure using the Business Report Manager, follow these steps:

- 1 Start SAS Management Console.
- 2 Navigate to **Business Report Manager/BIP Tree/**.
- 3 Select the **BIP Tree** folder, and then select **Actions ► New Directory** and name the new folder **ReportStudio**.

Because the server's operating system might be case sensitive, follow the casing presented here. Both SAS Web Report Studio and SAS Report Viewer are configured by default to look for folders with these exact names.

- 4 Select the **ReportStudio** folder you previously created, and then use the **Actions ► New Directory** command to create the following subfolders:
 - BannerImages**
 - Maps**
 - Shared**
 - Shared/Reports**
 - Shared/Reports/StoredProcesses**
 - Users**

Your folder structure should look like the folder structure in Display 17.1 on page 312.

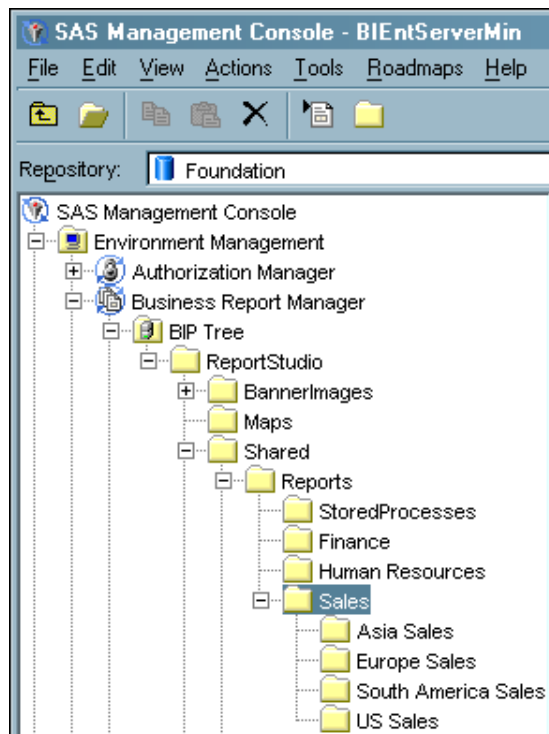
- 5 Navigate to **Authorization Manager/Resource Management/By Application/BIP Service/BIP Tree** and verify that the **ReportStudio** folder structure was created.
- 6 Assign the appropriate permissions to ensure that all users have access to the **Shared** folder. For more information about the Shared folder see "Understanding the ReportStudio Folder Structure" on page 312. For more information about setting permissions for report folders, see "Planning Your Access Controls" on page 204.

Customizing the Reporting Folder Structure

Customizing the reporting environment involves defining subfolders under the ReportStudio folder structure. Reports, information maps, and stored processes inherit permissions from the folder in which they are stored. Subfolders can simplify the organizing process because it is easier to apply permissions to folders than to individual reports or report information sources.

You might want to create specific content areas for information maps and reports that are generated by groups of similar users or departments. For reports, these content areas are represented by subfolders in `/ReportStudio/Shared/Reports/`. For example, you could create folders for Human Resources, Finance, and Sales, and then set permissions for the groups of users that access those folders. Within the Sales folder, additional subfolders could be created for each sales region (see Display 17.2 on page 315). A similar subfolder structure could be defined in the Maps folder for the information maps that are used to build the reports for each of these departments. Note, however, that SAS Web Report Studio does not display to the user any folder name from the Maps folder. Instead, all information maps that the user has permission to see at the root level and in subfolders inside the Maps folder are displayed in a single list.

Display 17.2 A Customized Report Repository



You should not move or rename folders that contain information maps after report writers and information architects begin storing information maps in them. Moving or renaming folders that contain information maps can break metadata associations and can result in reports that cannot be rendered.

Users might want to copy or create their reports in the top level of the Shared folder because it seems like as an easy way to share data. However, this practice can create clutter that is difficult to secure and to navigate. Instead, create subfolders in the Shared folder for storage of individual reports. As an administrator you could restrict write permission in the Shared folder to make sure users couldn't store individual reports in that folder, but such restrictive permissions would also prevent users from creating subfolders.

Another customization option involves renaming the main folders in the ReportStudio folder structure. When you define a new server definition for SAS Web Report Studio in SAS Management Console, you can define the folder names for the report repository. You can also define a top level folder that contains the folders for shared reports, user reports, information maps, and banner images (replacing the BIP

Tree folder). If you want to customize the folder names, navigate in SAS Management Console to **Application Management/Report Studio Configuration/Web Report Studio**. The SAS Report Studio Configuration plug-in is installed when SAS Query and Reporting Services is installed.

Working with Banner Images in SAS Management Console

Adding Images to the BannerImages Folder

The BannerImages folder in the report repository is where SAS Web Report Studio looks for banner images when you are building a report. To add images to this folder, follow these steps:

- 1 From SAS Management Console, navigate to the **Business Report Manager/BIP Tree/ReportStudio/BannerImages** folder.
- 2 Select the **BannerImages** folder.
- 3 Select **Actions ► Import**.
- 4 From the resulting dialog box, choose a file or set of files to import into the BannerImages folder. Use the CTRL key and the SHIFT key to select multiple or contiguous files, respectively.

If you select a folder for import, the entire folder structure is recursively imported. While this can be useful in some situations, SAS Web Report Studio detects only the images in the top level of the BannerImages folder and does not detect images in subfolders.

- 5 Restart the server on which SAS Web Report Studio is running so that the imported images can be seen from within SAS Web Report Studio.

Adding Descriptions to Images in the BannerImages Folder

When creating a report in SAS Web Report Studio, the banner images pull-down menu for header and footer images can list either the image filename or an image description, if an image description is present in the metadata for that image. If no description is present in the metadata, the filename extension is removed, and the filename is used to populate the pull-down menu. To add a description to an image in the BannerImages folder, follow these steps:

- 1 In SAS Management Console, navigate to **Authorization Manager/Resource Management/By Application/BIP Service/ BIP Tree/ReportStudio/BannerImages**.
- 2 Select an image, and then select **File ► Properties**.
- 3 Enter a description in the Properties window.

Image descriptions should be less than 20 characters. The description is viewable in the Business Report Manager folder structure, and from SAS Web Report Studio when you refresh your browser page.

Deleting Images from the BannerImages Folder in the Report Repository

To delete an image from the **BIP Tree/ReportStudio/BannerImages** folder, follow these steps:

- 1 From SAS Management Console, navigate to the **Business Report Manager/BIP Tree/ReportStudio/BannerImages** folder.
- 2 Select the image to delete.

3 Select **Edit ► Delete**.

Importing a Report

To import a report, follow these steps:

- 1 Open SAS Management Console using the metadata profile that contains the server where you want the report copied to.
- 2 Navigate to the Business Report Manager and select the folder into which you want the report imported.
- 3 Select **Actions ► Import**.
- 4 Select the report to import.
- 5 Click **Open**.

For the report to render properly, the underlying information map must be where the report references it. For instructions on moving information maps, see “Exporting and Importing Information Maps” on page 309.

Administering Batch Reporting

Reports can be pre-rendered in a batch process. The advantage of batch reporting is speed: because the queries needed to run a report have already occurred and the HTML or PDF files are already generated, the report renders quickly. Reports that are generated overnight are candidates for batch reporting. A batched report consists of the following files in the report repository:

- an XML file that describes the report layout
- HTML files that contain the data that was returned when the batched report was generated
- cascading style sheet (CSS) files that contain the styles for the HTML
- image files
- a PDF version of the report (if the user chooses to generate one).

Files Required for Batch Reporting

The files that are required for batch reporting are installed as part of SAS Query and Reporting Services. The files are located in the SAS Query and Reporting Services directory. For example, if you accepted the default values on a Windows system installation, the required files are in **C:\Program Files\SAS\SASQueryandReportingServices\9.1**.

The Java class that contains the Batched Report Generator application is **com.sas.report.render.batch.BatchExecution** in **sas.report.render.jar**.

The Batch Reporting Process

To create a batched report from within SAS Web Report Studio, follow these steps:

- 1 Create the report that you want to generate as a batched report by using SAS Web Report Studio.
- 2 Save the report as a batched report by specifying *manually refreshed* as the save option. Reports that are saved as manually refreshed have a batched version generated.

Or, if you do not want to save the report as a manually refreshed report from within SAS Web Report Studio, follow these steps to create a batched report by using the Batch Generation Tool:

- 1 Create the report that you want to generate as a batched report by using SAS Web Report Studio.
- 2 Optionally, run the Batch Generation Tool to extract information from the report repository.
- 3 Run the Batch Generation Tool to create one or more batched reports.
- 4 Secure the generated batched reports.

You do not need to perform any special task to display a batched report. If a batched report exists, it is returned by default when a user requests the report upon which it is based.

The Batch Generation Tool

There are two modes for running the Batch Generation Tool:

- The *extract mode* pulls information about reports from the report repository and then writes that information to a specified file. The extract mode is used to generate a file that is then used as input for the run mode. This two-part process enables you to select specific reports in a directory or edit prompt values before you generate a report or group of reports. Prompts enable users to input report parameters at run-time. For example, a prompt for the year could be used to narrow the focus of a sales report.
- The *run mode* generates the actual report files.

The use of the extract mode is optional. If you use the run mode of the command and specify a directory, batched reports are generated for every report in that directory (non-recursively) with the prompt values as they were last saved.

The command line for the Batch Generation Tool is

```
java com.sas.report.render.batch.BatchExecution <extract|run> <parameters>
```

The following table describes the parameters for the Batch Generation Tool.

Table 17.1 Batch Generation Tool Parameters

Parameter	Purpose	Mode
-username	The name of a user registered in the metadata repository that is specified with the -repository parameter. The user should have all the permissions necessary for generating the report, including permission to access the information maps that are used to generate the reports.	both
-password	The password that belongs to the user specified in -username .	both
-repository	The name of the SAS Metadata Repository.	both
-host	The network address of the host on which the SAS Metadata Server is running.	both
-port	The port on the host to which the SAS Metadata Server is listening.	both

Parameter	Purpose	Mode
-workspaceserver	The name of the logical workspace server that will be used to create a PDF version of the report.	both
-configFile	Optional. The name of a configuration file that contains all or some of the parameters for the Batch Generation Tool. This file can also contain advanced options that define, for example, specialized image loaders. If there is a conflict between the parameters in the configuration file and parameters entered on the command line, then the command line parameters are used.	both
-url -file	The -url parameter specifies an individual report or a directory in the report repository using an SBIP URL. An SBIP URL is a path through the metadata to a specific object, and is comprised of four pieces of information: the repository, the root folder, the path (which can be null), and the name of the object. The -file parameter specifies a file on disk that contains a list of URLs that specify reports and directories in the report repository. The file can also include prompt parameters. This file is generated using the extract mode of the Batch Generation Tool.	both
-nopdf	Optional. A Boolean that specifies whether a PDF file is generated for batched reports. Defaults to true.	Run
-logfile	Optional. The name of the log file for the logging service. Logging options can be changed in the file: sas_service_deployment_export_queryand Reporting_BatchGenerationTemplate.xml	both
-outputFile	The name of the file that will contain the extracted information. If this parameter is set in a configuration file, use double slashes in the path. For example, c:\\Program Files\\SAS\\.	Extract

Parameter	Purpose	Mode
-excludePrompts	Optional. Excludes prompts from extraction so that the output file (specified by -outputFile) contains only report URLs, parameter definitions, and their default values. This is useful when you want to generate a listing of a folder, and then edit that listing. Subsequently, you can use the edited file as input for another extract.	Extract
-recursive	Optional. Extracts information about the reports recursively so that all reports that are contained in a directory and in its subdirectories are included.	Extract

Configuring the Batch Generation Tool

To most effectively use the Batch Generation Tool, you must construct the BatchExecution command using information that is provided at installation. To do so, follow these steps:

- 1 The parameters that are needed to construct the BatchExecution command can be found in the **batchgen.ini** file. During the installation of SAS Query and Reporting Services, the batchgen.ini file is configured relative to the installation location. If you build the BatchExecution command using the unaltered values from the batchgen.ini file, you must run the BatchExecution command from the directory that contains the batchgen.ini file.

For example, installing SAS Query and Reporting Services on a Windows machine in the default location (c:\Program Files\SAS\SASQueryandReportingServices\9.1) produces the following batchgen.ini file:

```

;-----;
; The type of launcher behavior to execute ;
;-----;
[Launcher Type]
LauncherType=Launcher

;-----;
; LauncherType=Launcher ;
;-----;
[Launcher]

;-----;
; The command to execute (must be a fully qualified path) ;
; e.g.: C:\j2sdk1.4.1\bin\java.exe ;
;-----;
CommandToExecute=C:\Program Files\SAS\Shared Files\JRE\1.4.1\bin\java.exe

;-----;
; Any arguments to pass to the above command ;
; e.g. (for SAS SMC): ;
; -Djava.ext.dirs= -cp sas.smc.jar;. com.sas.console.visuals.MainConsole ;
;-----;
CommandLineArgs=-Djava.system.class.loader=com.sas.app.AppClassLoader

```



```

-Dsas.app.class.dirs="C:\Program Files\SAS\SASQueryandReportingServices\9.1"
-Dsas.app.class.path=sas.report.render.jar;.
-cp sas.launcher.jar
-Djava.security.policy=.\policy
-Djava.security.auth.policy=.\auth.policy
-Dcache.auth.policy=true
-DPFS_TEMPLATE=.\sas_service_deployment_export_queryandReporting_BatchGenerationTemplate.xml
-Djava.security.auth.login.config=.\login.config
  com.sas.report.render.batch.BatchExecution

;-----;
; The working directory the application specified by the above command ;
; should start in, e.g.: C:\Program Files\SAS\JDMS\9.1 ;
;-----;
WorkingDirectory=C:\Program Files\SAS\SASQueryandReportingServices\9.1

```

- 2 Using the **CommandToExecute** and **CommandLineArgs** parameters from the **batchgen.ini** file, construct the **BatchExecution** command (do not insert line breaks in the actual command):

```

"C:\Program Files\SAS\Shared Files\JRE\1.4.1\bin\java.exe"
-Djava.system.class.loader=com.sas.app.AppClassLoader
-Dsas.app.class.dirs="C:\Program Files\SAS\SASQueryandReportingServices\9.1"
-Dsas.app.class.path=sas.report.render.jar;.
-cp sas.launcher.jar
-Djava.security.policy=.\policy
-Djava.security.auth.policy=.\auth.policy
-Dcache.auth.policy=true
-DPFS_TEMPLATE=.\sas_service_deployment_export_queryandReporting_BatchGenerationTemplate.xml
-Djava.security.auth.login.config=.\login.config
  com.sas.report.render.batch.BatchExecution

```

- 3 Append to the command the relevant extract or run mode information. For example:

```

run
-configFile "C:\Program Files\SAS\SASQueryandReportingServices\9.1\config1.properties"
-url "SBIP://Foundation/ReportStudio/Shared/Reports/Report.srx"

```

- 4 To keep the console window open long enough to read the results, append a newline and a **pause** command.

The full command is now:

```

"C:\Program Files\SAS\Shared Files\JRE\1.4.1\bin\java.exe"
-Djava.system.class.loader=com.sas.app.AppClassLoader
-Dsas.app.class.dirs="C:\Program Files\SAS\SASQueryandReportingServices\9.1"
-Dsas.app.class.path=sas.report.render.jar;.
-cp sas.launcher.jar
-Djava.security.policy=.\policy
-Djava.security.auth.policy=.\auth.policy
-Dcache.auth.policy=true
-DPFS_TEMPLATE=.\sas_service_deployment_export_queryandReporting_BatchGenerationTemplate.xml
-Djava.security.auth.login.config=.\login.config

```

```

    com.sas.report.render.batch.BatchExecution
run
-configFile "C:\Program Files\SAS\SASQueryandReportingServices\9.1\config
1.properties"
-url "SBIP://Foundation/ReportStudio/Shared/Reports/Report.srx"

pause

```

5 Save the command to a .cmd file for simple execution.

Ensure that the .cmd file is saved with the encoding that matches the default codepage for the command-prompt, or change the default codepage for the command-prompt in which the .cmd file is executed so that it fits the encoding of the .cmd file. To change the codepage from within the .cmd file, add to the top of the file the `chcp` command followed by a newline. For example, `chcp 865`.

Viewing and Editing Batched Reports

If a batched report exists, it is returned by default when a user requests a report on which it is based. If the user edits the report, and has the permissions required to do so, then the batched report is replaced in the Web browser with the refreshed, edited report. If the user saves changes to this new, edited report, the existing batched report is deleted. If the edited report is saved with the manually refresh option, or with a different name, a new batch version of the report is generated and saved with the edited or renamed report.

If the user saves the edited report under a different name, then the batched report that is based on the unedited report is not deleted, but the new report does *not* have a batched report associated with it. If the edited report is saved under a different name as a manually refreshed report, a new batched report *is* created. Users can also produce a batched report that is based on the new report, by using the Batch Generation Tool.

If the new report is saved with the automatic refresh option, no new batched report is created, the original batched report is deleted, and no batched report is associated with the new report.

Example Batch Generation Tool Configuration File

The following is an example config file for the Batch Generation Tool. Note the escaped back slashes. If the URL has national characters (such as æ, ø, and à), the config file must be saved in UTF-8 format.

```

username=omruser
password=DemoDemol
repository=Foundation
host=svr01.xyz.sas.com
port=8561
workspaceserver=Pooled Workspace Server - Logical Workspace Server
url=SBIP://Foundation/MyDir
outputFile=c:\\Program files\\SAS\\SASQueryandReportingServices\\9.1\\output.lst

```

Example 1: Extracting a Report

Executing the following command (as one line, without any line breaks), extracts information about MyReport.srx in the MyDir folder:

```

java com.sas.report.render.batch.BatchExecution extract
-configFile "c:\batch.cfg"
-URL "SBIP://MyDir/MyReport.srx"

```

```
-outputFile "c:\batch-reports.lst"
```

Upon completion, the output file (c:\batch-reports.lst) contains information about the locale the file is extracted from, the location of the report, and a description of prompt values that are saved within the report:

```
# Prompt values in an extracted list will be formatted
# according to the Locale at the extraction time.
# An exception is Prompts of type Date which always should be
# in ddMMMyyyy format, with en_US Locale.

LOCALE=en_US

URL1.report=SBIP://MyDir/MyReport.srx
# URL1.parm.SexUserInput
# Description :
# Type : java.lang.String
URL1.parm.SexUserInput.value=F
```

Example 2: Extracting a Directory of Reports

In this example, the `-ur1` parameter specifies a directory that contains the three reports: `MyReport.srx`, `MyReport2.srx`, and `MyReport3.srx`. Executing the following command (as one line, without any line breaks), extracts information about all the reports in the SBIP folder `MyDir`:

```
java com.sas.report.render.batch.BatchExecution extract
  -configFile "c:\batch.cfg"
  -URL "SBIP://MyDir"
  -outputFile "c:\batch-reports.lst"
```

Upon completion, the output file (c:\batch-reports.lst) contains information about all the reports in the `SBIP://MyDir` folder, including the prompts for the reports, and their values. Notice that `MyReport2.srx` does not contain any prompts, and the `DatePrompt` for `MyReport3.srx` has a list of Valid Values.

```
# Prompt values in an extracted list will be formatted
# according to the Locale at the extraction time.
# An exception is Prompts of type Date which always should be
# in ddMMMyyyy format, with en_US Locale.

LOCALE=en_US

URL1.report=SBIP://MyDir/MyReport.srx
# URL1.parm.SexUserInput
# Description :
# Type : java.lang.String
URL1.parm.SexUserInput.value=F

URL2.report=SBIP://MyDir/MyReport2.srx

URL3.report=SBIP://MyDir/MyReport3.srx
# URL3.parm.DatePrompt
# Description :
# Type : java.sql.Date (Format = ddMMMyyyy, Locale = en_US)
# Valid Values : [16Jan2000, 13Jan2000, 15Jan2000, 02Jun2000, 03Jun2000]
URL3.parm.DatePrompt.value=16Jan2000
```

Example 3: Extracting While Excluding Prompts

The `-excludePrompts` option can be used to extract information about reports while excluding prompt information. In this example, the `-url` parameter specifies a directory that contains three reports: `MyReport.srx`, `MyReport2.srx`, and `MyReport3.srx`. Executing the following command (as one line, without any line breaks), extracts information about all the reports in the `MyDir` folder, but does not include any of the prompt information:

```
java com.sas.report.render.batch.BatchExecution extract
  -configFile "c:\batch.cfg"
  -URL "SBIP://MyDir"
  -outputFile "c:\batch-reports.lst"
  -excludePrompts
```

Upon completion, the output file (`c:\batch-reports.lst`) contains the following (compare to the output in Example 2):

```
# Prompt values in an extracted list will be formatted
# according to the Locale at the extraction time.
# An exception is Prompts of type Date which always should be
# in ddMMMyyyy format, with en_US Locale.

LOCALE=en_US

URL1.report= SBIP://MyDir/MyReport.srx

URL2.report= SBIP://MyDir/MyReport2.srx

URL3.report= SBIP://MyDir/MyReport3.srx
```

Example 4: Extracting Recursively

The `-recursive` option can be used to recursively extract information about reports from a folder and its subfolders. Executing the following command (as one line, without any line breaks), extracts information recursively:

```
java com.sas.report.render.batch.BatchExecution extract
  -configFile "c:\batch.cfg"
  -URL "SBIP://MyDir"
  -outputFile "c:\batch-reports.lst"
  -excludePrompts
  -recursive
```

If `MyDir` contains the three reports `MyReport.srx`, `MyReport2.srx` and `MyReport3.srx`, and a subfolder named "SubFolder" contains the two reports `MyReport4.srx` and `MyReport5.srx`, the output file (`c:\batch-reports.lst`) contains the following:

```
# Prompt values in an extracted list will be formatted
# according to the Locale at the extraction time.
# An exception is Prompts of type Date which always should be
# in ddMMMyyyy format, with en_US Locale.

LOCALE=en_US

URL1.report=SBIP://MyDir/MyReport.srx

URL2.report=SBIP://MyDir/MyReport2.srx
```

```

URL3.report=SBIP://MyDir/MyReport3.srx

URL4.report=SBIP://MyDir/SubFolder/MyReport4.srx

URL5.report=SBIP://MyDir/SubFolder/MyReport5.srx

```

Example 5: Using Edited Extract Data as Input for Run Mode

You can edit the output of one extraction, use that edited output as the input for another extraction, and finally use the resulting output from that extraction in a run-mode operation. In this example, the output from Example 4 is edited to include only three reports, and then saved as `c:\batched-reports-input.lst`, which is then used as input for a run mode command. The edited file contains the following data:

```

# Prompt values in an extracted list will be formatted
# according to the Locale at the extraction time.
# An exception is Prompts of type Date which always should be
# in ddMMMyyyy format, with en_US Locale.

LOCALE=en_US

URL2.report=SBIP://MyDir/MyReport2.srx

URL3.report=SBIP://MyDir/MyReport3.srx

URL4.report=SBIP://MyDir/SubFolder/MyReport4.srx

```

With the `-file` option, use the edited report list as input for the `extract` command. When run, this command will generate a report list with prompt information for reports 2, 3, and 4 only:

```

java com.sas.report.render.batch.BatchExecution extract
  -configFile "c:\batch.cfg"
  -file "c:\batched-reports-input.lst"
  -outputFile "c:\batched-reports.lst"

```

The resulting file contains the following:

```

# Prompt values in an extracted list will be formatted
# according to the Locale at the extraction time.
# An exception is Prompts of type Date which always should be
# in ddMMMyyyy format, with en_US Locale.

LOCALE=en_US

URL1.report=SBIP://MyDir/MyReport2.srx

URL2.report=SBIP://MyDir/MyReport3.srx
# URL2.parm.DatePrompt
# Description :
# Type : java.sql.Date (Format = ddMMMyyyy, Locale = en_US)
# Valid Values : [16Jan2000, 13Jan2000, 15Jan2000, 02Jun2000, 03Jun2000]
URL2.parm.DatePrompt.value=16Jan2000

URL3.report=SBIP://MyDir/SubFolder/MyReport4.srx
# URL3.parm.SexUserInput

```

```
# Description :
# Type : java.lang.String
URL3.parm.SexUserInput.value=M
```

You can now edit the prompt values of the report (for example, change the DatePrompt of MyReport3.srx to 02Jun2000) before you use the report list as input for a run command. Lines that start with # are comments, and are not processed in run mode.

To use the **batched-reports.lst** file to generate reports, use the **-file** option, as in the following command:

```
java com.sas.report.render.batch.BatchExecution run
  -configFile "c:\batch.cfg"
  -file "c:\batched-reports.lst"
```

The result of the command is that batched reports are generated for all reports and directories listed in the **batched-reports.lst** file, using the values for prompt parameters defined in the file, or, if a value is not present in the file, using the value currently saved in the repository within a report.

Example 6: Running a Single Report

To run a single report via the **-url** option, execute the following command (as one line, without any line breaks):

```
java com.sas.report.render.batch.BatchExecution run
  -configFile "c:\batch.cfg"
  -url "SBIP://MyDir/MyReport.srx"
```

A batch version of the report at **//MyDir/MyReport.srx** is generated and stored in the report repository.

Example 7: Running a Directory of Reports

You can also specify a directory with the **-url** option. To generate batched reports for all reports in a directory, execute the following command (as one line, without any line breaks):

```
java com.sas.report.render.batch.BatchExecution run
  -configFile "c:\batch.cfg"
  -url "SBIP://MyDir"
```

Example 8: Running Without the **-configFile** Option

The options specified in the configuration file via the **-configFile** option can also be entered at the command line. Values entered at the command line override values in the configuration file. If all options are entered at the command line, the **-configFile** option can be omitted, for example:

```
java com.sas.report.render.batch.BatchExecution run
  -username "omruser"
  -password "DemoDemo1"
  -domain "EUROPE"
  -repository "Hope"
  -host "svr01.xxx.yyy.com"
  -port "6411"
  -workspaceserver "IOM - Logical Workspace Server"
```

```
-URL "SBIP://MyDir"
```

Registering Fonts for Use from SAS Web Report Studio

When printing or creating PDF files from SAS Web Report Studio, the fonts specified in the report must be registered with the SAS server. Fonts that are not registered cannot be used for output. The following code registers all the available fonts:

```
PROC FONTREG mode=all;
  FONTPATH 'c:\winnt\fonts'; /* Path to the system fonts on the server */
RUN;
```

Printing Non-Latin1 Languages from SAS Web Report Studio

To print non-Latin1 characters from SAS Web Report Studio to an Output Delivery System (ODS) destination, you must designate in the `styles.xml` file the fonts capable of rendering the appropriate glyphs. To make such an alternate font available, follow these steps:

- 1 Register the host fonts with the ODS Printer using Proc FONTREG. The following code registers the fonts in the host's default fonts path. Add additional fontpath statements if you have fonts stored in multiple locations.

```
PROC FONTREG;
  FONTPATH '?CSIDL_FONTS';
RUN;
```

- 2 Change the default assignments for font names within the printer registry. This change enables SAS Web Report Studio to use the newly registered fonts rather than the default printer-resident fonts. To change the default assignments, place the following information in a flat file:

```
[CORE\PRINTING\ALIAS\FONTS\PDF]
"Trebuchet MS"    = "<ttf> Arial"
"Arial"           = "<ttf> Arial"
"Times New Roman" = "<ttf> Times New Roman"
```

- 3 Inside SAS, change the printer registry to import the fonts specified in the flat file by using the following code:

```
proc registry import='<flatfileName>'; run;
```

Both changes above are stored in the SAS SASUSER information (as opposed to the SAS Web Report Studio SASUSER information), and remain in effect until SASUSER is removed.

Using Xythos Software's WebFile Server with SAS Web Report Studio

You can use Xythos Software's WebFile Server as a DAV repository when using SAS Web Report Studio, but you must create the necessary directory structure. When Xythos was installed, a `/sasdav` directory was created for you. You must create (using the Xythos administration interface) the required `/wrs` directory inside the `/sasdav` directory. This is in contrast to using Apache HTTP Server, Tomcat, or Microsoft IIS in which the creation of the directory is accomplished through the operating system. For more information on using Xythos Software's WebFile Server, refer to the Xythos Software documentation.

Customizing the SAS Web Report Studio Banner and Title

You can configure SAS Web Report Studio to display your company's logo and Web page title by specifying values in the `<webreportstudio>` section of the `WebReportStudioProperties.xml` deployment file. This file is located on the server on which SAS Web Report Studio is deployed and executing. If you are using Tomcat, this location should be similar to `\Tomcat\webapps\SASWebReportStudio\WEB-INF\`. If you are using WebLogic with Windows, this location should be similar to `\bea\webapps\SASWebReportStudio`. The properties that are available in the `<webreportstudio>` section are described in the following table.

Table 17.2 Properties for Customizing SAS Web Report Studio

Property	Purpose
<code>product.logo.url</code>	A reference to an image with a height of 56 pixels. The logo must be in the top 34 pixels of the image so that the tabs do not overlap the logo (see Figure 17.2 on page 329).
<code>product.logo.text</code>	The window title and the tooltip text for the <code>product.logo</code> image.
<code>company.logo.url</code>	A reference to an image with a height of 34 pixels.
<code>company.logo.text</code>	The tooltip text for the <code>company.logo</code> image.
<code>banner.tile.url</code>	A reference to an image with a height of 63 pixels. This image is tiled between the <code>product.logo</code> image and the <code>company.logo</code> image. This image should be a dark color so that the View Report tab and Create/Edit Report tab are easily visible (see Figure 17.2 on page 329).

Specifying an Image Reference

There are three ways to reference an image for use in the `<webreportstudio>` section of the `WebReportStudioProperties.xml` deployment file:

banners/companyLogo.jpg

The preferred way to reference an image. Create a folder called "banners" in the SAS Web Report Studio deployment area, and place the referenced images in that folder.

http://www.xyz.com/images/companyLogo.jpg

Referencing an image with an HTTP protocol that is different from the protocol that was used to access SAS Web Report Studio (for example, using HTTPS to access SAS Web Report Studio and using HTTP to access the images) will result in security warning messages. To avoid such messages, either reference the banner images using a method other than HTTP and HTTPS, or ensure that all users access SAS Web Report Studio using the same protocol, and use that protocol to reference the banner images.

file:///C:/public/banners/companyLogo.jpg

Using the `C:/` syntax to reference an image should be used only in test or verification environments because an image referenced using `C:/` is available only from the server machine on which SAS Web Report Studio is deployed. However, if the disk is shared as a network drive, the image is available to any machine that accesses that drive using the standard Windows shared drive syntax of two

leading forward slashes. Thus, the following references are acceptable in a production environment:

- `//Machine123/public/images/companyLogo.jpg`
- `file:///Machine123/public/images/companyLogo.jpg`

Example: Customizing the SAS Web Report Studio Banner

The following is a sample `<webreportstudio>` section of the `WebReportStudioProperties.xml` file:

```
<webreportstudio>
  <product.logo>
    <url>file:///machine/public/images/imageTest/productLogo.jpg
    </url>
    <text>Product.Logo Text
    </text>
  </product.logo>
  <company.logo>
    <url>http://www.sas.com/includes/headers/images/companyLogo.jpg
    </url>
    <text>Company.Logo Text
    </text>
  </company.logo>
  <banner.tile.url>file:///machine/public/images/imageTest/tile.jpg
  </banner.tile.url>
</webreportstudio>
```

The XML above produces the following banner (for comparison, the default banner is also shown).

Figure 17.2 A Customized Banner in SAS Web Report Studio

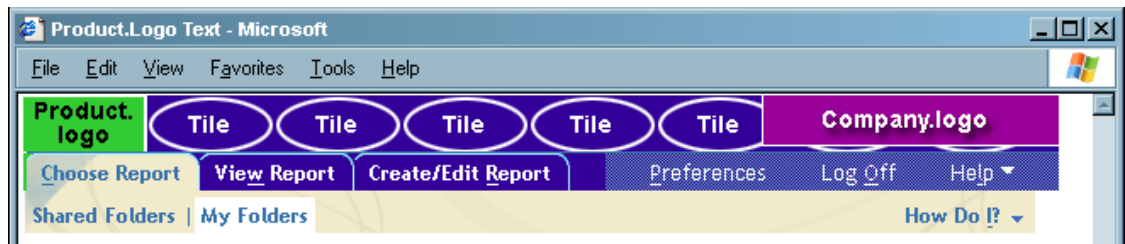
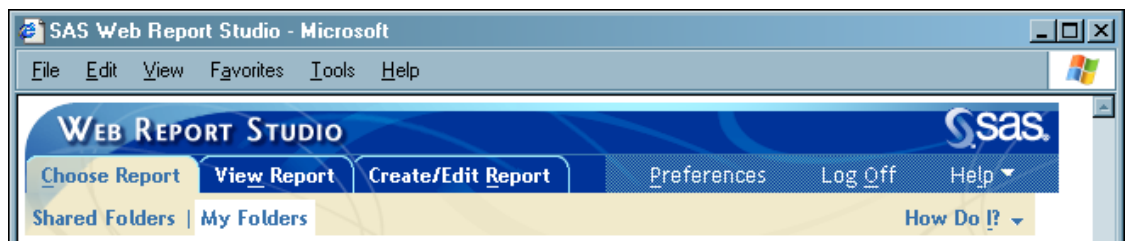


Figure 17.3 The Default Web Report Studio Banner



You do not need to restart the server to change an existing image that is referenced while SAS Web Report Studio is running; refreshing your browser will update the image. However, to use a different image name (filename or URL) or text value, you must restart the server on which SAS Web Report Studio is executing.

Printing Large Reports

To print large reports, the Java memory settings might need to be adjusted. To increase the amount of memory that is accessible to the Java Virtual Machine (JVM), follow these steps:

- 1 In SAS Management Console, navigate to **Server Manager/SASMain/SASMain - Logical Workspace Server/SASMain - Workspace server**.

Note that the server name **SASMain** might have been changed at installation.

- 2 Select the **SASMain - Workspace server**, and then select **File ► Properties**.
- 3 Click the **Options** tab.
- 4 In the **Command** field, append the following code to the existing **sas** startup command:

```
-JREOPTIONS '-Xms1024m -Xmx1500m'
```

- 5 Click the **OK** button.

JREOPTIONS passes memory settings to the JVM running inside of the spawned SAS session. The **-xms** value is the initial heap size, and the **-xmx** value is the maximum heap size. Increasing the heap size should improve printing performance overall, even when you are not printing large reports. Do not set either of these values to a value that is larger than the amount of RAM available on the machine.

Adjusting JVM Memory

The *vm.args* file both describes and specifies the arguments passed to the JVM. Each client that starts the JVM has a **vm.args** file, or something that functions in its place. For a standard install of SAS Web Report Studio, the *vm.args* file is located in **C:\Program Files\SAS\SASWebReportStudio\9.1\wrsstaging\config**.

Increasing the Memory Available to the Tomcat JVM

If you experience an out of memory error when running multiple applications in the same servlet container (such as the SAS Information Portal and SAS Web Report Studio), try increasing the amount of memory available to the Tomcat JVM **permanent area**, a block of memory that holds classes and other permanent objects. The JVM uses only the amount of memory that it actually needs to load the objects. If additional memory is not needed, the JVM does not increase the size of the permanent area to the specified value.

To increase the Tomcat JVM permanent area memory allocation, edit the file that is used to start Tomcat, **startServletContainer.bat** (which is located in a default Windows installation in **\SAS\BIEntServerMin\Lev1\web**). Add the following option to the **set CATALINA_OPTS** line:

```
-XX:MaxPermSize=128M
```

Writing ODS Output to a Report Repository

By using the SAS Report XML tagset and the SASXPGRP access method on the FILENAME statement, you can write ODS output directly to a repository. SAS Web

Report Studio will view the output as a report, allowing a user to view, move, rename, and delete the output like any other report. However, the report cannot be edited from SAS Web Report Studio.

When using the SASXPGRP access method on the FILENAME statement, an SBIP URL is used as the external file that you want to write to. If the specified repository entry already exists, it is overwritten. The SBIP URL can refer to a directory instead of to a specific file. This is useful when a process generates multiple files to the same location. A trailing slash in the SBIP URL is required when specifying a directory.

The following options to the SASXPGRP access method are required unless otherwise indicated:

userid="userID"

The userID to access the server.

password="password"

The password to access the server.

domain="domain"

The domain name for the server.

OMRHost="host"

The network name of the machine hosting the metadata repository.

OMRPort="nnnn"

The port number for accessing the repository.

OMRUser="userID"

The userID to access the repository. This can be the same as the server userID, or it can be different.

OMRPassword="password"

The password to access the repository. This can be the same as the server password, or it can be different.

OMRReposName="name"

The name of the repository.

Example: Writing ODS Output to a Repository

The following code outputs SAS Report XML to the specified repository:

```
filename dest sasxpgrp "SBIP://RepName/Bip Tree/ReportStudio/Users/xyz/Reports"
  userid="xyz" password="bip2004" domain="thisDomain"
  OMRHost="bipsvrxyz.na.sas.com" OMRPort="9999" OMRUser="xyz"
  OMRPassword="bip2004" OMRReposName="RepName"
;

option noovp;
ods sasreport file="myreport.xml" path=dest;
proc print data=sashelp.class; run;
ods sasreport close;
```

Using a File-Based Content Server

Using a file-based content server can reduce server overhead and result in faster response times. This advantage is because the report read and write requests no longer passing through an HTTP/DAV server. Although several variables influence how much of an improvement can be realized, a performance increase of 10–15% is typical.

However, the performance increase comes at the cost of flexibility. A DAV-based content server enables access to content without direct operating system support or shared network areas. This type of access is especially important if an installation requires that content be accessible by tools or applications running on several and/or widely dispersed machines. In such a diverse environment, a DAV-based content server is most likely a necessity. If content is accessed only from one machine or a very small number of machines, sharing the content space may not be an issue, and a file-based content server with the resulting performance increase is the better choice.

To configure SAS Web Report Studio to use a file-based content server, follow these steps:

- 1 In SAS Management Console, navigate through Business Report Manager to the root folder (which might or might not already have a content server defined for it).
- 2 Select the root folder, and then select **File ► Properties**.
- 3 In the **Content Server** field, select **[FILE SYSTEM]**.
- 4 In the **Content Base Path** field, type the full path to the content directory.
If the content needs to be accessible from applications that are running on multiple machines, this path must be network accessible.
- 5 Click the **OK** button.

When using a file-based server, authentication is performed by the server operating system and the **Content Server Authentication** fields in the properties dialog box have no effect.

Although the metadata server does not need to be restarted, any client that uses the reports (notably, SAS Web Report Studio) should be restarted.

Managing Stored Processes

Configuring Stored Processes to Work with SAS Web Report Studio

Stored processes, via ODS, are frequently used to create text and graphical output that are shown in a client application or that are associated with an information map (via SAS Information Map Studio) to pre-process data prior to the reporting of that data. Regardless of the intended use, existing SAS programs can easily be converted to stored processes.

Stored processes must be registered in the SAS Open Metadata Repository, either by an administrator using SAS Management Console, or by users through SAS Enterprise Guide. For more information on creating and registering stored processes, see support.sas.com/rnd/itech/doc9/dev_guide/stprocess/. For information on securing stored processes, see “Access Requirements for Working with Stored Processes” on page 206.

SAS Web Report Studio can execute a stored process that is chosen from the Reports Selection window and have it rendered in your browser, or it can use a stored process as a section of a report. Despite how stored processes are used, if you want to modify the output that is generated by a stored process, you must modify the stored process source code.

Converting an Existing SAS Program to a Stored Process

Existing SAS programs that create reports can be converted to stored processes so that they can be used in SAS Web Report Studio. The programs can also be

parameterized, enabling users to input data when prompted. Prompted parameter values are transferred to the stored process as macro variables.

All output from the stored process that is intended for use in a report must be generated through the Output Delivery System (ODS). Output that is generated in other ways, such as with PUT statements, would appear in a SAS log or another external file, but would not be part of the report data that is accessible from SAS Web Report Studio. Additionally, the ODS output type cannot be controlled by the stored process code (neither by setting the value of the stored process input parameter `_RESULT`, nor by explicit ODS statements). Controlling the ODS output type is determined by the manner in which the stored process is registered and executed.

To convert an existing program to a stored process, follow these steps:

- 1 Insert a `*ProcessBody` statement as the first line of the program.
- 2 Insert a `%stpbegin` statement prior to a section of the code that produces output.
- 3 Insert a `%stpend` statement after a section of the code that produces output.
- 4 Store the program on the server and register it as a stored process, either from SAS Management Console or SAS Enterprise Guide.

Programs that do not produce output (such as programs that only update data) do not need `%stpbegin` or `%stpend` statements.

Example of Converting a SAS Program to a Stored Process

The original SAS program:

```
Title "Sports & Outdoors Sales 2002";
Proc print data=sashelp.orsales;
  Where year=2002;
Run;
```

The same program altered to become a stored process:

```
*ProcessBody;
%stpbegin;

Title "Sports & Outdoors Sales &year";
Proc print data=sashelp.orsales;
  Where year=&year /* &year is a parameter from a user prompt */
Run;

%stpend;
```

For more information on registering stored processes in SAS Management Console, see the SAS Management Console Help, which is available from within the product, or the documentation on the Web, which is available at support.sas.com/rnd/itech/doc9/admin_oma/stprocess/. If you are using SAS Enterprise Guide to register a stored process, see the SAS Enterprise Guide Help, which is available from within the product.

Requirements for Using Stored Processes in SAS Web Report Studio

When building a new report and adding a stored process section, SAS Web Report Studio searches for available stored processes in a folder called `StoredProcesses` in the shared reports location. Although the shared reports location is configurable, if you used the default location at installation, the `StoredProcesses` folder is in `BIP Tree/ReportStudio/Shared/Reports/`. Although this is the initial search location, a SAS Web Report Studio user can navigate up the hierarchy one level and search the shared reports to find additional stored processes.

When accessing stored processes as reports from the **Choose Report Page**, a user has access to any stored process in the folder hierarchy for which they have permissions.

Stored Process Output Style

Stored processes that use the `%stpbegin` and `%stpend` macros in the source code are formatted with the ODS. For text output (such as PROC PRINT listings) this causes the output to use the same style as is set in the user's SAS Web Report Studio preferences, for example "Seaside".

For graphical output (like PROC GCHART graphs), SAS Web Report Studio defaults to an ActiveX device: `Options device=ActiveX;`. Using the ActiveX device also enables you to render stored process graphs from the SAS Add-In to Microsoft Office. The `%stpbegin` macro also defaults to ActiveX, ensuring that stored processes that do not specify a device are compatible between SAS Web Report Studio and the SAS Add-In to Microsoft Office.

Associating Stored Processes with Information Maps

Stored processes that are associated with an information map are executed prior to any queries generated against that information map. This processing order enables you to use SAS tools such as the DATA step or the macro language to process the data used as input for the information map.

Because pre-processing data usually involves subsetting or updating the data on a per-user basis (for instance, when a user is prompted to enter parameters), the WORK library of the IOM Workspace Server can be used to store temporary copies of report data on a per-user basis so that no original data is over-written, and multiple, concurrent users of a stored process each have access to a private version of the resulting data. As a result, although the stored process reads the data from a permanent data source, it writes the data to the WORK library of the IOM Workspace Server. Similarly, the information map references permanent metadata that describes the data from the stored process.

To associate a stored process with an information map, follow these steps:

- 1 Write and execute the SAS program that modifies the existing data and creates the custom data sets. Use a permanent library to store the custom data sets. These data sets need only to contain header information about the columns and their attributes; rows are not required.
- 2 Register in the metadata the custom data sets and the library in which they reside, using either SAS ETL Studio or the metadata LIBNAME engine.
- 3 Convert the SAS program from step 1 to a stored process by inserting a `*ProcessBody` statement as the first line of the program. You do not need to include the `%stpbegin` or `%stpend` statements.
- 4 Insert into the SAS program a LIBNAME statement that references the location of the custom data sets.
- 5 Insert another LIBNAME statement to write the data from the stored process to the WORK library.

Because output that is concatenated to a library is written to the first LIBNAME entry listed, list the WORK library first to ensure that the custom data set is written to the WORK library. For example:

```
*ProcessBody;
Libname source 'path-to-source-data';
Libname custom (work); /* Custom library and datasets must already be */
```

```

/* registered in metadata */

Data custom.result_set;
  Set source.data1;
  /* more code */
run;

```

- 6 Save the source code of the stored process. Register it for execution in the IOM Workspace Server with an output type of NONE using either SAS Management Console or SAS Enterprise Guide.
- 7 Open SAS Information Map Studio and create an information map that is based on the data in the custom library.
- 8 In SAS Information Map Studio, select **Tools ► Set Stored Process** and then select the stored process that you created in step 6. Save the information map. The stored process is now associated with the information map.

The information map with the associated stored process can now be used for reporting.

Securing Your Reporting Environment

Although the ReportStudio folder structure that is automatically created by SAS Web Report Studio automatically implements a basic level of security by creating the Shared and Users folders, additional steps should be taken in order to protect your reports. When securing a report, you must consider both how to protect the report definition and how to protect the underlying report data. Because the type of report determines what security measures must be taken, you must consider whether the report is automatically refreshed, manually refreshed, or a batched report.

If a report's underlying data, information maps, stored processes, and output are secure, the report is considered secure. Although there is no embedded data of a sensitive nature in a report definition, batched reports and some reports created through ODS can contain data, and should be secured through the operating system. For more information on how users and administrators should secure their report content and resources, see Chapter 12, "Developing Your Security Plan," on page 193.

You can apply security in the metadata environment as well as in the physical data storage environment. The following table identifies the basic security considerations for reports.

Table 17.3 Report Security Considerations

In order to protect	You must secure
report definitions	<ul style="list-style-type: none"> <input type="checkbox"/> the metadata objects that are associated with the reports <input type="checkbox"/> the physical location in the report repository that contains the report definitions
underlying report data	<ul style="list-style-type: none"> <input type="checkbox"/> the metadata objects that are associated with the report data <input type="checkbox"/> the physical storage location of the report data <input type="checkbox"/> the information maps that reference the report data <input type="checkbox"/> the stored processes that reference the report data

If your organization uses publication channels to deliver reports, the reports can also be secured by setting access controls on the publication channel.

For additional information about securing reports, see “Securing Batched Reports” on page 336.

For specific information about the access controls that are recommended for securing reports, stored processes, and information maps within the metadata environment, see “Access Requirements for Common Tasks” on page 205. For information about securing the physical storage locations of your data and your report repository, see your operating system or data storage software documentation.

Securing Information Maps

Because information maps can reference many physical data sources, it is recommended that the information maps themselves be secured by setting access controls on the folders in which the information maps are saved. Securing the information maps with folder permissions enables you to group information maps that should have similar access controls. For example, most of the employees in a Human Resources department would likely use the same information maps, and see the same data. Therefore, you might define a group of Human Resources users and a folder for Human Resources information maps. If you need to restrict certain users from sensitive data such as payroll information, you can then set access controls for the user group on the Human Resources folder and then limit the access that certain users have to the information maps within that folder. For more information on users and groups, see “Planning Your User Groups” on page 201. For more information on setting access controls for information maps, see “Access Requirements for Working with Information Maps” on page 207.

Securing Batched Reports

Batched reports contain data that has already been generated from the report’s underlying information maps. Because viewing a batched report does not invoke the security that is placed on an information map or stored process, users who are not

granted access to an information map or stored process might be able to view a batched report that is generated from those sources.

For batched reports you must design report-level security to parallel the security set on the information maps and stored processes that are used to generate the report. Do not rely on restricting access to the underlying information maps or stored processes to ensure that batched reports are viewed only by the appropriate users.

For information about assigning permissions to users, see “Planning Your Access Controls” on page 204.

Securing Temporary Files

As part of typical operation, SAS Web Report Studio writes temporary files to the middle tier (the server on which SAS Web Report Studio is executing). Specifically, temporary files are stored in three locations:

- Two folders, `tmpnull` and `tmpuser`, are created in the folder where SAS Web Report Studio is deployed. If you are using Tomcat, this location should be similar to `Tomcat\webapps\SASWebReportStudio\`. If you are using the BEA WebLogic Server with Windows, this location is most likely `bea\webapps\SASWebReportStudio`.
- In Java’s temporary folder (as defined by the Java property `java.io.tmpdir`).

These temporary files might contain data that needs to be secured. To secure these temporary files, the middle tier must be secured both by physical location and by password protection via the operating system. In addition, the folders that contain the temporary files must be secured so that only essential users have access. This can be accomplished by changing permissions on the root level folders via the operating system. It is assumed that a system administrator who should not necessarily have access to the data will be able to view the temporary files, and that this is acceptable.

Using SAS Web Report Studio with SSL

To enhance the security of the reporting environment, you can choose to use Secure Sockets Layer (SSL). SSL is a protocol for client/server communication that prevents tampering, forgery, and eavesdropping. If you are deploying SAS Web Report Studio as part of the SAS Information Portal, consult the documentation for the SAS Information Portal.

If you are running SAS Web Report Studio as a stand-alone service, you can enable SSL by following these steps:

- 1 Get an SSL certificate from a certifying authority.

For testing purposes, you can generate a certified key using the `keytool` command available in the `JSDK\bin` directory.

To use the `keytool` command with Windows 2000 and Windows XP, enter the following command and answer the questions as they appear on the screen:

```
keytool -genkey -alias tomcat -keyalg RSA
```

To use the `keytool` command with Unix, assuming that Tomcat is installed at `/usr/share/tomcat4`, enter the following command with no line breaks:

```
JAVA_HOME/bin/keytool -genkey -alias tomcat -keyalg RSA
-keystore /usr/share/tomcat4/.keystore
```

- 2 Enable the SSL listening port on the J2EE Application Server.

If you are using Tomcat, open the `\Tomcat_Home\conf\server.xml` file and uncomment the block under

```
<-- Define an SSL HTTP/1.1 Connector on port 8443 -->
```

If you are using the BEA WebLogic Server, use the WebLogic Console to enable and specify a listen port.

- 3 Enable SSL in the Web application by adding a `<user-data-constraint>` section to the `web.xml` file as follows:

```
<security-constraint>

  <user-data-constraint>
    <transport-guarantee>Confidential</transport-guarantee>
  </user-data-constraint>
</security-constraint>
```

For more information on configuring the BEA WebLogic Server, see “Configuring Security in Web Applications” at e-docs.bea.com/wls/docs70/webapp/security.html.

For more general information on implementing security, see

- “Web FORM-Based Authentication” at www.onjava.com/pub/a/onjava/2001/08/06/webform.html
 - “Specifying Security Constraints” at java.sun.com/webservices/docs/1.3/tutorial/doc/Security4.html
- 4 Direct your Web browser to SAS Web Report Studio using the HTTPS protocol via the SSL listening port, for example `https://myserver:8443/SASWebReportStudio`.

After you access SAS Web Report Studio via HTTPS, HTTPS is used for the entire session.

Preventing the DAV Navigator Portlet from Viewing Unauthorized Data

Because the DAV navigator portlet does not use metadata permissions as the report navigator portlet does, a user can, on a publicly accessible DAV location, navigate to another user’s report folder using the DAV navigator portlet and view the XML of a report. To prevent this intrusion, protect the `/sasdav/wrs` area with a user ID and password, and deny public access to it.

To ensure that SAS Web Report Studio users have access to the DAV area, enter the user ID and password of the DAV area in the Business Report Manager. The report navigator portlet will automatically access and use the needed user ID and password from the metadata. The DAV navigator is not aware of metadata and because it cannot offer a valid user ID and password it cannot search this area for reports.

Delivering Reports

Reports that are generated in the SAS Intelligence Platform can be delivered in either HTML or PDF formats. Options for delivering the reports include Web-based delivery using SAS Web Report Studio or the SAS Information Delivery Portal, possibly embedded in Microsoft Word or Microsoft Excel documents. Report delivery options include the following:

- *delivering reports using the SAS Information Delivery Portal.*

If your organization has installed the SAS Web Report Viewer, then you can use the Portal to deliver reports that were created in SAS Web Report Studio and are

saved in the report repository. The SAS Web Report Viewer is bundled with the SAS Web Infrastructure Kit and is installed separately from the SAS Information Delivery Portal.

For more information about installing the software, see Chapter 7, “Installing and Configuring Your Software,” on page 79. For more information about using the SAS Web Report Viewer, see the SAS Web Report Viewer Help, which is available from within the product. For more information about adding a report to a portlet or a channel on the portal, see the *SAS Web Infrastructure Kit: Administrator’s Guide*, which is available at support.sas.com/rnd/itech/doc9/portal_admin/.

- *delivering reports that are produced from stored processes.*

Both SAS Web Report Studio and the SAS Add-in for Microsoft Office enable you to generate and deliver reports that are produced from stored processes. In SAS Web Report Studio, you can execute a stored process from the Choose Report page. Once it is executed, the report is converted into XML that can be delivered in HTML, PDF, or Microsoft Excel file format. You can use the SAS Add-in for Microsoft Office to dynamically execute stored processes and embed the results in Microsoft Word documents, Microsoft Excel spreadsheets, and rich text documents.

For more information about delivering reports as rich text documents, Microsoft Word documents, or Microsoft Excel spreadsheets, see the SAS Add-in for Microsoft Office Help, which is available from within the product.

- *delivering reports as PDF documents using SAS Web Report Studio.*

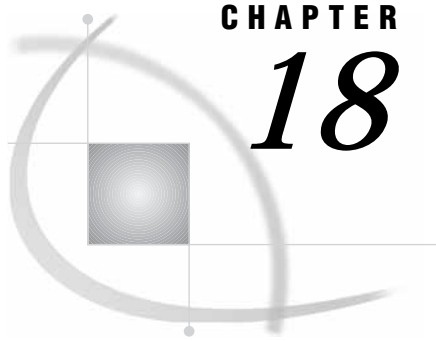
When reports are saved in the report repository and then printed using SAS Web Report Studio, the XML report definition is sent to the ODS server that is defined in your intelligence platform, which converts the XML to a PDF document. This enables report viewers to save the document to a file or to print it and produce a hardcopy report. You can also export the report from SAS Web Report Studio for use in Microsoft Excel.

For more information about printing and exporting reports using SAS Web Report Studio, see the SAS Web Report Studio Help, which is available from within the product.

- *delivering reports as Web pages that can be accessed using any Web browser:*

SAS Web Report Studio enables you to deliver reports across your organization to any user who has access to a Web browser. The SAS Web Report Studio application enables those users to view reports that are stored in the report repository as Web pages.

For more information about viewing the reports using SAS Web Report Studio, see the SAS Web Report Studio Help, which is available from within the product.



CHAPTER

18

Preparing SAS Enterprise Miner for Use

<i>Overview of Preparing SAS Enterprise Miner for Use</i>	341
<i>Configuring SAS Enterprise Miner</i>	342
<i>Launching the SAS Enterprise Miner Configuration Wizard</i>	342
<i>Completing the SAS Enterprise Miner Configuration Wizard</i>	343
<i>Configuring the SAS Enterprise Miner Client</i>	345
<i>Customizing SAS Workspace Server Settings</i>	348
<i>Setting Required Variables in UNIX Shell Scripts</i>	351
<i>Customizing the Apache Tomcat HTTP Server</i>	351
<i>Securing SAS Enterprise Miner Metadata</i>	354
<i>Securing Access at the SAS Enterprise Miner Folder Level</i>	354
<i>Securing Access at the Projects Folder Level</i>	355
<i>Securing Access at the Individual Project Level</i>	356
<i>Securing Access at the SAS Workspace Server Level</i>	357

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:

- *Personal workstation.* In this deployment, Java files, configuration files, and documentation are stored locally on the client computer. The client communicates directly with the SAS Workspace Server and the SAS Metadata Server and must remain connected for the duration of a model training session. The personal workstation installation is appropriate for single-user configurations.
- *Enterprise client and shared platform server.* The SAS Enterprise Miner enterprise client installation includes only the Java files that are needed for display on the client computer. All other files are installed on a SAS Enterprise Miner shared platform server. Enterprise clients connect directly to the shared platform 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 enterprise client files on demand from a link in a Web document. Java Web Start automatically downloads the most

recent versions of required files from an application server and stores the files on the client computer.

Note: For more information about how to deploy SAS Enterprise Miner, see the SAS Enterprise Miner Help, which is available from within the product. Δ

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. For more information, see Chapter 7, “Installing and Configuring Your Software,” on page 79.

In addition, if this is not a personal workstation installation, you must perform these tasks:

- Create additional SAS Enterprise Miner users (see “Planning Your Users” on page 195).
- 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 that are used for non-Windows systems.
- Configure the Apache Tomcat HTTP Server for use with SAS Enterprise Miner.

Configuring SAS Enterprise Miner

When SAS Enterprise Miner 5.1 is delivered, it is preconfigured as a personal workstation, specifically to run on the same computer with the metadata server and workspace servers.

If you are using SAS Enterprise Miner on your computer as a personal workstation or as an enterprise (thin) client to connect to a SAS Enterprise Miner shared platform server, then you can skip the following sections—“Launching the SAS Enterprise Miner Configuration Wizard” and “Completing the SAS Enterprise Miner Configuration Wizard”—and proceed to “Completing the SAS Enterprise Miner Client.”

If the computer hosts the SAS Enterprise Miner shared platform server for multi-client access, you may need to adjust the configuration, as described in the following subsection.

Launching the SAS Enterprise Miner Configuration Wizard

To launch the SAS Enterprise Miner configuration wizard, complete the tasks that are applicable to your operating environment. See the following table.

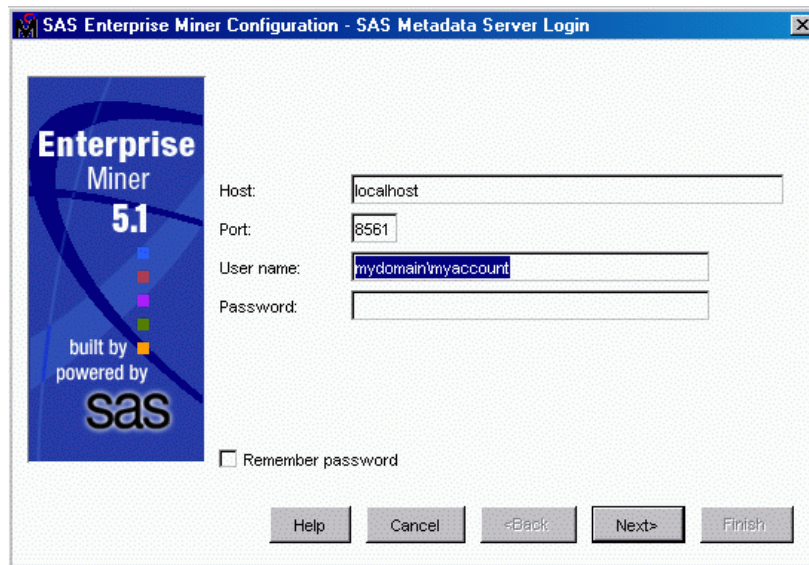
Table 18.1 Tasks for Launching the SAS Enterprise Miner Configuration Wizard in Different Environments

Name of Operating Environment	Tasks to Perform
UNIX	Execute the emconfig script by entering [emroot]/emconfig .
Windows	Select Start ► Programs ► SAS ► Enterprise Miner EM 5.1 Advanced Configuration .

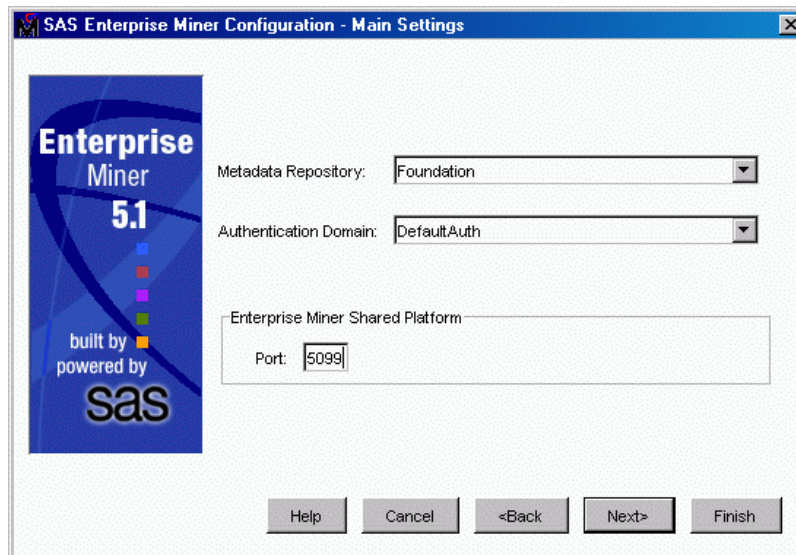
Completing the SAS Enterprise Miner Configuration Wizard

Here are the steps that you must follow to complete the SAS Enterprise Miner configuration wizard:

- 1 In the first wizard window, 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. The default port is 8561.
 - c Enter the name of the user of this installation of SAS Enterprise Miner. The user must have a login name for the specified SAS Metadata Server. For the 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, enter **mydomain\myaccount**.
 - d Enter the **Password** for the user name that you entered.
 - e Select **Remember password** to save the user's password in encoded format along 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 a SAS Enterprise Miner Shared Platform Server or Personal Workstation is launched.
- When you are finished, click **Next**.
- 2 Verify the repository name, authentication domain, and the shared platform server port.

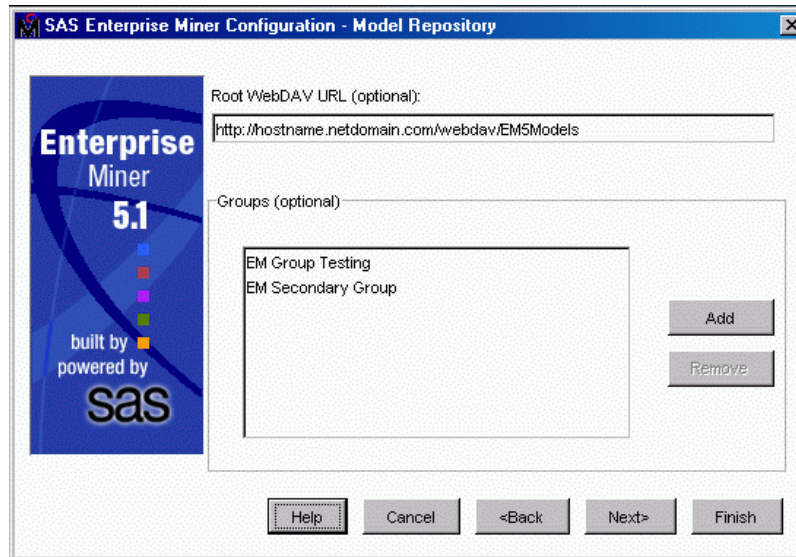


- a The name of the foundation repository is shown for verification. If multiple repositories are present, they might be displayed for additional verification.
- b Select the **Authentication Domain** that contains the user name that was specified in the first wizard window. Typically, all SAS Enterprise Miner users are assigned to the same authentication domain.
- c If you are configuring the SAS Enterprise Miner Shared Platform, you can enter the **Enterprise Miner Shared Platform** server port, which is the port on which the Remote Method Invocation (RMI) registry will be listening on this machine. This port is typically 5099, so it is generally safe to leave it at its default setting. If an RMI registry is already running on that port, the SAS Enterprise Miner Shared Platform Server will use it. Otherwise, the Enterprise Miner Shared Platform Server will launch an internal registry to use that port.

Note: RMI is a standard Java-based protocol provided by Sun Microsystems. It is used by SAS Enterprise Miner to manage communication between SAS Enterprise Miner clients and the SAS Enterprise Miner Shared Platform Server. \triangle

When you are finished, click **Next**.

- 3 Enter optional model storage parameters.



- 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. For more information, see the SAS Enterprise Miner 5.1 Help, which is available from within the product. The **Root WEBDAV URL (Optional)** field contains the URL of the WebDAV storage application. The URL should contain a port number when a number is applicable, such as `http://hostname.netdomain.com:8080/webdav/EM5Models`.
- b Users can group saved models for investigation and study purposes. To add a group, click **Add** 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 finished, click **Next**.

- 4 Verify the settings in the Summary window, then click **Finish**.
- 5 If the computer hosts the SAS Enterprise Miner Shared Platform Server for multi-client access, you must launch the server. The server's operation depends on the metadata server that was launched previously.

Table 18.2 Tasks for Launching the SAS Enterprise Miner Shared Platform Server in Different Operating Environments

Name of the Operating Environment	Tasks to Perform
UNIX	Execute the emserver script by entering <code>[emroot]/emserver</code> .
Windows	Execute the emserver script by entering <code>[emroot]\emserver.bat</code> .

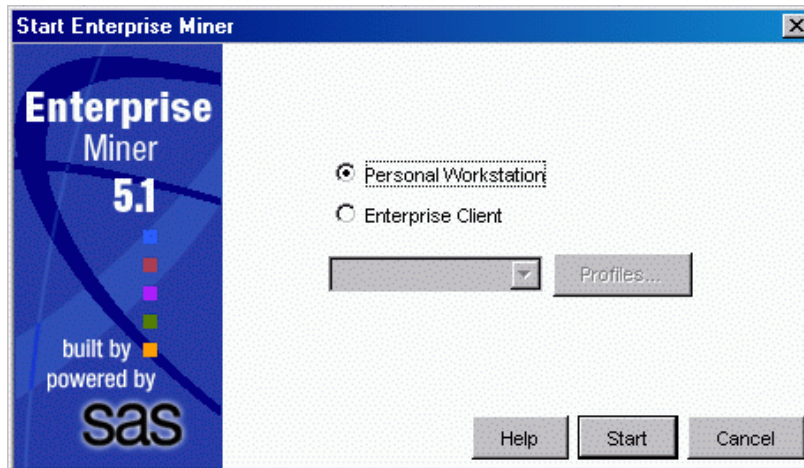
Configuring the SAS Enterprise Miner Client

To launch the SAS Enterprise Miner client, complete the tasks that are applicable to your operating environment.

Table 18.3 Tasks for Launching the SAS Enterprise Miner Client in Different Operating Environments

Name of the Operating Environment	Tasks to Perform
UNIX	Execute the <code>em</code> script by entering <code>[emroot]/em</code> .
Windows	Select Start ► Programs ► SAS ► Enterprise Miner EM 5.1 Client .

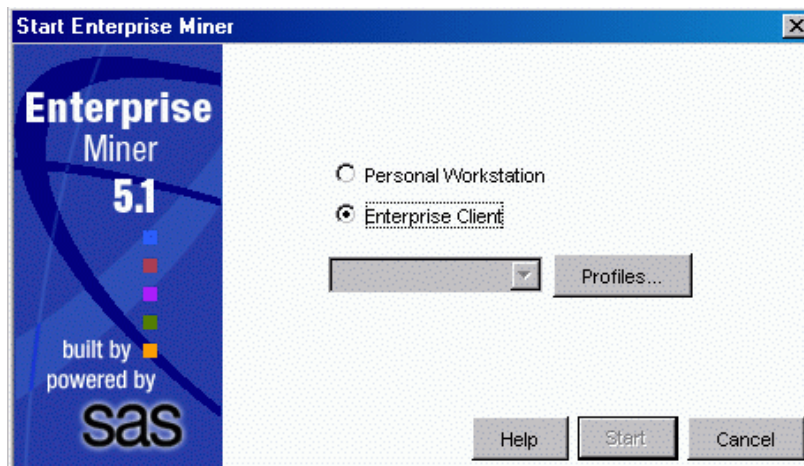
- 1 The client launcher appears.



This example shows the **Personal Workstation** selected. The launch window always begins with the most recent selection.

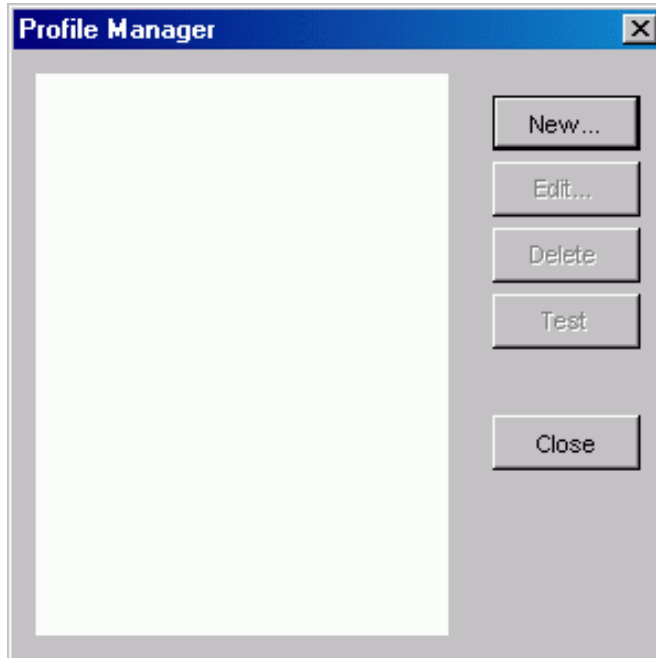
Note: If this is a personal workstation, click **Start** to launch SAS Enterprise Miner. You can skip this step for the personal workstation launch by adding the parameter `-noprompt` to the UNIX script or Windows shortcut. △

- 2 If you will be using this machine as a client to access one or more SAS Enterprise Miner Shared Platform Servers, select **Enterprise Client**.



- a This selection activates the **Profiles** button. If no enterprise clients have been defined, the **Start** button will be unavailable.

- b Click **Profiles** to add and manage profiles of enterprise clients.
- 3 When you click **Profiles**, the **Profile Manager** window appears.

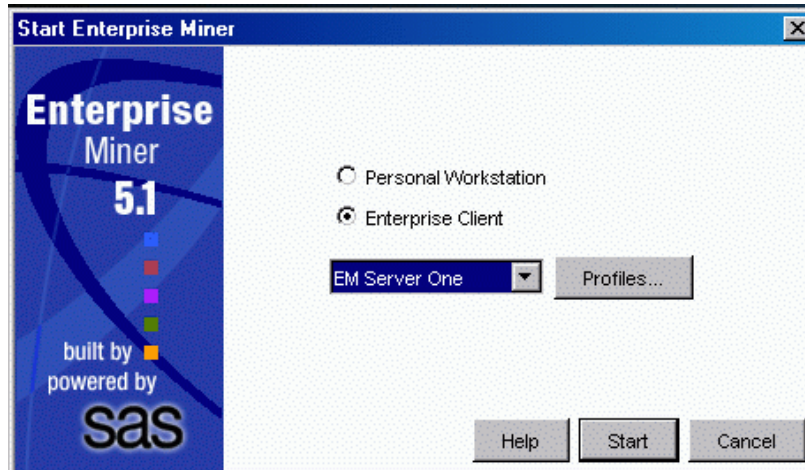


- a If the list of profiles is empty, only the **New** button will be active. If the list contains items, the **Edit** or **Delete** buttons will also be active.
 - b You can test whether a SAS Enterprise Miner Shared Platform Server is available by selecting the entry and clicking **Test**. A dialog box will appear that shows the condition of the server.
 - c Click **New** to add an enterprise client profile.
- 4 When you add or edit an enterprise client profile, the New Shared Platform Profile window appears.



- a Enter a convenient name for the shared platform server into the **Name** field.
- b Enter the host address of the shared platform server in the **Host** field. This might take the form of an IP address (for example, 123.123.123.123) or a domain name (for example, myserver.mynetdomain.com).

- c Enter the host port of the shared platform server's RMI registry in the **Port** field. This is typically **5099**.
 - d You can test whether a SAS Enterprise Miner Shared Platform Server is available by selecting the entry and clicking **Test**. A dialog box will appear that shows the condition of the server.
 - e Save the new or edited entry by clicking **OK**.
- 5 When enterprise client entries are present, they are selectable in the **Start** window.



- a Select the enterprise client that you would like to launch.
- b Click **Start** to launch the client and connect to the SAS Enterprise Miner Shared Platform Server.

Customizing SAS Workspace Server Settings

You can set extended attributes on each logical SAS Workspace Server that you are using at your site.

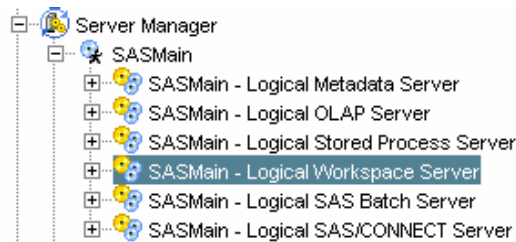
Table 18.4 Optional SAS Enterprise Miner Extended Attributes for the SAS Workspace Server

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.

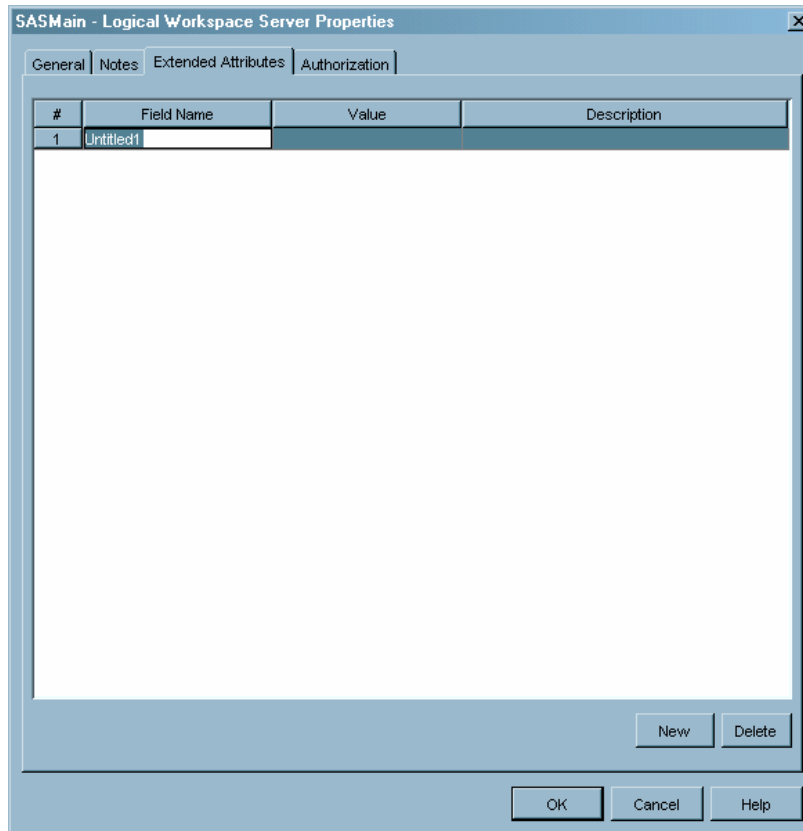
Extended Attribute	Description
EM_SERVER_INIT_CODE	The path on the server from which SAS Enterprise Miner retrieves SAS code to execute when a project is opened. For example, you might write code to force users to log their use into an audit trail.
EM_MAX_CPU	The number of CPUs to exercise during a process 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.
EM_SASCMD	The command that SAS Enterprise Miner uses to start a SAS MPConnect process to run a node. The default value is <code>!sascmdv -noobjectserver -nosyntaxcheck -noasynchio</code>

To add these attributes to a logical SAS Workspace Server, complete these steps in SAS Management Console:

- 1 Use your metadata profile to log on to the SAS Metadata Server that contains the SAS Metadata Repository connection that you want to use.
- 2 In the SAS Management Console navigation tree, select **Environment Management ► Server Manager**.
- 3 Select a SAS application server 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 18.5 Sample Values for SAS Enterprise Miner Extended Attributes on the SAS Workspace Server

Field Name	Sample Value*	Description
EM_PROJECT_ROOT	<i>d:\my\project\path\</i>	Project tree storage path.
EM_ENFORCE_PROJECT_LOCATION		User cannot change the project storage path.
EM_SERVER_INIT_CODE	<i>d:\my\server\initialCode.sas</i>	Path to server initialization code.
EM_MAX_CPU	-1	Use all available processors.
EM_SASCMD	sas -config <i>e:\my\special\sasv9.cfg</i>	Use an alternate SAS configuration.

* User-specified values appear in italic.

- 9 When you are finished, 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 18.6 UNIX Shell Scripts

Shell Script	Description
<code>\$EM_HOME/em</code>	Java client script
<code>\$EM_HOME/emhelp</code>	Java client script
<code>\$EM_HOME/emserver</code>	Java application server (middle tier) script
<code>\$EM_HOME/emconfigure</code>	SAS Enterprise Miner configuration wizard script

Also, set the execute permission for each script file.

Customizing the Apache Tomcat HTTP Server

The HTTP server that is 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 SAS Enterprise Miner Model Viewer application for Tomcat.

Note: Servers other than Tomcat 4.1.18 might have different instructions. Δ

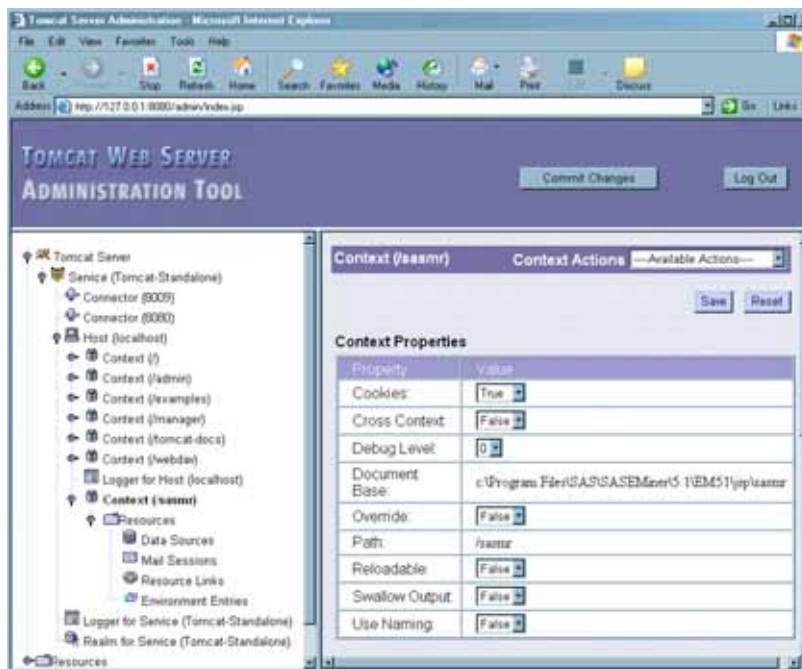
Note: These instructions assume that the default Windows and UNIX installation paths were used. The default path for Windows is `c:\Program Files\SAS\SASEminer\5.1\EM51`. The default path for UNIX is `/opt/SAS/SASEminer/5.1/EM51`. Δ

- 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. Δ

```
<!--
  <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** \blacktriangleright **Programs** \blacktriangleright **Apache Tomcat 4.1** \blacktriangleright **Tomcat Administration**.
 - a In the navigation tree on the left, select **Tomcat Server** \blacktriangleright **Service** \blacktriangleright **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 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 -->
<Context path="/sasmr"
  docBase="c:\Program Files\SAS\SASEMiner\5.1\EM51\jsp\sasmr"
  crossContext="false"
  debug="0"
  reloadable="false">
</Context>
```

Note: If the SAS Enterprise Miner installation path is other than that shown, change the location of the Document Base (`docBase=` value). \triangle

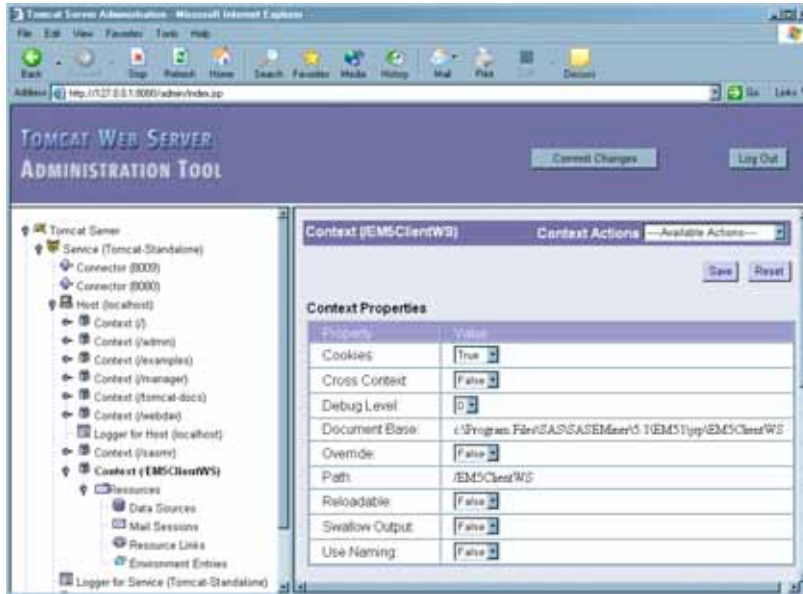
If the SAS Enterprise Miner Model Viewer is installed on a Tomcat server on which Tomcat security is enabled, the installer will need to grant permissions to Catalina for the model viewer. To accomplish this, you need to add the following item to the Catalina permissions file, typically located at `C:\Program`

Files\Apache Group\Tomcat 4.1\conf\catalina.policy or **C:\Tomcat 4.1\conf\catalina.policy**. Append these lines to the end of the module:

```
grant codeBase "file:C:/Program Files/SAS/SASEMiner/5.1/EM51/jsp/sasmr/-"
{
    permission java.security.AllPermission;
};
```

where **C:/Program Files/SAS/SASEMiner/5.1/EM51/jsp/sasmr/** (or its UNIX counterpart) is the docBase path to the **sasmr** application on the local disk, but with forward slashes even for Windows installations.

- 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** \blacktriangleright **Service** \blacktriangleright **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 an alternative to using the Tomcat Web Server Administration Tool, you can manually add the following context coding beneath the SAS Enterprise Miner Model Viewer in the Tomcat server configuration XML file:

```
<!-- EM Java Web Start Client -->
<Context path="/EM5ClientWS"
docBase="c:\Program Files\SAS\SASEMiner\5.1\EM51\jsp\EM5ClientWS"
crossContext="false"
debug="0"
reloadable="false">
</Context>
```

Note: If the SAS Enterprise Miner installation path is other than that shown, change the location of the Document Base (**docBase=** value). Δ

- 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 `em5jha11.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. Δ

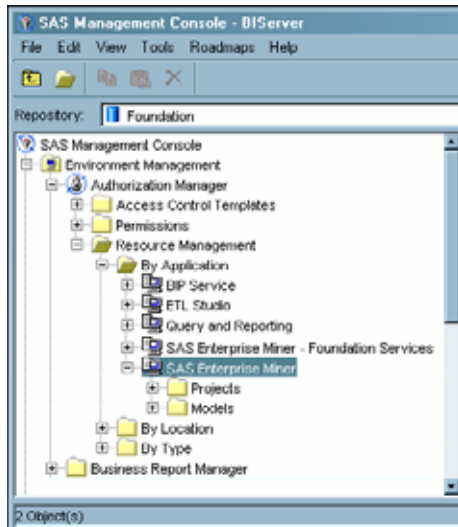
Note: Because all SAS Intelligence Platform applications use the metadata server when accessing resources, permissions that are enforced by the metadata server provide the strongest protections that are available in the metadata authorization layer. For more information, see “Planning Your Access Controls” on page 204. Δ

Note: Currently, you cannot secure SAS Enterprise Miner models. Δ

Securing Access at the SAS Enterprise Miner Folder Level

To access the **Authorization** tab for the **SAS Enterprise Miner** folder in the SAS Management Console navigation tree, complete these steps:

- 1 Use your metadata profile to log on to the SAS Metadata Server that contains the SAS Metadata Repository connection that you want to use.
- 2 Select **Environment Management** ► **Authorization Manager** ► **Resource Management** ► **By Application**.
- 3 Select the **SAS Enterprise Miner** folder.



- 4 Select **File ► Properties**.
- 5 Click the **Authorization** tab.

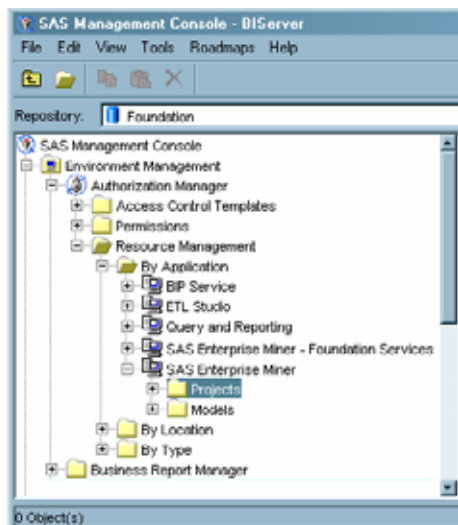
Note: For help on using the **Authorization** tab, click [Help](#). △

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 355).

Securing Access at the Projects Folder Level

To access the **Authorization** tab for the **Projects** folder in the SAS Management Console navigation tree, complete these steps:

- 1 Use your metadata profile to log on to the SAS Metadata Server that contains the SAS Metadata Repository connection that you want to use.
- 2 Select **Environment Management ► Authorization Manager ► Resource Management ► By Application ► SAS Enterprise Miner**.
- 3 Select the **Projects** folder.



- 4 Select **File ► Properties**.
- 5 Click the **Authorization** tab.

Note: For help on using the **Authorization** tab, click [Help](#). △

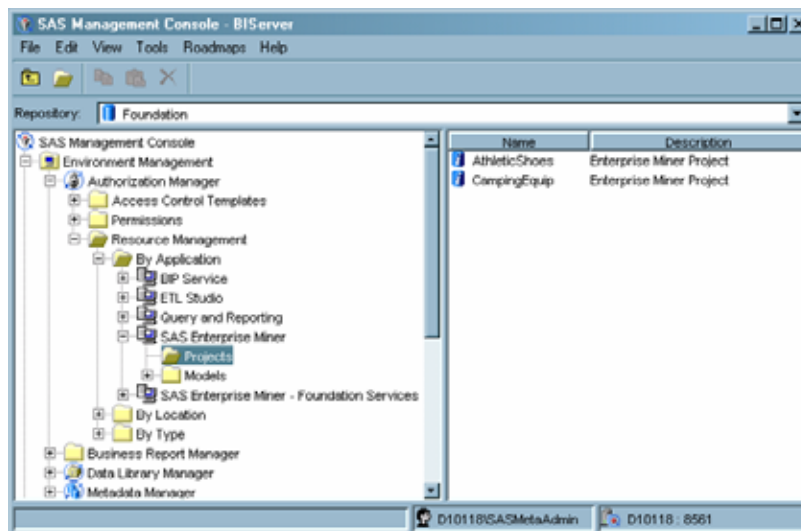
For example, if you deny ReadMetadata access to the **Projects** folder to UserA, then UserA will not be able to see any projects in SAS Enterprise Miner.

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 on to the SAS Metadata Server that contains the SAS Metadata Repository connection that you want to use.
- 2 Select **Environment Management ► Authorization Manager ► Resource Management ► By Application ► SAS Enterprise Miner**.
- 3 Select the **Projects** folder.
- 4 In the display area to the right of the navigation tree, select the project that you want to secure.



- 5 Select **File ► Properties**.
- 6 Click the **Authorization** tab.

Note: For help on using the **Authorization** tab, click [Help](#). △

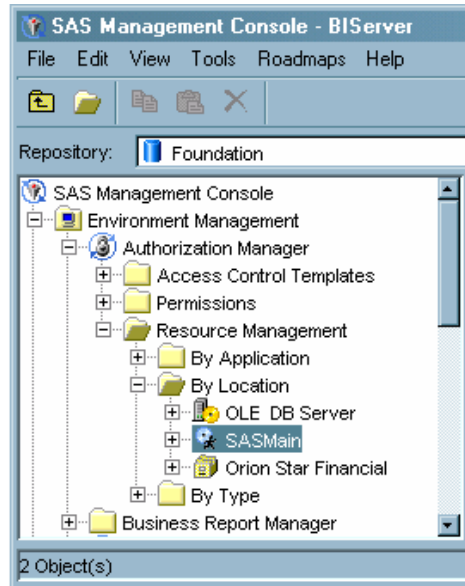
For example, if you deny ReadMetadata access to the **CampingEquip** project to UserA, then UserA will not be able to see the **CampingEquip** project in SAS Enterprise Miner. If you grant ReadMetadata but deny WriteMetadata, then UserA will be able to open the project but will not be able to save changes 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:

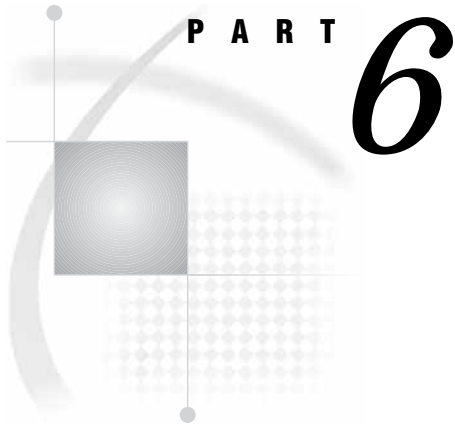
- 1 Use your metadata profile to log on to the SAS Metadata Server that contains the SAS Metadata Repository connection that you want to use.
- 2 Select **Environment Management ► Authorization Manager ► Resource Management ► By Location**.
- 3 Select the SAS application server that contains the SAS Workspace Server that is associated with your SAS Enterprise Miner projects.



- 4 Select **File ► Properties**.
- 5 Click the **Authorization** tab.

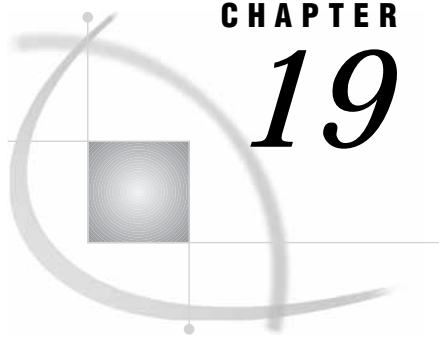
Note: For help on using the **Authorization** tab, click [Help](#). △

For example, if you deny ReadMetadata access to the **SASMain** application server to UserA, then UserA will not have access to any projects associated with that server.



Advanced Topics

- Chapter 19* **Configuring Your Servers for Better Performance** 361
- Chapter 20* **Promoting and Replicating Metadata** 393
- Chapter 21* **Managing an Environment** 421



CHAPTER

19

Configuring Your Servers for Better Performance

- Overview of Configuring Your Servers for Better Performance* 362
- Tuning a Workspace Server for Use with SAS Web Report Studio* 363
 - Which System Options to Add* 363
 - Adding System Options to the Workspace Server Launch Command* 364
- Workspace Server Pooling for SAS Web Report Studio and SAS Information Delivery Portal* 366
 - Configuring the Workspace Server and Pool* 367
 - Converting the Workspace Server to Pooling* 367
 - Setting Configuration Options for the Pool* 368
 - Adding an Authentication Domain to the SAS Trusted User's Login* 369
 - Configuring the Foundation Services Manager User Service* 370
 - Verifying That Connection Pooling Is Working for SAS Web Report Studio* 372
- Tuning Your J2EE Server or Servlet Container for Use with SAS Web Report Studio and SAS Information Delivery Portal* 373
 - Tuning the Java Virtual Machine* 373
 - Which JVM Are You Using?* 373
 - Where to Specify JVM Options* 374
 - Quick Start Settings* 375
 - Setting Just-in-Time Compiler and Memory Options* 376
 - Selecting a Garbage Collector* 377
 - Configuring the Garbage Collector* 377
 - Tuning the J2EE Server or Servlet Container* 378
 - Detecting Changes in JavaServer Pages and Servlets* 378
 - Setting the Number of Available Worker Threads (SAS Information Delivery Portal Only)* 379
- Load Balancing Workspace Servers for Desktop Applications* 379
 - Installing the Software* 379
 - Configuring the Workspace Server and Object Spawner* 380
 - Start the SAS Management Console* 381
 - Defining Your Application Server and Workspace Server* 381
 - Define Your Object Spawner* 381
 - Start the Object Spawner* 381
 - Converting the Logical Workspace Server to Load Balancing* 381
 - Setting Load Balancing Parameters for Each Workspace Server* 382
- Overview of the Initial Load Balancing Setup for Stored Process Servers* 383
- Load Balancing Stored Process Servers on Multiple Hosts* 385
 - Installing the Software* 385
 - Configuring the Stored Process Server and Object Spawner* 386
 - Start the SAS Management Console* 386
 - Defining Your Application Server and Workspace Server* 387
 - Define Your Stored Process Server* 387
 - Define Your Object Spawner* 387

<i>Load SAS Stored Process Samples</i>	387
<i>Start the Object Spawner</i>	388
<i>Setting Logical Stored Process Server Properties</i>	388
<i>Setting the Load Balancing Properties for Each Stored Process Server</i>	390

Overview of Configuring Your Servers for Better Performance

This chapter explains how you can configure your SAS servers and your J2EE server to improve the performance of your SAS applications. Here are the specific areas that we will be looking at:

- If you are running SAS Web Report Studio at your site, it is a good idea to change the SAS system options that are used to start the workspace server processes that the application uses. This recommended setup enables the application to serve more clients. For information on how to tune your workspace server for use with SAS Web Report Studio, see “Tuning a Workspace Server for Use with SAS Web Report Studio” on page 363.
- If you have a large number of users who are running SAS Web Report Studio or SAS Information Delivery Portal, you can improve the performance of the application by setting up a pooling workspace server. Without pooling, each user session must establish a connection to a workspace server process, and establishing this connection takes time. If you set up a pooling workspace server, an application can create a set of connections from which users can simply get a reference to a connection. This setup greatly increases performance. For information on how to set up a pooling workspace server, see “Workspace Server Pooling for SAS Web Report Studio and SAS Information Delivery Portal” on page 366.
- If you are running SAS Web Report Studio or SAS Information Delivery Portal, you can use the information at “Tuning Your J2EE Server or Servlet Container for Use with SAS Web Report Studio and SAS Information Delivery Portal” on page 373 to tune your J2EE server or servlet container to achieve the best possible results with those two applications.
- If you need to support a large group of SAS ETL Studio users, you might want to set up load balanced workspace servers. Because each workspace server is a single-user server, each client request causes a new server process to be created. If too many clients require the use of workspace servers simultaneously, the performance of programs executing on the workspace server host will degrade. You can address this problem by
 - 1 setting up a workspace server on a second host
 - 2 configuring the server to be a part of your existing logical workspace server
 - 3 converting the logical workspace server to load balancing.

For step-by-step instructions on how to perform these tasks, see “Load Balancing Workspace Servers for Desktop Applications” on page 379.

- 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 installation. Three MultiBridge connections are set up so that the object spawner can start up to three stored process server processes, and the object spawner balances the workload across these processes. For more information about this initial configuration, see “Overview of the Initial Load Balancing Setup for Stored Process Servers” on page 383.

To some extent, you can scale your system by adding MultiBridge connections to the existing stored process server. However, at some point, you will need to add a second stored process server running on a second host to improve performance. For information on how to perform this task, see “Load Balancing Stored Process Servers on Multiple Hosts” on page 385.

Note: All of the servers in a load balanced cluster must belong to the same SAS authentication domain. Δ

Tuning a Workspace Server for Use with SAS Web Report Studio

To obtain the best performance from SAS Web Report Studio, you should tune the workspace server that is being used by that product as described in this section. The changes you should make include specifying

- an appropriate work folder
- a buffer size for writing files to the work area
- a limit on the total amount of memory that SAS uses at any one time.

“Which System Options to Add” on page 363 lists the system options that you should set and recommends values to use with those options. “Adding System Options to the Workspace Server Launch Command” on page 364 explains how to add the system options to the command that starts the workspace server.

Note: In addition to tuning your workspace server by following the directions in this section, you should convert your workspace server to pooling as discussed in “Workspace Server Pooling for SAS Web Report Studio and SAS Information Delivery Portal” on page 366. Δ

Which System Options to Add

This section explains which system options the object spawner should use when it launches a workspace server process to be used by SAS Web Report Studio. The arguments to the system options that are shown in the following table are values that are useful on a system with the following characteristics:

- four CPUs, 2.0 GHz
- 3.5 GB RAM
- Windows Server 2003

Table 19.1 System Options for the Workspace Server

System Option	Explanation
-RSASUSER	Opens the SASUSER library in read-only mode. Declaring this library read only makes the workspace server much faster for SAS Web Report Studio.
-work <i>work-folder</i>	Specifies the pathname for the directory that contains the Work data library. This directory should reside on a disk that emphasizes fast write performance, not, for example, on a RAID-5 device.
-ubufsize 64K	Specifies a buffer size for writing files to the work area.
-memsize 192M	Specifies a limit on the total amount of memory that SAS uses at any one time.

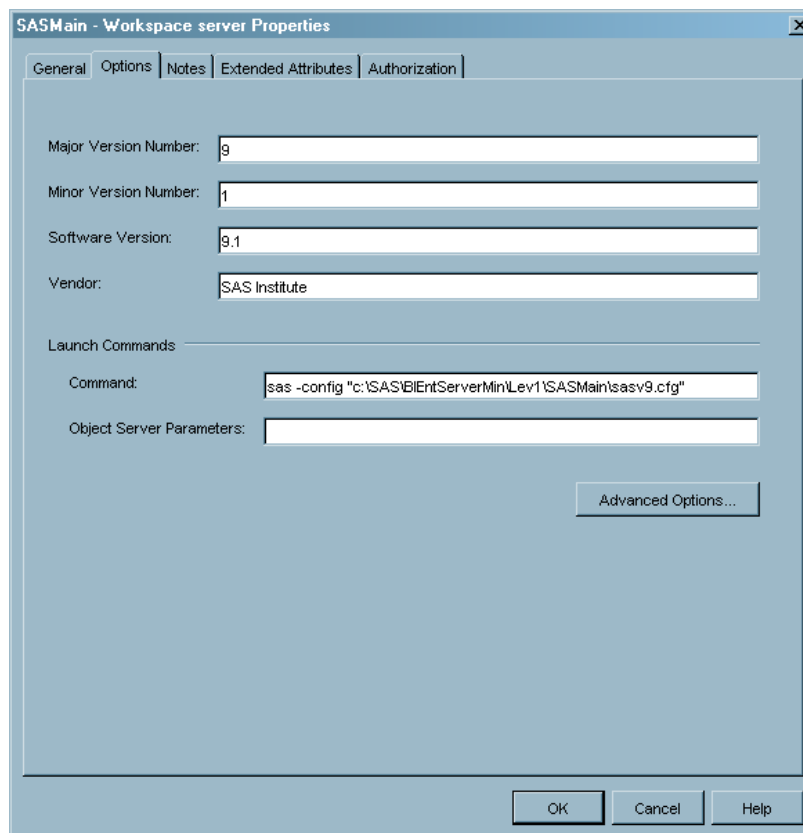
System Option	Explanation
-realmemsize 128M	Indicates the amount of RAM available to a process before it begins to page. Keeping this number low limits the amount of RAM consumed by a SAS server in order to reduce paging activity.
-sortsize 128M	Limits the amount of memory available to the SORT procedure.
-cpucount 2	Specifies the number of processors that thread-enabled applications should assume will be available for concurrent processing. This setting maximizes the effectiveness of the SAS Web Report Studio sorting algorithm.

Note that the arguments to these options will be site and job specific. Take care in choosing these values, and consult a SAS representative if necessary.

Adding System Options to the Workspace Server Launch Command

After you have determined the system options that you want to use to start your workspace server, follow the directions in this section to edit the `sas` command that starts the server.

- 1 In SAS Management Console, expand the Server Manager; then, expand the SASMain—Logical Workspace Server. You will see a tree node that represents the physical workspace server.
- 2 Right-click the icon for the physical workspace server, and select **Properties** from the pop-up menu. A Workspace Server Properties dialog box appears.
- 3 Click the **Options** tab. You will see the information that is shown in the following display.



- 4 Edit the text in the **Command** text box, which by default is set to

```
sas -config "path-to-config-dir\Lev1\SASMain\sasv9.cfg"
```

The edited command should look like this:

```
sas -config "path-to-config-dir\Lev1\SASMain\sasv9.cfg" -rsasuser
-work work-folder -ubufsize 64K -memsize 192M -realmemsize 128M
-sortsize 128M -cpucount 2
```

CAUTION:

Do not add the options to the configuration file that are specified in this command. Setting values in `Lev1\SASMain\sasv9.cfg` affects every server launched. This includes the metadata server, the OLAP server, and every workspace server and stored process server. Δ

- 5 Click **OK** in the Workspace Server Properties dialog box.

Note: At the end of this procedure, you will have optimized your workspace server for use with SAS Web Report Studio. If you are using other applications and these applications would benefit from a differently configured workspace server, you must create a new logical workspace server (under SASMain) and add a workspace server to it. Δ

Workspace Server Pooling for SAS Web Report Studio and SAS Information Delivery Portal

This section explains how to set up pooling for your workspace server for use with SAS Web Report Studio and SAS Information Delivery Portal. The result of this configuration is that the users of these Web applications will see much better performance than they would if they were connecting to a standard workspace server.

Before we explain how to perform this configuration, it is important that you understand a couple of concepts. For example, what does it mean to set up a pooling workspace server? If a workspace server has not been converted to pooling, each time a SAS Web Report Studio or SAS Information Delivery Portal user starts a session, a workspace server process must be created and the user must establish a connection to this process. This can be a time-consuming sequence of events. When you set up a pooling workspace server, a pool (or set) of connections to workspace servers are opened when SAS Web Report Studio or SAS Information Delivery Portal makes its first request for a workspace server. A user can then obtain a preexisting connection from the pool instead of having to establish the connection.

Another important concept is that of a *puddle*. A puddle is a subset of the connections in a pool. Setting up puddles enables you to associate a different set of users with different puddles. Typically, the reason for setting up different groups of users is to give the different groups different levels of access.

One of the advantages of using a pooling workspace server is to enable SAS Web Report Studio and SAS Information Delivery Portal users to obtain an existing connection to a workspace server. Other advantages include the following:

- You can limit the number of clients that can connect to workspace servers simultaneously. As a result, you can ensure acceptable response times for all connected clients.
- You can add server hosts to the pool to accommodate increased demand.
- You can grant certain groups greater access to the pool than others.
- You can divide the pool into puddles of connections that have unique login credentials and access to server-side resources.

Note: In this section, we assume that you are starting with a standard workspace server and that you will have a single puddle of connections. In addition, we present a straightforward method of setting up the pool. For information about setting up a pool with multiple puddles and for detailed information about pooling security, see the *SAS Integration Technologies: Server Administrator's Guide*, which is available at support.sas.com/rnd/itech/doc9/admin_oma/. \triangle

To set up your pooling workspace server for use with SAS Web Report Studio, follow the directions in these sections:

- “Configuring the Workspace Server and Pool” on page 367
- “Adding an Authentication Domain to the SAS Trusted User’s Login” on page 369
- “Configuring the Foundation Services Manager User Service” on page 370

To set up your pooling workspace server for use with SAS Information Delivery Portal, follow the directions in these sections:

- “Configuring the Workspace Server and Pool” on page 367
- “Adding an Authentication Domain to the SAS Trusted User’s Login” on page 369

It is not necessary to perform the steps in “Configuring the Foundation Services Manager User Service” on page 370 because the User Service that is employed by the portal application is configured by default in a way that will work with a pool of workspace servers.

Finally, to verify that connection pooling is working, see “Verifying That Connection Pooling Is Working for SAS Web Report Studio” on page 372.

Configuring the Workspace Server and Pool

The first step in configuring connection pooling is to convert your logical workspace server to a pooling configuration and to configure a pool and puddle for this server. Follow the directions in the next two subsections.

Converting the Workspace Server to Pooling

To convert your workspace server to pooling and to define a puddle, perform these steps:

- 1 Log on to SAS Management Console as the SAS Administrator (**sasadm**).
- 2 In SAS Management Console, expand the Server Manager tree node and the node for the SASMain application server. One of the tree nodes under SASMain will be SASMain—Logical Workspace Server.
- 3 Right-click the icon for the logical workspace server, and select **Convert To ► Pooling**. You will see an Information dialog box that asks if you want to continue with the conversion. Click **Yes**. The Pooling Options dialog box appears.
- 4 In the Pooling Options dialog box, click **New** to indicate that you want to define a puddle. The New Puddle dialog box appears.

- 5 Fill out the fields in the dialog box as shown in the previous display. The following table explains what the values in the dialog box mean.

Table 19.2 New Puddle Information

Field	Explanation
Name	The name of the puddle, for example, Puddle1.
Minimum Available Servers	Specifies the number of workspace servers to start if the minimum number of servers is already in use.
Minimum Number of Servers	Specifies the initial number of started workspace servers that are available to satisfy client connection requests from SAS Web Report Studio and SAS Information Delivery Portal users.

Field	Explanation
Login	<p>Contains the user ID of the user who will start the connections in the pool. Enter the ID of the SAS General Server User (sassrv) here.</p> <p>Note: A login for this user was added to the group SAS General Servers when your system was first configured.</p>
Grant Access To Group	<p>This field specifies which group of users can use connections from the pool. In the current example, this is the SASUSERS group. If you want to allow all users to access the system, you should change this value to PUBLIC.</p> <p>Note: If you do not want members of PUBLIC to be able to use SAS Web Report Studio <i>and</i> you want to present such users with a clear and user-friendly error message, follow these directions. First, set the value of the Grant Access To Group field to PUBLIC. Then, edit the properties file WEB-INF\WebReportStudioProperties.xml, and change the value of the <code>wrs.pfs.allowPublicUsers</code> element to false.</p>

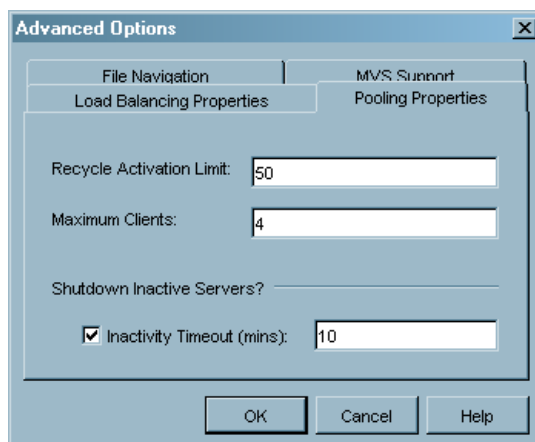
Note: If you are not logged on to SAS Management Console as the SAS Administrator (**sasadm**), you might not see **sassrv** in the **Login** drop-down list box. In this case, click **Cancel** in the New Puddle dialog box. Then, reconnect to the metadata server using the metadata profile for the SAS Administrator. △

- 6 Click **OK** in the New Puddle dialog box.
- 7 Click **OK** in the Pooling Options dialog box.

Setting Configuration Options for the Pool

Next, set the configuration options for the pool itself by following these directions:

- 1 In SAS Management Console, expand the SASMain—Logical Workspace Server to reveal the icon for the physical workspace server.
- 2 Right-click the workspace-server icon, and select **Properties** from the pop-up menu. A Workspace Server Properties dialog box appears.
- 3 In the Workspace Server Properties dialog box, select the **Options** tab.
- 4 On the **Options** tab, click **Advanced Options**. The Advanced Options dialog box appears.
- 5 In the Advanced Options dialog box, select the **Pooling Properties** tab.



- 6 Fill the fields in the dialog box as shown in the previous display. The following table explains the meaning of each value.

Table 19.3 Pooling Properties

Field	Explanation
Recycle Activation Limit	Places a limit on how often workspace server processes are reused to satisfy puddle connections.
Maximum Clients	Each client requires a workspace server process on the workspace server host. These processes constitute the pool that will be available to SAS Web Report Studio and SAS Information Delivery Portal. Each process requires approximately 150 MB of memory and one CPU for efficient operation. As a guide, we recommend one process per CPU. If the server is <i>not</i> also acting as the metadata server, you can add one or two to this maximum. If you have long-running queries, you can add one or two servers. This will make the system seem more responsive to short-running queries, at the expense of total throughput. A typical setting is between 4 and 6.
Inactivity Timeout	A workspace server process can have an inactivity timeout. Having a short timeout reduces the workload on the server host, but might reduce response time for client connection requests.

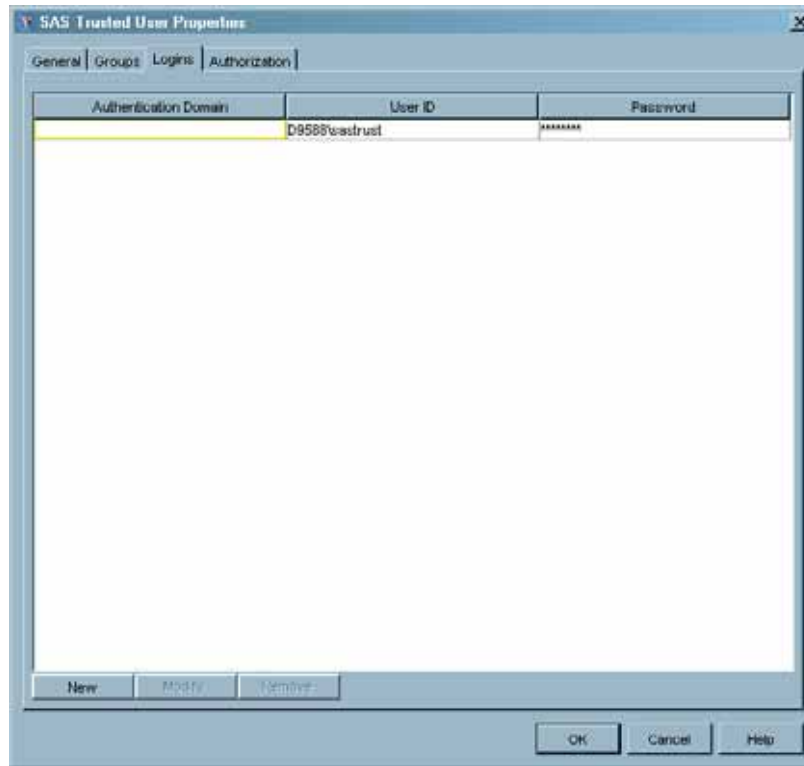
- 7 Click in the Advanced Options dialog box.
 8 Click in the Workspace Server Properties dialog box.

Adding an Authentication Domain to the SAS Trusted User's Login

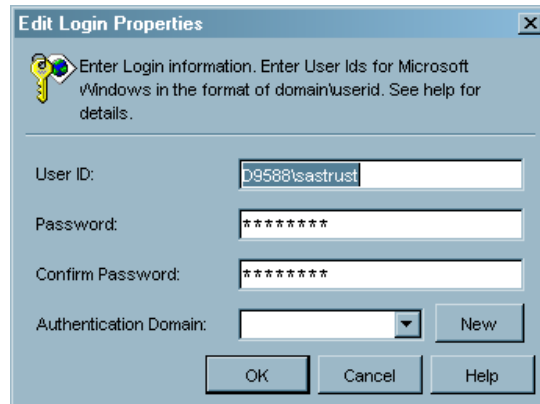
When you configure the Foundation Services Manager User Service—as described in the next section—you are asked to specify a *pool administrator*, and the configuration instructions tell you to specify the SAS Trusted User (**sastrust**) as this user. In order for the SAS Trusted User to be used as the pool administrator, you must add the proper authentication domain to that user's login.

To define an authentication domain for the SAS Trusted User, perform the following steps in SAS Management Console:

- 1 Select the icon for the User Manager plug-in so that a list of users and groups appears in the console.
- 2 Double-click the entry for the SAS Trusted User. A SAS Trusted User Properties dialog box appears.
- 3 In this dialog box, select the **Logins** tab.



- 4 Select the existing login, and click **Modify**. An Edit Login Properties dialog appears.



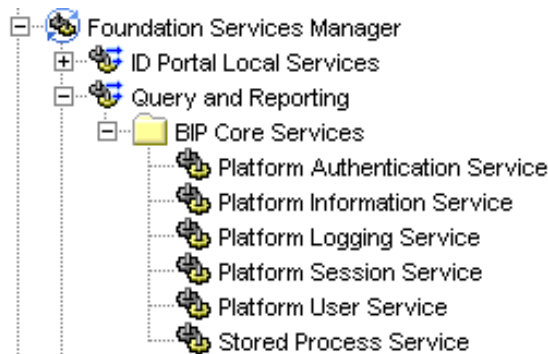
Notice that no authentication domain has been specified.

- 5 Select **DefaultAuth** from the **Authentication Domain** drop-down list, and click **OK**. The login for the SAS Trusted User is updated in the SAS Trusted User Properties dialog box.
- 6 Click **OK** in the SAS Trusted User Properties dialog box.

Configuring the Foundation Services Manager User Service

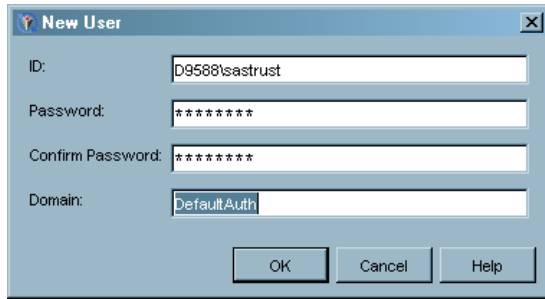
SAS Web Report Studio and SAS Information Delivery Portal use the Platform User Service, which is one of the SAS Query and Reporting Services. Configure this service so that it knows about the pool administrator (**sastrust**) by following these directions:

- 1 In SAS Management Console, expand the Foundation Services Manager. (SAS Foundation Services should have been installed when your system was first set up.) Most likely, you will see nodes for ID Portal Local Services and Remote Services. If this is the case, continue with step 2 to create the necessary query-and-reporting service deployment. (If you already have a Query and Reporting node, you can skip to step 6.)
- 2 Right-click the Foundation Services Manager icon, and select **Import Service Deployment** from the pop-up menu. An Import Service Deployment dialog box appears.
- 3 In the Import Service Deployment dialog box, click **Add**. An Open dialog box (a file-system browser) appears.
- 4 The service-deployment file that you need to select resides on the host where SAS Web Report Studio or SAS Web Report Viewer is deployed. Look in your J2EE server's deployment area for the appropriate application. From there, the path to the file is
WEB-INF\pfsconfig\config\host-name_sas_pfs_queryandreporting.xml.
 Select this file and click **Open**. The name of the file will display in the Import Service Deployment dialog box.
- 5 Click **OK** in the Import Service Deployment dialog box. The service deployment will be imported, and if you fully expand the Query and Reporting node, you should see the following tree nodes.



- 6 Right-click the Platform User Service icon, and select **Properties** from the pop-up menu. A Platform User Service Properties dialog box appears.
- 7 In the Platform User Service Properties dialog box, select the **Service Configuration** tab.
- 8 On the **Service Configuration** tab, click **Edit Configuration**. A User Service Configuration dialog box appears.
- 9 In the User Service Configuration dialog box, select the **Users** tab.
- 10 On the **Users** tab, click **Add**. A New User dialog box appears. This is where you specify the user ID of the pool administrator, who by default is the SAS Trusted User.

Note: For information about the pool administrator and how applications employ this user's ID, see "Planning the Pooling Security (IOM Bridge only)" in the *SAS Integration Technologies: Server Administrator's Guide*, which is available at support.sas.com/rnd/itech/doc9/admin_oma. △



11 Fill in the fields in the New User dialog box as shown in the previous display, changing the domain qualifier for the user ID as necessary. (Note that the user ID is case sensitive.) Then, click **OK**. The SAS Trusted User's user ID is now listed in the User Service Configuration dialog box.

12 Click **OK** in the User Service Configuration dialog box.

13 Click **OK** in the Platform User Service Properties dialog box.

Verifying That Connection Pooling Is Working for SAS Web Report Studio

If you have not configured connection pooling correctly, SAS Web Report Studio will continue to work; however, it will not be able to take advantage of a connection pool. Therefore, it is important that you verify that your system is configured correctly. You can verify this by temporarily changing the logging level for connections, viewing a SAS Web Report Studio report, and then checking the contents of the SAS Web Report Studio log file. Detailed instructions follow:

- 1 Change the logging level for connections by adding the following element to the file `web-server-deployment-area\WEB-INF\DefaultLoggerProperties.xml`.

```
<LoggingContext name      = "com.sas.services.connection"
                priority   = "INFO"
                chained    = "false">
  <OutputRef outputID = "WRS" />
</LoggingContext>
```

You can add this element directly below the existing LoggingContext element in the file.

CAUTION:

We recommend that you create a back-up copy of this XML file before editing it. An error in the XML syntax in this file can prevent SAS Web Report Studio from starting up properly. Δ

- 2 Start SAS Web Report Studio and log on.
- 3 View a report that causes a relational database query.
- 4 View the SAS Web Report Studio log file (`path-to-config-dir\Lev1\web\Deployments\WebReportStudio\logs\WebReportStudio.log`). If pooling is working, you will see information that is similar to this information about the connection to the workspace server:

```
privileged user name: D9588\sastrust
pd#0: putting cx#8 on the available queue
rq#0 routed to pd#0
```

If pooling is not working correctly, you will see a message similar to this message:

```
request shared by unshared connection #9
```

After you have confirmed that connection pooling is working, you can undo the changes that you made to `DefaultLoggerProperties.xml`.

Tuning Your J2EE Server or Servlet Container for Use with SAS Web Report Studio and SAS Information Delivery Portal

This section explains how you can best tune your J2EE server or servlet container, and the Java Virtual Machine (JVM) within that server, for use with SAS Web Report Studio and SAS Information Delivery Portal. The suggestions made are based on extensive performance testing and represent a reasonable balance between speed, scalability, and stability.

Note: SAS Web Report Viewer is configured similarly to SAS Web Report Studio. △

Tuning the Java Virtual Machine

Your J2EE server or servlet container's JVM can be started with a number of options that affect its behavior. For a quick overview of where to set these options and a list of generally applicable settings, see the following subsections:

- "Which JVM Are You Using?" on page 373
- "Where to Specify JVM Options" on page 374
- "Quick Start Settings" on page 375

For more details about the various JVM options, see the following subsections:

- "Setting Just-in-Time Compiler and Memory Options" on page 376
- "Selecting a Garbage Collector" on page 377
- "Configuring the Garbage Collector" on page 377
- "Tuning the J2EE Server or Servlet Container" on page 378
- "Detecting Changes in JavaServer Pages and Servlets" on page 378
- "Setting the Number of Available Worker Threads (SAS Information Delivery Portal Only)" on page 379

Which JVM Are You Using?

The information in this section applies both to the JVM that is supplied by Sun Microsystems and to the JVM that is supplied by IBM. Note that these two JVMs use different parameters. Most installations that use BEA WebLogic should use Sun's JVM and its corresponding parameters. Most installations that use IBM WebSphere should use IBM's JVM and its parameters. At the time this is being written, two exceptions are known:

- 1 On IBM's AIX operating system, only the JVM that is supplied by IBM should be used, even for BEA WebLogic.
- 2 On Sun's Solaris operating system, only the JVM that is supplied by Sun should be used, even for IBM WebSphere.

Where to Specify JVM Options

BEA WebLogic

The following procedures will enable you to set the JVM arguments for WebLogic version 8.1. If you are running the Node Manager, you can use the WebLogic Server administration console to set these options:

- 1 From the BEA WebLogic console, on the right panel under **Domain Configurations**, click on **Network Configuration**.
- 2 Click on **Servers**.
- 3 The resulting screen shows the managed servers (JVM processes). Select the appropriate server.
- 4 The next screen has two levels of tabs. Select **Configuration** (which should be first). Look for the **Remote Start** tab. That screen has an **Arguments** field where you can insert your JVM options. After inserting your these options, click .

If you are not using the Node Manager, you can set your JVM options in the `startManagedWebLogic.extension` script, which is located in `WebLogic-install-dir\user_projects\domains\domain`:

- 1 Open the script in a text editor.
- 2 Uncomment the line reserved for setting the `JAVA_OPTIONS` environment variable, and set this variable as explained later in this section.
- 3 Save your changes, and close the file.

Note: If you are running the Node Manager, you actually have the option of using the administration console or the `startManagedWebLogic` script to set JVM options. Use whichever option you use to set other WebLogic options. \triangle

IBM WebSphere

The IBM WebSphere administrative console is used to set Java startup parameters. The following procedure will enable you to set the parameters for IBM WebSphere version 5.1. For more information, see the IBM WebSphere documentation.

- 1 From the IBM WebSphere console, on the left panel, select **Servers**.
- 2 Select **Application Servers**.
- 3 On the right will be displayed a list of your servers. Select the appropriate server.
- 4 On the resulting screen, select **Process Definition**, then **Java Virtual Machine**.
- 5 Enter your Java parameters into the **Generic JVM Arguments** field.

Note: Some parameters can be specified either in other boxes on this screen, or in the **Generic JVM Arguments** field as recommended here. Avoid placing the same information in both areas; this can have unpredictable results. \triangle

Apache Tomcat

When the SAS Configuration Wizard runs, it creates a script that you can use to start your servlet container or J2EE server. This script is called `startServletContainer.bat` or `startServletContainer.sh` and resides in the directory `path-to-config-dir\Lev1\web`. This script provides one place in which you can specify JVM options. For example, if you are using the Apache Tomcat servlet container on a Windows system, the contents of the `startServletContainer.bat` script will look something like this:

```
set JAVA_HOME=C:\jdk1.4.2_04
set CATALINA_HOME=C:\Tomcat4.1
```

```
set CATALINA_OPTS=-Xms512m -Xms1024m -server -XX:-UseOnStackReplacement
-Djava.awt.headless=true
```

```
call "%CATALINA_HOME%\bin\catalina.bat" run -security
```

In this case, you can change the options that are used to start the JVM by changing the value of the environment variable `CATALINA_OPTS`. You can add JVM options to other versions of this script in a similar manner.

Quick Start Settings

If you want to start with a group of settings that will provide you with a convenient starting place, find the description in the following examples that matches your situation. Then, use the options that follow that description. (You can later fine tune these settings as necessary.)

If you will have 10 or fewer concurrent users and are using a Sun-based JDK whose revision level is 1.4.2_01 or later, use these settings:

```
-server -Xms512m -Xmx512m -XX:NewSize=64m -XX:MaxNewSize=64m
-XX:MaxPermSize=128m -Xss128k -XX:-UseTLAB -XX:+UseConcMarkSweepGC
-XX:+UseCMSCompactAtFullCollection -XX:CMSFullGCsBeforeCompaction=0
-XX:+DisableExplicitGC -Dsun.rmi.dgc.client.gcInterval=3600000
-Dsun.rmi.dgc.server.gcInterval=3600000 -Djava.awt.headless=true
```

If you will have 10 or fewer concurrent users and are using a Sun-based JDK whose revision level is between 1.4.1 and 1.4.2_00, use these settings:

```
-server -Xms512m -Xmx512m -XX:NewSize=64m -XX:MaxNewSize=64m
-XX:MaxPermSize=128m -Xss128k -XX:-UseTLAB -XX+UseParallelGC
-XX:+DisableExplicitGC -Dsun.rmi.dgc.client.gcInterval=3600000
-Dsun.rmi.dgc.server.gcInterval=3600000 -Djava.awt.headless=true
```

If you will have more than 10 concurrent users and are using a Sun-based JDK whose revision level is 1.4.2_01 or later, use these settings:

```
-server -Xms1280m -Xmx1280m -XX:NewSize=160m -XX:MaxNewSize=160m
-XX:MaxPermSize=128m -Xss128k -XX:-UseTLAB -XX:+UseConcMarkSweepGC
-XX:+UseCMSCompactAtFullCollection -XX:CMSFullGCsBeforeCompaction=0
-XX:+DisableExplicitGC -Dsun.rmi.dgc.client.gcInterval=3600000
-Dsun.rmi.dgc.server.gcInterval=3600000 -Djava.awt.headless=true
```

If you will have more than 10 concurrent users and are using a Sun-based JDK whose revision level is between 1.4.1 and 1.4.2_00, use these settings:

```
-server -Xms1280m -Xmx1280m -XX:NewSize=160m -XX:MaxNewSize=160m
-XX:MaxPermSize=128m -Xss128k -XX:-UseTLAB -XX+UseParallelGC
-XX:+DisableExplicitGC -Dsun.rmi.dgc.client.gcInterval=3600000
-Dsun.rmi.dgc.server.gcInterval=3600000 -Djava.awt.headless=true
```

If you will have 10 or fewer concurrent users and are using an IBM-based JDK whose revision level is 1.4.1 or later, use these settings:

```
-Xms512m -Xmx512m -Xgcpolicy:optavgpause -Xpartialcompactgc
-Dsun.rmi.dgc.client.gcInterval=3600000
-Dsun.rmi.dgc.server.gcInterval=3600000 -Djava.awt.headless=true
```

If you will have more than 10 concurrent users and are using an IBM-based JDK whose revision level is 1.4.1 or later, use these settings:

```
-Xms1280m -Xmx1280m -Xgcpolicy:optavgpause -Xpartialcompactgc
-Dsun.rmi.dgc.client.gcInterval=3600000
```

```
-Dsun.rmi.dgc.server.gcInterval=3600000 -Djava.awt.headless=true
```

Setting Just-in-Time Compiler and Memory Options

A number of JVM options affect which compiler the JVM uses and the amount of memory that the JVM uses for such things as the object heap.

Sun-Based JVM

You should always use the Just-in-Time compiler if one is available. You enable the JIT compiler with the `-server` option. The following list includes the relevant memory settings and their respective options:

- Minimum heap size (`-Xms`)
- Maximum heap size (`-Xmx`)
- Thread stack size (`-Xss`)
- Minimum new generation size (`-XX:NewSize`)
- Maximum new generation size (`-XX:MaxNewSize`)
- Maximum permanent generation size (`-XX:MaxPermSize`)

Minimum and maximum heap size, `-Xms` and `-Xmx` respectively, define the total amount of memory that the JVM has at its disposal. To eliminate heap growth overhead, set both of these options to the same value. The recommended value for heap size depends on which applications will be running and how much load will be supported. For 10 or fewer concurrent users, set the heap size to 512 MB (`-Xms512m -Xmx512m`). For more than 10 users, set the heap size to 1280 MB (`-Xms1280m -Xmx1280m`).

Although a 1280 MB heap is the maximum possible heap size on a 32-bit Windows system, most versions of UNIX allow the JVM to address up to 3 GB of memory. If your J2EE server is running on UNIX and you anticipate a load of hundreds of users, you might want to scale the implementation. However, large heap sizes can cause longer garbage collection times, so it is important to expand the heap only as much as necessary.

Thread stack size defines the default amount of memory allocated to each native thread spawned by the JVM. Keeping this value as low as possible allows the JVM to dedicate more process memory to the heap, which in turn increases scalability. The recommended setting is 128 KB (`-Xss128k`) for Windows and Solaris systems. You might need to adjust this number for other platforms or for additional load. As an added optimization, you can include the `-XX:-UseTLAB` option, which tells the JVM to minimize thread stack usage. This option is Sun JDK-specific and might not be supported on all platforms.

Finally, you can size different memory regions, or generations, appropriately to ensure efficient garbage collector performance. You use the minimum new generation size (`-XX:NewSize`), maximum generation size (`-XX:MaxNewSize`), and maximum permanent generation size (`-XX:MaxPermSize`) to control the sizes of the various memory regions. The new generation should receive about 12.5% of total memory, up to a maximum of about 256 MB. Both the minimum and maximum new generation settings should be set to the same value to avoid generation growth overhead. Assuming a 1280 MB heap, the calculation and associated options would look like this:

```
1280 x 12.5% = 160 MB (-XX:NewSize=160m -XX:MaxNewSize=160m)
```

The permanent generation is used to store loaded Java class definitions and extremely long-lived objects. Given the variety of components, services, and frameworks in use by SAS applications, the maximum permanent generation size should be set at 128 MB (`-XX:MaxPermSize=128m`).

IBM JVM

The recommendations for the Sun family of JDKs for the `-xms` and `-xmx` parameters are applicable here. The other options are not.

Selecting a Garbage Collector

The following subsections explain how to specify a garbage collector.

Sun-Based JVM

The two preferred garbage collectors are the concurrent mark and sweep collector and the parallel collector. Which collector you should choose depends on the version of the JDK that you are using.

If you are using JDK version 1.4.2_01 or later, you should use the concurrent mark and sweep collector. You enable this collector by specifying the `-XX:+UseConcMarkSweepGC` option. If you are using an earlier version of the JDK, do not use this collector. Earlier versions of the collector contained significant flaws, which result in poor performance.

If you are using a version of the JDK between 1.4.1 and 1.4.2_00, the parallel collector is the best choice. You enable this collector by using the `-XX:+UseParallelGC` option. It does not perform as well as the concurrent mark and sweep collector, but its performance is still much better than that of the other collectors available.

IBM JVM

You control the IBM JVM's garbage collection behavior using the `-Xgcpolicy` switch. The `optavgpause` policy provides garbage collection behavior similar to that of Sun's concurrent mark and sweep collector. Select this policy (`-Xgcpolicy:optavgpause`) to ensure stable server response times.

Configuring the Garbage Collector

After you have chosen a garbage collector, configure the collector appropriately.

Sun-Based JVM

If you are using the concurrent mark and sweep garbage collector, force the collector to compact memory regions at each collection. This results in a reduced number of garbage collection runs and also reduces the duration of each run because overall memory pool fragmentation will be reduced. Always use the

`-XX:+UseCMSCompactAtFullCollection` option. It tells the collector to compact memory during full collection runs. Also, configure the collector to perform compaction on each full collection run. Do this by using the option `-XX:CMSFullGCsBeforeCompaction=0`.

If you are using the parallel garbage collector, the JVM's default settings provide good performance for most loads. However, because the parallel garbage collector uses multiple threads to achieve parallel operation under heavy loads, it might help to increase the number of threads. Use the `-XX:ParallelGCThreads` option to control the number of garbage collector threads (which are separate from application threads and are dedicated to garbage collections tasks). By default, the JVM creates as many threads as there are installed CPUs. For example, the default setting on a four-way symmetric multiprocessing server is effectively `-XX:ParallelGCThreads=4`. For a very large server with multiple JVMs running (for example, eight or more CPUs), you might want to reduce the number of threads. It is NOT recommended that you raise this value higher than the number of CPUs on your server.

You can also exercise control over how often the garbage collector runs by using one of three options. The `-XX:+DisableExplicitGC` option prevents Java code from invoking the garbage collector. You should always use this option. In addition, there are two Java

properties that you can set to control distributed garbage collection. (This is important because most SAS Java applications use Remote Method Invocation, which in turn uses distributed garbage collection.) Setting the `sun.rmi.dgc.client.gcInterval` and `sun.rmi.dgc.server.gcInterval` command line properties to 3600000 will reduce distributed garbage collection runs from once a minute (the default) to once an hour.

IBM JVM

Select the `-Xpartialcompactgc` parameter, which spreads the time of compacting the memory across multiple collections, rather than waiting until memory is completely fragmented. If this parameter is not selected, excessive garbage collection times can be experienced, especially under heavy load.

Tuning the J2EE Server or Servlet Container

In addition to specifying Java Virtual Machine options, you can improve the performance of SAS Web Report Studio and SAS Information Delivery Portal by configuring other aspects of your servlet container or J2EE server's behavior. For example, two obvious ways to improve the performance of any Web application are

- to limit the frequency with which servers check for updated JavaServer pages and servlets
- to make sure that the server can create sufficient threads to service incoming requests.

The following sections explain how to change these settings on the Web servers that are supported by the SAS Intelligence Platform.

Detecting Changes in JavaServer Pages and Servlets

Most servlet containers and J2EE servers perform a periodic check of compiled class files and source files to determine whether a servlet or JavaServer page has been edited. This behavior is only appropriate while applications are under development. In your production environment, you should disable these features. The following subsections explain how to do this for the supported J2EE servers and Apache Tomcat.

BEA WebLogic Server

BEA WebLogic does not require tuning for JSP changes for use with SAS Web Report Studio and SAS Information Delivery Portal. The `weblogic.xml` file that is shipped with both applications performs all necessary tuning functions.

IBM WebSphere

IBM WebSphere does not require tuning for JSP changes or the number of threads available.

Apache Tomcat

- 1 Open the file `Tomcat-install-dir\conf\web.xml`.
- 2 Add the following XML to the `<init-param>` block for the servlet with the servlet name `jsp`:

```
<init-param>
  <param-name>reloading</param-name>
  <param-value>>false</param-value>
</init-param>
<init-param>
  <param-name>development</param-name>
  <param-value>>false</param-value>
```

```
</init-param>
```

- 3 Restart the Tomcat server.

Setting the Number of Available Worker Threads (SAS Information Delivery Portal Only)

Because each user request requires a server thread to service it, the server must be able to start a sufficient number of threads to service the expected load. The following sections explain how to control the number of available of threads on the different supported servers.

WebLogic Server

- 1 Log on to the administration console.
- 2 Open the Servers node of the tree.
- 3 Right-click a server definition.
- 4 Select the **View Execute Queues** menu option.
- 5 Select the **Configure a new Execute Queues** menu option.
- 6 Enter `sas.portal.default` as the queue name.
- 7 Click .

Note: The default **Thread Count** value of 25 should be sufficient for most loads. The BEA Web site (e-docs.bea.com/platform/docs81/index.html) contains specific information about tuning execute queues. △

Apache Tomcat

- 1 Open the file `Tomcat-install-dir\conf\server.xml`.
- 2 Locate the **Connector** element for the server's listening port. By default, this is port 8080.
- 3 Change the value of the `maxProcessors` attribute to a number greater than 75 (the default).
- 4 Restart the server.

Note: The default value of 75 provides good performance for most loads. You should need to change this setting only for very heavy loads. △

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 an object spawner and 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

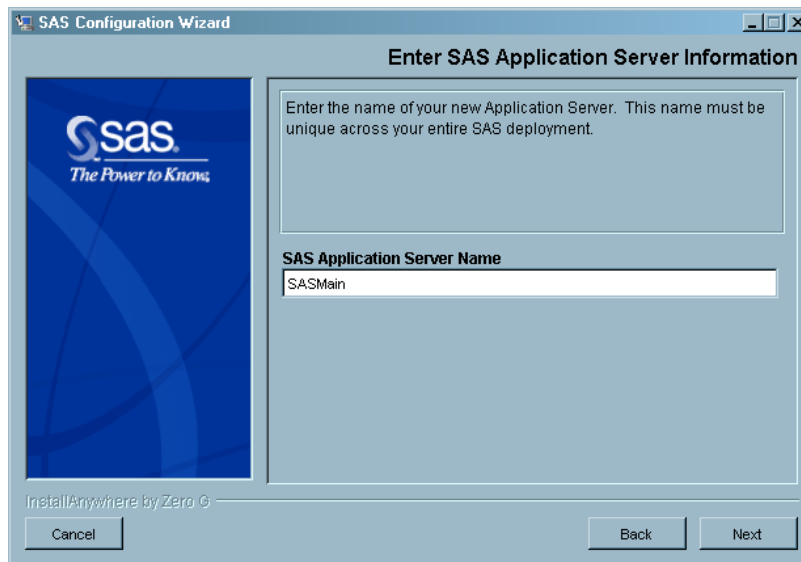
to your intelligence system and that you want to run a second workspace server on that host. Your SAS representative will be able to get you the software that you need—SAS Foundation and SAS Management Console—and to prepare the planning file that you need for the installation.

For information on how to perform a project installation using the SAS Software Navigator, see “Installing Software on a Machine” on page 87.

Configuring the Workspace Server and Object Spawner

After you have installed SAS Foundation software and SAS Management Console software, you should run the SAS Configuration Wizard on the new host. The wizard will prompt you for information about a configuration directory and about the credentials for certain users. Because you are not installing a metadata server on this machine, the wizard will also prompt you for an application server as shown in the following display.

Display 19.1 Enter SAS Application Server Information Window



The assumption behind this prompt is that your workspace server will be part of a new application server. In the case we are considering, however, this assumption is incorrect; you want to add the new server to an existing logical workspace server, which is part of an existing application server (probably called **SASMain**). The best practice is to enter the name of your new host computer when you are prompted for an application server. This will not have any effect—except on the HTML instructions that the SAS Configuration Wizard generates.

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. \triangle

Start the SAS Management Console

SAS Management Console should start automatically. If it does not, start it by following the instructions that were generated by the SAS Configuration Wizard. You will need to use this application to define your workspace server and object spawner in the metadata.

Defining Your Application Server and Workspace Server

Do not follow the instructions in the section “Defining Your Application Server and Workspace Server.” You have already defined an application server in your metadata—when you first installed your system—so you do not need to define one now, and you need to define your new workspace server in a manner different from the one described in `instructions.html`.

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 that 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 that are detailed in the sections “Converting the Logical Workspace Server to Load Balancing” on page 381 and “Setting Load Balancing Parameters for Each Workspace Server” on page 382.

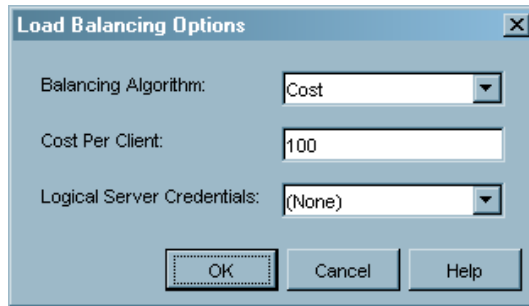
Converting the Logical Workspace Server to Load Balancing

After you have defined your new workspace server and object spawner in the metadata, you can use SAS Management Console to convert your logical workspace server to load balancing. During this process, you set some options that apply to all of the physical workspace servers in the logical workspace server.

Note: If you do not convert the logical workspace server to use load balancing, clients will be able to use only the first physical workspace server in the logical workspace server. Δ

To convert the logical workspace server to load balancing, perform these steps:

- 1 Right-click **SASMain - Logical Workspace Server**, and select **Convert To ► Load Balancing**. You will be asked whether you want to continue. Click **Yes**. The Load Balancing Options dialog box appears.



- 2 Set the parameters in this dialog box using the explanations in the following table, and click **OK**.

Table 19.4 Load Balancing Options

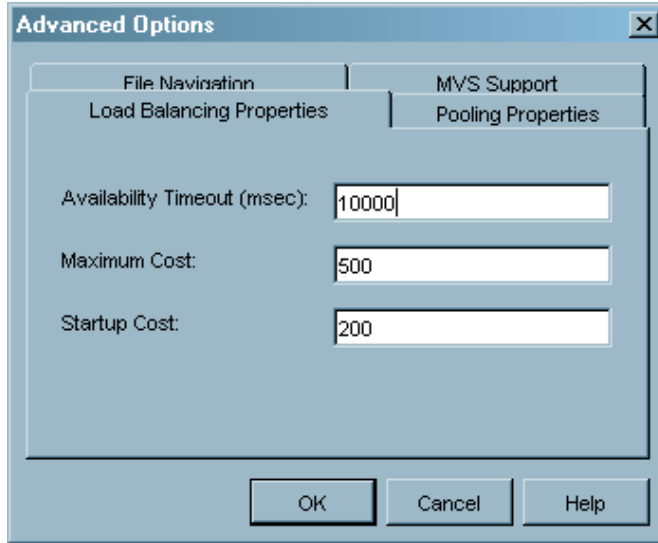
Parameter	Explanation
Balancing Algorithm	Specifies the algorithm that the object spawners should use to control load balancing. The only valid choice when you are load balancing workspace servers is Cost. The Cost algorithm specifies that client requests are processed based on the current server cost and the start-up cost of a new server. The clients' costs are added to and subtracted from the server cost as they connect and disconnect. For more information on the Cost algorithm, see the <i>SAS Integration Technologies: Server Administrator's Guide</i> at support.sas.com/rnd/itech/doc9/admin_oma/ .
Cost Per Client	Specifies the default amount of weight (cost) that each client adds to (on connection) or subtracts from (on disconnection) the total cost of the server.
Logical Server Credentials	Shows the credentials that the object spawners on the two hosts will use to communicate about load balancing. We recommend that you use the SAS General Servers group login (sassrv) for this purpose. This account will be used in both directions, so it must be a network account that will be valid on both spawner hosts.

Setting Load Balancing Parameters for Each Workspace Server

There are also some load-balancing properties that you set for each of your physical workspace servers. To set these properties, perform these steps:

- 1 In SAS Management Console, right-click the icon for the physical server, and select **Properties** from the pop-up menu. A Properties dialog box appears.
- 2 Select the **options** tab.

- 3 Click **Advanced Options**. An Advanced Options dialog box appears.



Select the **Load Balancing Properties** tab if it is not already selected.

- 4 Set the load-balancing properties using the information in the following table; then, click **OK**.

Table 19.5 Load Balancing Properties

Property	Explanation
Availability Timeout (msec)	Specifies the number of milliseconds to wait for an available server. The wait can be caused by the time that is required for a server to start or the time that is required for a running server to become available.
Maximum Cost	Specifies the maximum cost allowed on the server before requests to the server are denied. Use the value of the Cost Per Client field on the logical server to determine this value based on the number of client connections allowed.
Startup Cost	Specifies the start-up cost of the server. When a request is made to the load balancer, the load balancer assigns this start-up cost value to inactive servers. A new server is not started unless it is determined that its cost (the start-up cost) is less than the cost of the rest of the servers in the cluster. This field enables the administrator to control the order in which servers are started. After a server is started, the cost value is 0. When a client connects to the server, the server's cost value is increased.

- 5 Click **OK** in the Properties dialog box.

You can now start your object spawner.

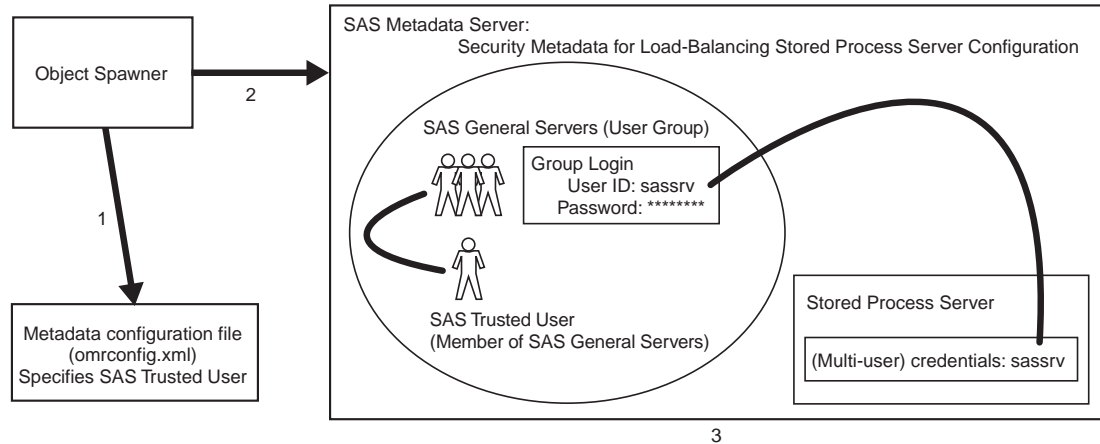
Overview of the Initial Load Balancing Setup for Stored Process Servers

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 19.1 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*. Δ

In the preceding figure, the object spawner obtains the metadata it needs to start a load balancing stored process server as follows:

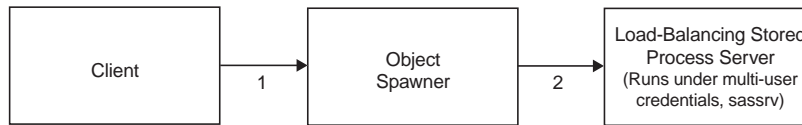
- 1 When the spawner is started, it reads a metadata configuration file named **omrconfig.xml** that contains information required to access the metadata server. This metadata configuration file specifies
 - the location of the metadata server
 - the user ID that the spawner will use to connect to the 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. Δ

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 19.2 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. Δ

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 385.

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

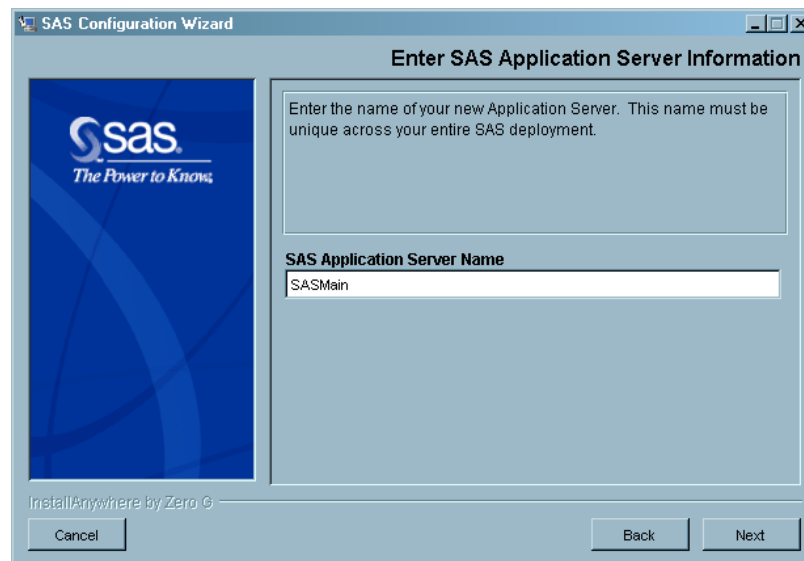
Installing the software is easy. Just let your SAS representative know that you want to add a new host to your intelligence system and that you want to run a second stored process server on that host. Your SAS representative will be able to get you the software that you need—SAS Foundation and SAS Management Console—and to prepare the planning file that 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 87.

Configuring the Stored Process Server and Object Spawner

After you have installed SAS Foundation software and SAS Management Console software, you should run the SAS Configuration Wizard on the new host. The wizard will prompt you for information about a configuration directory and about the credentials for certain users. Because you are not installing a metadata server on this machine, the wizard will also prompt you for an application server as shown in the following display.

Display 19.2 Enter SAS Application Server Information Window



The assumption behind this prompt is that your stored process server will be part of a new application server. In the case we are considering, this assumption is incorrect. You want to add the new server to an existing logical stored process server, which is part of an existing application server (probably called **SASMain**). The best practice is to enter the name of your new host when you are prompted for an application server. This will not have any effect—except on the HTML instructions 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. Δ

Start the SAS Management Console

SAS Management Console should start automatically. If it does not start automatically, start the application by following the instructions in this section. You

will need this application to define your stored process server and object spawner in the metadata.

Defining Your Application Server and Workspace Server

Skip the instructions in the sections “Defining Your Application Server and Workspace Server” and “Edit the SAS Command for the Workspace Server.” You have already defined an application server in your metadata—when you first installed your system—and you are not adding a workspace server.

Define Your Stored Process Server

Also, skip the instructions in the sections “Define Your Stored Process Server” and “Define the Stored Process Server as Supporting Load Balancing.” We are assuming that you have already defined a logical stored process server called **SASMain - Logical Stored Process Server** and that you have configured that logical server for load balancing.

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 388 and “Setting the Load Balancing Properties for Each Stored Process Server” on page 390.

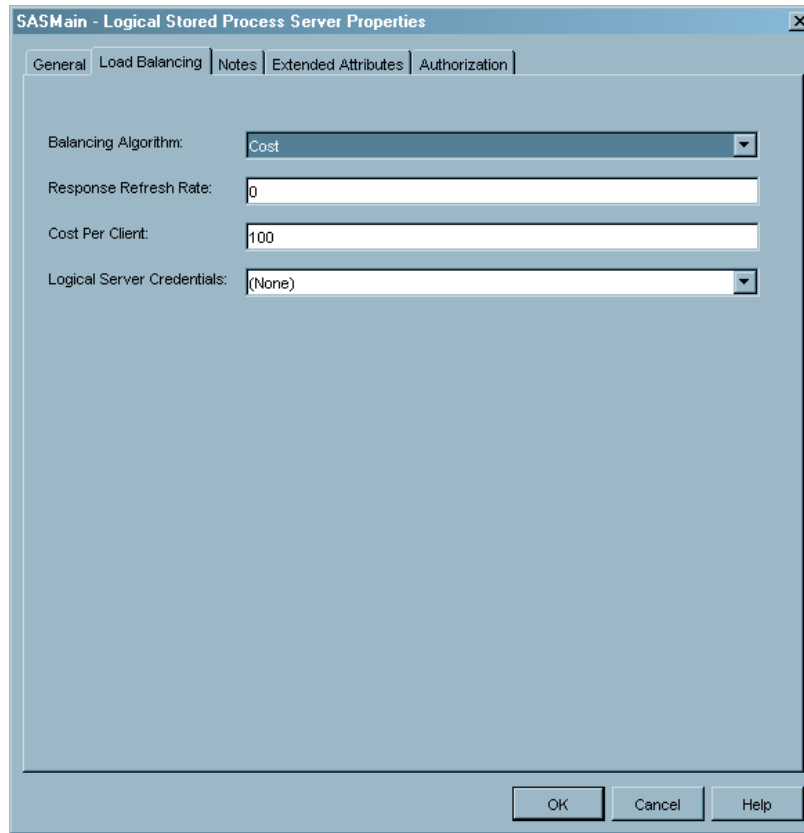
Setting Logical Stored Process Server Properties

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. \triangle

- 2 Right-click the icon for your logical stored process server, and select **Properties** from the pop-up menu. A properties dialog box appears.
- 3 Select the **Load Balancing** tab. This tab contains the parameters that you can set.



4 Set these parameters using the explanations in the following table, and click **OK**.

Table 19.6 Load Balancing Parameters

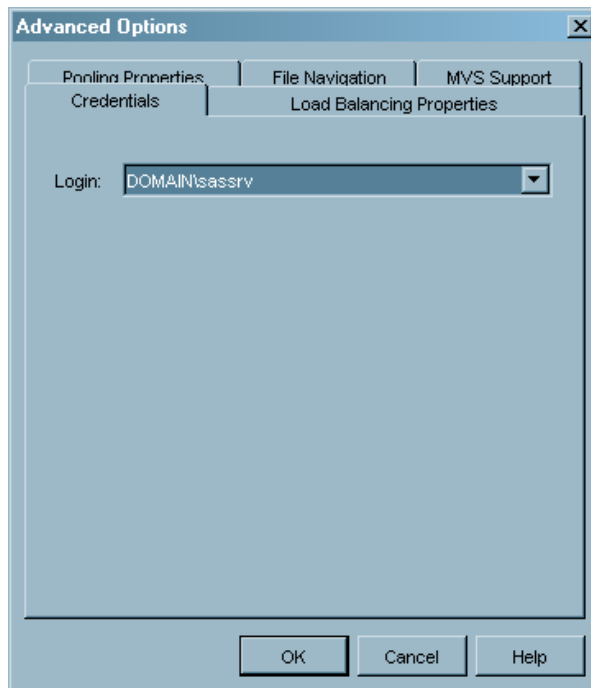
Parameter	Explanation
Balancing Algorithm	Specifies the algorithm that the load balancer should use to control load balancing. Valid values are Cost and Response Time. Selecting the Cost algorithm specifies that client requests are processed based on the current server cost and the start-up cost of a new server. The clients' costs are added to or subtracted from the server cost as they connect and disconnect. Selecting Response Time specifies that client requests are allocated based on server response times. For more information about these load balancing algorithms, see the <i>SAS Integration Technologies: Server Administrator's Guide</i> at support.sas.com/rnd/itech/doc9/admin_oma/ .
Response Refresh Rate	Specifies how often the server response times are checked. You only enter a value in this field if you selected Response Time in the Balancing Algorithm field, and the value should always be set to -1.

Parameter	Explanation
Cost Per Client	Specifies the default amount of weight (cost) that each client adds to (on connection) or subtracts from (on disconnection) the total cost of the server. (Cost algorithm only.)
Logical Server Credentials	Shows the credentials that the object spawners on the two hosts will use to communicate about load balancing. We recommend that you use the SAS General Servers group login (sassrv) for this purpose. This account will be used in both directions, so it must be a network account that will be valid on both spawner hosts.

Setting the Load Balancing Properties for Each Stored Process Server

There are also some load balancing properties that you should set for each of your physical stored process servers. Follow these steps to set these properties:

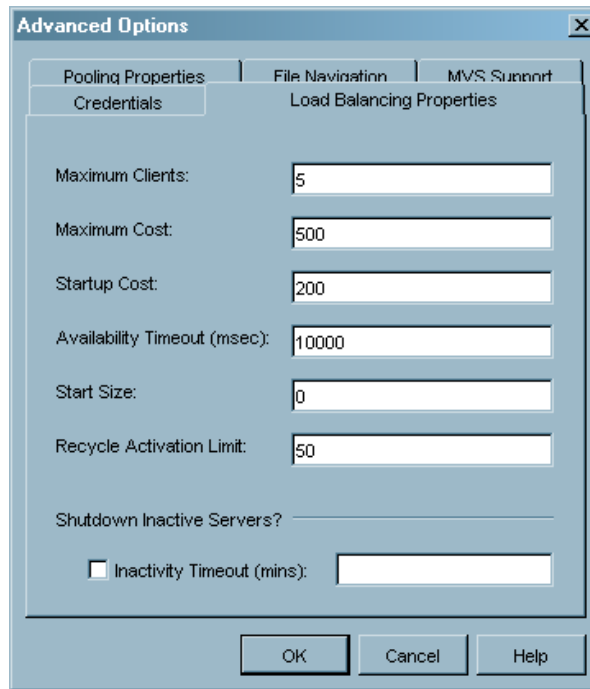
- 1 In SAS Management Console, right-click the icon for the stored process server, and select **Properties** from the pop-up menu that appears. A Properties dialog box appears.
- 2 Select the **Options** tab.
- 3 Click **Advanced Options**. An Advanced Options dialog box appears, and the **Credentials** tab displays.



- 4 From the **Login** list box, select the account that will be used to start stored process servers. We recommend that you use the SAS General Servers group login (**sassrv**) for this purpose.

Note: This should be a network account so that the stored process servers will be able to access data resources on file servers on the network. Δ

- 5 Select the **Load Balancing Properties** tab.



6 Set the load balancing properties using the information in the following table; then, click **OK**.

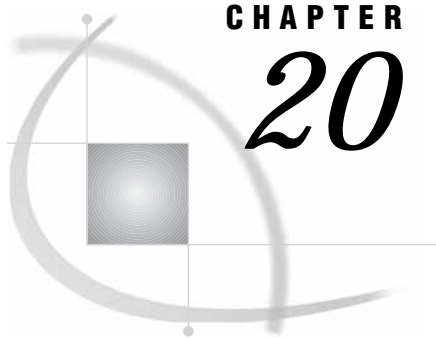
Table 19.7 Load Balancing Properties

Property	Explanation
Maximum Clients	The maximum number of simultaneous clients connected to this server. (Response Time algorithm only.)
Maximum Cost	The maximum cost allowed on each server before requests to the server are denied. (Cost algorithm only.)
Startup Cost	The cost of starting a server. (Cost algorithm only.)
Availability Timeout (msec)	The number of milliseconds to wait for a load balancing server to become available. This parameter is used (1) when all servers have allocated the maximum number of clients per server and (2) when the load balancer is waiting for a server to start and become available for its first client.
Start Size	The number of MultiBridge connections to start when the spawner starts.
Recycle Activation Limit	The number of times a connection to the server will be reused before it is disconnected ("recycled"). If the value is 0, then there will be no limit on the number of times a connection to the server can be reused. This property is optional. The default value is 0.

Property	Explanation
Shutdown Inactive Servers?	Indicates what you want a server to do when it is not currently serving a client. Select this check box to indicate that you want the process to terminate; otherwise, the server will remain active.
Inactivity Timeout (mins)	If you elected to shut down inactive servers, this field specifies how many minutes of inactivity must pass before the server terminates.

7 Click in the Properties dialog box.

You can start your object spawner now.



CHAPTER 20

Promoting and Replicating Metadata

<i>Overview of Promoting and Replicating Metadata</i>	393
<i>Preparing for Replication and Promotion</i>	394
<i>An Overview of the Required Steps</i>	395
<i>Assumptions</i>	396
<i>Step 1. Verifying SAS 9.1 Installations</i>	396
<i>Step 2. Configuring the Administration Metadata Server</i>	396
<i>Step 3. Creating the SAS Replication Administrator Operating System Account</i>	397
<i>Step 4. Defining the SAS Replication Administrator in SAS Management Console</i>	398
<i>Step 5. Creating the Metadata Access File on the Source Metadata Server Machine</i>	399
<i>Server Status Check</i>	399
<i>Step 6. Authorizing the Source Metadata Server Administrator on the Source Metadata Server Machine</i>	399
<i>Step 7. Defining the Target Metadata Server Administrator in the Source Metadata Repository in SAS Management Console</i>	400
<i>Step 8. Configuring the Target Metadata Server</i>	401
<i>Step 9. Defining a SAS/CONNECT Server for the Target Metadata Server Machine</i>	402
<i>Step 10. Authorizing the Target Metadata Server Administrator on the Target Metadata Server Machine</i>	402
<i>Step 11. Defining the Source Metadata Server in SAS Management Console</i>	403
<i>Step 12. Adding a SAS Workspace Server Component to the Source Metadata Server Definition</i>	403
<i>Step 13. Defining the Target Metadata Server in SAS Management Console</i>	404
<i>Step 14. Adding a SAS/CONNECT Server Component to the Target Metadata Server Definition</i>	405
<i>Step 15. Defining an IOM Object Spawner for the SAS Workspace Server Component</i>	406
<i>Creating a Promotion Job</i>	407
<i>Running a Promotion Job</i>	413
<i>Making Modifications After Promotion to the Target Metadata Server</i>	413
<i>Modifications Related to the Server Definitions</i>	413
<i>Modifications Related to Cubes</i>	415
<i>Creating a Replication Job</i>	415
<i>Running a Replication Job</i>	420
<i>Troubleshooting Replication and Promotion</i>	420

Overview of Promoting and Replicating Metadata

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*. \triangle

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.

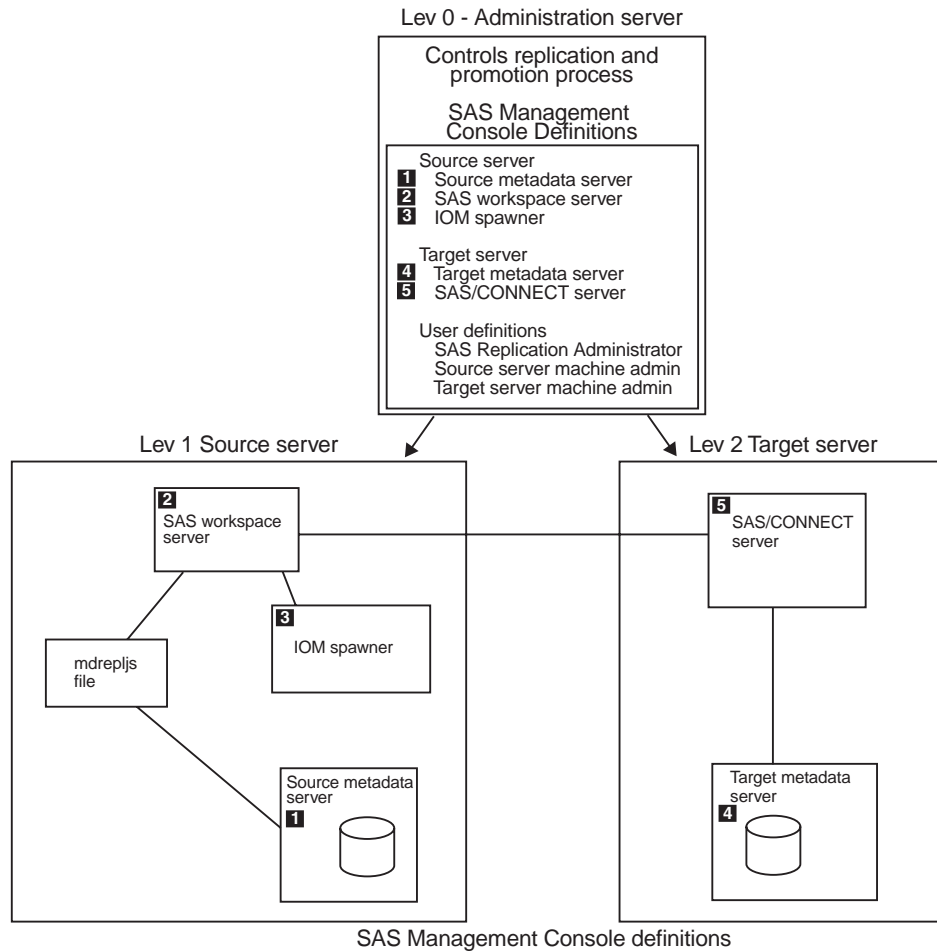
Note: For more information about SAS application servers, see Chapter 2, “Servers in the SAS Intelligence Platform,” on page 15. \triangle

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 431):

- \square the administration metadata server from which the replication or promotion job is run. The production level that is identified as Lev0 represents the administration metadata server in the metadata and in the directory structure that the SAS Configuration Wizard creates.
- \square the source metadata server from which metadata is promoted or replicated. The production level that is 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.
- \square the target metadata server to which the metadata is promoted or replicated. The production level that is 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 that are 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 20.1 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 that was 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 Create the metadata access file on the source metadata server machine.
- 6 Authorize the source metadata server administrator on the source metadata server machine.
- 7 Define the target metadata server administrator in the source metadata repository in SAS Management Console.
- 8 Configure and start the target metadata server.
- 9 Define and start the SAS/CONNECT server for the target metadata server machine.

- 10 Authorize the target metadata server administrator on the target metadata server machine.
- 11 Define the source metadata server in SAS Management Console.
- 12 Add a SAS Workspace Server component to the source metadata server definition.
- 13 Define the target metadata server in SAS Management Console.
- 14 Add a SAS/CONNECT server component to the target metadata server definition.
- 15 Define an IOM Object Spawner for the SAS Workspace Server component.

Assumptions

The tasks that you are instructed to perform assume that these conditions are true:

- 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.
- 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 that were provided in the `instructions.html` file that the SAS Configuration Wizard displayed during the configuration process.

Note: These instructions are for migrating metadata to a target environment; they do not address migrating your data into the target environment. To fully test the metadata migration—which might include load processes that update data—you should use your established data copying procedures. You can perform a full copy or move a data sample. In addition, you can move DBMS data to different schemas so that it can be tested without updating production data. Δ

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 the same information that you entered to create the source metadata server, including using the same configuration name and the same paths.

The differences between the configuration of the source metadata server and the administration metadata server are listed in these steps:

- 1 Launch the SAS Configuration Wizard by using the CD Index, which is part of a basic installation.

Note: For more information about basic installations, see Appendix 2, “Software Index Installations,” on page 443. Δ

- 2 On the Select Install Set window, select **Custom** and then choose to install only the metadata server.

Note: The **Custom** option is available only when you perform a basic installation. Δ

- 3 On the Enter SAS Metadata Server Information window, enter **8560** instead of **8561** as the port number.
- 4 On the Advanced Properties Editor window, click **Edit Properties**. In the text file that opens, make the changes that are shown in the following table.

Table 20.1 Values for the Administration Metadata Server

Replace Occurrences of These	With This Value
Lev1	Lev0

- 5 Save the changes to the file and close the file to return to the SAS Configuration Wizard.
- 6 Click **Next** to display the Finish window, then click **Finish**.
- 7 When the **instructions.html** file appears in your browser, close the window. Do not follow the instructions in the **instructions.html** file. Instead, you must follow the instructions in this chapter.
- 8 If SAS Management Console launches, exit the application.
- 9 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** \blacktriangleright **Programs** \blacktriangleright **Sas** \blacktriangleright **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**.

Note: Do *not* insert an asterisk before **hostname\sasrpadm**. The **sasrpadm** cannot be an unrestricted user. Δ

- 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 login, the user ID must belong to an administrative user on the associated machine. (The target metadata server administrator should be an unrestricted user.) Each administrative user must have the security permissions that are required to perform these tasks:

- write to the applicable directories that are 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** \blacktriangleright **User Manager**.
- 4 Select **Actions** \blacktriangleright **New** \blacktriangleright **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 (we recommend using *admin server hostname\sasrpadm*. 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
 - be entered in the form *source server hostname\userid* (for example, **D1234\srcadmin**)
 - 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 that is located on the source (Lev1) machine. (Do *not* include an asterisk before the name.)
- 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
 - be entered in the form *target server hostname\userid* (for example, **D5678\trgadmin**)
 - be an unrestricted user
 - be in the **adminUsers.txt** file that is located on the target (Lev2) machine. (An asterisk *should* be prepended to the name.)

Note: For detailed help in creating a new user, click **Help** in the User Manager plug-in application. Δ

Note: For information about creating unrestricted users, see the *SAS Metadata Server: Setup Guide*, which is available at support.sas.com/rnd/eai/openmeta/v9/setup. △

Step 5. 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 `mdrep1js.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 `xxxxxx` 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\`.

Server Status Check

At this point, the following server conditions must exist on the source metadata server machine:

- the metadata server must be running
- the IOM Object Spawner must be running and the SAS Workspace Server must be available for services.

Step 6. Authorizing the Source Metadata Server Administrator on the Source Metadata Server Machine

In step 8 under “Step 4. Defining the SAS Replication Administrator in SAS Management Console” on page 398, you selected a user (the example was

D1234\srcadmin) when you defined a login for the SAS Replication Administrator on the source metadata server machine.

As explained previously, this user must have write permissions to these directories:

- the directories under the **ReplicationWorkArea** directory (typically located in **C:\SAS\9.1\Lev1\SASMain\MetadataServer\ReplicationWorkArea**)
- the directory of the repository that is being promoted (typically located in **C:\SAS\9.1\Lev1\SASMain\MetadataServer\MetadataRepositories\Foundation**).

Operating system administrators have write permission to all directories, so, if the user is defined as an operating system administrator on the source metadata server machine, then the user is properly authorized.

If the user is *not* an operating system administrator, then complete these steps:

- 1 In Windows Explorer, select the **ReplicationWorkArea** directory.
- 2 Select **File** \blacktriangleright **Properties**.
- 3 In the Properties dialog box, select the **Security** tab.
- 4 Click **Add** to add the user to the **Group or user names** box.
- 5 With the user selected in the **Group or user names** box, select the **Allow** check box for the Modify permission in the **Permissions for** box.
- 6 Click **OK** to save your changes and close the Properties dialog box.

Perform similar steps to give the same user Modify permission to the directory of the repository that is being promoted, then complete these steps:

- 1 Stop the source (Lev1) metadata server service.
- 2 Add the user to the **adminUsers.txt** file on the source metadata server machine. Use the form *source server hostname\userid* (for example, **D1234\srcadmin**). (Do *not* include an asterisk before the name.)
- 3 Restart the source (Lev1) metadata server.
- 4 If it is not already running, start the Object Spawner.

Step 7. Defining the Target Metadata Server Administrator in the Source Metadata Repository in SAS Management Console

In step 9 under “Step 4. Defining the SAS Replication Administrator in SAS Management Console” on page 398, you selected a user as the target server administrator (the example was **D5678\trgadmin**). In SAS Management Console, you must ensure that this user has ReadMetadata and WriteMetadata access to the source metadata repository. To do this, you define the target metadata server administrator on the source metadata server machine and then you assign the permissions.

Note: After promotion, the target metadata server administrator will exist in the target metadata server repository. Δ

- 1 Use SAS Management Console to connect to the source metadata server as SAS Administrator (**sasadm**).

Note: The SAS Administrator is one of the user accounts that you created when you completed the pre-installation checklist. Δ

- 2 In the SAS Management Console navigation tree, select **Environment Management** \blacktriangleright **User Manager**.
- 3 Select **Actions** \blacktriangleright **New** \blacktriangleright **User**.

- 4 On the **General** tab, enter **SAS Target Replication Administrator** as the name.
- 5 Click the **Logins** tab.
- 6 Click **[New]** and complete the New Login Properties dialog box to create the login for the source metadata server. Enter the user ID in the form *source server hostname\userid* (for example, **D1234\trgadmin**). Leave the **Password** field blank. Do not specify an authentication domain.
- 7 Select **Environment Management ► Authorization Manager ► Access Control Templates ► Default ACT**.
- 8 Select **File ► Properties**.
- 9 In the Properties dialog box, select the **Users and Permissions** tab.
- 10 Click **[Add]** to open the Add Users and/or Groups dialog box. Move the **SAS Target Replication Administrator** to the **Selected Identities** box. Click **[OK]** to close the dialog box.
- 11 On the **Users and Permission** tab, select the **SAS Target Replication Administrator**.
- 12 In the Permissions dialog box, grant ReadMetadata and WriteMetadata to the **SAS Target Replication Administrator**.
- 13 Click **[OK]** to close the dialog box and save your settings.
- 14 Exit SAS Management Console.

Step 8. 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 Launch the SAS Configuration Wizard by using the CD Index, which is part of a basic installation.

Note: For more information about basic installations, see Appendix 2, “Software Index Installations,” on page 443. △
- 2 On the Select Install Set window, select **Custom** and then choose to install the metadata server and SAS/CONNECT server.

Note: The **Custom** option is available only when you perform a basic installation. △
- 3 On the Enter SAS Metadata Server Information window, enter **8562** instead of **8561** as the port number.
- 4 When the Advanced Properties Editor window appears, click **[Edit Properties]**. In the text file that opens, make these changes.

Table 20.2 Values for the Target Metadata Server

Replace Occurrences of These	With This Value
8591	8592
7551	7552
Lev1	Lev2

- 5 Save the changes to the file and close it to return to the SAS Configuration Wizard.

- 6 Click **Next** to display the Finish window, then click **Finish**.
- 7 When the `instructions.html` file appears in your browser, close the window. Do not follow the instructions in the `instructions.html` file. Instead, you must follow the instructions in this chapter.
- 8 If SAS Management Console launches, exit the application.
- 9 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** \blacktriangleright **Programs** \blacktriangleright **Sas** \blacktriangleright **name of your configuration directory**. Δ

Step 9. 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 Locate the line that begins `echo Manual start` and check to see if the `-security` parameter is present at the end of the start command. If it is not present, then add it. For example, here is the command with `-security` present:

```
echo Manual start of the Spawner using default port of %CONNPORT%
start /b 'Connect Spawner' '%sasdir%\spawner' -service %CONNPORT%
-SASCMD %CONNCMD% -security
```

- 5 Save your changes.
- 6 Run the `StartConnectServer.bat` file to start the SAS/CONNECT server.

Step 10. Authorizing the Target Metadata Server Administrator on the Target Metadata Server Machine

In step 9 under “Step 4. Defining the SAS Replication Administrator in SAS Management Console” on page 398, you selected a user (the example was `D5678\trgadmin`) when you defined a login for the SAS Replication Administrator on the source metadata server machine (see step 9 under “Step 4. Defining the SAS Replication Administrator in SAS Management Console” on page 398).

As explained previously, this user should be an unrestricted user.

- 1 Stop the target (Lev2) metadata server service.
- 2 Add the user to the `adminUsers.txt` file on the target metadata server machine. Use the form `target server hostname\userid` (for example, `D5678\trgadmin`). (Include an asterisk before the name.)
- 3 Restart the target (Lev2) metadata server.

Step 11. 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:

<i>Server type</i>	SAS Application Server
<i>Name</i>	Lev 1 - <i>application server name</i> (for example, Lev 1 - SASMain)
<i>SAS server type</i>	Metadata Server
<i>Authentication Domain</i>	ReplicationSourceAuth
<i>Host Name</i>	Source server host name
<i>Port</i>	8561

The following display illustrates how the Server Manager navigation tree appears after the source metadata server has been defined.

Display 20.1 Source Metadata Server Defined



For more information about defining a server, see the *SAS Management Console: User's Guide*.

Step 12. 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 11. Defining the Source Metadata Server in SAS Management Console” on page 403. Complete these steps to launch the wizard:

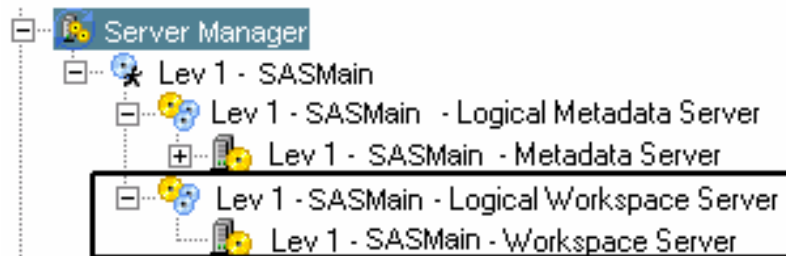
- 1 In the SAS Management Console navigation tree, select **Environment Management ► Server Manager ► source metadata server name**.
For example, select **Environment Management ► Server Manager ► Lev 1 - SASMain**.
- 2 Select **Actions ► Add Application Server Component**.

When defining the server component in the New Application Server Component wizard, specify these properties:

<i>SAS server type</i>	Workspace Server
<i>Configuration</i>	Custom
<i>Authentication Domain</i>	ReplicationSourceAuth
<i>Host Name</i>	Source server host name
<i>Port</i>	8591

The following display illustrates how the Server Manager navigation tree appears after the SAS Workspace Server component has been added.

Display 20.2 SAS Workspace Server Component Added to the SASMain Application Server



For more information about defining an application server component, see the *SAS Management Console: User's Guide*.

Step 13. 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:

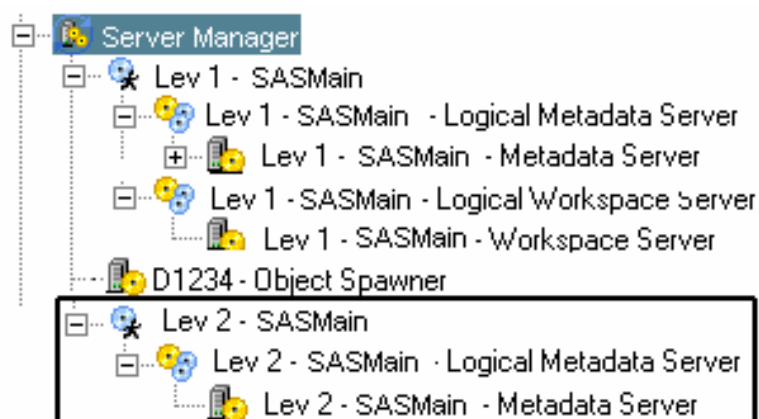
- 1 In the SAS Management Console navigation tree, select **Environment Management ► Server Manager**.
- 2 Select **Actions ► New Server**.

When defining the server in the New Server wizard, specify these properties:

<i>Server type</i>	SAS Application Server
<i>Name</i>	Lev 2 - sas application server (for example, Lev 2 - SASMain)
<i>SAS server type</i>	Metadata Server
<i>Authentication Domain</i>	ReplicationTargetAuth
<i>Host Name</i>	Target server host name
<i>Port</i>	8562

The following display illustrates how the Server Manager navigation tree appears after the target metadata server has been defined.

Display 20.3 Target Metadata Server Defined



For more information about defining a server, see the *SAS Management Console: User's Guide*.

Step 14. Adding a SAS/CONNECT Server Component to the Target Metadata Server Definition

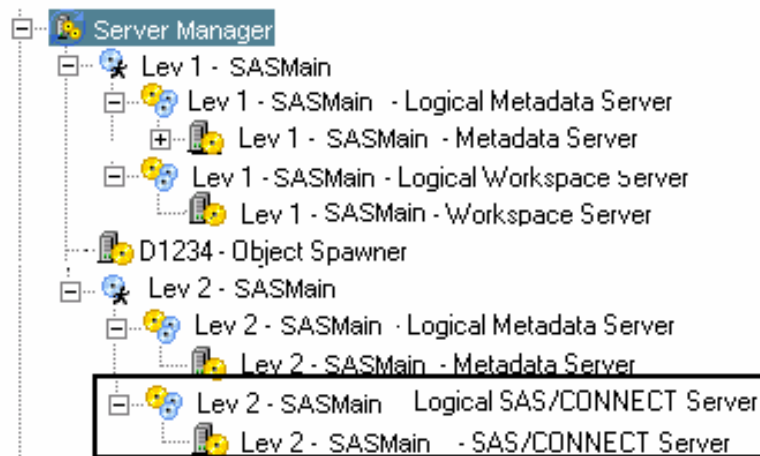
In SAS Management Console, use the New Application Server Component wizard to add a SAS/CONNECT server component to the target metadata server that you defined in “Step 13. Defining the Target Metadata Server in SAS Management Console” on page 404. Complete these steps to launch the wizard:

- 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 - SASMain**.
- 2 Select **Actions ► Add Application Server Component**.

When defining the server component in the New Application Server Component wizard, specify these properties:

<i>SAS server type</i>	SAS/CONNECT Server
<i>Configuration Type</i>	Basic
<i>Authentication Domain</i>	ReplicationTargetAuth
<i>Host Name</i>	Target server host name
<i>Port Number</i>	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 20.4 SAS/CONNECT Server Component Added

For more information about defining an application server component, see the *SAS Management Console: User's Guide*.

Step 15. Defining an IOM Object Spawner for the SAS Workspace Server Component

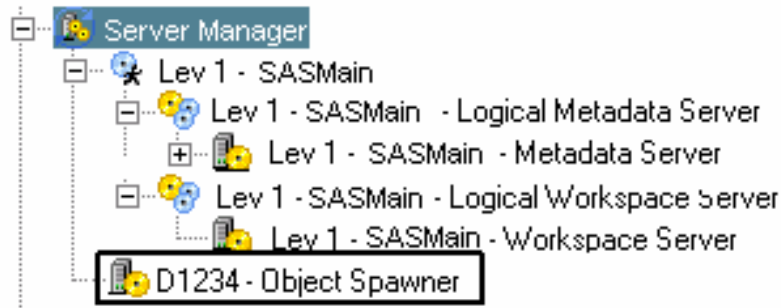
In SAS Management Console, use the New Server wizard to define the IOM Object Spawner that will be used to launch the SAS Workspace Server that you defined in “Step 12. Adding a SAS Workspace Server Component to the Source Metadata Server Definition” on page 403. Complete these steps to launch the wizard:

- 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:

<i>Server type</i>	Object Spawner
<i>Name</i>	<i>hostname</i> - Object Spawner (for example, D1234 - ObjectSpawner)
<i>Associated Machine</i>	Source server machine
<i>Selected Servers</i>	Lev 1 - sas application server - Workspace Server
<i>Authentication Domain</i>	ReplicationSourceAuth
<i>Host Name</i>	Source server host name
<i>Port</i>	8580

The following display illustrates how the Server Manager navigation tree appears after the object spawner has been defined.

Display 20.5 Object Spawner Defined

For more information about defining an Object Spawner, see the *SAS Management Console: User's Guide*.

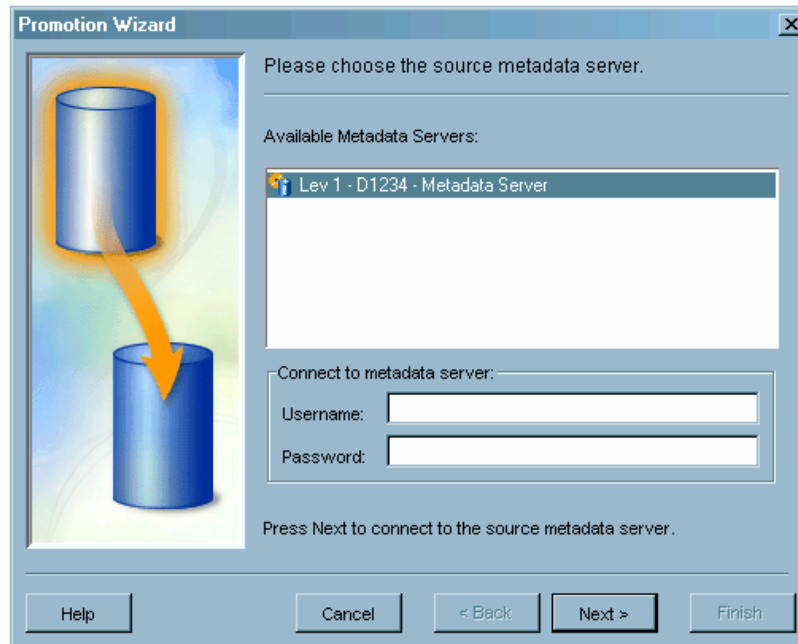
Creating a Promotion Job

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**.

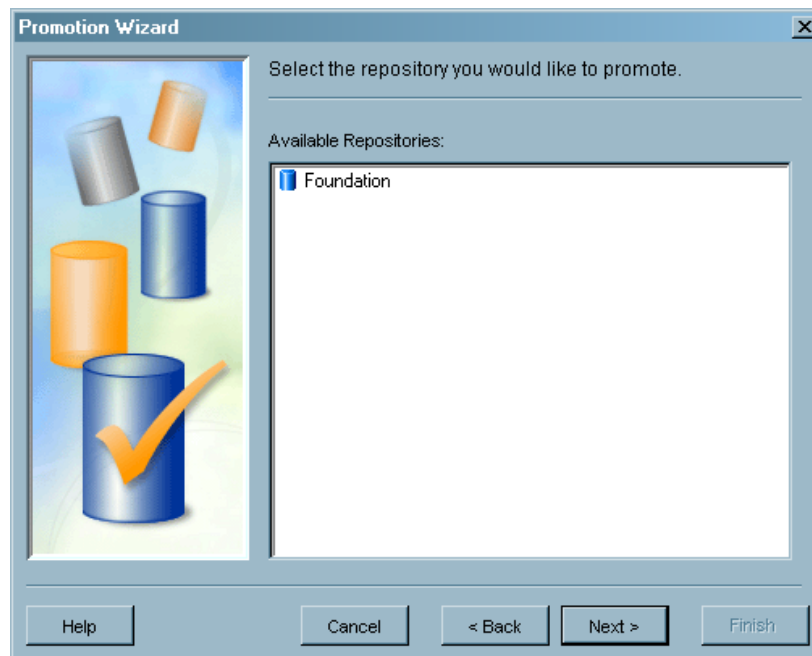
Note: You will create a promotion job to move metadata with substitutions from level 1 to level 2. Technically, however, moving metadata from a production level to a test level is a demotion. △

- 3 Select **Actions ► New Definition**.
- 4 In the Source Metadata Server Definition window, select the source metadata server.

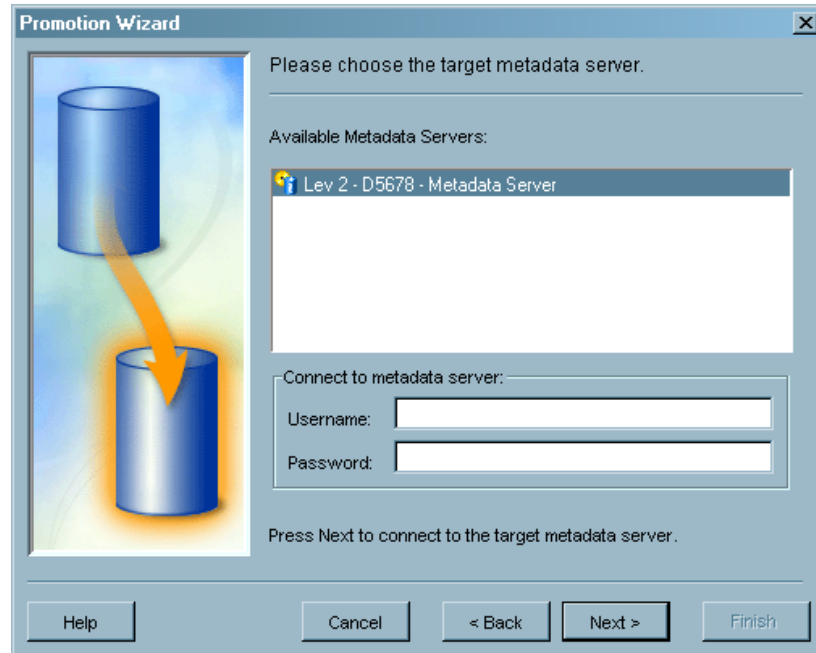
Display 20.6 Promotion Wizard—SourceMetadata Server Definition Window

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 398. Click **Next** to continue.

- 5 Select the **Foundation** repository and click **Next** to continue.

Display 20.7 Promotion Wizard—SelectRepository Window

- 6 In the Connect to Target Metadata Server window, select the target metadata server.

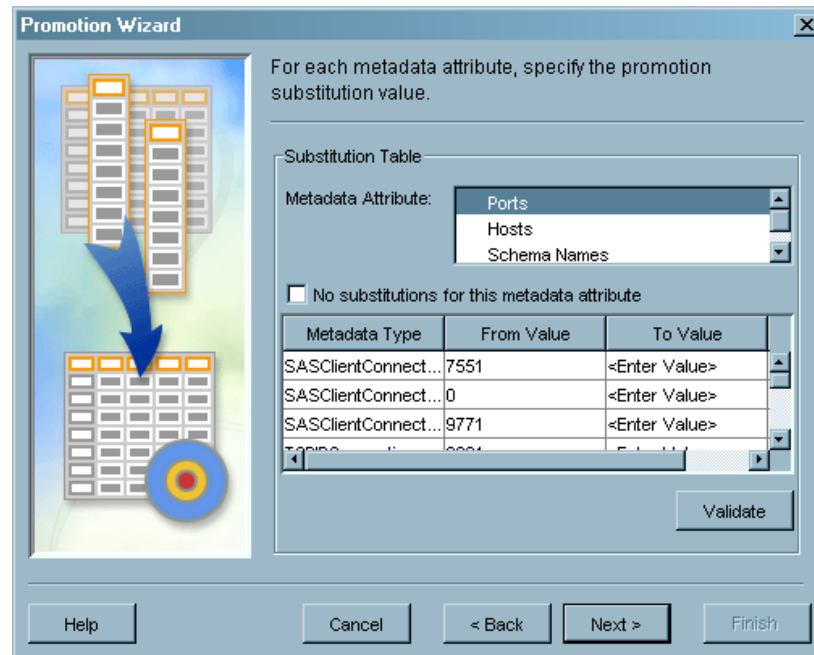
Display 20.8 Promotion Wizard—Connect to Target Metadata Server Window

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 398. 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.

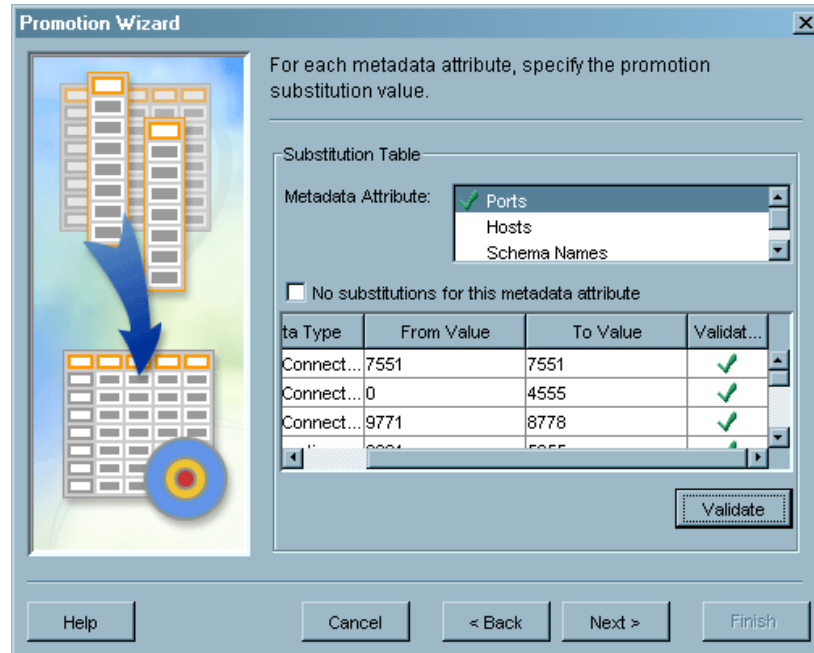
Display 20.9 Promotion Wizard—SubstitutionsWindow

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 20.3 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 20.10 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 20.11 Promotion Wizard—Define Work Directories Window

Promotion Wizard

Please define work and backup locations required for the job execution.

Enter work directory location for source metadata server:

MetadataServer\ReplicationWorkArea\work

Enter work directory location for target metadata server:

MetadataServer\ReplicationWorkArea\work

Enter location to store backup of repository on target server:

MetadataServer\ReplicationWorkArea\tempBackup

Help Cancel < Back Next > Finish

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 20.12 Promotion Wizard—Run or Save Job Definition Window

Promotion Wizard

Please enter a name for this job definition. You can run the promotion now or save the job for later execution.

Enter job definition name:

Promotion Action

Do not run. Only save job definition
This option will save the promotion job definition for later execution.

Run promotion now and save job definition
Running the promotion job will pause both servers while the chosen repository is promoted. The promotion job definition will also be saved for future executions.

Help Cancel < Back Next > Finish

- 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. Δ

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** \blacktriangleright **Job Definitions** \blacktriangleright **Promotion**.
- 2 In the display area to the right, select a job to run.
- 3 Select **Actions** \blacktriangleright **Run Job**.

Note: To save the SAS code for the promotion job to a file, select **Save to File** instead of **Run Job**. Δ

Making Modifications 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. You might also need to make changes such as re-creating cubes and redeploying scheduled jobs.

Modifications Related to the Server Definitions

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, “Servers in the SAS Intelligence Platform,” on page 15. Δ

- 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 398.
- 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 SAS Management Console.

Modifications Related to Cubes

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:

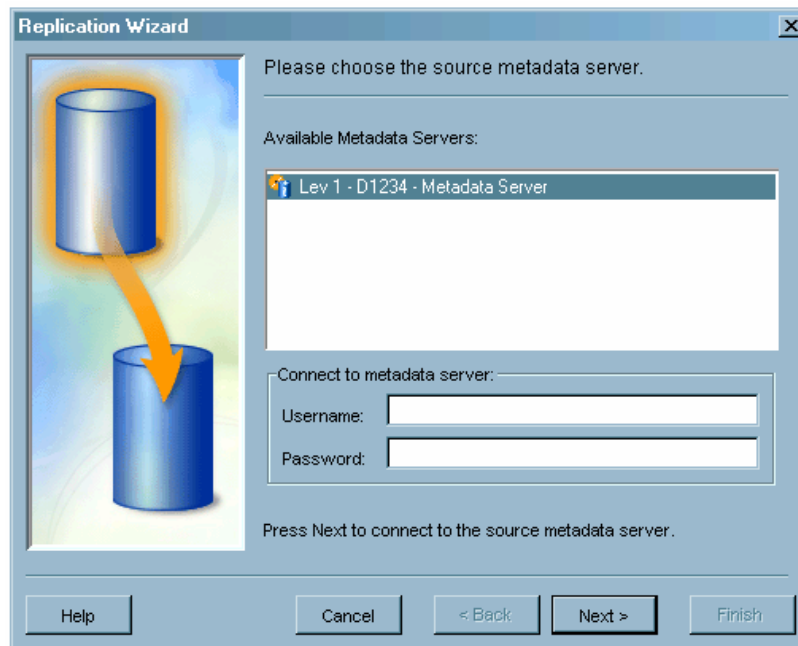
- 1 Use SAS ETL Studio to connect to the target metadata server. You can use the login information for any user who has WriteMetadata access to the metadata repository that contains the cubes that you need to edit, as well as WriteMetadata access to the cube metadata that must be modified.
- 2 In the SAS ETL Studio inventory tree, select the cube name.
- 3 To launch the Cube Designer wizard, right-click on the cube name and select **Edit Cube Structure**.
- 4 On the General window, change the **Work Path** to the correct path for the target machine.
- 5 Click **Next** until the Generated Aggregations window appears. On that window, click **Advanced** to open the Performance Options dialog box. On the Performance Options dialog box, make these changes on the **Global** tab:
 - a Change the value of the **Location of index component files** to the correct path for the target machine.
 - b Change the value of the **Location of partitions in which to place aggregation table data** to the correct path for the target machine.
 - c Click **OK** to close the dialog box and return to the Cube Designer wizard.
- 6 Click **Next** until the Finish window appears.
- 7 Depending upon whether you want to save the metadata and rebuild the cube or just save the metadata, select the applicable radio button.
- 8 Click **Finish**.

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** \blacktriangleright **Job Definitions** \blacktriangleright **Replication**.
- 3 Select **Actions** \blacktriangleright **New Definition**.
- 4 In the Source Metadata Server Definition window, select the source metadata server.

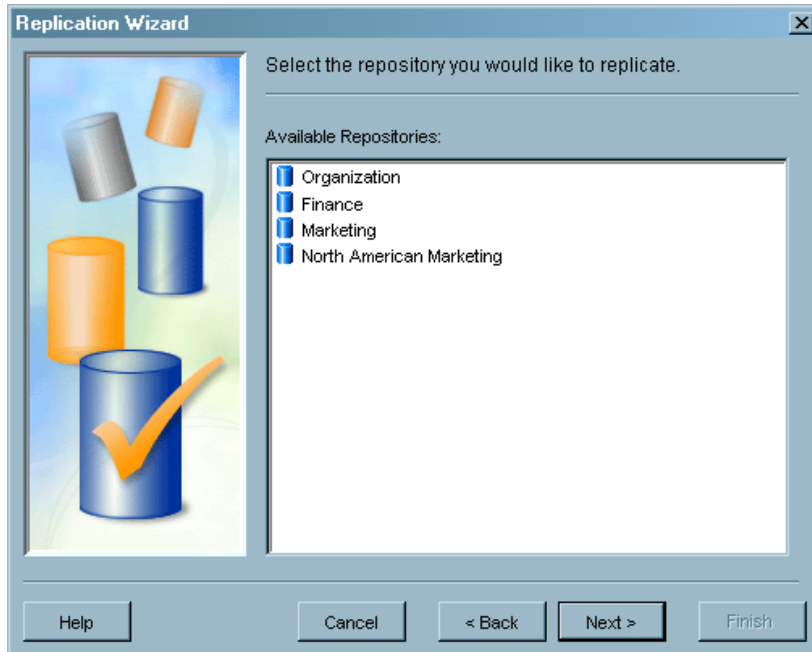
Display 20.13 Replication Wizard—SourceMetadata Server Definition Window



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 398. Click **Next** to continue.

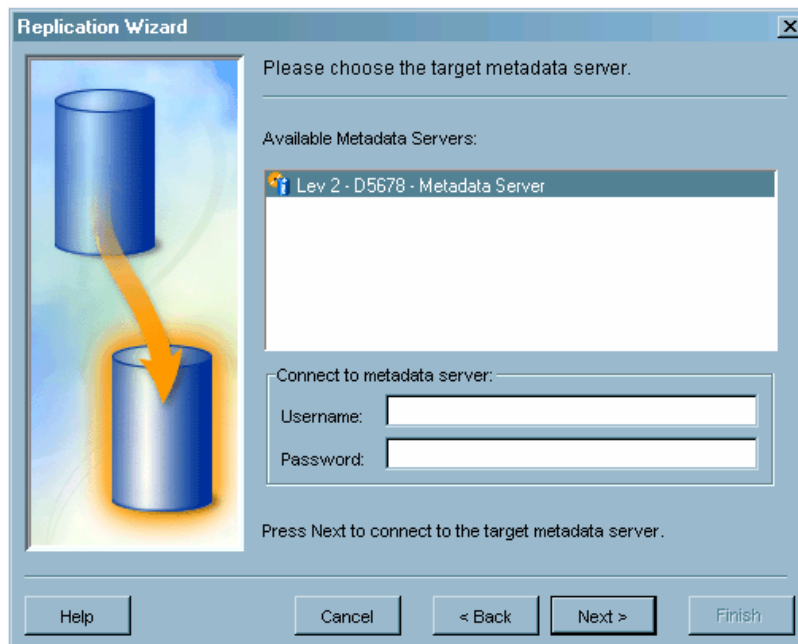
- 5 Use the Select Repository window to select the repository that you want to replicate.

Note: The target metadata server cannot contain an existing repository with the same name as the selected repository. Δ

Display 20.14 Replication Wizard—SelectRepository Window

Click **Next** to continue.

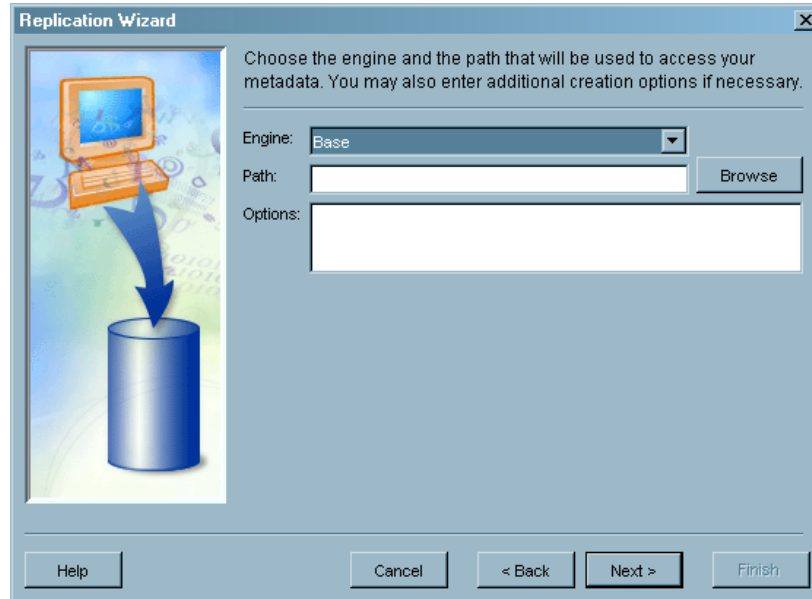
- 6 In the Connect to Target Metadata Server window, select the target metadata server.

Display 20.15 Replication Wizard—Connect to Target Metadata Server Window

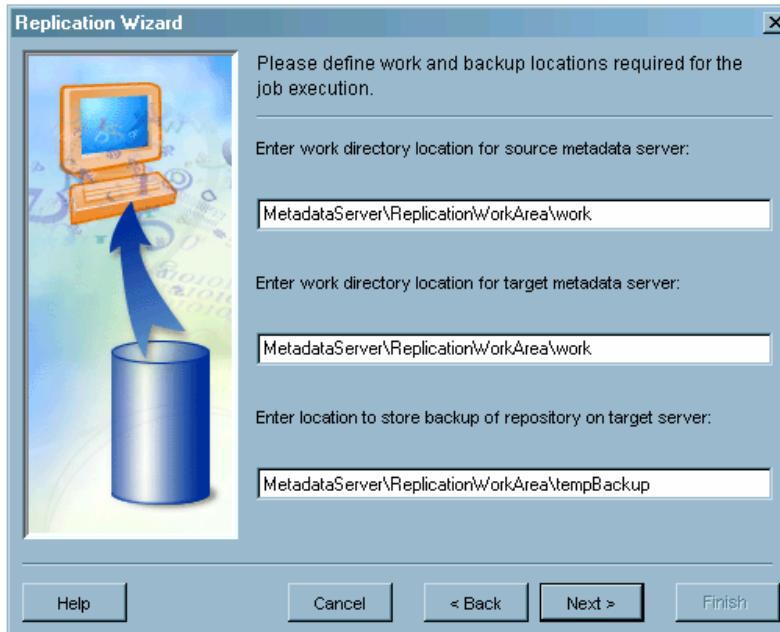
Enter the target metadata server login for the SAS Replication Administrator. This login was defined in “Step 4. Defining the SAS Replication Administrator in SAS Management Console” on page 398. Click **Next** to continue.

- 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 20.16 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 20.17 Replication Wizard—Define Work Directories Window


Replication Wizard

Please define work and backup locations required for the job execution.

Enter work directory location for source metadata server:

MetadataServer\ReplicationWorkArea\work

Enter work directory location for target metadata server:

MetadataServer\ReplicationWorkArea\work

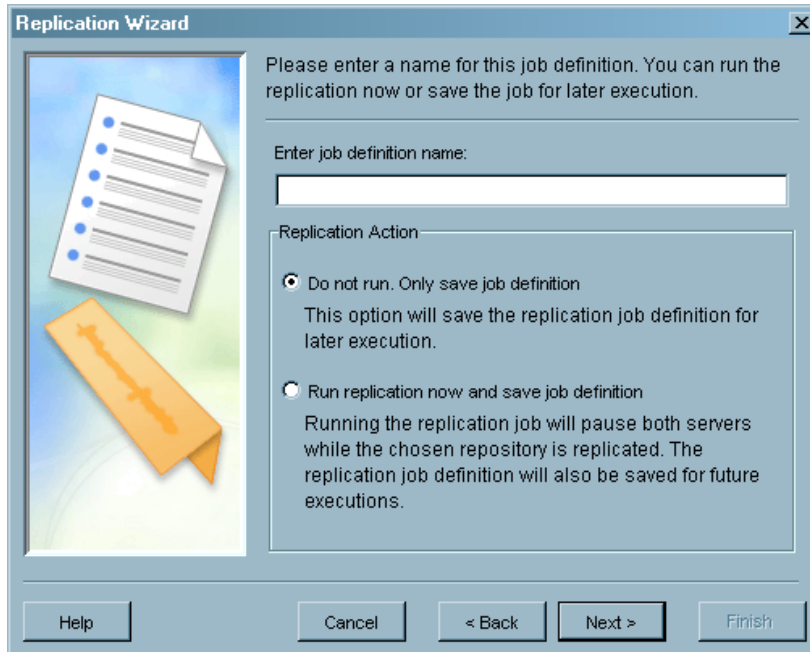
Enter location to store backup of repository on target server:

MetadataServer\ReplicationWorkArea\tempBackup

Help Cancel < Back **Next >** Finish

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 20.18 Replication Wizard—Run or Save Job Definition Window


Replication Wizard

Please enter a name for this job definition. You can run the replication now or save the job for later execution.

Enter job definition name:

Replication Action

Do not run. Only save job definition
This option will save the replication job definition for later execution.

Run replication now and save job definition
Running the replication job will pause both servers while the chosen repository is replicated. The replication job definition will also be saved for future executions.

Help Cancel < Back **Next >** Finish

- 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** \blacktriangleright **Job Definitions** \blacktriangleright **Replication**.
- 2 In the display area to the right, select a job to run.
- 3 Select **Actions** \blacktriangleright **Run Job**.

Note: To save the SAS code for the replication job to a file, select **Save to File** instead of **Run Job**. Δ

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.

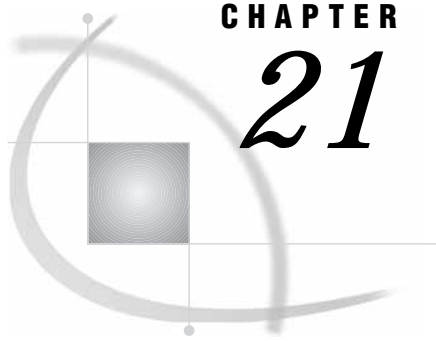
For example, an entry in the error log similar to the following text indicates that the **mdrepjls.sas** file was not found or that there are errors in the options that are specified in the file. See “Step 5. Creating the Metadata Access File on the Source Metadata Server Machine” on page 399.

```
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** \blacktriangleright **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** \blacktriangleright **Save to File**.
- 5 Specify a name for the job in the Save window and click **OK**.



CHAPTER

21

Managing an Environment

<i>Overview of Managing an Environment</i>	421
<i>Customizing the Properties of a New Environment</i>	421
<i>What Is a Configuration Property?</i>	421
<i>Ways to Modify the Configuration Properties File for a New Environment</i>	422
<i>Using a Command-Line Option to Specify a Custom Properties File</i>	422
<i>Editing the Configuration File from Within the SAS Configuration Wizard</i>	423
<i>Manually Changing the Properties of an Existing Environment</i>	423
<i>Special Considerations for Client Applications</i>	424
<i>Special Considerations for Web Applications</i>	424
<i>Adding to an Environment</i>	425
<i>Adding a SAS Application Server to an Existing Environment</i>	425
<i>Including the Same Servers</i>	425
<i>Including Different Servers</i>	425
<i>Adding an Application Server Component to an Existing SAS Application Server</i>	425
<i>Re-Creating an Existing Environment</i>	426
<i>Uninstalling an Environment</i>	427

Overview of Managing an Environment

A SAS environment identifies an entire set of related information such as levels, SAS application servers, scripts, utilities, and documentation. This chapter explains how to perform these tasks:

- customize the properties of a new environment
- manually change the properties of an existing environment
- add to an existing environment
- re-create an existing environment
- uninstall an environment.

Customizing the Properties of a New Environment

What Is a Configuration Property?

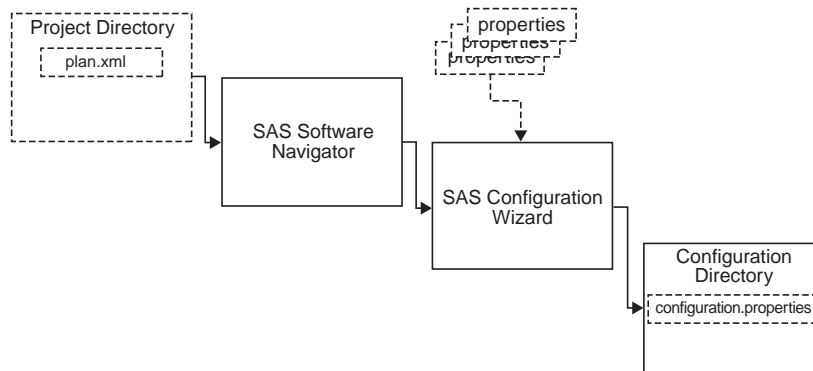
A configuration property is a simple name/value pair. The pairs are specified in a properties file, which contains one property pair (**name=value**) per line. Throughout the configuration process, property values are substituted for property names.

Note: The properties file is one of two data inputs to the SAS Configuration Wizard. The other input is the planning file or project directory. For more information about

project installations, see Chapter 7, “Installing and Configuring Your Software,” on page 79. Δ

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 21.1 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 **OVERWRITE_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.

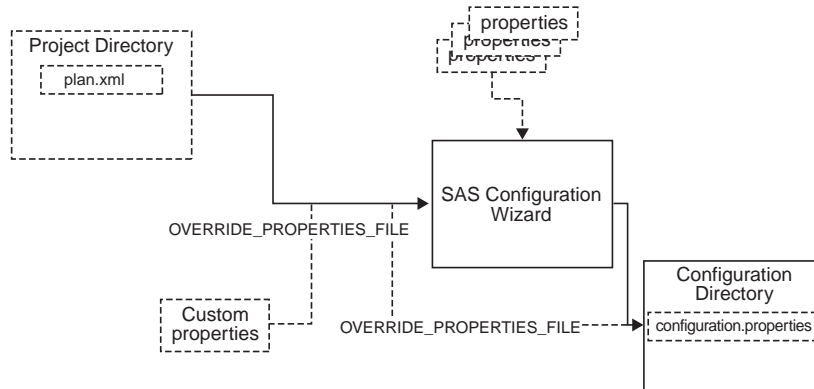
To use this option, you launch the SAS Configuration Wizard by using a command such as

```
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. Δ

The following figure shows at which point the custom properties are introduced into the configuration process.

Figure 21.2 Using the `OVERWRITE_PROPERTIES_FILE` Command to Specify a Custom Property File

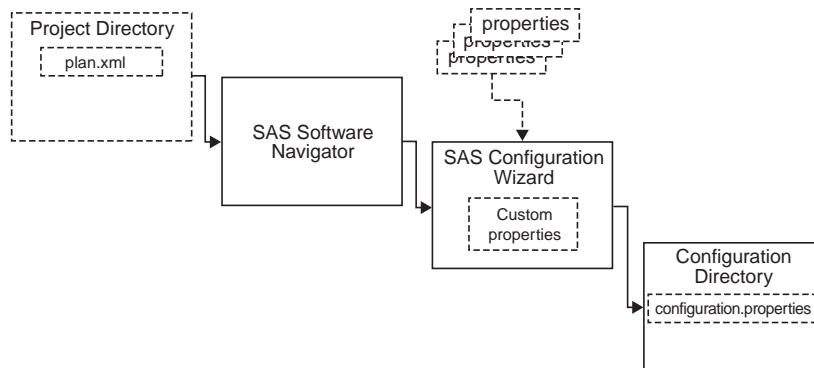


Editing the Configuration File from Within the SAS Configuration Wizard

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 21.3 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. \triangle

Note: For information about how to manually reset passwords, see “Updating Passwords” on page 225. \triangle

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 20, “Promoting and Replicating Metadata,” on page 393. △

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 **OVERWRITE_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.

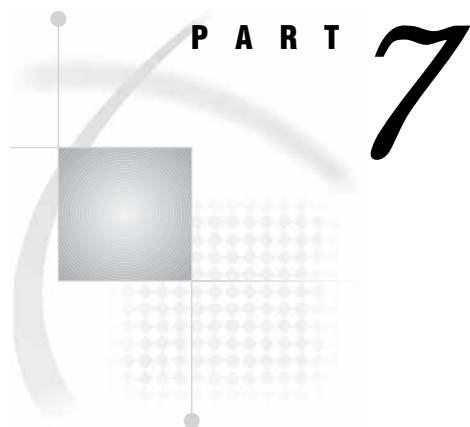
- 3 Review the remaining contents of the **C:\SAS\configuration directory name** directory and decide which files that you want to keep.
- 4 Copy the files that you want to keep to another location.
- 5 Delete the entire directory structure. For example, you can use the following commands:

- Windows:

```
rmdir /S/Q configuration directory location
```

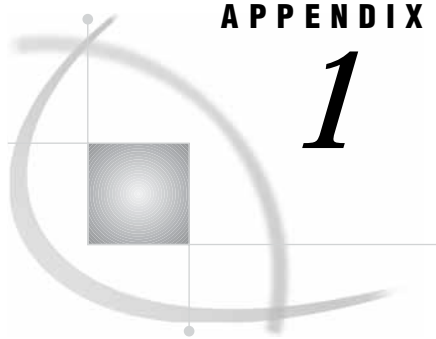
- UNIX:

```
rm -r -f configuration directory location
```

Appendixes

- Appendix 1* **Understanding the SAS Configuration Environment** 431
- Appendix 2* **Software Index Installations** 443
- Appendix 3* **Upgrading a SAS 9.1 or 9.1.2 System to a SAS 9.1.3 System** 453
- Appendix 4* **Recommended Reading** 473



APPENDIX

1

Understanding the SAS Configuration Environment

<i>Overview of Understanding the SAS Configuration Environment</i>	431
<i>The Basic Concepts</i>	431
<i>The Main Directory Structure</i>	433
<i>The Lev1 Directory Structure</i>	434
<i>SASMain Contents</i>	435
<i>About the SAS Configuration File</i>	435
<i>About the SAS Metadata Server Configuration File</i>	436
<i>The SASMain Subdirectories</i>	437
<i>Web Contents</i>	440
<i>Default Directory Permissions</i>	440

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:

<i>Environment</i>	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.
<i>Level</i>	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 Lev1 for Production, Lev2 for Test, and Lev3 for Development.

Note: User-written scripts can take advantage of the consistent format of the level directory names. For example, a script that

searches for a particular piece of SAS code could search the lowest level (or the level where the request originates) and then continue to increment through the other levels, moving towards production. \triangle

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 287.

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

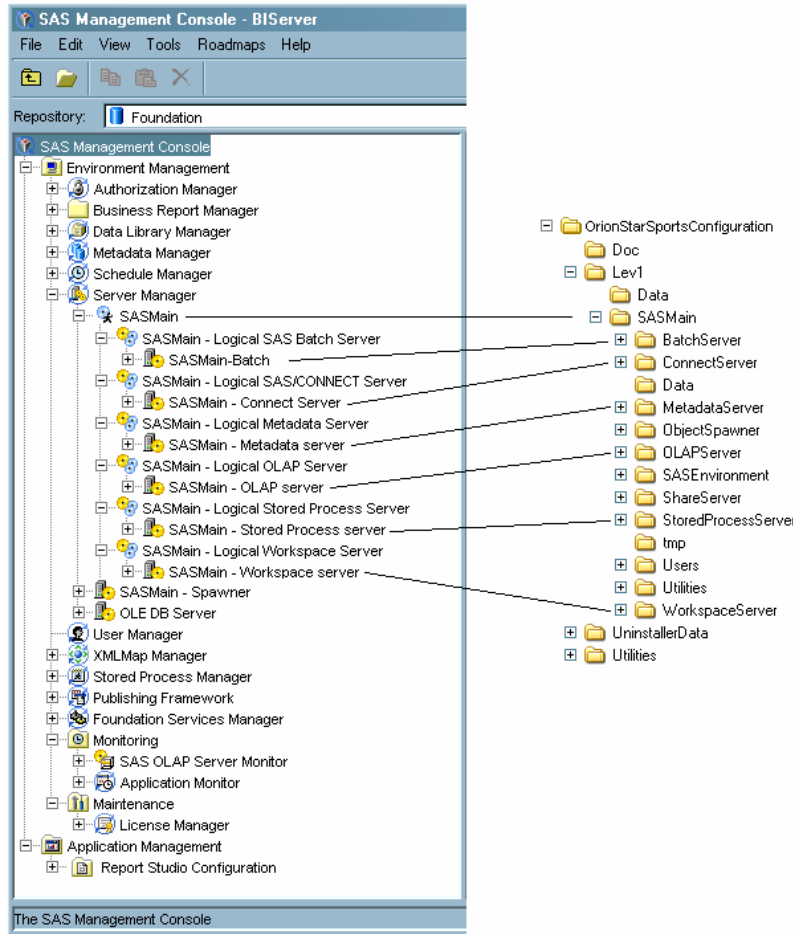
within SAS application server metadata, a logical server groups similar servers that run under the SAS application server. A logical server is referenced when a specific launch mode is requested, such as interactive or batch. Logical server definitions contain one or more server definitions.

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



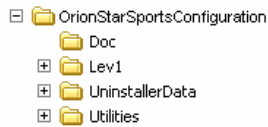
For an explanation of the **SASMain** directory structure, see “SASMain Contents” on page 435.

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 that is shown in the illustration are

- Doc* provides a location for you to save your own documents, which is related to the current environment.
- Lev1* as explained in “The Lev1 Directory Structure” on page 434, 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.
- 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.

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. △

The Lev1 Directory Structure

For all configurations, the **Lev1** directory contains **Data** and **SASMain** subdirectories. If you are configuring a middle tier machine, then the **Lev1** directory also includes a **web** subdirectory.

Note: Depending on your operating system and your installed products, your site might also have subdirectories that are not shown in the following display. △

Display 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 435.

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 440.

SASMain Contents

On machines with an installed SAS Metadata Server (and perhaps other SAS servers), this directory represents a SAS application server that contains configuration files, a variety of subdirectories, and utilities that support the SAS application server. On Web-tier only machines, this directory is empty.

About the SAS Configuration File

Regardless of the mechanism used to invoke SAS, all SAS sessions must begin with the same SAS configuration file. This practice has the following advantages:

- It enables the same SAS application to be executed via any of the SAS server technologies that support code submission, including SAS Workspace Servers, SAS/CONNECT servers, and batch processes.

- It ensures that the resources that are required to execute a SAS application are properly configured.

Depending on your operating system and your installed products, the `sasv9.cfg` file might contain the following information:

- the path to the directory that contains `sasv9.cfg` file
- the path to the root directory for the current SAS application server
- the path to the directory that will contain the SAS formats used by the current SAS application server
- the path to the directory that will contain the SAS macros used by the current SAS application server
- a list of the locales used during data cleansing and the location of the SAS Data Quality Server setup file
- SAS Metadata Server information such as port number, foundation repository name, server name, and connection protocol.

For example, the `sasv9.cfg` file for the `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\Levl\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>

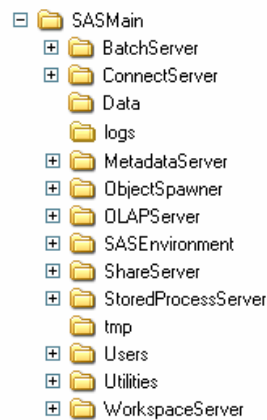
Note: For information about the `omacnfig.xml` file, see the *SAS Metadata Server: Setup Guide* available at support.sas.com/rnd/eai/openmeta/v9/setup. △

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 that are shown in the display:

<i>BatchServer</i>	contains log files and SAS code that are associated with the current SAS application server running in batch mode.
<i>ConnectServer</i>	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: <ul style="list-style-type: none"> <input type="checkbox"/> one or more start-up scripts that are appropriate for the operating system. <input type="checkbox"/> the logs subdirectory that stores any log files generated by this server invocation. <input type="checkbox"/> any utility files that are used to manage the server.
<i>Data</i>	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.
<i>logs</i>	contains any log files that are specific to the current SAS application server.
<i>MetadataServer</i>	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 131).
- 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 435):

- the `Q1tyKb` 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. \triangle

Default Directory Permissions

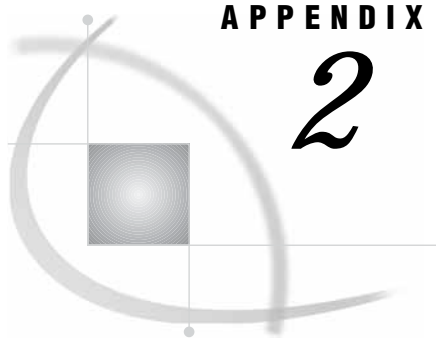
For UNIX and z/OS systems, here are the default permissions for the directories, files, and scripts that are created in a project installation.

Note: There are no default permissions set on Windows systems. All directories, files, and scripts are owned by the user who performs the installation. \triangle

Note: For information about the users and groups that are created during a project SAS deployment, see "Setting Up Required User Accounts" on page 68. \triangle

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
Levl/SASMain/ StoredProcessServer	Read, write, execute	Read, write, execute	No access
Levl/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

Software Index Installations

<i>Overview of Software Index Installations</i>	443
<i>Software Index Installations</i>	444
<i>Licensed Software Installations</i>	446
<i>CD Index Installations</i>	447
<i>Using the SAS Configuration Wizard in Software Index Installations</i>	448

Overview of Software Index Installations

In this document, we have recommended the use of *planned* installations, both Personal and Advanced, 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 easiest to perform. However, it is also possible to perform what is called a Software Index installation, an installation for which you have no planning file. Obviously, during a Software Index installation, you need to know which software to install and configure on each host.

There are two types of Software Index installations: Licensed Software installations and CD Index installations. In a Licensed Software installation, you install the required components of a technology package on a machine. In that case, the SAS Software Navigator gives you guidance as to the order in which you should install products. In a CD Index installation, you select individual products to install those products from a set of CDs.

When would you perform a Software Index installation? Well, there are two main cases. First, an expert installer might use a Licensed Software installation, followed by a CD Index installation of a couple of optional products, to install the software on a machine. If you have the knowledge to do this, you can avoid creating a planning file. Second, an installer might want to use a CD Index installation 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 Software Index installation, you can use the SAS Configuration Wizard to configure the software that 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 software or middle-tier servers, you must perform the pre-installation tasks that are described in Chapter 6, “Pre-Installation Tasks,” on page 47—with the exception of setting up a project directory.

For more information about the topics that were introduced in this overview, see the following sections:

- “Software Index Installations” on page 444
- “Licensed Software Installations” on page 446

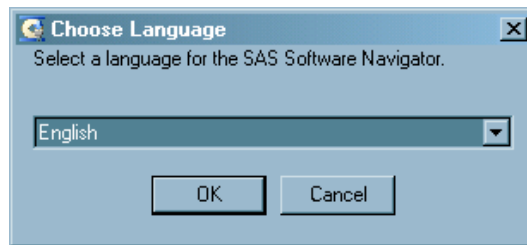
- “CD Index Installations” on page 447
- “Using the SAS Configuration Wizard in Software Index Installations” on page 448

Software Index Installations

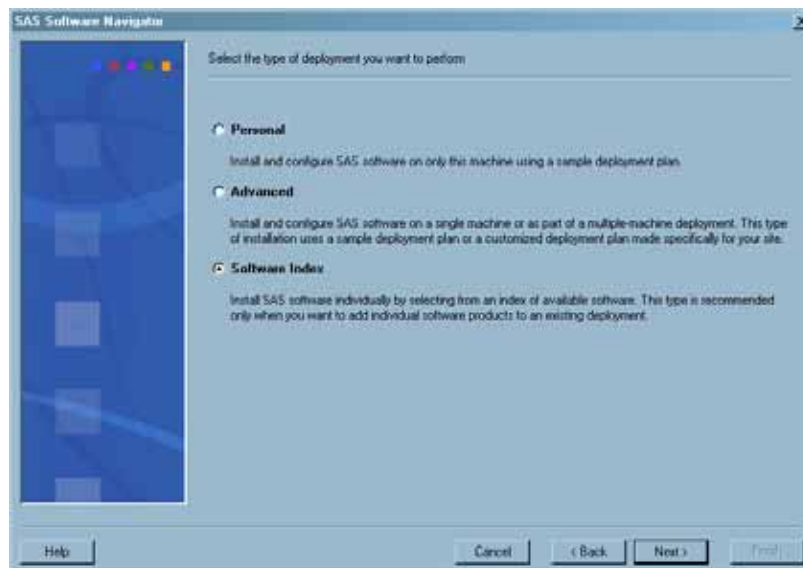
Log on to the computer on which you will be installing software. On Windows systems, you can log on as any user who belongs to the Administrators group. On UNIX systems, log on as the SAS user (recommended name `sas`), which you defined as one of your pre-installation tasks.

Note: We recommend that you do not log on as `root` to install software on a UNIX system. \triangle

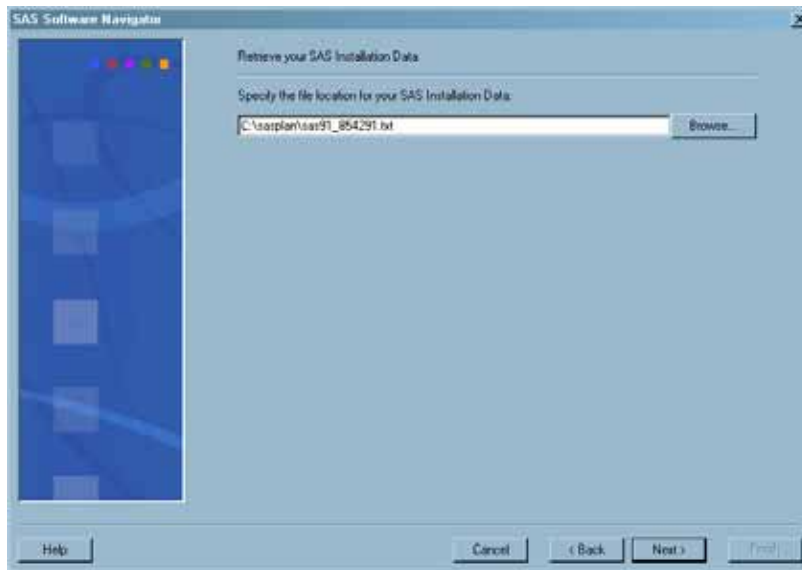
- 1 Start the SAS Software Navigator from your SAS Software Depot or by using the CD that contains the navigator. (For information on starting the SAS Software Navigator, see “Starting the SAS Software Navigator” on page 81.) A Choose Language dialog box appears.



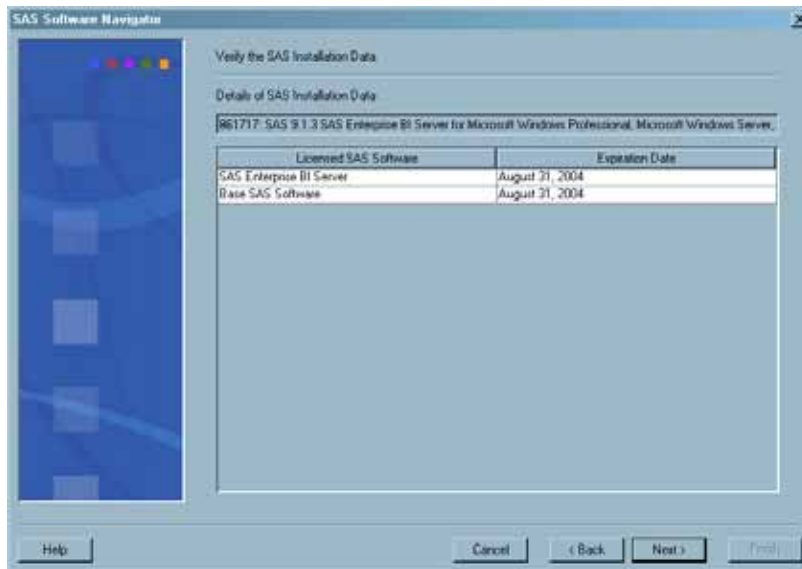
- 2 From the language drop-down list, select the language that you want the SAS Software Navigator to use. Click **OK**. The SAS Software Navigator Wizard starts.



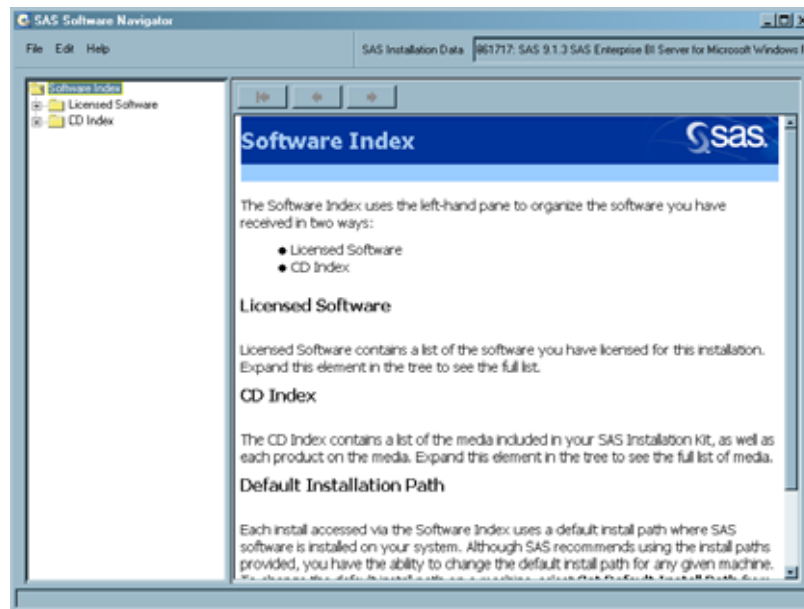
- 3 Select the **Software Index** radio button, and click **Next**. The next screen in the wizard prompts you for the location of your SAS Installation Data file.



- 4 Specify the path to the SAS Installation Data file. You should have received this file with your Software Order E-mail and saved it on your system. 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**. You will see a screen that lists the SAS software that you have licensed.



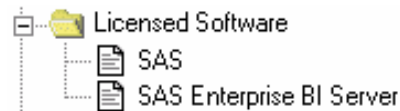
- 5 Verify that the list of licensed software is correct. Then, click **Next**. The SAS Software Navigator window appears.



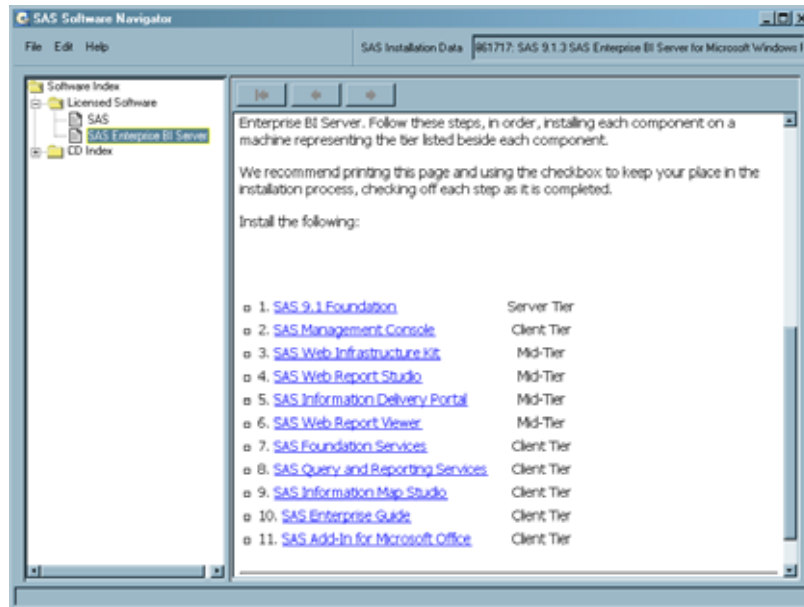
The Software Index folder in the left pane contains two subfolders: a Licensed Software folder and a CD Index folder.

Licensed Software Installations

- 1 To do a Licensed Software installation, expand the Licensed Software folder to display a list of the technology packages that you have licensed for the current machine.



- 2 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 and an indication of the tier on which each product should be installed.



Note: To install additional products on this machine, you can perform a CD Index installation, as explained in “CD Index Installations” on page 447. △

- 3 Click the link for the product that you want to install, and an HTML page will appear in the right pane. 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.
- 4 Read the installation instructions (if necessary), and run the installation program.

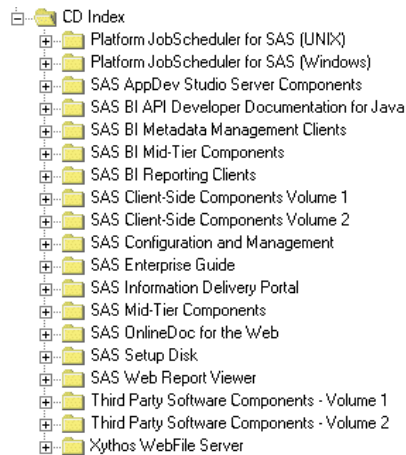
Repeat the last two steps for each product in the list.

For information about using the SAS Configuration Wizard to configure products that you have installed using a Licensed Software installation, see “Using the SAS Configuration Wizard in Software Index Installations” on page 448.

CD Index Installations

After performing the steps listed in “Software Index Installations” on page 444, 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 **Insta11** 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 installation, see the following section.

Using the SAS Configuration Wizard in Software Index Installations

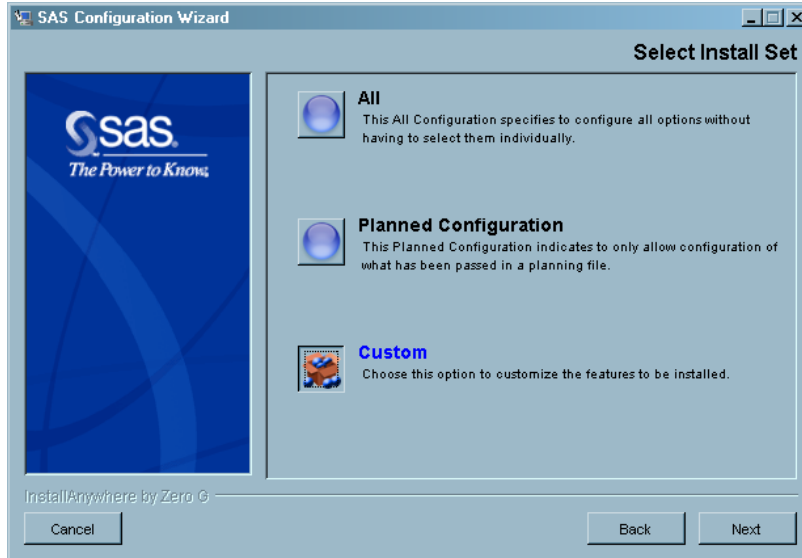
Although the SAS Configuration Wizard is designed primarily to be used in Personal and Advanced installations, it can also be used to configure software after a Software Index installation. If you use the wizard after a Software Index installation, it will create a configuration directory on all server hosts and provide you with instructions for configuring your software just as it would in a planned installation. (For information about configuration directories, see Appendix 1, “Understanding the SAS Configuration Environment,” on page 431.)

To use the SAS Configuration Wizard following a Software Index installation, perform these steps:

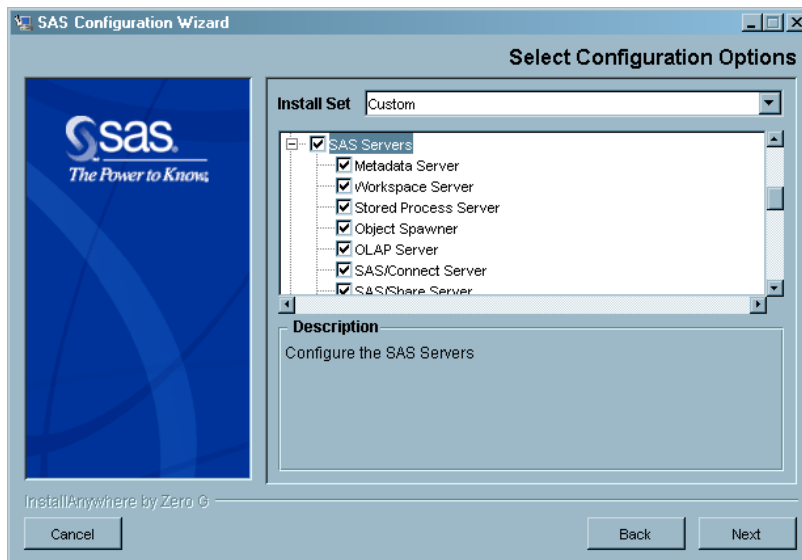
- 1 To start the SAS Configuration Wizard look in the CD Index for the CD titled **SAS Configuration and Management**, open the folder for that CD, selecting **SAS Configuration Wizard**, and then click the **Configure** link in the right pane of the navigator. The SAS Configuration Wizard starts and 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, 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.



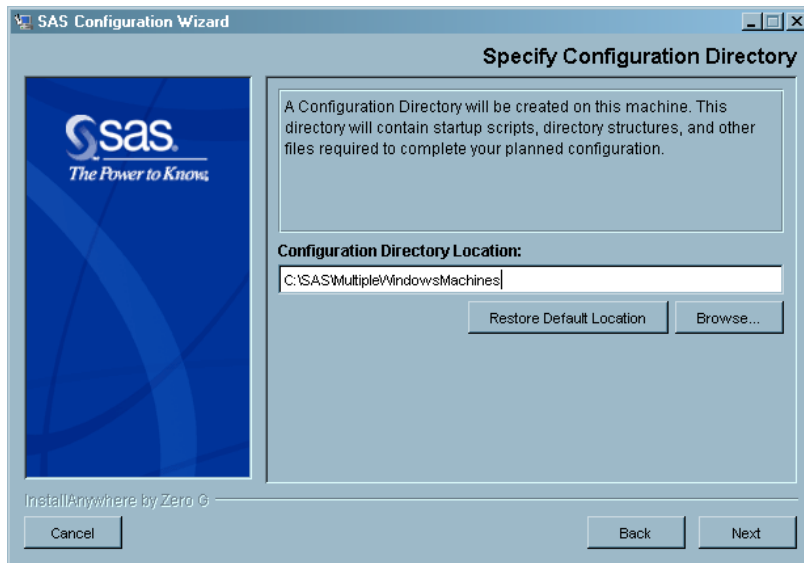
This window displays a list of the components that the SAS Configuration Wizard can configure:

- SAS Clients

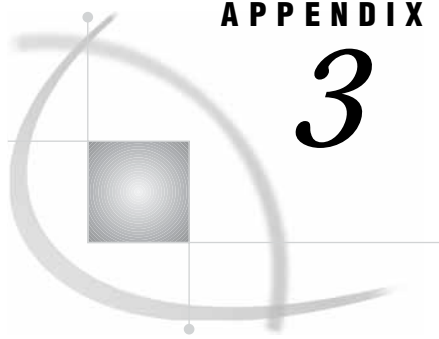
- SAS Add-In for Microsoft Office
 - SAS Enterprise Miner
 - SAS Middle Tier
 - SAS Web Infrastructure Kit
 - SAS Web Report Studio
 - SAS Web Report Viewer
 - SAS BI Web Services for Java
 - SAS BI Web Services for .Net
 - SAS Online Doc for the Web
 - SAS Remote Help for z/OS
 - Apache Tomcat
 - BEA WebLogic Server
 - IBM WebSphere Application Server
 - SAS Servers
 - Metadata Server
 - Workspace Server
 - Stored Process Server
 - Object Spawner
 - OLAP Server
 - SAS/CONNECT Server
 - SAS/SHARE Server
 - Job Scheduler
 - SAS Services
 - SAS Foundation Services
 - HTTP/DAV Servers
 - Apache HTTP Server
 - Xythos WebFile Server
 - Planned
- 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. Δ

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 planned installation. Go to step 3 in “Running the Configuration Wizard on Windows and UNIX Systems” on page 100 for further step-by-step instructions.



APPENDIX

3

Upgrading a SAS 9.1 or 9.1.2 System to a SAS 9.1.3 System

<i>Overview of Upgrading a SAS 9.1 or 9.1.2 System to a SAS 9.1.3 System</i>	454
<i>Assumptions</i>	454
<i>Before You Begin the Upgrade</i>	454
<i>Your Two Options for Upgrading</i>	454
<i>Upgrading in Place</i>	454
<i>Upgrading After Testing in a Test Environment</i>	456
<i>Building a Test System</i>	456
<i>Updating Your Production System</i>	457
<i>Upgrading Each Machine</i>	458
<i>Metadata Server on Machine</i>	458
<i>Stop the Metadata Server</i>	458
<i>Copy Files from the Original Configuration Directory</i>	458
<i>Start the Metadata Server</i>	458
<i>Upgrade Your Metadata</i>	458
<i>Re-establishing Existing Schedule Flows</i>	459
<i>Server Machine, No Metadata Server</i>	459
<i>Copy Files from the Original Configuration Directory</i>	459
<i>Start Your Servers</i>	460
<i>Middle-Tier Machine, No Metadata Server</i>	460
<i>Copy Files from the Original Configuration Directory</i>	460
<i>Upgrade User and Group Definitions (9.1 to 9.1.3 only)</i>	460
<i>Define the SAS Foundation Services in the Metadata (9.1 to 9.1.3 only)</i>	461
<i>Update the SAS Web Infrastructure Kit Metadata</i>	461
<i>Copy Custom Portlets from the Original Installation Directory</i>	461
<i>Migrate Your WebDAV Data (SAS Information Delivery Portal)</i>	462
<i>Migrate Your WebDAV Data (SAS Web Report Studio)</i>	464
<i>Start the SAS Services Application</i>	465
<i>Deploy Your Web Applications</i>	465
<i>Middle-Tier Machine, Metadata Server</i>	465
<i>Stop the Metadata Server</i>	466
<i>Copy Files from the Original Configuration Directory</i>	466
<i>Start the Metadata Server</i>	466
<i>Upgrade Your Metadata</i>	466
<i>Start the Other Servers</i>	467
<i>The Remaining Steps</i>	467
<i>Upgrading a SAS 9.1 or 9.1.2 z/OS System to a SAS 9.1.3 System</i>	467
<i>Installing and Configuring the SAS 9.1.3 Software</i>	467
<i>Replacing Your Production System</i>	470

Overview of Upgrading a SAS 9.1 or 9.1.2 System to a SAS 9.1.3 System

This appendix explains how to upgrade from a 9.1 or 9.1.2 business intelligence system to a 9.1.3 system. For all platforms except z/OS, see “Your Two Options for Upgrading” on page 454 for pointers to the upgrade instructions. For instructions on upgrading a z/OS system, see “Upgrading a SAS 9.1 or 9.1.2 z/OS System to a SAS 9.1.3 System” on page 467.

The following subsections discuss

- the assumptions we are making about your current installation and the software you plan to install
- the tasks that you must perform prior to following the steps in this appendix
- your options for upgrading your non-z/OS machines.

Assumptions

We are making the following assumptions about your system:

- Your initial installation was a Project installation. (A Project installation is a planned installation, similar to an Advanced or a Personal installation of the 9.1.3 software.)
- You are not installing any new products. You are simply upgrading your existing environment.

Before You Begin the Upgrade

Before you actually begin the upgrade, you should perform the following tasks:

- 1 Plan a time to do the upgrade.
- 2 If you will be performing an Advanced installation of the 9.1.3 software, set up a new project directory. You will need a 9.1.3 planning file and a 9.1.3 SID file(s).
- 3 Back up your current configuration.

Your Two Options for Upgrading

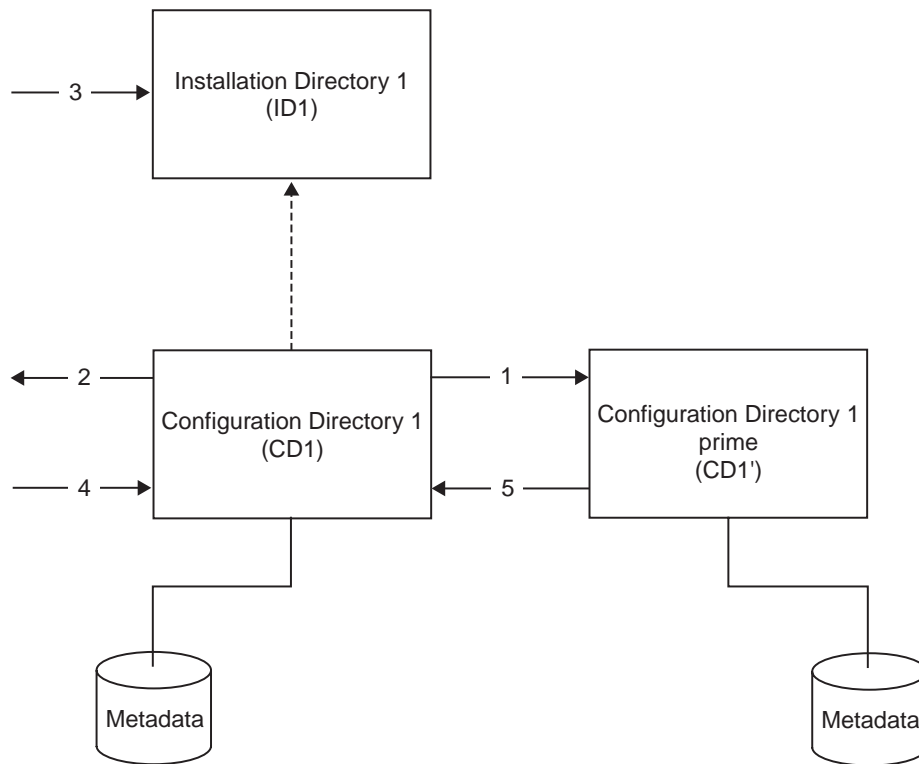
You have two options for upgrading your system. You can simply replace your 9.1 or 9.1.2 software with your 9.1.3 software, or you can temporarily maintain separate 9.1[.2] and 9.1.3 systems so that you can test your 9.1.3 system before taking your 9.1 or 9.1.2 system out of production. For information about these options, see the following sections:

- “Upgrading in Place” on page 454.
- “Upgrading After Testing in a Test Environment” on page 456.

Upgrading in Place

The upgrade-in-place procedure is designed to allow you to minimize the amount of time involved in upgrading. The procedure is illustrated in the following figure.

Figure A3.1 Upgrading in Place



For each machine in your system, you should stop any servers that are running on that machine and then perform the following tasks. (The following numbered steps correspond to the numbers in the previous figure.) Begin work on your metadata-server machine; then move on to other server-tier machines, middle-tier machines, and client machines, in that order.

- 1 Copy your 9.1[.2] configuration directory (CD1) to a backup location (CD1').
- 2 Uninstall the 9.1[.2] configuration by running the appropriate uninstall program. On Windows systems, run the executable `configuration-directory\UninstallerData\Uninstall SAS Configuration Wizard.exe`. On UNIX systems, run `configuration-directory/UninstallerData/Uninstall_SAS_Configuration_Wizard`.
- 3 Install the 9.1.3 software—including the 9.1.3 clients—in ID1. There is no need to uninstall the 9.1[.2] software first.

For third-party software, such as the BEA WebLogic Server, make sure that you have the supported version of the product installed. You can find information about the supported version of each such product at support.sas.com/thirdpartysupport.

- 4 Run the SAS Configuration Wizard using the original configuration directory (CD1).

CAUTION:

Do not follow the HTML instructions that are generated by the SAS Configuration Wizard. These generated instructions are meant for a new configuration. Follow the instructions referred to in step 5 instead. △

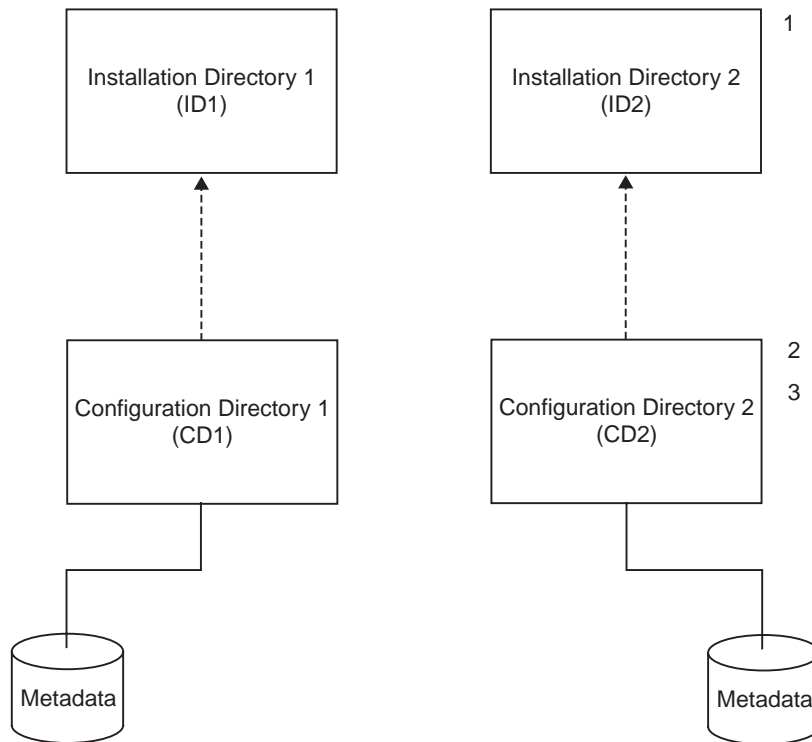
- 5 Execute the manual upgrade instructions presented in the section “Upgrading Each Machine” on page 458. Choose the correct set of instructions for the machine that you are currently working on.

Upgrading After Testing in a Test Environment

If you want to maintain both 9.1[.2] and 9.1.3 business intelligence systems for a testing period, before you make the 9.1.3 system your production system, see the following sections.

Building a Test System

Build a test system by following the steps that are shown in the following figure.



For each machine in your configuration, perform the following steps. (Start with your metadata server machine. Then proceed to your other server-tier machines, your middle-tier machine, and your client machines, in that order.)

- 1 Install the 9.1.3 software on a different machine of the same type (or possibly on the same machine) in ID2.

For each Windows machine in your installation, you must install the SAS 9.1.3 software on a different Windows machine. For each UNIX machine, we recommend that you install the SAS 9.1.3 software on a different UNIX machine. However, it is possible to install the SAS 9.1.3 software on the same machine as SAS 9.1[.2] software.

CAUTION:

If you choose to install SAS 9.1[.2] and SAS 9.1.3 software on the same UNIX system, be aware that when you installed your SAS 9.1[.2] software, a preference file was created that will prevent you from installing the SAS 9.1.3 software in a new location. Therefore, before you can install SAS 9.1.3 software on a host where SAS 9.1[.2] software is installed, you must rename this preferences file. The file `vpd.properties` is located in either the `sas` user's home directory (Solaris and HP-UX IPF) or `/usr/lib/objrepos` (AIX). Rename this file `vpd.properties91` or `vpd.properties912`.

Note that renaming this file will prevent you from uninstalling your SAS 9.1[.2] software later. To uninstall that software, perform the following steps.

- a Rename the SAS 9.1.3 `vpd.properties` file to `vpd.properties913`.
- b Change the name of the SAS 9.1[.2] properties file back to `vpd.properties`.
- c Uninstall your SAS 9.1[.2] software.
- d Delete the SAS 9.1[.2] properties file.
- e Change the name of the SAS 9.1.3 properties file back to `vpd.properties`.

△

For third-party software, such as the BEA WebLogic Server, make sure that you install the supported version of the product. You can find information about the supported version of each such product at support.sas.com/thirdpartysupport.

- 2 Run the SAS Configuration Wizard to produce CD2. If you are installing a 9.1.3 side by side with a 9.1[.2] system, be sure to use different port numbers than the port numbers that you used to configure your 9.1[.2] system.

CAUTION:

Do not follow the HTML instructions that were generated by the SAS Configuration Wizard. These generated instructions are meant for a new configuration. Follow the instructions in step 3 instead. △

- 3 Execute the manual upgrade instructions that are documented in the section “Upgrading Each Machine” on page 458. Choose the correct set of instructions for the machine that you are currently working on.

Then test the entire system.

Updating Your Production System

After you are satisfied with your testing, you will need to update your production system. To do this, you follow the same steps that you follow to upgrade a system in place—see “Upgrading in Place” on page 454—with the following caveat. Depending on where you installed your new software in building your test system, you might not need to reinstall the 9.1.3 software. Instead, you might be able to use the existing installation from your test system. Your hardware deployment and standard operating procedures will dictate whether this is feasible. For example, you might be able to use UNIX NFS mount points to accomplish this.

CAUTION:

Do not copy your test system's configuration directory and metadata repositories. In the process of creating that system, you made several edits to that directory that might or might not be appropriate for your production system. △

Upgrading Each Machine

The manual steps that are needed to update each machine are dependent on the software that is configured on that machine. See the appropriate section:

- “Metadata Server on Machine” on page 458.
- “Server Machine, No Metadata Server” on page 459.
- “Middle-Tier Machine, No Metadata Server” on page 460.
- “Middle-Tier Machine, Metadata Server” on page 465.

Metadata Server on Machine

On the machine where the SAS Metadata Server is configured, perform the steps in the following sections:

- 1 “Stop the Metadata Server” on page 458.
- 2 “Copy Files from the Original Configuration Directory” on page 458.
- 3 “Start the Metadata Server” on page 458.
- 4 “Upgrade Your Metadata” on page 458.
- 5 “Re-establishing Existing Schedule Flows” on page 459.

Note: If your middle-tier servers and metadata server are installed on the same machine, see “Middle-Tier Machine, Metadata Server” on page 465 instead. \triangle

Stop the Metadata Server

The SAS Configuration Wizard automatically starts the metadata server. To make sure that there are no file contention issues, stop the metadata server before you copy any files into the new configuration directory.

Copy Files from the Original Configuration Directory

In the typical use of the configuration, you will have created and modified files in your original configuration directory. For example, you might have added SAS system options to the configuration file *path-to-config-dir\Lev1\SASMain\sasv9.cfg*. You need to determine which of these new or modified files you should copy to the new configuration. Use the following rules to make this decision:

- If you changed a host name, pathnames, or port numbers that are associated with the configuration, you need to ensure that the new configuration directory contains the correct information.
- If you added tuning options or other options to your original configuration, you should add those options to (or maintain those options in) the new configuration.
- You must always copy the SAS Metadata Server files to the new configuration. These files are contained in the `MetadataServer/reposmgr` and `MetadataServer/MetadataRepositories` directories. You should place these files in the same location in the new configuration directory.

Start the Metadata Server

You need to restart the metadata server before continuing this procedure.

Upgrade Your Metadata

Using SAS Management Console, upgrade your existing metadata repositories:

- 1 Log on to SAS Management Console as an unrestricted user.
- 2 Expand the Metadata Manager node in the SAS Management Console tree.
- 3 Right-click the Active Server icon, and select **Upgrade Metadata** from the pop-up menu.

All non-project repositories on the active server will be upgraded.

Note: If you have changed a host name, pathnames, or port numbers that are associated with the configuration, you will need to ensure that the metadata definitions contain the new information. To validate host names and port numbers, use the Server Manager plug-in to SAS Management Console: select the physical server definition; then, examine the properties of the connection shown in the right-hand view of the console. To verify pathnames, use the same plug-in to edit the properties of the physical server definition. △

Note: If you are working on a Windows machine, have registered local users (*host-name\user-ID*) in the metadata in your original configuration, and are running the new configuration on a different machine, you must modify the host-name qualifier on each user ID. You do this using the User Manager plug-in to SAS Management Console. Edit users and groups as appropriate, changing the user IDs on the **Logins** tab. △

Re-establishing Existing Schedule Flows

When you upgrade your metadata, existing scheduled flows are preserved. Although flows that you create in 9.1.3 are represented slightly differently in the metadata, your 9.1[.2] flows will continue to work. The 9.1.3 Schedule Manager supports both types of flows.

All you need to do is to reschedule the previously scheduled jobs in the Schedule Manager.

Note: If you have not installed your 9.1.3 scheduling software yet, wait to perform this step until after you have installed that software. △

For each flow that you want to schedule, perform the following steps:

- 1 Right-click the flow, and select **Schedule Flow** from the pop-up menu.
- 2 Click **Yes** in the Reschedule confirmation dialog box.
- 3 Choose the trigger to start the flow, and click **OK**. This will schedule the flow with Platform Computing's JobScheduler.

Server Machine, No Metadata Server

Follow the steps in this section on all machines that host SAS servers, but do not host the metadata server.

Copy Files from the Original Configuration Directory

In the typical use of the configuration, you will have created and modified files in your original configuration directory. For example, you might have added SAS system options to the configuration file *path-to-config-dir\Lev1\SASMain\sasv9.cfg*. You need to determine which of these new or modified files you should copy to the new configuration. Use the following rules to make this decision:

- If you changed a host name, pathnames, or port numbers that are associated with the configuration, you need to ensure that the new configuration directory contains the correct information.

- If you added any tuning or other options to your original configuration, you should add those options to (or maintain those options in) the new configuration.

Start Your Servers

You can now start the servers in your configuration. For details on how to perform this step, see the `instructions.html` file from your 9.1[.2] installation.

Middle-Tier Machine, No Metadata Server

You should perform the steps that are listed in the following sections on a machine where a middle tier is configured (for example, where the SAS Web Infrastructure Kit is installed) and where there is no SAS Metadata Server. Note that a couple of these steps apply only when you are upgrading from 9.1 to 9.1.3 (not from 9.1.2 to 9.1.3).

- 1 “Copy Files from the Original Configuration Directory” on page 460.
- 2 “Upgrade User and Group Definitions (9.1 to 9.1.3 only)” on page 460.
- 3 “Define the SAS Foundation Services in the Metadata (9.1 to 9.1.3 only)” on page 461.
- 4 “Update the SAS Web Infrastructure Kit Metadata” on page 461.
- 5 “Copy Custom Portlets from the Original Installation Directory” on page 461.
- 6 “Migrate Your WebDAV Data (SAS Information Delivery Portal)” on page 462.
- 7 “Migrate Your WebDAV Data (SAS Web Report Studio)” on page 464.
- 8 “Start the SAS Services Application” on page 465.
- 9 “Deploy Your Web Applications” on page 465.

Copy Files from the Original Configuration Directory

In the typical use of the configuration, you will have created and modified files in your original configuration directory. You need to determine which of these new or modified files you should copy to the new configuration. Use the following rules to make this decision:

- If you changed the a host name, pathnames, or port numbers that are associated with the configuration, you need to ensure that the new configuration directory contains the correct information.
- If you added any tuning or other options to your original configuration, you should add those options to (or maintain those options in) the new configuration.

Upgrade User and Group Definitions (9.1 to 9.1.3 only)

In SAS Management Console, add the SAS Web Administrator (`saswbadm`) to the SAS System Services group.

- 1 Select the User Manager plug-in to SAS Management Console. You will see a list of groups and users on the right side of the GUI.
- 2 Right-click the SAS System Services group, and select **Properties** from the pop-up menu. A properties dialog box appears.
- 3 Select the **Members** tab.
- 4 Move the SAS Web Administrator from the **Available Members** list to the **Current Members** list.
- 5 Click **OK** to save this group information.

Define the SAS Foundation Services in the Metadata (9.1 to 9.1.3 only)

You need to load the SAS Foundation Services into the metadata before anyone uses the SAS 9.1.3 Web applications. To do this, you use the Foundation Services Manager plug-in to SAS Management Console.

- 1 Right-click the Foundation Services Manager, and select **Import Service Deployment** from the pop-up menu. An Import Service Deployment dialog box appears.
- 2 Click **Add** in this dialog box. A file browser (an Open dialog box) appears.
- 3 Use the file browser to display the contents of the directory *configuration-directory\Lev1\web\Deployments\Portal*.
- 4 Select the following files and click **Open**.
 - *sas_services_idp_remote_omr.xml*
 - *sas_services_idp_local_omr.xml*

The names of these files will be displayed in the Import Services Deployment dialog box.

- 5 In the same way, add the file *configuration-directory\Lev1\web\Deployments\WebReportStudio\server-name_sas_pfs_queryandreporting.xml* to the list.
- 6 Click **OK** in the Import Service Deployment dialog box.

Note: If you have installed the SAS BI Web Services for Java, you should also use the Foundation Services Manager to import the file *configuration-directory\Lev1\web\Deployments\WebServices\sas_services_websvc_local_omr.xml*. △

Update the SAS Web Infrastructure Kit Metadata

You must also upgrade the metadata for the SAS Web Infrastructure Kit before anyone uses the SAS 9.1.3 Web applications. In the directory *SAS-install-dir\Web\Portal2.0.1\OMR*, you will find a set of SAS programs. You must run a subset of these programs—in the order indicated below—and check the log for any errors. There should be no errors.

If you are upgrading a 9.1.2 system to 9.1.3, run the following programs:

- 1 **Remove912PreferenceDefs.sas**
- 2 **LoadDefaultPreferences.sas**

If you are upgrading a 9.1 system to 9.1.3, run the following programs:

- 1 **LoadDefaultPreferences.sas**
- 2 **LoadPreferencesConnection.sas**
- 3 **LoadThemeConnection.sas**

On Windows systems, you can run these programs from the Windows Explorer. Right-click the SAS file, and select **Batch Submit with SAS 9.1** from the pop-up menu. After executing each SAS program, check the corresponding log file. (This file will have the same name as the program, but will end with a *.log* extension.) Open each log file and search for the string "ERROR:". You should only find program statements that contain this string, not log messages.

For more details about this step, see "Install Metadata" in the *SAS Web Infrastructure Kit Deployment Notes*.

Copy Custom Portlets from the Original Installation Directory

Portlets are mini-applications that run inside the framework that is provided by the SAS Web Infrastructure Kit. If you have created any custom portlets, you will need to

copy them from the original installation directory to the new installation directory. You will find these portlets in the directory

SAS-install-dir\Web\Portal2.0.1\DeployedPortlets.

Note: You do not need to copy portlets that were supplied with your SAS software. You need to copy only portlets that you developed yourself. \triangle

Migrate Your WebDAV Data (SAS Information Delivery Portal)

If you are using Xythos WebFile Server as your WebDAV server and your current base path root is /, you should change the base path root to */sasdav*. This change will result in better performance and will make the system consistent with any newly installed 9.1.3 systems. To make the change, perform the following steps.

- 1 Make sure that you have the latest User Management code installed in your Xythos installation area. You can obtain the latest JAR files and scripts from the 9.1.3 Xythos installation CD. Unzip *saswfs.zip* into *Xythos-install-dir\wfs-4.0.48*.
- 2 Run the *WFSInstaller.bat* script located in *Xythos-install-dir\wfs-4.0.48*. In the installation screen presented, fill in the appropriate values for each field. You can look at your existing *Xythos-install-dir\wfs-4.0.48\saswfs.properties* file to see how the values were set during the previous installation. In the **Enter the path to the user area** field, enter */sasdav*.

Note: For new installations, we recommend defining a user area */sasdav/Users*. We do not recommend */sasdav/Users* for an upgrade, because you will lose access to content in your personal repository, such as alerts. \triangle

Note: The base path root specified in the Xythos WebFile Server installation must match the value specified during the SAS Web Infrastructure Kit installation.

\triangle

- 3 Run the *WFSInstaller2.bat* script located in the same directory. This script will create a */sasdav* top-level directory on the Xythos server. This directory is referred to as the base path root. If you set the user area to */sasdav/Users*, the script will also create a */sasdav/Users* directory to store each user's personal content. If you did not set the user area this way, each user's personal content will be stored in */sasdav*.
- 4 Stop and restart the Xythos server so that the new settings take effect.
- 5 When you unpacked the file *saswfs.zip* in step 1, the migration utilities **MovePersons** and **MoveResources** were placed in the directory *wfs-4.0,48\utils*. The usage information for the two utilities is as follows:

```

-----
MovePersons (Version 1.0.001)
-----
Usage: MovePersons [/v] dir1 dir2
eg. MovePersons / /sasdav/Users
where
    dir1 is the source directory for person entries
    dir2 is the target directory
options
    /v verify operation, but do not actually do the move
-----
MoveResources (Version 1.0.001)

```

```

-----
Usage: MoveResources [/v] dir1 regex dir2
eg. MoveResources /v / "*" /sasdav
where
    dir1 is the source directory for person entries
    dir2 is the target directory
options
    /v verify operation, but do not actually do the move

```

- 6 In both of these scripts, make sure that the **XYTHOS_HOME** and **JDBC_DRIVER** environment variables are set correctly. For example, you might need to set their values like this:

```

set XYTHOS_HOME=c:\xythos
set JDBC_DRIVER=c:\cygwin\usr\share\postgresql\java\postgresql.jar

```

It is also a good idea to define the environment variable **JAVA_HOME** at the top of both scripts to make sure that you know which version of the JDK you are using:

```

set JAVA_HOME=c:\j2sdk1.4.2

```

- 7 Migrate user folders from / to **/sasdav** using the **MovePersons.cmd** script. For example, you might use the command:

```

C:\xythos\wfs-4.0.48\utils>MovePersons.cmd / /sasdav

```

If you set the user area to **/sasdav/Users** in step 2, then the second argument to this script should be **/sasdav/Users**.

- 8 Migrate other top level SAS content folders using the **MoveResources.cmd** script. For instance, the example below shows how you might move a **/Sales** folder containing Microsoft Office documents, a **/sas/publish** folder with publication/subscription content, and a **/ReportStudio** folder containing reports:

```

C:\xythos\wfs-4.0.48\utils>MoveResources / "Sales" /sasdav
C:\xythos\wfs-4.0.48\utils>MoveResources / "sas" /sasdav
C:\xythos\wfs-4.0.48\utils>MoveResources / "ReportStudio" /sasdav

```

- 9 If you have defined your WebDAV server in your metadata, bring up SAS Management Console, and go to the Server Manager plug-in. Right-click the icon representing your WebDAV server, and select **Properties** from the pop-up menu that appears. In the Properties dialog box, select the **Options** tab. In the **Base Path(s)** list, change / to **/sasdav**.
- 10 If you have defined channels with a WebDAV persistent store in the metadata, bring up SAS Management Console, and go to the Publishing Framework plug-in. Expand the tree to list the channels. Right-click each channel that uses WebDAV for a persistent store, and select **Properties** from the pop-up menu that appears. In the Properties dialog box, select the **General** tab. In the **Content Base Path** drop-down list, change / to **/sasdav**.
- 11 Make sure that the **SDAV_BASE\$** name/value pair located in the file *SAS-install-dir\Web\Portal2.0.1\PortalConfigure\install.properties* is set to **/sasdav**. This value is the DAV base bath root used to configure the Web Infrastructure Kit (WIK). The base path root is used in the WIK's local services deployment definition: there is a repository definition entry for the DAV server in the information services configuration.
- 12 Run the script **configure_wik**. This script is located in the directory *SAS-install-dir\Web\Portal2.0.1*.

- 13 If the services deployment definitions are loaded in your metadata repository, reload the service definitions to make sure that the DAV repository definition entry has the correct base path.

Migrate Your WebDAV Data (SAS Web Report Studio)

If you are using SAS Web Report Studio at your site, you might be using either the Xythos WebFile Server or the Apache HTTP Server as a WebDAV server. In either case, you should make your existing WebDAV content available at the content base path `/sasdav/wrs` instead of (or in addition to) `/dav/sas`. This section explains how to migrate your content for each of these servers.

If you are using the Xythos WebFile Server as your WebDAV server, follow the procedure below:

- 1 Use the Xythos WFS administration console to create the folder structure `/sasdav/wrs`.
- 2 Use the Xythos WFS administration console to grant the SAS Web Administrator (`saswbadm`) full access to the `wrs` folder and to deny access to this folder to other users.
- 3 Copy your existing WebDAV content from `/dav/sas` to `/sasdav/wrs`.
- 4 In SAS Management Console, use the Server Manager plug-in to change the properties for your WebDAV server (HTTP DAV Server). Right-click the icon that represents this server, and select **Properties** from the pop-up menu. A properties dialog box appears. On the **Options** tab, create a new base path: `/sasdav/wrs`.
- 5 In SAS Management Console, use the Business Report Manager plug-in to update the properties of the BIP Tree. Expand the Business Report Manager. Then right-click the BIP Tree icon, and select **Properties** in the pop-up menu that appears. A properties dialog box appears. Set the **Content Base Path** to `/sasdav/wrs`, and enter the user ID and password for the SAS Web Administrator in the **Content Server Authentication** area.

If you are using the Apache HTTP Server as your WebDAV server, use the following procedure::

- 1 Edit the `httpd.conf` configuration file to create an alias for the content base path. This way, you do not have to actually copy your WebDAV content to a new location.

When you first configured your system, you should have added an element like this to this file:

```
<Directory "C:\Program Files\Apache Group\Apache2\htdocs\dav\sas">
  SetEnv redirect-carefully 1
  Dav On
  DavDepthInfinity On
  Options None
  AllowOverride None
</Directory>
```

Add the following line just before this element:

```
Alias /sasdav/wrs "C:\Program Files\Apache Group\Apache2\htdocs\dav\sas"
```

Then, save the file.

- 2 In SAS Management Console, use the Server Manager plug-in to change the properties for your WebDAV server (HTTP DAV Server). Right-click the icon that represents this server, and select **Properties** from the pop-up menu. A properties dialog box appears. On the **Options** tab, create a new base path: `/sasdav/wrs`.
- 3 In SAS Management Console, use the Business Report Manager plug-in to update the properties of the BIP Tree. Expand the Business Report Manager. Then

right-click the BIP Tree icon, and select **Properties** in the pop-up menu that appears. A properties dialog box appears. Set the **Content Base Path** to `/sasdav/wrs`, and enter the user ID and password for the SAS Web Administrator in the **Content Server Authentication** area.

Start the SAS Services Application

You should now start the SAS Services application. For more information about how to perform this step, see the directions in your initial configuration instructions, `instructions.html`.

Deploy Your Web Applications

Your Web applications will have been upgraded. Redeploy these new versions to your Web server.

- 1 If necessary, uninstall the current version of each SAS Web application. If you are using Tomcat as your Web server, delete the directory structure for a Web application from the `webapps` directory. If you are using the BEA WebLogic server as your Web server:
 - Use the WebLogic Server administration console to delete an application. (To see the currently deployed Web applications, expand the Deployments tree node, and select the Web Applications node. In the resulting list of Web applications, select the Delete icon for the appropriate application.)
 - Remove the directory structure for the Web application from the `webapps` directory.
- 2 To deploy the new Web applications, follow the directions in the section “Deploy Your Web Applications” in your configuration instructions, `instructions.html`.

Note: For more detailed information about deploying your Web applications, see the section “Getting More Information” in `instructions.html`. Δ

- 3 If you are using Tomcat as your servlet container, start Tomcat. For directions on how to do this, see “Start Your Tomcat Server” in `instructions.html`.

Middle-Tier Machine, Metadata Server

On a machine where a middle tier is configured (for example, where the SAS Web Infrastructure Kit is installed) *and* where a SAS Metadata Server is configured, you should perform the steps that are presented in the following sections.

- 1 “Stop the Metadata Server” on page 466.
- 2 “Copy Files from the Original Configuration Directory” on page 466.
- 3 “Start the Metadata Server” on page 466.
- 4 “Upgrade Your Metadata” on page 466.
- 5 “Start the Other Servers” on page 467.

You should then perform most of the steps discussed in “Middle-Tier Machine, No Metadata Server” on page 460. Specifically, perform the steps in these sections (as appropriate):

- 1 “Upgrade User and Group Definitions (9.1 to 9.1.3 only)” on page 460.
- 2 “Define the SAS Foundation Services in the Metadata (9.1 to 9.1.3 only)” on page 461.
- 3 “Update the SAS Web Infrastructure Kit Metadata” on page 461.

- 4 “Copy Custom Portlets from the Original Installation Directory” on page 461.
- 5 “Migrate Your WebDAV Data (SAS Information Delivery Portal)” on page 462.
- 6 “Migrate Your WebDAV Data (SAS Web Report Studio)” on page 464.
- 7 “Start the SAS Services Application” on page 465.
- 8 “Deploy Your Web Applications” on page 465.

Stop the Metadata Server

The SAS Configuration Wizard automatically starts the metadata server. To make sure that there are no file contention issues, stop the metadata server before you copy any files into the new configuration directory.

Copy Files from the Original Configuration Directory

In the typical use of the configuration, you will have created and modified files in your original configuration directory. You need to determine which of these new or modified files you should copy to the new configuration. Use the following rules to make this decision:

- If you changed the a host name, pathnames, or port numbers that are associated with the configuration, you need to ensure that the new configuration directory contains the correct information.
- If you added any tuning or other options to your original configuration, you should add those options to (or maintain those options in) the new configuration.
- You must always copy the SAS Metadata Server files to the new configuration. These files are contained in the `MetadataServer/reposmgr` and `MetadataServer/MetadataRepositories` directories. You should place these files in the same location in the new configuration directory.

Start the Metadata Server

You need to restart the metadata server before continuing this procedure.

Upgrade Your Metadata

Using SAS Management Console, upgrade your existing metadata repositories:

- 1 Log on to SAS Management Console as an unrestricted user.
- 2 Expand the Metadata Manager node in the SAS Management Console tree.
- 3 Right-click the Active Server icon, and select **Upgrade Metadata** from the pop-up menu.

All repositories on the active server will be upgraded.

Note: If you have changed the a host name, pathnames, or port numbers that are associated with the configuration, you will need to ensure that the metadata definitions contain the new information. To validate host names and port numbers, use the Server Manager plug-in to SAS Management Console: select the physical server definition; then, examine the properties of the connection shown in the right-hand view of the console. To verify pathnames, use the same plug-in to edit the properties of the physical server definition. \triangle

Note: If you are working on a Windows machine, have registered local users (`host-name\user-ID`) in the metadata in your original configuration, and are running the new configuration on a different machine, you must modify the host-name qualifier on

each user ID. You do this using the User Manager plug-in to SAS Management Console. Edit users and groups as appropriate, changing the user IDs on the **Logins** tab. Δ

Start the Other Servers

You can now start the servers in your configuration. (The metadata server should be running already.) For more details on how to perform this step, see the directions in your initial configuration instructions, `instructions.html`.

The Remaining Steps

As mentioned earlier, you should now perform all except the first step presented in the section “Middle-Tier Machine, No Metadata Server” on page 460.

Upgrading a SAS 9.1 or 9.1.2 z/OS System to a SAS 9.1.3 System

The following two sections explain how to create a SAS z/OS business intelligence installation that you can swap into a functioning SAS 9.1 or SAS 9.1.2 business intelligence installation. Be certain to follow the steps in order.

Installing and Configuring the SAS 9.1.3 Software

Follow these instructions to install and configure a 9.1.3 system that you can test before you put it into production.

- 1 Install SAS 9.1.3 Foundation for z/OS using the “Action A” instructions for installing a completely new SAS Foundation for z/OS. You can use the SAS Installation Wizard for z/OS for this task. The high level prefix that you use for this installation is *not* critical.

Note: The “Action A” instructions are located in your Installation Kit, behind the Installation Guide tab. Δ

- 2 Perform all of the necessary post-installation customization tasks for the new SAS 9.1.3 Foundation for z/OS. These customizations are discussed in appendixes in the Configuration Guide section of the Installation Kit, and should mirror the customizations that you performed when installing the SAS 9.1 or SAS 9.1.2 Foundation for z/OS.
- 3 In the control data set used for installing the new SAS 9.1.3 Foundation for z/OS (hereafter referred to as the **CNTLDSN**), open an editor on the member **UGTARGET**. Edit the following line:

```
UGPFX=SAS.V91.PROD.PFX          <<==SUPPLY YOUR PRODUCTION SAS PREFIX
```

Substitute the high-level prefix of your production or existing SAS 9.1 or SAS 9.1.2 Foundation for z/OS system for **SAS.V91.PROD.PFX**. Then save the member.

- 4 Open the editor on **CNTLDSN(MAKEUPG)**. Locate the following JCL near the beginning of the job:

```
//INSTEDT1 PROC CNTLDSN='your_ctrl_data_set',
//          SASPROC='SAS'                <=== << SUPPLY >>
//          SASDTP='SASEDTP',           <=== << VERIFY >>
//          SYSOUT='*',                  <=== << VERIFY >>
//          DISKUNI='DISK'               <=== << VERIFY >>
```

Then, follow these steps:

- a In the line with **SASPROC='SAS'**, substitute the procedure name of a working SAS 9.1, 9.1.2, or 9.1.3 PROC.
 - b Do not change **SASEDITP=parm** unless directed to do so by SAS Technical Support.
 - c Verify the **SYSOUT=** and **DISKUNI=parms** DD statements, which should have been set correctly from the main install.
 - d Save the edit.
- 5 Submit **CNTLDSN(MAKEUPG)** for execution. This will create four utilities to be used later. The names will contain the *actual* TS level identifiers of the production (old) and newly installed systems, obtained by analyzing the executables library of the production and new systems. These two TS level identifiers will also be used in the new data set names created by running the utilities.

In the section “Replacing Your Production System” on page 470, “TSpMp” (your production or old system) and “TSnMn” (your newly installed system) may have actual values of TS1M0 and TS1M3 as one of a number of possible TS level combinations. Substitute your actual TS level values for **TSpMp** and **TSnMn**.

- 6 After the new system has been deemed production quality, and during system down time, perform the following steps:
 - a Stop the version 9.1 or 9.1.2 servers.
 - b Copy the version 9.1 or 9.1.2 configuration directory to the version 9.1.3 configuration directory. You can do this by running the following command from the UNIX System Services shell:

```
cp -R 9.1_config_dir 9.1.3_config_dir
```

- c Remove old log files from the 9.1.3 configuration directory. Use a command similar to this one:

```
rm 9.1.3_config_dir/Levl/SASMain/*/logs/*
```

- d Edit the COPYIA job with the location of the 9.1.3 configuration directory. (Supply the value for the CONFIG_DIR variable.) The COPYIA job is located in the *sas_v913_installed_prefix.w0.SRVCNTL* data set. Run the COPYIA job, and verify a successful completion.)
 - e Copy **configuration.properties** from your 9.1 or 9.1.2 configuration directory to your 9.1.3 configuration directory. From the USS shell, issue the command:

```
cp 9.1_config_dir/Utilities/zOS_config/configuration.properties
9.1.3_config_dir/Utilities/zOS_config/configuration.properties
```

- f Change directories so that you are in the **zOS_config** directory inside the 9.1.3 configuration directory. To get there, use the USS shell command:

```
cd 9.1.3_config_dir/Utilities/zOS_config
```

- g Edit the 9.1.3 version of **configuration.properties**, supplying new values for the following properties:
 - \$CONFIG_DIRS - Point to the 9.1.3 configuration directory.
 - \$APPSERVER_DIRS - Point to the new application-server directory, for example, *9.1.3_config_dir/Levl/SASMain*.
 - h You need to change only the following values if your site wants to run the 9.1.3 servers in parallel with the 9.1 or 9.1.2 servers during a testing period. This is the recommended approach. If your site wants to run the 9.1.3 servers right out of the box, just as they were configured in 9.1 or 9.1.2, then you do not need to change any of the following values:

- \$OMAPORTS\$ - Enter a new port number.
 - \$OMA_STCNAMES\$ - Enter a new started task name (different from the 9.1 or 9.1.2 name).
 - \$OMA_CFGNAMES\$ - Enter a new configuration file name (same as the started task name).
 - \$OMA_TKENVNMS\$ - Enter the new tk environment file name (same as the started task name).
 - \$SPAWNER_OPERATOR_PORTS\$ - Enter a new port number.
 - \$SPAWNER_LOADBALANCING_PORTS\$ - Enter a new port number.
 - \$IOM_PORTS\$ - Enter a new port number.
 - \$SPAWNER_STCNAMES\$ - Enter a new started task name.
 - \$SPAWNER_PRMNAMES\$ - Enter a new parm name.
 - \$STP_PORTS\$ - Enter a new port number.
 - \$STP_PORT1\$ - Enter a new port number.
 - \$STP_PORT2\$ - Enter a new port number.
 - \$STP_PORT3\$ - Enter a new port number.
 - \$OLA_PORTS\$ - Enter a new port number.
 - \$OLA_STCNAMES\$ - Enter a new started task name.
 - \$OLA_CFGNAMES\$ - Enter a new configuration file name.
 - \$OLA_TKENVNMS\$ - Enter a new tk environment file name.
 - \$CONNECT_PORTS\$ - Enter a new port number.
 - \$SHAREPORTS\$ - Enter a new port number.
 - \$SHR_STCNAMES\$ - Enter a new started task name.
 - \$SHR_CFGNAMES\$ - Enter a new configuration file name.
 - \$SHR_TKENVNMS\$ - Enter a new tk environment file name.
 - \$CNT_STCNAMES\$ - Enter a new started task name.
 - \$CNT_PRMNAMES\$ - Enter a new parm file name.
- i Run the **deploy_IA.sh** script.
- ```
deploy_IA.sh -p configuration.properties
```
- If you are asked whether you want to overwrite the environment, answer **Yes**.
- j Copy the new started task names to your site's started task library.
- k Start the metadata server.
- l Using SAS Management Console, upgrade your existing metadata repositories.
- i Expand the Metadata Manager plug-in to SAS Management Console.
  - ii Right-click the Active Server icon, and select **Upgrade Metadata** from the pop-up menu.
- m Using SAS Management Console, update the metadata for the 9.1.3 system (using the 9.1.3 **instructions.html** as a guide). Change
- the Launch Command for the workspace server
  - the Launch Command for the stored process server
  - the Launch Command for the SAS/CONNECT server
  - all the port numbers listed previously.
- n Start the remaining servers.
- o Test all of the servers.

## Replacing Your Production System

In the following steps, you will rename your production and newly installed SAS system data sets in order to facilitate the changeover between your existing production SAS environment and the newly installed SAS system. You will accomplish this by running the **RNMTSpMp** and **RNMTSnMn** utilities, which will rename the existing SAS data sets using the following guidelines:

- $\square$  **SAS.V91.PROD.PFX.\*** becomes **SAS.V91.PROD.PFX.TSpMp.XXX**
- $\square$  **SAS.V913.PFX.\*** becomes **SAS.V91.PROD.PFX.TSnMn.XXX**

Alias names will be created for the originally installed data set names, and they will now point to the recently renamed data set names.

### CAUTION:

**Note that the alias names are alternate names for data sets that are defined in the catalog.** RACF does not allow alias names because it uses the RACF database, not the catalog, for its processing. If you have existing RACF profiles for your SAS installation, you might need to adjust these profiles based on the new data set names.  $\triangle$

### CAUTION:

**If your installation currently uses, or has future plans to run, a SAS/CONNECT spawner or object spawner, note that the SAS executable library and the SAS/C transient library must be put under RACF program control.** Because RACF does not support alias names, you must use the renamed, fully qualified data set names in the program control table.  $\triangle$

### CAUTION:

**After the RNMTSpMp utility is executed, the SAS/CONNECT and object spawners will not function until the SAS.V91.PROD.PFX.TSpMp.DLR (the renamed SAS/C transient library) and SAS.V91.PROD.PFX.TSpMp.DLD (the renamed SAS executable library) have been added to the program control table.**  $\triangle$

- 1 During system downtime, submit the utility **CNTLDSN(RNMTSpMp)**. This utility will rename the production system data sets and create aliases from the existing names to the new names. If you need to run this utility outside of system downtime, you must follow a locally appropriate procedure to ensure that data sets are not in use or enqueued, which will cause the process to fail.
- 2 Add the **SAS.V91.PROD.PFX.TSpMp.DLR** and **SAS.V91.PROD.PFX.TSpMp.DLD** libraries to the program control table, if your existing production SAS installation uses a SAS/CONNECT spawner or object spawner.
- 3 Submit the utility **CNTLDSN(RNMTSnMn)**. This utility will rename the new system data sets and create aliases from the existing names to the new names.
- 4 Add the **SAS.V91.PROD.PFX.TSnMn.DLR** and **SAS.V91.PROD.PFX.TSnMn.DLD** libraries to the program control table, if your existing production SAS installation uses a SAS/CONNECT spawner or object spawner.

*Note:* The **TSxxx** portion of these names is taken from the TS level that is obtained from the installed systems and will vary depending on the version of the installed systems.  $\triangle$

At this point you are ready to perform a swap.

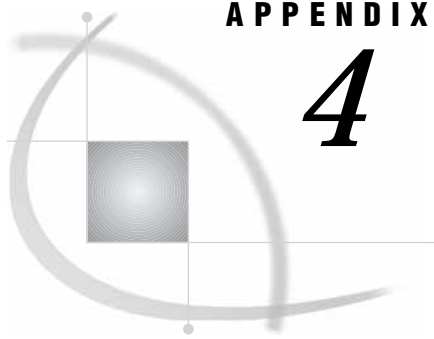
- 5 During system downtime, or during a testing period where access to and enqueues upon the SAS system can be controlled, submit the utility **CNTLDSN(USSETSnmn)**, which will cause the production-named aliases to point to the new SAS system of that TS level. Note that once you have submitted the **USSETSnmn** utility, the 9.1.3 SAS system will be the production environment. Therefore, only the BI server started tasks that were created in step 6j are valid for use. The 9.1 or 9.1.2 BI

server started tasks that were previously defined in your started tasks library will not be valid unless you change the system back to 9.1 or 9.1.2 by running the **CNTLDSN(USETSpMp)** utility.

- 6 Perform tests as needed or scheduled to verify that the new production system is functioning properly.
- 7 You can perform an emergency restoration of the previous system by submitting the utility **CNTLDSN(USETSpMp)**.







## APPENDIX

## 4

## Recommended Reading

---

*Recommended Reading* 473

---

### 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 Integration Technologies: Server Administrator's Guide*
- *SAS Language Reference: Concepts*
- *SAS Language Reference: Dictionary*
- *SAS Management Console: User's Guide*
- *SAS Metadata LIBNAME Engine: User's Guide*
- *SAS Metadata Server: Setup 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](mailto:sasbook@sas.com)  
 Web address: [support.sas.com/pubs](http://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 determining which users have which permissions for which resources. The outcome of the authorization process is an authorization decision that either permits or denies a specific action on a specific resource, based on the requesting user's identity and group memberships.

**basic installation**

a method of installing a SAS business intelligence system that requires you to specify what software should be installed on each host. The basic installation method uses an installation tool called the SAS Software Navigator. See also project installation.

**batch mode**

a method of running SAS programs in which you prepare a file that contains SAS statements plus any necessary operating system control statements and submit the file to the operating system. Execution is completely separate from other operations at your terminal. Batch mode is sometimes referred to as running in the background.

**buffer**

a portion of computer memory that is used for special holding purposes or processes. For example, a buffer might simply store information before sending that information to main memory for processing, or it might hold data after the data is read or before the data is written.

**cache**

a small, fast memory area that holds recently accessed data. The cache is designed to speed up subsequent access to the same data.

**change management**

in the SAS Open Metadata Architecture, a facility for metadata source control, metadata promotion, and metadata replication.

**channel**

a virtual communication path for distributing information. In SAS, a channel is identified with a particular topic (just as a television channel is identified with a particular radio frequency). Using the features of the Publishing Framework, authorized users or applications can publish digital content to the channel, and authorized users and applications can subscribe to the channel in order to receive the content. See also *publish*, *subscribe*.

**cleanse**

to improve the consistency and accuracy of data by standardizing it, reorganizing it, and eliminating redundancy.

**client application**

an application that runs on a client machine.

**client tier**

the portion of a distributed application that requests services from the server tier. The client tier typically uses a small amount of disk space, includes a graphical user interface, and is relatively easy to develop and maintain.

**cluster**

a group of machines that participate in load balancing. Each machine in the cluster runs an object spawner that handles client requests for connections.

**COM (Component Object Model)**

an object-oriented programming model that defines how software components interact within a single process or between processes. For example, COM includes standard rules of communication that enable a user-interface object to be dragged and dropped from one application window to another.

**component**

a self-contained, reusable programming object that provides some type of service to other components in an object-oriented programming environment.

**controller**

a computer component that manages the interaction between the computer and a peripheral device such as a disk or a RAID. For example, a controller manages data I/O between a CPU and a disk drive. A computer can contain many controllers. A single CPU can command more than one controller, and a single controller can command multiple disks.

**CORBA (Common Object Request Broker Architecture)**

a standard API for distributed object communication. CORBA was created by the Object Management Group. It is the most widely used distributed object standard for connecting operating system platforms from multiple vendors.

**credentials**

the user ID and password for a particular user account that has been established either in the operating system or with an alternative authentication provider such as Microsoft Active Directory or Lightweight Directory Access Protocol.

**cube**

a logical set of data that is organized and structured in a hierarchical, multidimensional arrangement. A cube is a directory structure, not a single file. A cube includes measures, and it can have numerous dimensions and levels of data.

**cube loading**

the process of building a logical set of data that is organized and structured in a hierarchical, multidimensional arrangement. See also *cube*.

**custom repository**

in the SAS Open Metadata Architecture, a metadata repository that must be dependent on a foundation repository or custom repository, thus allowing access to metadata definitions in the repository or repositories on which it depends. A custom repository is used to specify resources that are unique to a particular data collection. For example, a custom repository could define sources and targets that are unique to a particular data warehouse. The custom repository would access user definitions, group definitions, and most server metadata from the foundation repository. See also foundation repository, project repository.

**daemon**

a process that starts and waits either for a request to perform work or for an occurrence of a particular event. After the daemon receives the request or detects the occurrence, it performs the appropriate action. If nothing else is in its queue, the daemon then returns to its wait state.

**data mart**

a collection of data that is optimized for a specialized set of users who have a finite set of questions and reports.

**data partition**

a physical file that contains data and which is part of a collection of physical files that comprise the data component of a SAS Scalable Performance Data Engine data set. See also partition, partitioned data set.

**data quality**

the relative value of data, which is based on the accuracy of the knowledge that can be generated using that data. High-quality data is consistent, accurate, and unambiguous, and it can be processed efficiently.

**data warehouse**

a collection of data that is extracted from one or more sources for the purpose of query, reporting, and analysis. In contrast to a data mart, a data warehouse is better suited for storing large amounts of data that originates in other corporate applications or which is extracted from external data sources such as public databases.

**database library**

a collection of one or more database management system files that are recognized by SAS and that are referenced and stored as a unit. Each file is a member of the library.

**database management system (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 is used to specify metadata for global resources that can be shared by other repositories. For example, a foundation repository is used to store metadata that defines users and groups on the metadata server. Only one foundation repository should be defined on a metadata server. See also custom repository, project repository.

**global resource**

a widely used resource, such as a server that is used to access many tables in a data warehouse. See also resource.

**hierarchy**

an arrangement of members of a dimension into levels that are based on parent-child relationships. Members of a hierarchy are arranged from more general to more specific. For example, in a Time dimension, a hierarchy might consist of the members Year, Quarter, Month, and Day. In a Geography dimension, a hierarchy might consist of the members Country, State or Province, and City. More than one hierarchy can be defined for a dimension. Each hierarchy provides a navigational path that enables users to drill down to increasing levels of detail. See also member, level.

**HTTP (HyperText Transfer Protocol)**

a protocol for transferring data to the Internet. HTTP provides a way for servers and Web clients to communicate. It is based on the TCP/IP protocol.

**HTTP server**

a server that handles an HTTP request from a client such as a Web browser. Usually the client's HTTP request indicates that the client wants to retrieve information that is pointed to by a URL. An example of a popular HTTP server is the Apache HTTP Server from the Apache Software Foundation.

**HTTPS (HyperText Transfer Protocol Secure)**

an HTTP protocol that enables secure connections to be made between a Web browser and a server.

**identity**

See metadata identity.

**inbound login**

a login that is used to determine your metadata identity. The login is inbound to a SAS Metadata Server. A login that functions only as an inbound login does not need to include a password or to specify an authentication domain.

**information map**

a collection of data items and filters that describes and provides a business-relevant view of physical data. Users of query and reporting applications such as SAS Web Report Studio can easily build business reports by using information maps as the building blocks for their reports.

**Integrated Object Model**

See IOM (Integrated Object Model).

**Integrated Object Model server**

See IOM server.

**IOM (Integrated Object Model)**

the set of distributed object interfaces that make SAS software features available to client applications when SAS is executed as an object server.

**IOM bridge**

a software component of SAS Integration Technologies that enables Java clients and Windows clients to access an IOM server.

**IOM server**

a SAS object server that is launched in order to fulfill client requests for IOM services. See also IOM (Integrated Object Model).

**JAR file**

a Java Archive file. The JAR file format is used for aggregating many files into one file. JAR files have the file extension .jar.

**Java**



a set of technologies for creating software programs in both stand-alone environments and networked environments, and for running those programs safely. Java is a Sun Microsystems trademark.

**Java application**

a stand-alone program that is written in the Java programming language.

**Java Database Connectivity**

See JDBC (Java Database Connectivity).

**Java Development Kit**

See JDK (Java Development Kit).

**Java Virtual Machine**

See JVM (Java Virtual Machine).

**JavaServer page**

See JSP (JavaServer page).

**JDBC (Java Database Connectivity)**

a standard interface for accessing SQL databases. JDBC provides uniform access to a wide range of relational databases. It also provides a common base on which higher-level tools and interfaces can be built.

**JDK (Java Development Kit)**

a software development environment that is available from Sun Microsystems, Inc. The JDK includes a Java Runtime Environment (JRE), a compiler, a debugger, and other tools for developing Java applets and applications.

**job**

a metadata object that specifies processes that create output.

**JSP (JavaServer page)**

a type of servlet that enables users to create Java classes through HTML.

**JVM (Java Virtual Machine)**

a program that interprets Java programming code so that the code can be executed by the operating system on a computer. The JVM can run on either the client or the server. The JVM is the main software component that makes Java programs portable across platforms. A JVM is included with JDKs and JREs from Sun Microsystems, as well as with most Web browsers.

**key**

a value that uniquely identifies a specific record in a database.

**LDAP (Lightweight Directory Access Protocol)**

a protocol that is used for accessing directories or folders. LDAP is based on the X.500 standard, but it is simpler and, unlike X.500, it supports TCP/IP.

**LDAP directory**

a repository that contains data about an enterprise's users and resources, as well as related security information, and that stores this data and information in a format that clients on a network can access by using the Lightweight Directory Access Protocol (LDAP).

**LDAP server**

a server that provides access to one or more LDAP directories.

**level**

in a multidimensional database (or cube), an element of a dimension hierarchy. Levels describe the dimension from the highest (most summarized) level to the lowest (most detailed) level. For example, possible levels for a Geography dimension are Country, Region, State or Province, and City.

**Lightweight Directory Access Protocol**

See LDAP (Lightweight Directory Access Protocol).

**load balancing**

for IOM bridge connections, a program that runs in the object spawner and that uses an algorithm to distribute work across object server processes on the same or separate machines in a cluster.

**locale**

a value that reflects the language, local conventions, and culture for a geographic region. Local conventions can include specific formatting rules for dates, times, and numbers, and a currency symbol for the country or region. Collating sequences, paper 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.

**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 (MDDDB)**

another term for cube. See cube.

**object**

in object-oriented programming, an instantiation or specific representation of a class.

**Object Linking and Embedding**

See OLE (Object Linking and Embedding).

**object server**

another term for IOM server. See IOM server.

**object spawner**

a program that instantiates object servers that are using an IOM bridge connection. The object spawner listens for incoming client requests for IOM services. When the spawner receives a request from a new client, it launches an instance of an IOM server to fulfill the request. Depending on which incoming TCP/IP port the request was made on, the spawner either invokes the administrator interface or processes a request for a UUID (Universal Unique Identifier).

**OLAP (online analytical processing)**

a software technology that enables users to dynamically analyze data that is stored in cubes.

**OLAP schema**

a group of cubes. A cube is assigned to an OLAP schema when it is created, and an OLAP schema is assigned to a SAS OLAP Server when the server is defined in the metadata. A SAS OLAP Server can access only the cubes that are in its assigned OLAP schema.

**OLE (Object Linking and Embedding)**

a method of interprocess communication supported by Windows that involves a client/server architecture. OLE enables an object that was created by one application to be embedded in or linked to another application.

**OLE DB**

an open specification that has been developed by Microsoft for accessing both relational and nonrelational data. OLE DB interfaces can provide much of the same functionality that is provided by database management systems. OLE DB evolved from the Open Data base Connectivity (ODBC) application programming interface. See also OLE (Object Linking and Embedding).

**OLE DB for OLAP (ODBO)**

an extension to OLE DB that enables users to access multidimensional databases in addition to relational databases. See also OLE DB, OLAP (online analytical processing).

**outbound login**

a login that applications can retrieve from a SAS Metadata Server and send to other systems that need to verify a user's identity. The login is outbound from the SAS Metadata Server to the other systems. An outbound login must specify an authentication domain and must include credentials that are appropriate for the systems to which the login provides access.

**package**

a container for data that has been generated or collected for delivery to consumers by the SAS Publishing Framework. Packages can contain SAS files (SAS catalogs; SAS data sets; various types of SAS databases, including cubes; and SAS SQL views), binary 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). Packages also contain metadata such as a description, an abstract, and user-specified name/value pairs.

**page size**

the number of bytes of data that SAS moves between external storage and memory in one input/output operation. Page size is analogous to buffer size for SAS data sets.

**parallel I/O**

a method of input and output that takes advantage of multiple CPUs and multiple controllers, with multiple disks per controller to read or write data in independent threads.

**parallel processing**

a method of processing that uses multiple CPUs to process independent threads of an application's computations. See also *threading*.

**partition**

part or all of a logical file that spans devices or directories. In the SPD Engine, a partition is one physical file. Data files, index files, and metadata files can all be partitioned, resulting in data partitions, index partitions, and metadata partitions, respectively. Partitioning a file can improve performance for very large data sets. See also *data partition*, *partitioned data set*.

**partitioned data set**

in the SPD Engine, a data set whose data is stored in multiple physical files (partitions) so that it can span storage devices. One or more partitions can be read in parallel by using threads. This improves the speed of I/O and processing for very large data sets. See also *parallel processing*, *partition*, *thread*.

**permanent package**

a container for content that was produced by a SAS program or by a third-party application, and that is written to a specific location. Permanent packages remain in existence even after the stored process completes execution and the client disconnects from the server. See also *transient package*.

**permanent result package**

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*.

**planning file**

an XML file that contains a list of the products to be installed and the components to be configured at a site. This file serves as input to both the SAS Software Navigator and the SAS Configuration Wizard.

**pool**

a group of server connections that can be shared and reused by multiple client applications. A pool consists of one or more puddles. See also *puddle*.

**pooling**

the act or process of creating a pool. See *pool*.

**portlet**

a Web component that is managed by a Web application and which is aggregated with other portlets to form a page within the application. A portlet processes requests from the user and generates dynamic content.

**pre-installation checklist**

a checklist that enumerates the tasks a customer must perform before installing the business intelligence platform. The primary task is to create a set of operating system user accounts on the metadata server host.

**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**

a repository that must be dependent on a foundation repository or custom repository that will be managed by the Change Management Facility. A project repository is used to isolate changes from a foundation repository or from a custom repository. The project repository enables metadata programmers to check out metadata from a foundation repository or custom repository so that the metadata can be modified and tested in a separate area. Project repositories provide a development/testing environment for customers who want to implement a formal change management scheme. See also foundation repository, custom repository.

**promotion**

See metadata promotion.

**Public Kiosk**

a public page that is displayed when a user starts the SAS Information Delivery Portal but has not yet logged on.

**publish**

to deliver electronic information, such as SAS files (including SAS data sets, SAS catalogs, and SAS data views), other digital content, and system-generated events to one or more destinations. These destinations can include e-mail addresses, message queues, publication channels and subscribers, WebDAV-compliant servers, and archive locations.

**puddle**

a group of servers that are started and run using the same login credentials. Each puddle can also allow a group of clients to access the servers. See also pool.

**RAID (redundant array of independent disks)**

a type of storage system that comprises many disks and which implements interleaved storage techniques that were developed at the University of California at Berkeley. RAIDs can have several levels. For example, a level-0 RAID combines two or more hard drives into one logical disk drive. Various RAID levels provide various levels of redundancy and storage capability. A RAID provides large amounts of data storage inexpensively. Also, because the same data is stored in different places, I/O operations can overlap, which can result in improved performance. See also redundancy.

**redundancy**

a characteristic of computing systems in which multiple interchangeable components are provided in order to minimize the effects of failures, errors, or both. For example, if data is stored redundantly (in a RAID, for example), then if one disk is lost, the data is still available on another disk. See also RAID (redundant array of independent disks).

**relational database management system**

a database management system that organizes and accesses data according to relationships between data items. The main characteristic of a relational database management system is the two-dimensional table. Examples of relational database management systems are DB2, Oracle, SYBASE, and Microsoft SQL Server.

**Remote Library Services (RLS)**

a feature of SAS/SHARE and SAS/CONNECT software that enables you to read, write, and update remote data as if it were stored on the client. RLS can be used to access SAS data sets on computers that have different architectures. RLS also provides read-only access to some types of SAS catalog entries on computers that have different architectures. See also architecture.

**replication**

See metadata replication.

**repository access control template**

the access control template (ACT) that controls access to a particular repository and to resources for which access controls are not specified. You can designate one repository ACT for each metadata repository. The repository ACT is also called the default ACT.

**resource**

any object that is registered in a metadata repository. For example, a resource can be an application, a data store, a dimension in an OLAP cube, a metadata item, an access control template, or a password.

**resource template**

an XML file that specifies the information that is needed for creating a metadata definition for a SAS resource.

**result type**

the kind of output that is produced by a 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 log**

a file that contains a record of the SAS statements that you enter as well as messages about the execution of your program.

**SAS Management Console**

a Java application that provides a single user interface for performing SAS administrative tasks.

**SAS Metadata Model**

a collection of metadata types that are used for saving information about application elements.

**SAS Metadata Repository**

one or more files that store metadata about application elements. Users connect to a SAS Metadata Server and use the SAS Open Metadata Interface to read metadata from or write metadata to one or more SAS Metadata Repositories. The metadata types in a SAS Metadata Repository are defined by the SAS Metadata Model.

**SAS Metadata Server**

a multi-user server that enables users to read metadata from or write metadata to one or more SAS Metadata Repositories. The SAS Metadata Server uses the Integrated Object Model (IOM), which is provided with SAS Integration Technologies, to communicate with clients and with other servers.

**SAS OLAP Cube Studio**

a Java interface for defining and building OLAP cubes in SAS System 9 or later. Its main feature is the Cube Designer wizard, which guides you through the process of registering and creating cubes.

**SAS OLAP Server**

a SAS server that provides access to multidimensional data. The data is queried using the multidimensional expressions (MDX) language.

**SAS Open Metadata Architecture**



a general-purpose metadata management facility that provides metadata services to SAS applications. The SAS Open Metadata Architecture enables applications to exchange metadata, which makes it easier for these applications to work together.

**SAS procedure**

a program that produces reports, manages files, or analyzes data and which is accessed with a PROC statement. Many procedures are included in SAS software.

**SAS Report Model**

an XML specification that defines a standard reporting format and provides common reporting functions for SAS applications.

**SAS statement**

a string of SAS keywords, SAS names, and special characters and operators that instructs SAS to perform an operation or that gives information to SAS. Each SAS statement ends with a semicolon.

**SAS Stored Process**

a SAS program that is stored in a central location and which can be executed from the SAS Information Delivery portal at the user's request. When a stored process is executed, it creates a report that includes the most current data that is available. Stored processes can display input forms that enable users to customize the contents of reports.

**SAS Stored Process Server**

a SAS IOM server that is launched in order to fulfill client requests for SAS stored processes. See also IOM server.

**SAS system option**

an option that affects the processing of an entire SAS program or interactive SAS session from the time the option is specified until it is changed. Examples of items that are controlled by SAS system options include the appearance of SAS output, the handling of some files that are used by SAS, the use of system variables, the processing of observations in SAS data sets, features of SAS initialization, and the way SAS interacts with your host operating environment.

**SAS table**

another term for SAS data set. See SAS data set.

**SAS Workspace Server**

a SAS IOM server that is launched in order to fulfill client requests for IOM workspaces. See also IOM server, workspace.

**SAS/ACCESS software**

a group of software interfaces, each of which makes data from a particular external database management system (DBMS) directly available to SAS, as well as making SAS data directly available to the DBMS.

**SAS/CONNECT server**

a server that provides SAS/CONNECT services to a client. When SAS 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 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 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 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 package**

a container for content that was produced by a SAS program or by a third-party application for immediate use, and that is not saved. After the client program disconnects from the server, the transient package disappears. See also permanent package.

**transient result package**

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.



# Your Turn

---

If you have comments or suggestions about *SAS 9.1.3 Intelligence Platform: 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  
E-mail: [yourturn@sas.com](mailto:yourturn@sas.com)

For suggestions about the software, please return the photocopy to

SAS Institute Inc.  
Technical Support Division  
SAS Campus Drive  
Cary, NC 27513  
E-mail: [suggest@sas.com](mailto:suggest@sas.com)