



THE
POWER
TO KNOW.



SAS[®] 9.1.3 Intelligence Platform

System Administration Guide

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

SAS® 9.1.3 Intelligence Platform: System Administration Guide

Copyright © 2006, SAS Institute Inc., Cary, NC, USA

ISBN-13: 978-1-59994-285-8

ISBN-10: 1-59994-285-2

All rights reserved. Produced in the United States of America.

For a hard-copy book: 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.

For a Web download or e-book: Your use of this publication shall be governed by the terms established by the vendor at the time you acquire this publication.

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, November 2006

2nd printing, February 2007

3rd printing, April 2007

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.

Other brand and product names are registered trademarks or trademarks of their respective companies.

Contents

<i>What's New</i>	<i>ix</i>
Overview	ix
Documentation Enhancements	ix

PART 1 Before You Begin 1

Chapter 1 △ Introduction to the SAS Intelligence Platform: System Administration Guide 3

About the SAS Intelligence Platform: System Administration Guide	3
Prerequisites for Using the SAS Intelligence Platform: System Administration Guide	4
Accessibility Features in the SAS Intelligence Platform Products	4

Chapter 2 △ Understanding the State of Your System 5

Overview of the State of Your System	5
Overview of the SAS Metadata Server and Its Initial Configuration	6
Overview of SAS Application Server Components and Their Initial Configuration	8
Overview of Ports	9
Overview of the Configuration Directory Structure	9
Overview of Configuration Files	12
Overview of Initial Users and Groups	14

Chapter 3 △ What to Do Next: Summary of System Administration Tools and Tasks 21

Overview of SAS Management Console	21
Modifying Your Initial Configuration to Optimize Performance	23
Making Other Changes to Your Initial Configuration	24
Performing Ongoing System Administration Tasks	25

PART 2 System Operation 27

Chapter 4 △ Starting, Stopping, and Pausing Servers 29

About Starting, Stopping, and Pausing Servers	29
Starting and Stopping SAS Servers on Windows	31
Starting and Stopping SAS Servers on UNIX	32
Starting and Stopping SAS Servers on z/OS	33
Use SAS Management Console to Stop the SAS Metadata Server	33
Use SAS Management Console to Pause or Resume the SAS Metadata Server	33
Use SAS Management Console to Close Active Sessions on a SAS OLAP Server and to Stop the Server	34
Starting Servlet Containers and J2EE Application Servers	35

Chapter 5 △ Checking the Status of Servers 37

Checking the Access State of a SAS Metadata Repository	37
--	----

Checking the Access State of the Repository Manager	39
Checking the Access State of a SAS Metadata Server	40
Checking the Status of the Object Spawner, SAS Workspace Servers, and SAS Stored Process Servers	41
Check the Status of a SAS OLAP Server	42

PART 3 Backing Up and Restoring the SAS Metadata Server 43

Chapter 6 △ Planning SAS Metadata Server Backups and Restores 45

About %OMABAKUP	45
Why Use %OMABAKUP?	46
Best Practices for Using %OMABAKUP to Perform Backups	46
Best Practices for Using %OMABAKUP to Perform Restores	48
Using PROC COMPARE To Compare Backup Datasets to Production Datasets	50

Chapter 7 △ Using the %OMABAKUP Macro to Perform Backups and Restores 51

Requirements for Using %OMABAKUP	51
Using %OMABAKUP on Windows or UNIX	52
Using %OMABAKUP on z/OS	56
How the %OMABAKUP Backup Process Works	57
Warning Messages	58
%OMABAKUP Macro: Reference	59

Chapter 8 △ Manually Restoring the SAS Metadata Server 63

Manually Restoring an Unresponsive SAS Metadata Server	63
Manually Restore a Running SAS Metadata Server	64

PART 4 System Monitoring and Logging 67

Chapter 9 △ Administering the SAS Metadata Server Log and Audit Trail 69

Administering the SAS Metadata Server Log	69
Administering the SAS Metadata Server Audit Trail	74

Chapter 10 △ Using the ARM_OMA Subsystem to Obtain Raw Performance Data for the SAS Metadata Server 77

About the ARM_OMA Subsystem	77
Enabling ARM Logging	78
Understanding the Records Written by the ARM_OMA Subsystem	79
ARM_OMA Log Examples	82
Post-processing the ARM_OMA Log	88

Chapter 11 △ Administering Logging for the SAS Object Spawner, SAS Workspace Server, SAS Stored Process Server, and SAS/CONNECT Server 89

Enabling Logging for the SAS Object Spawner	89
Enabling Logging for SAS Workspace Servers	92
Administering Logging for SAS Stored Process Servers	93
Enabling Increased Logging for the SAS/CONNECT Spawner	94

Chapter 12 △ SAS OLAP Server Monitoring and Logging	97
Overview of SAS OLAP Server Monitoring and Logging	97
Administering the OLAP Server Log	97
Using ARM to Monitor SAS OLAP Server Performance	98
Using the SAS OLAP Server Monitor	99

PART 5 **Optimizing the Performance of the SAS Metadata Server** **101**

Chapter 13 △ Using Configuration Settings to Optimize Metadata Server Performance	103
Configuring the Number of Threads Used by the Metadata Server	103
Using Workunit Journaling to Optimize Performance on Multiprocessor Hosts	104
Avoiding Changes to the Repository That Adversely Affect Performance	107
Chapter 14 △ Managing Memory for Performance	109
Ensuring Adequate Memory	109
Using %OMARUNAN (or the RUNANALYSIS Option of %OMABAKUP) to Optimize Memory Usage	113
%OMARUNAN Autocall Macro: Reference	115
Recovering Memory on the SAS Metadata Server	115

PART 6 **Managing SAS Metadata Repositories** **117**

Chapter 15 △ Creating, Moving, Copying, and Deleting Metadata Repositories	119
Creating Repositories	119
Copying, Promoting, Importing, and Exporting Your Metadata	120
Moving a Repository to a New Location on the Same SAS Metadata Server	122
Porting a Repository to a Different Host Environment	123
Deleting a Metadata Repository	125
Chapter 16 △ Creating a Metadata Repository in a Database Management System	127
About Creating a Metadata Repository in a Database Management System	127
Prerequisites for Setting Up a DBMS Repository	127
Create the Repository Manager and a Metadata Repository in an Oracle or a DB2 Database	128
<RPOSMGR> Attributes Used to Connect to an Oracle Database	130
<RPOSMGR> Attributes Used to Connect to a DB2 Database	131
Chapter 17 △ Using the Promotion and Replication Wizards to Copy an Entire Metadata Repository	135
When to Use the Promotion and Replication Wizards	136
Features and Capabilities of the Promotion and Replication Wizards	137
Special Considerations for Using the Promotion and Replication Wizards	137
Overview of the Promotion or Replication Process	139
Promoting Dependent Repositories	139
Preparing Your Site for a Promotion Job	140
Creating and Running a Promotion Job	152

What Happens When You Run a Promotion Job?	163
Moving Associated BI Content Following a Promotion Job	165
Performing a Demotion	167
Chapter 18 △ Promoting Individual BI Objects	169
Overview of Promoting Individual BI Objects	169
How to Access the Export and Import Wizards	174
Special Considerations for Promoting Individual BI Objects	175
Preparing to Use the Export and Import Wizards	176
Using the Export and Import Wizards: Example Usage Scenario	179
Troubleshooting the Export and Import Wizards	188

PART 7 **Scheduling Jobs and Reports** **191**

Chapter 19 △ SAS Scheduling Overview	193
About SAS Scheduling	193
About Schedule Manager in SAS Management Console	193
About Scheduling Metadata	194
About Platform Product Suite for SAS	196
About Configuring and Using SAS Scheduling	197
Chapter 20 △ Installing and Configuring Platform Product Suite for SAS	199
Summary of Platform Product Suite Installation and Configuration Tasks	199
Step 1: Create the Necessary Users, Groups, and Logins	199
Step 2: Install Platform Product Suite for SAS	200
Step 3: Give the LSF Administrative User Full Control of Scheduling Folders	200
Step 4: Define the Process Manager Server	201
Resolving Issues Related to Scheduling With Platform Product Suite for SAS	201
Further Resources	204
Chapter 21 △ Scheduling Jobs from SAS Data Integration Studio	205
Overview of Scheduling for SAS Data Integration Studio	205
Enabling the Scheduling of SAS Data Integration Studio Jobs	206
Deploying Data Integration Studio Jobs for Scheduling	208
Scheduling Deployed SAS Data Integration Studio Jobs	208
Troubleshooting the Scheduling of SAS Data Integration Studio Jobs	209
Chapter 22 △ Using an Operating System Scheduling Server	211
Capabilities and Limitations of Operating System Scheduling Servers	211
Summary of Setup Tasks	212
Step 1: Assign Local Administration Rights to User (Windows Only)	212
Step 2: Define an Operating System Scheduling Server	212
Step 3: Assign a Spawner to the Operating System Scheduling Server	213
Using Operating System Scheduling on Windows Systems	213
Using Operating System Scheduling on UNIX Systems	215
Using Operating System Scheduling on z/OS Systems	216

Chapter 23 △ **Enabling the Scheduling of Reports** 219

How Report Scheduling Works 219

Overview of Tasks for Enabling Report Scheduling 220

Step 1: Create the Necessary Users, Groups, and Logins 220

Step 2: Define a SAS Java Batch Server 222

Step 3: Update the SAS Web Report Studio Configuration File 223

Step 4: Update the LSF Password File (Windows Only) 224

Step 5: Create a Log Directory for the Output Generation Tool, and Apply Appropriate Directory Permissions 224

Step 6: Edit the Initialization File for the Output Generation Tool (Windows Only) 225

Step 7: Verify the Configuration File for the Output Generation Tool 225

Step 8: Define Archive Paths for Publishing Reports to Channels 226

Step 9: Test the Configuration 227

Troubleshooting the Configuration 227

Defining a SAS Java Batch Server to Run SAS Web Report Studio 2.1 and 3.1 Reports 228

PART 8 **Troubleshooting** 231**Chapter 24** △ **Troubleshooting the SAS Metadata Server** 233

Overview of Troubleshooting the SAS Metadata Server 233

How to Respond When the SAS Metadata Server Will Not Start 233

How to Respond When a Client Cannot Connect to SAS Metadata Server 235

PART 9 **Appendixes** 237**Appendix 1** △ **Default Operating System Users and Groups** 239

Users 239

Appendix 2 △ **Default SAS Ports** 243

Overview 243

Appendix 3 △ **Default Log Locations** 247

Default Locations for Server Logs 247

Default Locations for Application Logs 248

Appendix 4 △ **Configuration Files** 251

Location of Configuration Files for SAS Servers 251

Using adminUsers.txt to Configure New Privileged Users of the SAS Metadata Server 253

Using trustedUsers.txt to Configure New Trusted Users 253

Reference Information for omaconfig.xml 254

Appendix 5 △ **Options for the Metadata Server Invocation Command** 259

Recommended Parameters 259

Optional Parameters 262

Appendix 6	△ Configuration Directories	265
Overview of the SAS Configuration Environment		265
Basic Concepts of the SAS Configuration Environment		265
The Main Directory Structure		267
The Lev1 Directory Structure		268
Default Directory Permissions		273
Appendix 7	△ Managing an Environment	275
Overview of Managing an Environment		275
Customizing the Properties of a New Environment		275
Re-Creating an Existing Environment		277
Adding to an Environment		278
Uninstalling an Environment		279
Appendix 8	△ Recommended Reading	281
Recommended Reading		281
Glossary		283
Index		295

What's New

Overview

SAS creates and delivers enterprise intelligence through the SAS Intelligence Platform, which provides the end-to-end infrastructure that is necessary for exploring, analyzing, optimizing, reporting, and understanding your data.

The *SAS Intelligence Platform: System Administration Guide* is a new document that covers administrative tasks that you need to perform on an ongoing basis after the initial installation and configuration of the SAS Intelligence Platform. The document covers tasks that apply to the SAS Intelligence Platform as a whole. It is one of several new documents that are being published to replace the *SAS Intelligence Platform: Administration Guide*.

Documentation Enhancements

This document contains the following new information:

- a summary of system administration tasks
- instructions for starting and stopping servers and for checking the status of servers
- instructions for setting up and using an operating system scheduling server
- instructions for setting logging options for SAS servers
- reference information about the location of configuration files and log files

This document contains the following enhancements:

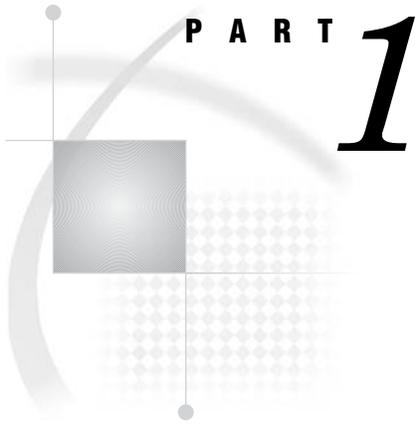
- The information about performing SAS Metadata Server backups has been reorganized and enhanced.
- The documentation about scheduling has been reorganized and enhanced.

This document contains the following information that was previously in the *SAS Intelligence Platform: Administration Guide*:

- a description of the default system configuration that is put in place by the SAS Configuration Wizard
- administrative tasks for SAS Metadata Servers and SAS Metadata Repositories

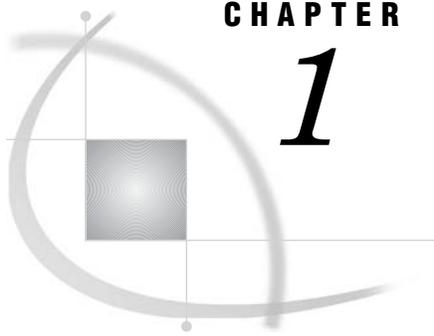
- instructions for setting up scheduling for jobs and reports
- reference information regarding initial users, ports, and the structure and content of configuration directories

This document also contains information about starting and monitoring SAS OLAP Servers that was previously in the *SAS OLAP Server: Administrator's Guide*.



Before You Begin

- Chapter 1* **Introduction to the SAS Intelligence Platform: System Administration Guide 3**
- Chapter 2* **Understanding the State of Your System 5**
- Chapter 3* **What to Do Next: Summary of System Administration Tools and Tasks 21**



CHAPTER

1

Introduction to the SAS Intelligence Platform: System Administration Guide

About the SAS Intelligence Platform: System Administration Guide 3

Prerequisites for Using the SAS Intelligence Platform: System Administration Guide 4

Accessibility Features in the SAS Intelligence Platform Products 4

About the SAS Intelligence Platform: System Administration Guide

This guide covers administrative tasks that you will need to perform on an ongoing basis after the initial installation and configuration of the SAS Intelligence Platform.

The guide covers tasks that apply to the SAS Intelligence Platform as a whole, including the following:

- starting and stopping servers
- checking the status of servers
- setting server logging options
- administering the SAS Metadata Server, including the following tasks:
 - performing backups and restores
 - monitoring the server
 - optimizing the server's performance
 - troubleshooting the server
- administering SAS Metadata Repositories, including the following tasks:
 - creating repositories
 - importing, exporting, and copying metadata
 - promoting and replicating metadata
- scheduling jobs and reports

The appendixes of this guide provide reference information about log files, configuration files and directories, required user accounts, and default ports.

This guide is one of several documents that SAS publishes to support administration of the SAS Intelligence Platform. Additional guides are available to cover the following administration topics:

- SAS application server administration
- data administration
- security administration
- Web application administration
- desktop application administration

To access these documents, go to <http://support.sas.com/913administration>.

Prerequisites for Using the SAS Intelligence Platform: System Administration Guide

Before using this guide, you should be familiar with the concepts and terminology that are introduced in the *SAS Intelligence Platform: Overview*.

In addition, you should have completed all of the installation tasks that are described in the *SAS Intelligence Platform: Installation Guide*.

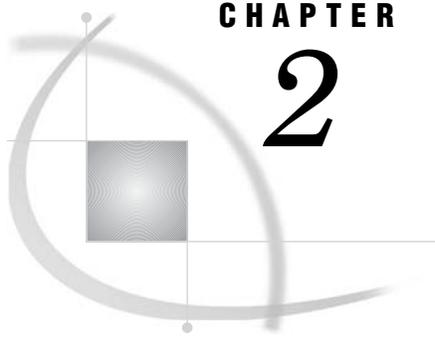
To access these documents, go to <http://support.sas.com/913administration>.

Accessibility Features in the SAS Intelligence Platform Products

For information about accessibility for any of the products mentioned in this book, see the documentation for that product.

For Windows-based assistive technologies to interoperate with SAS Java-based desktop applications, you need to download and install the Java Access Bridge (JAB) from Sun Microsystems. For instructions, see “Accessibility Features in the SAS Desktop Clients” in the *SAS Intelligence Platform: Desktop Application Administration Guide*.

If you have questions or concerns about the accessibility of SAS products, send email to accessibility@sas.com.



CHAPTER

2

Understanding the State of Your System

<i>Overview of the State of Your System</i>	5
<i>Overview of the SAS Metadata Server and Its Initial Configuration</i>	6
<i>What Is the SAS Metadata Server?</i>	6
<i>About the SAS Metadata Repositories and the Repository Manager</i>	6
<i>How the SAS Metadata Server Works</i>	7
<i>The Initial Configuration of the SAS Metadata Server</i>	7
<i>Overview of SAS Application Server Components and Their Initial Configuration</i>	8
<i>Overview of Ports</i>	9
<i>Overview of the Configuration Directory Structure</i>	9
<i>About the Configuration Directory</i>	9
<i>About the Lev1 Subdirectory</i>	11
<i>About the SASMain or SASApp Subdirectory</i>	11
<i>About the web Subdirectory</i>	12
<i>Overview of Configuration Files</i>	12
<i>About the SAS Metadata Server Configuration File (omaconfig.xml)</i>	12
<i>About Other SAS Server Configuration Files</i>	13
<i>About the Security-Related Configuration Files</i>	13
<i>About Configuration Files for Client Applications</i>	14
<i>Overview of Initial Users and Groups</i>	14
<i>About the Initial Operating System User Accounts</i>	14
<i>About the Initial Operating System Groups</i>	16
<i>About Administrative Users</i>	17
<i>About Unrestricted Users and Trusted Users</i>	17
<i>About Required User and Group Metadata Objects</i>	18

Overview of the State of Your System

Before you begin administering the SAS Intelligence Platform, you should be aware of the following components that have been put in place during the installation process:

- the SAS Metadata Server, which serves metadata from one or more SAS Metadata Repositories to all SAS Intelligence Platform client applications
- the SAS Application Server components, including SAS Workspace Servers and SAS Stored Process Servers, that might have been installed and configured in your environment to run SAS processes
- the ports that are used by each server to listen for incoming requests
- the configuration directory that stores configuration files, scripts, and special-purpose SAS data sets on each server machine and middle-tier machine
- the main configuration files that control the operation of the system

- the initial SAS users and groups that have been defined, both on your host operating systems and in the SAS Metadata Repository

Read the following topics to learn about the starting configuration for these components.

Overview of the SAS Metadata Server and Its Initial Configuration

What Is the SAS Metadata Server?

The SAS Metadata Server is a multi-user server that serves metadata from one or more SAS Metadata Repositories to all of the SAS Intelligence Platform client applications in your environment.

The functionality of the SAS Metadata Server is provided through the SAS Open Metadata Architecture, which is a general-purpose metadata management facility that provides common metadata services to applications. One metadata server supports all of the applications in your environment and can support hundreds of concurrent users. This architecture enables the following:

- the exchange of metadata between applications, so that applications can work together more easily.
- centralized management of metadata resources. Because there is a common framework for creating, accessing, and updating metadata, it is easier to manage the applications that rely on this metadata.

About the SAS Metadata Repositories and the Repository Manager

The SAS Metadata Server is a SAS process that enables clients to read metadata from and write metadata to one or more SAS Metadata Repositories. A *repository* is a physical location in which a collection of related metadata objects is stored.

The SAS Metadata Server supports three types of metadata repositories:

foundation

a basic stand-alone repository that does not depend on other repositories. Every metadata server has a foundation metadata repository. A metadata server can have only one foundation metadata repository.

project

a repository that is dependent on another repository and that is used to test and evaluate development changes before they are applied to the parent repository. A project repository uses resource templates and user definitions from its parent repository. Each project repository has a specified owner (a user or a group) that is allowed to make changes to the repository contents.

A project repository cannot be dependent on another project repository, and can only be dependent on foundation or custom repositories that are change managed.

custom

a repository that is dependent on a foundation repository or another custom repository. For example, you could create a custom repository that contains the metadata for a set of tables and columns that are maintained by a particular application.

Note: Custom repositories are supported only by SAS Data Integration Studio, SAS OLAP Cube Studio, and solutions that use SAS Solutions Services (for example, SAS

Financial Management and SAS Strategic Performance Management). Project repositories are supported only by SAS Data Integration Studio. △

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

How the SAS Metadata Server Works

SAS Intelligence Platform applications connect to the SAS Metadata Server and issue SAS Open Metadata Interface method calls that access SAS Metadata Repositories.

To enable high-speed access by clients, the SAS Metadata Server is an “in-memory” server. As clients submit queries and updates, the requested records are read from repository data sets into the server’s memory, where they remain until the server is paused and resumed or until the server is stopped.

When metadata updates are requested, the server’s default behavior is to update both the in-memory copy and the permanent copy of the metadata before informing clients that the request was successfully completed. This ensures that both copies contain the same information. You can change this default behavior if necessary to optimize memory usage.

The Initial Configuration of the SAS Metadata Server

The SAS Configuration Wizard, which was used to install the SAS Intelligence Platform in your environment, provides the following initial configuration for your metadata server:

- Your metadata server contains a foundation metadata repository, which is a basic stand-alone repository that does not depend on other repositories. Your foundation repository contains the following items:
 - metadata definitions for initial users and groups
 - metadata definitions for your servers
 - default authorization settings
 - a full set of resource templates that can be used to create additional metadata objects

The data sets that make up the foundation repository are located in the following path:

SAS-config-dir\Lev1\SASMain\MetadataServer\MetadataRepositories

In this path, *SAS-config-dir* refers to the SAS configuration directory (see “About the Configuration Directory” on page 9).

- The maximum number of threads that the metadata server can use has been set to a value representing the number of processors on the metadata server host machine.
- Workunit journaling has not been enabled. Workunit journaling is a feature that improves the performance of the metadata server on multi-processor hosts. When this feature is enabled, metadata updates that are requested by client applications are recorded immediately in a workunit journal file—enabling clients to continue their work—and written to disk later.
- Two special users of the metadata have been set up: the SAS Administrator and the SAS Trusted User.

- The metadata server has been configured to write daily logs to the directory `SAS-config-dir\Lev1\SASMain\MetadataServer\logs`. By default, the following information is written to the log:
 - SAS Metadata Server start and stop information, including the invoking user ID, SAS long version number, SAS Metadata Model version, the directory where the metadata server was started, and configuration file options
 - user connection and disconnection events
 - repository creation, deletion, open, close, pause, resume, and refresh events
 - errors, including task/thread exceptions, memory allocation errors, I/O errors, application logic errors, authentication errors, and authorization failures
- The metadata server has been configured to authenticate users with the host operating system's authentication mechanism.
- The metadata server has been configured to use SAS proprietary encryption to encrypt user credentials that are sent over the network.

For information about changing this configuration, see the following sections:

- “Modifying Your Initial Configuration to Optimize Performance” on page 23
- “Making Other Changes to Your Initial Configuration” on page 24

Overview of SAS Application Server Components and Their Initial Configuration

During the installation process, the SAS Configuration Wizard created metadata definitions and configuration files for a set of SAS servers that are referred to collectively as a SAS Application Server. The name of the SAS Application Server is assigned as follows:

- The name **SASMain** is assigned on machines that have a SAS Metadata Server in addition to the SAS Application Server.
- The name **SASApp** is assigned on machines that do not have a SAS Metadata Server installed.

An application server can contain several types of components. The most common components are SAS Workspace Servers and SAS Stored Process Servers. An application server can also contain SAS/CONNECT servers, SAS OLAP Servers, and SAS batch servers. By default, these application server components are configured as follows:

- The initial workspace server has the following configuration:
 - The workspace server is configured as a standard workspace server (without workspace pooling or load balancing). With this configuration, each client must establish a connection to a single-user server process, use the server, and then disconnect.
 - Workspace server logging is disabled.
- The initial stored process server has the following configuration:
 - The stored process server is configured to be load balanced. By default, the object spawner balances a workload across three stored process server processes.
 - Logging for stored process servers is enabled by default.
- By default, only minimal logging is enabled for the SAS/CONNECT spawner.

- Logging for SAS OLAP Servers and batch servers is enabled by default.

For information about changing this configuration, see the following sections:

- “Modifying Your Initial Configuration to Optimize Performance” on page 23
- “Making Other Changes to Your Initial Configuration” on page 24

For detailed information about administering your SAS Application Server components, see the *SAS Intelligence Platform: Application Server Administration Guide*.

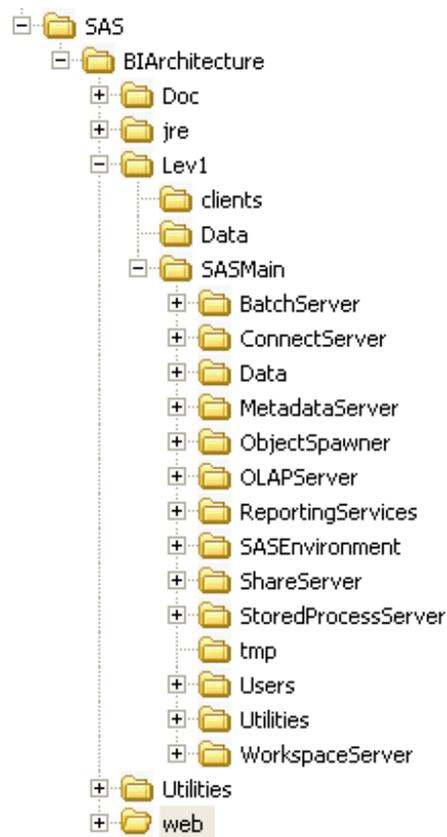
Overview of Ports

The servers in the SAS Intelligence Platform communicate with clients and other servers using TCP/IP. Thus, each server listens on a particular port or ports for incoming requests. The SAS Configuration Wizard configures the servers to use a standard set of ports, unless you tell it to do otherwise. For information about the default ports, see Appendix 2, “Default SAS Ports,” on page 243.

Overview of the Configuration Directory Structure

About the Configuration Directory

Every server-tier and middle-tier host has a configuration directory which is the root of the SAS Intelligence Platform environment. In this directory, the SAS Configuration Wizard builds a directory structure that contains important files for managing your system. Here is an example of the configuration directory structure for a single-machine Windows implementation:



The structure and contents of the directory vary depending on the host operating system, which products are installed, and whether the host machine is a server-tier host or a middle-tier host.

Throughout the SAS Intelligence Platform administration documentation, the variable *SAS-config-dir* is used to designate your configuration directory path. The default paths are as follows:

- on Windows systems:

drive:\SAS\deployment-plan-name

- on UNIX systems:

installer's-home-directory/SAS/deployment-plan-name

- on z/OS systems, the directory specified in the **CONFIG_DIR** environment variable

During the installation process, the SAS Configuration Wizard enables the installer to override the default configuration path with a different path.

On server-tier machines, the configuration directory includes the following:

- a **Doc** subdirectory
- a **jre** subdirectory
- a **Lev1** subdirectory
- an **UninstallerData** subdirectory
- a **Utilities** subdirectory

For a description of all of the subdirectories, see Appendix 6, “Configuration Directories,” on page 265. The following topics provide information about the **Lev1** subdirectory, which is the one you will use most frequently in the administration of the SAS Intelligence Platform.

About the Lev1 Subdirectory

In the configuration directory, **Lev1** is the subdirectory that you will access most frequently for information related to your configuration. **Lev1** identifies information that is at production status.

If you set up separate development and test environments, you can use **Lev2** and **Lev3** as the directory names for their configuration information. **Lev0** is used for administration environments that you set up in order to use the Replication and Promotion Wizard.

Depending on what software is installed on the machine, the **Lev1** directory can contain the following:

- a **Data** subdirectory
- a **SASMain** or **SASApp** subdirectory (see “About the SASMain or SASApp Subdirectory” on page 11)
- a **Utilities** subdirectory
- a **web** directory, which exists only on machines that have middle-tier components installed (see “About the web Subdirectory” on page 12)

About the SASMain or SASApp Subdirectory

The **SASMain** or **SASApp** subdirectory contains information related to your SAS servers. The name of the directory depends on what is installed on the machine, as follows:

- The **SASMain** subdirectory exists on machines that have a SAS Metadata Server (and perhaps other SAS servers) installed.
 - Note:* Middle-tier machines also have a **SASMain** subdirectory, but it is empty. Δ
- The **SASApp** subdirectory exists on machines that do not have a SAS Metadata Server installed, but which do have other SAS servers.

If additional application servers are created, then the directory name will correspond to the application server name.

The SASMain or SASApp subdirectory contains the following items on server-tier machines:

- the file **sasv9.cfg**, which is the SAS configuration file
- the file **omaconfig.xml**, which is the SAS Metadata Server configuration file (present only if the machine has a SAS Metadata Server installed)
- a **MetadataServer** subdirectory, if the machine has a SAS Metadata Server installed. This subdirectory contains the following items:
 - three security-related configuration files called **adminUsers.txt**, **trustedUsers.txt**, and **trustedPeers.xml**
 - a directory called **MetadataRepositories**, which is the location where your foundation metadata repository resides
- some or all of the following subdirectories, depending on which SAS servers have been configured. These subdirectories contain start-up scripts and configuration files for your SAS servers, as well as **logs** directories for the servers’ log files.
 - BatchServer**
 - ConnectServer**
 - ObjectSpawner**
 - OLAPServer**

- ShareServer**
- StoredProcessServer**
- WorkspaceServer**
- a **Data** subdirectory
- a **logs** subdirectory
- a **SASEnvironment** subdirectory
- a **tmp** subdirectory
- a **Users** subdirectory
- a **Utilities** subdirectory

For information about these directories, see Appendix 6, “Configuration Directories,” on page 265.

For more information about the configuration files in **SASMain** or **SASApp**, see “Overview of Configuration Files” on page 12.

About the web Subdirectory

On middle-tier machines (that is, machines with installed Web components), the **Lev1** directory contains a subdirectory called **web** which contains these items:

- a servlet container start-up script called **startServletContainer**
 - Note:* Before using this start-up script, refer to important information in “Starting Servlet Containers and J2EE Application Servers” on page 35. Δ
- a **Deployments** subdirectory that contains information related to deploying Web applications, including policy files and service definition files
- a **Utilities** subdirectory for any utilities related to installed Web applications
- a **webapps** subdirectory that contains the Web application archive (WAR) files for your Web applications such as the SAS Information Delivery Portal

For additional information about the contents of this subdirectory, see Appendix 6, “Configuration Directories,” on page 265. For information about administering middle-tier applications and components, see the *SAS Intelligence Platform: Web Application Administration Guide*.

Overview of Configuration Files

About the SAS Metadata Server Configuration File (omaconfig.xml)

On the machine that hosts the SAS Metadata Server, the **SASMain** directory contains a file called **omaconfig.xml**, which contains SAS Metadata Server settings such as the following:

- values for the libref and path of the metadata server’s repository manager
- the engine used to create the repository manager
- the names and locations of the **adminUsers.txt** and **trustedUsers.txt** files, which contain information about special users of the metadata server
- the number of threads that are allowed to run concurrently on the SAS Metadata Server (contingent on the number of available processors)
- workunit journaling options

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>
```

For details about the **omaconfig.xml** file, see “Reference Information for omaconfig.xml” on page 254.

About Other SAS Server Configuration Files

On server-tier machines, the **SASMain** or **SASApp** directory contains the following:

- configuration files for specific servers. These files are contained in subdirectories that correspond to each server. For example:
 - the **MetadataServer** subdirectory contains a configuration file called **sasv9_MetadataServer.cfg**
 - the **StoredProcessServer** subdirectory contains a configuration file called **sasv9_StoredProcSrv.cfg**
 - the **OlapServer** subdirectory contains a file called **sasv9_OLAPServer.cfg**
- a file called **sasv9.cfg**, which contains SAS system options that are shared by the SAS Metadata Server, SAS Workspace Servers, SAS OLAP Servers, and SAS Stored Process Servers.

To incorporate these shared options, the configuration files for the metadata server, OLAP servers, and stored process servers call **sasv9.cfg**. In turn, **sasv9.cfg** calls a configuration file that is located in the SAS installation directory.

By default, the SAS Workspace Server does not have its own configuration file. Instead, it directly uses the options contained in **sasv9.cfg**

For a list of server configuration files, see “Location of Configuration Files for SAS Servers” on page 251. For detailed information about configuring SAS servers other than the SAS Metadata Server, see the *SAS Intelligence Platform: Application Server Administration Guide*.

About the Security-Related Configuration Files

On the machine that hosts the SAS Metadata Server, the **MetadataServer** directory under **SASMain** contains three configuration files that affect security:

adminUsers.txt

defines your system’s *administrative users* and *unrestricted users*. The configuration wizard creates one unrestricted user called the SAS Administrator (sasadm). In the file, this user name is preceded by an asterisk. For information about these users, see “Overview of Initial Users and Groups” on page 14.

trustedUsers.txt

defines the system’s *trusted users*. The configuration wizard creates one trusted user called the SAS Trusted User (sastrust). For information about this user, see “Overview of Initial Users and Groups” on page 14.

trustedPeers.xml

can be used to specify a list of trusted domains for peer servers (or sessions) that run on the server tier and are connecting from environments other than Windows. You might need to update this file if your metadata server is authenticating clients against an alternative authentication provider.

Note: In most cases, it is not necessary to edit this file. By default, the metadata server is configured to trust connecting workspace and stored process servers as peers. \triangle

About Configuration Files for Client Applications

Configuration files for client applications are described in the following documents:

- \square *SAS Intelligence Platform: Web Application Administration Guide*
- \square *SAS Intelligence Platform: Desktop Application Administration Guide*

Overview of Initial Users and Groups

About the Initial Operating System User Accounts

When your system was installed, a number of operating system user accounts were created. The SAS users that might have been created, and their recommended user names, are as follows:

SAS Administrator (sasadm)

An unrestricted account that has full access to all metadata (other than passwords, which this account can overwrite but cannot read) and can administer all SAS servers. Use this account only with SAS Management Console, and use it only to perform tasks that require unrestricted privileges (such as resetting a password for another user). Do not use this account to define a database library or to perform other tasks that involve reading passwords from the metadata. For details about this user's privileges, see "About Unrestricted Users and Trusted Users" on page 17.

SAS Trusted User (sastrust)

A trusted account that can impersonate other users on connections to the metadata server. Impersonation functionality is used by the SAS OLAP Server, report scheduling, and Web applications that are using Web authentication. In the default configuration, this account is used by the object spawner to read server definitions. In the documented workspace pooling configuration, this account functions as the pool administrator. For details about this user's privileges, see "About Unrestricted Users and Trusted Users" on page 17,

SAS Demo User (sasdemo)

A regular account that can serve as a generic end-user account to test any of the SAS client applications. This account is not required for configuration and can be removed safely.

SAS General Server User (sassrv)

The process owner for stored process servers. In addition, both stored process servers and SAS/SHARE servers use this account when they communicate with the metadata server.

SAS Web Administrator (saswbadm)

A service account that SAS Information Delivery Portal and SAS Web Report Studio use to perform administrative functions.

SAS Guest (sasguest)

A surrogate account that some Web applications use to define access for users who do not have individual metadata identities. SAS Information Delivery Portal uses this account to define the content of the Public Kiosk. In the recommended configuration, SAS Web Report Studio uses this account as a surrogate user for users who do not have their own metadata identities.

LSF Administrator (lsfadmin)

This user is required if you install Platform Suite for SAS to support scheduling or grid computing. This is the user under which Platform LSF services run. In addition, this user owns the LSF configuration and log files and has permission to perform cluster-wide operations, edit configuration files, and reconfigure a cluster.

LSF User (lsfuser)

This user is required if you are scheduling reports.

SAS Installer (sas)

This user is required only on UNIX and Z/OS systems. The installer uses this account to install and configure SAS software. In addition, this user is the owner of configuration directories and their contents and is the process owner for items such as the metadata server, the OLAP server, and the object spawner.

PostgreSQL User (postgres)

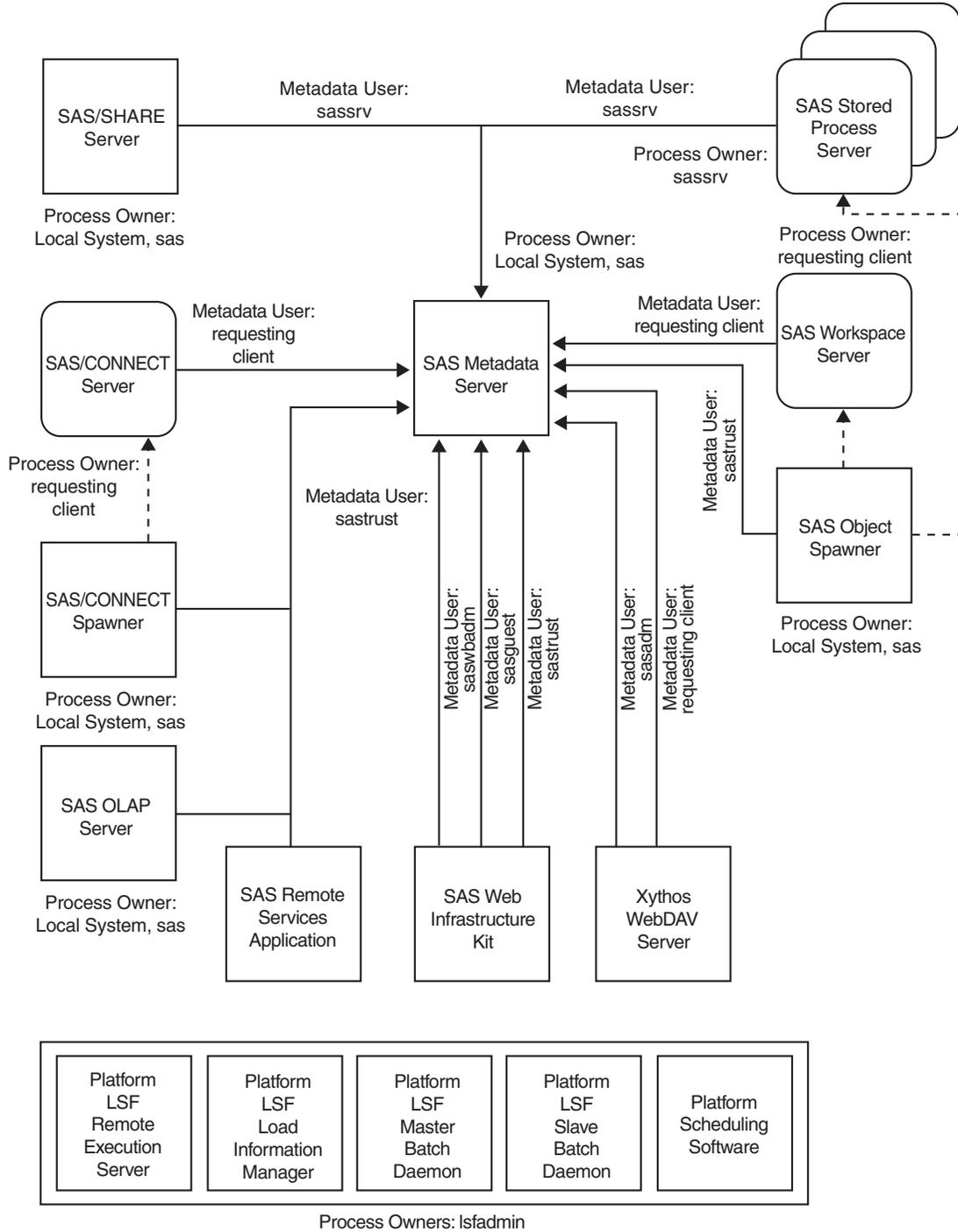
This user is required only if you are using a PostgreSQL database with Xythos WebFile Server.

The exact users that were created depend on the software that was installed and the platforms on which that software was installed.

For more information about these users, and about additional special users that might be set up on your system, see Appendix 1, “Default Operating System Users and Groups,” on page 239.

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

Figure 2.1 Process Owners



About the Initial Operating System Groups

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

SAS Server Users

This group might have been created if you have SAS servers running on Windows systems. This group should have been assigned whatever user rights are required

to start SAS servers on your Windows machines. You can grant these same rights to any users that you add to the system by making those users members of this group.

sas

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

About Administrative Users

A user who has *administrative user* status on the SAS Metadata Server 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

The SAS Configuration Wizard does not configure any *administrative users*. For information about how to designate an *administrative user*, see “User and Group Management” in the *SAS Intelligence Platform: Security Administration Guide*.

About Unrestricted Users and Trusted Users

The SAS Configuration Wizard configures the following highly privileged users of the SAS Metadata Server:

- one *unrestricted user* called the SAS Administrator, whose recommended user ID is sasadm.

A user who has *unrestricted user* status 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.

- one *trusted user* called the SAS Trusted User, whose recommended user ID is sastrust. A *trusted user* can perform the following tasks that are not allowed for “non-trusted” user connections:
 - acquire or release credential handles on behalf of other users
 - use the credential handles to make metadata and authorization requests as another user
 - generate one-time-use passwords to allow clients to connect to the metadata server directly

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

For information about how to designate a new *unrestricted user* or *trusted user*, see “User and Group Management” in the *SAS Intelligence Platform: Security Administration Guide*.

About Required User and Group Metadata Objects

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

Table 2.1 Summary of Metadata Identities

Metadata Identity	User ID*	Password**	Authentication Domain	Group Membership Information
User: SAS Administrator	sasadm	(not stored)		
User: SAS Trusted User	sastrust	(not stored)		member of: SAS System Services group, SAS General Servers group
User: SAS Guest	sasguest	*****	DefaultAuth	
User: SAS Demo User	sasdemo	*****	DefaultAuth	member of: Portal Demos
User: SAS Web Administrator***	saswbadm	*****	DefaultAuth	member of: Portal Admins group, SAS System Services group
Group: SAS System Services				members: SAS Trusted User, SAS Web Administrator
Group: SAS General Servers	sassrv	*****	DefaultAuth	members: SAS Trusted User
Group: Portal Admins***				members: SAS Web Administrator
Group: Portal Demos***				members: SAS Demo User
Group: WRS Administrator****				
Group: WRS Report Author****				
Group: WRS Advanced User****				

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

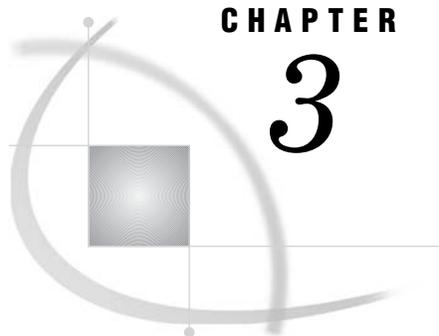
** If you are logged on to SAS Management Console as an *unrestricted user*, you will always see ***** in the password column, even if no password was specified.

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

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

For more information about why the SAS General Servers group must be set up this way, and about the problems you will see if it is not set up this way, see “Overview of

the Initial Load Balancing Setup for Stored Process Servers” in the *SAS Intelligence Platform: Application Server Administration Guide*.



CHAPTER

3

What to Do Next: Summary of System Administration Tools and Tasks

<i>Overview of SAS Management Console</i>	21
<i>About SAS Management Console</i>	21
<i>About SAS Management Console Plug-ins</i>	21
<i>Using the SAS Management Console Plug-Ins</i>	22
<i>Modifying Your Initial Configuration to Optimize Performance</i>	23
<i>Making Other Changes to Your Initial Configuration</i>	24
<i>Performing Ongoing System Administration Tasks</i>	25

Overview of SAS Management Console

About SAS Management Console

As you administer the SAS Intelligence Platform, your primary tool will be SAS Management Console, which is a Java application that provides a single point of control for managing metadata objects that are used throughout the SAS Intelligence Platform.

Whenever the SAS Metadata Server is running, you can use SAS Management Console to connect to the server and view and update the metadata objects that are stored in the server's metadata repositories.

About SAS Management Console Plug-ins

SAS Management Console is a framework in which a variety of *plug-ins* are installed. A plug-in is an application module that is designed to create and maintain metadata for a specific type of resource. For example, the Server Manager plug-in is used to create metadata to define SAS servers.

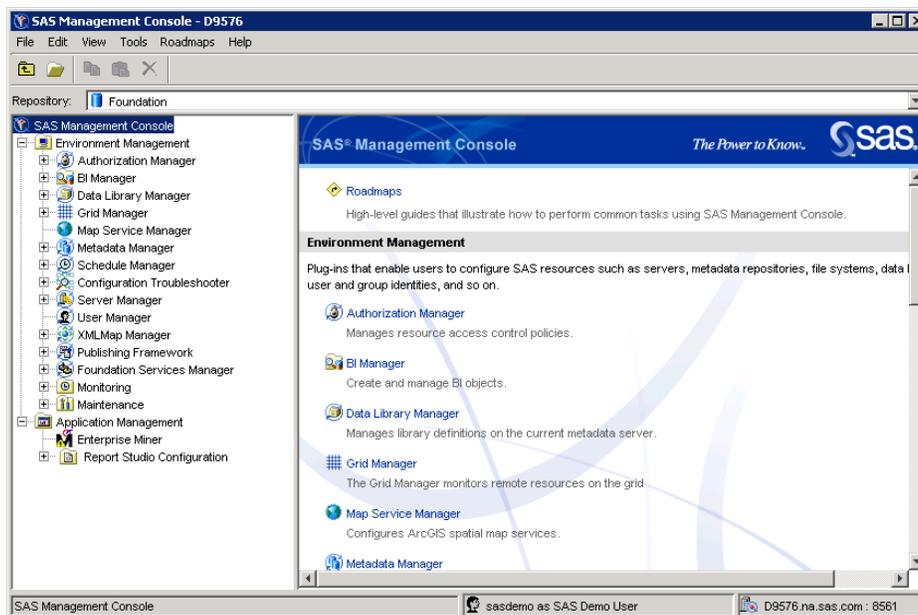
The following list provides the name and purpose of each of the most frequently used SAS Management Console plug-ins:

Authorization Manager	define and maintain access rules to control how users and groups can access metadata definitions
BI Manager	administer, promote, or replicate BI objects including stored processes, information maps, reports, data explorations, and folders
Data Library Manager	create and maintain definitions for SAS libraries and database schemas
License Manager	view information about the SAS licenses installed on a machine, as well as details about the machine

Metadata Manager	define metadata repositories on the active server, manage resource templates, and create jobs to promote or replicate a metadata repository
Publishing Framework	create and maintain definitions for channels, package subscribers, group subscribers, and event subscribers
Schedule Manager	create and maintain schedules for running jobs that are created in SAS Data Integration Studio and SAS Web Report Studio
Server Manager	create and maintain definitions for servers, including SAS Application Servers and their components, database servers, and enterprise servers
User Manager	create and maintain definitions for users and user groups
XMLMap Manager	import and create XMLMaps, which are XML files that describe how the SAS XML LIBNAME engine should interpret XML markup

Using the SAS Management Console Plug-Ins

When you open SAS Management Console and connect to a metadata server, the plug-ins appear in the left pane, as shown in this example:



To use the plug-ins, follow these general steps:

- To display the metadata objects and folders that are associated with a plug-in, click the plus sign next to the plug-in icon. The objects and folders are displayed in a navigation tree.
- To expand a folder or object in the navigation tree, click the plus sign next to it.
- To determine the actions that you can take with an item in the navigation tree (for example, a plug-in, a folder, or an object), either right-click the item, or highlight the item and select the **Actions** menu.
- To view the metadata definition for an object in the navigation tree, right-click the object and select **Properties**.

- To obtain help for a plug-in, select the plug-in (or any object associated with the plug-in) in the navigation tree, and select **Help ► Help on *plug-in name***.

Some of the plug-ins have wizards that assist you in creating new metadata objects. For example, Data Library Manager has a New Library Wizard that helps you create a new library. To access the wizard, you expand the Data Library Manager node, right-click the **SAS Libraries** folder, and select **New Library**.

For detailed information about using SAS Management Console and its plug-ins, see the product Help.

Modifying Your Initial Configuration to Optimize Performance

It might be necessary for you to modify your initial configuration to meet specific requirements in your environment. The following table summarizes some of the configuration changes that you might want to make in order to improve performance.

Table 3.1 Configuration Changes That Might Be Needed to Optimize Performance

Task	Documentation Reference
Adjust the maximum number of threads used by the metadata server.	“Configuring the Number of Threads Used by the Metadata Server” on page 103
Enable workunit journaling to improve performance on a multi-processor host. When this feature is enabled, metadata updates that are requested by client applications are recorded immediately in a workunit journal file—enabling clients to continue their work—and written to disk later.	“Using Workunit Journaling to Optimize Performance on Multiprocessor Hosts” on page 104
Convert a standard workspace server to a pooled server for use with SAS Web Report Studio or the SAS Information Delivery Portal. Using a pooled server is a good idea when 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.	“Convert a Workspace Server to Pooling” in the <i>SAS Intelligence Platform: Application Server Administration Guide</i>
Define a cluster of load-balanced workspace servers in order to balance a load across workspace servers that you have installed on multiple hosts. This type of configuration is most useful when you have a large number of users (such as data integration specialists) using a workspace server for relatively long-running jobs.	“Convert a Workspace Server to Load Balancing” in the <i>SAS Intelligence Platform: Application Server Administration Guide</i>

Task	Documentation Reference
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.	“Configure a Load-Balancing Cluster of Stored Process Servers” in the <i>SAS Intelligence Platform: Application Server Administration Guide</i> .
Configure middle-tier components for better efficiency and performance.	“Best Practices for Configuring Your Middle Tier” in the <i>SAS Intelligence Platform: Web Application Administration Guide</i> .

Making Other Changes to Your Initial Configuration

The following table lists other configuration changes that you might need to make to meet specific requirements in your environment.

Table 3.2 Other Configuration Changes That Might Be Needed

Task	Documentation Reference
Obtain additional log information for troubleshooting.	Chapter 9, “Administering the SAS Metadata Server Log and Audit Trail,” on page 69
	Chapter 10, “Using the ARM_OMA Subsystem to Obtain Raw Performance Data for the SAS Metadata Server,” on page 77
	Chapter 11, “Administering Logging for the SAS Object Spawner, SAS Workspace Server, SAS Stored Process Server, and SAS/CONNECT Server,” on page 89
	Chapter 12, “SAS OLAP Server Monitoring and Logging,” on page 97
Authenticate users through a different authentication provider (instead of using the host operating system’s authentication mechanism).	“Modifications to Support Alternative Authentication Mechanisms” in the <i>SAS Intelligence Platform: Security Administration Guide</i> .
Configure additional special users, especially <i>administrative</i> and <i>unrestricted users</i> .	“User and Group Management” in the <i>SAS Intelligence Platform: Security Administration Guide</i>

Task	Documentation Reference
Encrypt data in addition to user credentials, or use a different encryption algorithm.	“Enable Encryption” in the chapter “Securing a Deployment” in the <i>SAS Intelligence Platform: Security Administration Guide</i>
Store the metadata repository manager tables and repository tables in a third-party relational database. Note: For performance reasons, we strongly recommend that you use the default SAS data sets for the repository manager and for the foundation metadata repository. However, we recognize that your company might require you to store the repository manager tables and repository tables in a third-party relational database. If you are going to make this change, you should make it as soon as possible after installation to minimize the amount of metadata that you need to re-create.	Chapter 16, “Creating a Metadata Repository in a Database Management System,” on page 127

Performing Ongoing System Administration Tasks

The following table lists important system administration tasks that you will need to perform on a frequent, ongoing basis.

Table 3.3 Frequently Performed Ongoing Administration Tasks

Task	Documentation Reference
Start, stop, and pause the servers that are used in the system.	Chapter 4, “Starting, Stopping, and Pausing Servers,” on page 29
Perform regular full backups of the metadata server.	Chapter 6, “Planning SAS Metadata Server Backups and Restores,” on page 45

Task	Documentation Reference
Use system logs or set logging options.	Chapter 9, “Administering the SAS Metadata Server Log and Audit Trail,” on page 69
	Chapter 10, “Using the ARM_OMA Subsystem to Obtain Raw Performance Data for the SAS Metadata Server,” on page 77
	Chapter 11, “Administering Logging for the SAS Object Spawner, SAS Workspace Server, SAS Stored Process Server, and SAS/CONNECT Server,” on page 89
	Chapter 12, “SAS OLAP Server Monitoring and Logging,” on page 97
Administer the scheduling of jobs and reports.	Chapter 19, “SAS Scheduling Overview,” on page 193

The following table lists other ongoing administration tasks that you might need to perform:

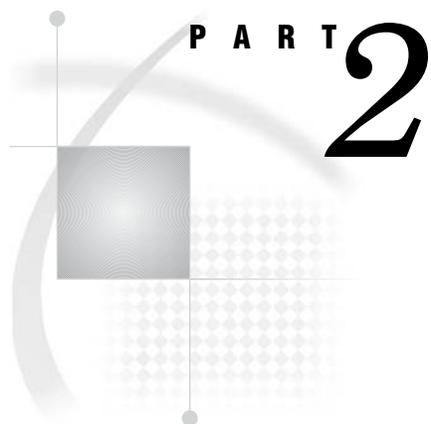
Table 3.4 Other Ongoing Administration Tasks

Task	Documentation Reference
Check the status of a server or of a metadata repository.	Chapter 5, “Checking the Status of Servers,” on page 37
Optimize memory usage on the SAS Metadata Server	“Using %OMARUNAN (or the RUNANALYSIS Option of %OMABAKUP) to Optimize Memory Usage” on page 113
Promote or replicate a metadata repository.	Chapter 17, “Using the Promotion and Replication Wizards to Copy an Entire Metadata Repository,” on page 135
Promote individual BI objects.	Chapter 18, “Promoting Individual BI Objects,” on page 169

Note: Other important administration tasks are described in the following documents:

- SAS Intelligence Platform: Security Administration Guide*
- SAS Intelligence Platform: Data Administration Guide*
- SAS Intelligence Platform: Application Server Administration Guide*
- SAS Intelligence Platform: Desktop Application Administration Guide*
- SAS Intelligence Platform: Web Application Administration Guide*

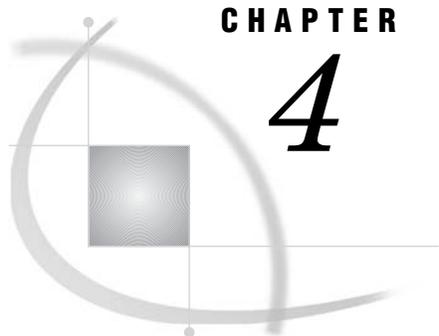
To access these documents, go to <http://support.sas.com/913administration>. Δ



System Operation

Chapter 4. **Starting, Stopping, and Pausing Servers** 29

Chapter 5. **Checking the Status of Servers** 37



CHAPTER

4

Starting, Stopping, and Pausing Servers

<i>About Starting, Stopping, and Pausing Servers</i>	29
<i>Overview of Server Start-up Requirements</i>	29
<i>Starting Servers in the Correct Order</i>	30
<i>Requirement for Stopping a SAS OLAP Server</i>	30
<i>About Server Start-up Parameters</i>	30
<i>Starting and Stopping SAS Servers on Windows</i>	31
<i>Start the SAS Servers If They Are Installed as Windows Services</i>	31
<i>Start and Stop the SAS Servers on Windows by Using Scripts</i>	31
<i>Start and Stop the SAS Servers on Windows by Using the Start Menu</i>	32
<i>Starting and Stopping SAS Servers on UNIX</i>	32
<i>Starting and Stopping SAS Servers on z/OS</i>	33
<i>Use SAS Management Console to Stop the SAS Metadata Server</i>	33
<i>Use SAS Management Console to Pause or Resume the SAS Metadata Server</i>	33
<i>Use SAS Management Console to Close Active Sessions on a SAS OLAP Server and to Stop the Server</i>	34
<i>Starting Servlet Containers and J2EE Application Servers</i>	35
<i>Starting the Apache Tomcat Servlet Container</i>	35
<i>Starting an IBM Websphere Application Server</i>	35
<i>Starting a BEA WebLogic Server</i>	35

About Starting, Stopping, and Pausing Servers

Overview of Server Start-up Requirements

In order for clients to access the SAS Intelligence Platform, the following services must be running on network-accessible machines:

- a SAS Metadata Server
- the SAS Object Spawner, which acts as a listener for SAS Workspace Servers and SAS Stored Process Servers

Depending on which SAS products you have installed, one or more instances of the following additional services might also be required to be running on network-accessible machines:

- SAS OLAP Server
- SAS Services Application
- SAS/CONNECT Spawner
- SAS/SHARE Server
- third-party WebDAV server

- third-party servlet container or J2EE application server

Occasionally, you will need to start, stop, or restart these services so that maintenance or troubleshooting activities can be performed. This chapter provides the following information:

- the order in which servers must be started
- how to start, stop, and restart SAS servers on Windows, UNIX, and z/OS
- how to use SAS Management Console to stop the SAS Metadata Server, or to pause and resume the SAS Metadata Server
- how to use SAS Management Console to close active sessions on a SAS OLAP Server and to stop the SAS OLAP Server
- starting your servlet container or J2EE application server

Starting Servers in the Correct Order

Because of dependencies, it is important to start the SAS servers, the SAS Object Spawner, and the SAS Services Application in the correct order. The dependencies are as follows:

- The SAS Metadata Server must be running before you start the SAS OLAP Server, the SAS Object Spawner, the SAS/CONNECT Spawner, or the SAS/SHARE Server.
- The SAS Metadata Server must be running before you start the SAS Services Application.
- The SAS Services Application and the WebDAV server must be running before you start the servlet container or the J2EE application server.

The following start-up order meets the dependency requirements:

- 1 Start the SAS Metadata Server.
- 2 Start the SAS Object Spawner.
- 3 Start the SAS Services Application.
- 4 Start the WebDAV server.
- 5 Start the servlet container or the J2EE application server.

The SAS OLAP Server, SAS/CONNECT Spawner, and SAS/SHARE Server can be started anytime after the SAS Metadata Server has been started.

Requirement for Stopping a SAS OLAP Server

Before you stop a SAS OLAP Server, it is important to close the active sessions. See “Use SAS Management Console to Close Active Sessions on a SAS OLAP Server and to Stop the Server” on page 34.

About Server Start-up Parameters

The start-up parameters for SAS servers are stored in configuration files. For a list of these configuration files and their locations, see “Location of Configuration Files for SAS Servers” on page 251.

Starting and Stopping SAS Servers on Windows

Start the SAS Servers If They Are Installed as Windows Services

Depending on the options that were selected during the installation process, the following might be installed as Windows services: the SAS/CONNECT Spawner, the SAS Metadata Server, the SAS Object Spawner, the SAS OLAP Server, and the SAS/SHARE Server. This installation option is highly recommended. The SAS Services Application also might be installed as a Windows service.

If these servers have been installed as services, then they start automatically when you restart the machines.

You can also stop, start, or restart these servers without restarting the machine. To do so, use either the scripts or the **Start** menu, as described in the following topics.

Start and Stop the SAS Servers on Windows by Using Scripts

To start or stop the SAS servers on a Windows machine by using scripts, follow these steps:

- 1 Log on as a user who is a member of the host operating system's **Administrators** group.
- 2 Run the script that is appropriate for the server that you are starting, stopping, or restarting. If you use a script whose name specifies only the server name, add the appropriate argument (**start**, **stop**, or **restart**).

The following table provides the location and name of each script.

Note: If your application server has a name other than **SASMain**, replace **SASMain** in the directory path with the appropriate name. For example, your application server might be called **SASApp** if it is installed on a machine other than the metadata server machine. Δ

Table 4.1 Scripts for Starting Servers on Windows Machines

Server	Directory Path	Script Name
SAS Batch Servers	No startup is needed.	
SAS/CONNECT Server	This server uses the SAS/CONNECT Spawner.	
SAS/CONNECT Spawner	<i>SAS-config-dir</i> \Lev1\SASMain\ ConnectServer	ConnectServer.bat StartConnectServer.bat
SAS Metadata Server	<i>SAS-config-dir</i> \Lev1\SASMain\ MetadataServer	MetadataServer.bat pauseServer.bat resumeServer.bat startMetadataServer.bat stopServer.bat
SAS Object Spawner	<i>SAS-config-dir</i> \Lev1\SASMain\ ObjectSpawner	ObjectSpawner.bat StartObjectSpawner.bat
SAS OLAP Server	<i>SAS-config-dir</i> \Lev1\SASMain\ OLAPServer	OLAPServer.bat startOLAPServer.bat

Server	Directory Path	Script Name
SAS Services Application	<i>SAS-config-dir</i> \Lev1\web\ Deployments\RemoteServices\WEB-INF	StartRemoteServices.bat
SAS/SHARE Server	<i>SAS-config-dir</i> \Lev1\SASMain\ ShareServer	ShareServer.bat StartShareServer.bat
SAS Stored Process Server	This server uses the SAS Object Spawner.	
SAS Workspace Server	This server uses the SAS Object Spawner.	

Start and Stop the SAS Servers on Windows by Using the Start Menu

To start or stop the SAS servers by using the Windows Start menu, follow these steps:

- 1 Log on as a user who is a member of the operating system's **Administrators** group.
- 2 Select **Start ▶ Programs ▶ SAS ▶ configuration-name**.
- 3 From the menu that appears, select one of the following options:
 - Start SAS Metadata Server**
 - Restart SAS Metadata Server**
 - Stop SAS Metadata Server**
 - Start SAS OLAP Server**
 - Restart SAS OLAP Server**
 - Stop SAS OLAP Server**
 - Start SAS Object Spawner**
 - Restart SAS Object Spawner**
 - Stop SAS Object Spawner**
 - Start SAS/CONNECT Server**
 - Restart SAS/CONNECT Server**
 - Stop SAS/CONNECT Server**
 - Start SAS/SHARE Server**
 - Restart SAS/SHARE Server**
 - Stop SAS/SHARE Server**
 - Start SAS Services Application**

Starting and Stopping SAS Servers on UNIX

On a UNIX system, follow these steps to start, stop, and restart the SAS servers:

- 1 Log on as the SAS User (for example, sas).
- 2 Change to the directory *SAS-config-dir/Lev1/SASMain/server-type*.
- 3 In that directory, execute the *server-type.sh* script. This script takes one of three parameters: **start**, **stop**, or **restart**.

For example, to stop the metadata server, you would use the command **MetadataServer.sh stop**.

It is 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. Then 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.

For more information, see your UNIX system administrator or the system administration guide for your platform.

Starting and Stopping SAS Servers on z/OS

On a z/OS system, follow these steps to start or stop a server:

- 1 Log on as the SAS User (for example, sas).
- 2 Each server is associated with a different started task. To start a server, use the following console command:

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

To stop a server, use the following console command.

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

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

Messages are written to the system log when the SAS Metadata Server, the SAS OLAP Server, or the SAS/SHARE Server is started or stopped. Automated tools can scan the system log to determine the current operational state of these servers. The log messages are displayed with the prefix **SAH**, as shown in the following examples:

```
+SAH0000I: Server SAS Metadata Server(port-number), Status starting.
+SAH0000I: Server SAS Metadata Server(port-number), Status running.
+SAH0000I: Server SAS Metadata Server(port-number), Status stopping.
+SAH0000I: Server SAS Metadata Server(port-number), Status stopped.
```

Use SAS Management Console to Stop the SAS Metadata Server

To use SAS Management Console to stop the SAS Metadata Server, follow these steps:

- 1 Expand the **Metadata Manager** node, and right-click **Active Server**.
- 2 From the menu that appears, select **Stop**.

You can also use the following procedures, which apply to all of the SAS servers:

- “Starting and Stopping SAS Servers on Windows” on page 31
- “Starting and SAS Stopping SAS Servers on UNIX” on page 32
- “Starting and Stopping SAS Servers on z/OS” on page 33

Note: You cannot start the SAS Metadata Server from SAS Management Console. Δ

Use SAS Management Console to Pause or Resume the SAS Metadata Server

When you use SAS Management Console to pause the SAS Metadata Server, the server does the following:

- retains all active client connections
- temporarily changes the access state of all repositories to offline status
- temporarily stops receiving client requests to read or write metadata. If a client application submits a new request, the client displays a message indicating that the server is paused and that the user should try the request later
- flushes updated metadata from the server's memory and writes the updated metadata to the metadata repository
- closes the SAS data sets that serve as repository containers

To pause the SAS Metadata Server, follow these steps:

- 1 Expand the **Metadata Manager** node, and right-click **Active Server**.
- 2 From the menu that appears, select **Pause**.

The **Resume** action resumes client activity. The repository containers are re-opened as clients request metadata from them. To resume the SAS Metadata Server after it has been paused, follow these steps:

- 1 Expand the **Metadata Manager** node, and right-click **Active Server**.
- 2 From the menu that appears, select **Resume**.

You can also use the following procedures to pause or resume the SAS Metadata Server:

- “Starting and Stopping SAS Servers on Windows” on page 31
- “Starting and Stopping SAS Servers on UNIX” on page 32
- “Starting and Stopping SAS Servers on z/OS” on page 33

Note: When you pause the metadata server from SAS Management Console, it stops accepting new client requests for all repositories. If you want to pause or resume one or more specific repositories, you must use PROC METAOPERATE. For instructions, see the documentation for the METAOPERATE procedure in the *SAS Open Metadata Interface: Reference*. △

Use SAS Management Console to Close Active Sessions on a SAS OLAP Server and to Stop the Server

Before you stop a SAS OLAP Server, you should first stop the active sessions, as follows:

- 1 In SAS Management Console, expand the **Monitoring** node and the **SAS OLAP Server Monitor** node.
- 2 Right-click the name of the logical SAS OLAP Server, and select **Connect**.
- 3 In the dialog box that appears, enter your user name and password and click **OK**.
- 4 Right-click the name of each session that you want to close, and select **Close Session**.

You can now stop the server, as follows:

- 1 Right-click the name of the logical SAS OLAP Server.
- 2 From the menu that appears, select **Stop**. This action stops the processing of queries.

You can also use the following procedures to stop any SAS server:

- “Starting and Stopping SAS Servers on Windows” on page 31
- “Starting and Stopping SAS Servers on UNIX” on page 32
- “Starting and Stopping SAS Servers on z/OS” on page 33

Starting Servlet Containers and J2EE Application Servers

Starting the Apache Tomcat Servlet Container

On Windows systems, follow these steps to start the Apache Tomcat servlet container:

- 1 Make sure that the SAS Services Application has been started.
- 2 Start Tomcat by using one of the following methods:
 - Select **Start ► Programs ► SAS ► *configuration-name* ► Start Tomcat**.
 - Run the following script:

```
SAS-config-dir\Lev1\web\startServletContainer.bat
```

CAUTION:

To activate the correct options, you must use one of the preceding methods. Do not start Tomcat by selecting **Start ► Programs ► Apache Tomcat 4.1 ► Start Tomcat**. △

On UNIX systems, follow these steps to start the Apache Tomcat servlet container:

- 1 Make sure that the SAS Services Application has been started.
- 2 Run the following script:

```
SAS-config-dir/Lev1/web/startServletContainer.sh
```

Note: For information about customizing your start-up options, see “Tuning the Java Virtual Machine” in the *SAS Intelligence Platform: Web Application Administration Guide*. △

Starting an IBM Websphere Application Server

If you are using an IBM WebSphere Application Server on a Windows system, follow these steps to start the server:

- 1 Make sure that the SAS Services Application has been started.
- 2 Start the Websphere Application Server by using one of the following methods:
 - Select **Start ► Programs ► IBM Websphere ► Application Server 5.1 ► Start the Server**.
 - Run the following script:

```
WAS_HOME\bin\startServer.bat server_name
```

If you are using an IBM WebSphere Application Server on a UNIX system, follow these steps to start the server:

- 1 Make sure that the SAS Services Application has been started.
- 2 Run the following script:

```
WAS_HOME/bin/startServer.sh server_name
```

Note: For information about customizing your start-up options, see “Tuning the Java Virtual Machine” in the *SAS Intelligence Platform: Web Application Administration Guide*. △

Starting a BEA WebLogic Server

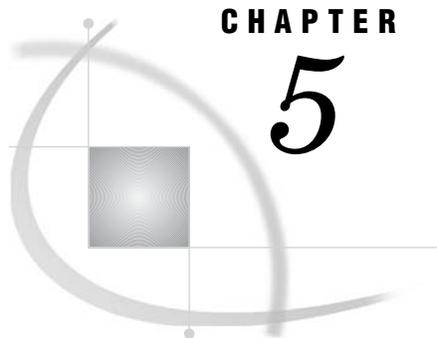
If you are using a BEA WebLogic Server on a Windows system, follow these steps to start the server:

- 1 Make sure that the SAS Services Application has been started.
- 2 If the WebLogic Administration Server is not running, start it by using the following script:
`WebLogic-install-dir\user_projects\domains\domain\startWebLogic.cmd`
- 3 Start the WebLogic Managed Server by using the following script:
`WebLogic-install-dir\user_projects\domains\domain\startManagedWebLogic.cmd`

If you are using an BEA WebLogic Server on a UNIX system, follow these steps to start the server:

- 1 Make sure that the SAS Services Application has been started.
- 2 If the WebLogic Administration Server is not running, start it by using the following script:
`WebLogic-install-dir/user_projects/domains/domain/startWebLogic.sh`
- 3 Start the WebLogic Managed Server by using the following script:
`WebLogic-install-dir/user_projects/domains/domain/startManagedWebLogic.sh`

Note: For information about customizing your start-up options, see “Tuning the Java Virtual Machine” in the *SAS Intelligence Platform: Web Application Administration Guide*. Δ



CHAPTER

5

Checking the Status of Servers

<i>Checking the Access State of a SAS Metadata Repository</i>	37
<i>About SAS Metadata Repository Access States</i>	37
<i>Use PROC METADATA to Check the Access State of a Repository</i>	38
<i>Sample PROC METADATA Code to Check the Access State of a Repository</i>	38
<i>Interpreting PROC METADATA Output to Determine a Repository's Access State</i>	38
<i>Checking the Access State of the Repository Manager</i>	39
<i>About Repository Manager Access States</i>	39
<i>Use PROC METADATA to Check the Access State of the Repository Manager</i>	40
<i>Checking the Access State of a SAS Metadata Server</i>	40
<i>About Metadata Server Access States</i>	40
<i>Use PROC METAOPERATE to Check the Access State of a SAS Metadata Server</i>	40
<i>Interpreting PROC METAOPERATE Output to Determine a Metadata Server's Access State</i>	41
<i>Checking the Status of the Object Spawner, SAS Workspace Servers, and SAS Stored Process Servers</i>	41
<i>Check the Status of a SAS OLAP Server</i>	42

Checking the Access State of a SAS Metadata Repository

About SAS Metadata Repository Access States

A SAS Metadata Repository can exist in one of three access states:

full

Clients can read, write, and update metadata.

read-only

Clients can read metadata but cannot write or update metadata. Through the PROC METAOPERATE command, repositories that were created with full access can be paused to read-only status.

offline

Clients can neither read nor write metadata to the repository. Through the PROC METAOPERATE command, repositories that were created with full access or read-only access can be paused to offline status.

When a repository is created, the intended access mode for the repository is recorded in the Access attribute of the repository's metadata definition. For example, if a repository was configured to be read-only at creation, then its Access attribute has the value **OMS_READONLY**. The default access state for new repositories is full access.

The most common reason for a repository's access state to change is that the internal process of an application needs temporary, exclusive use of a repository. Only a few situations have this requirement, including the following:

- The %OMABAKUP macro pauses all repositories when you perform a full backup or restore, and it pauses individual repositories when you perform a selective backup or restore. See Chapter 6, “Planning SAS Metadata Server Backups and Restores,” on page 45.
- An administrator might pause a repository in order to recover memory on the SAS Metadata Server (see “Recovering Memory on the SAS Metadata Server” on page 115).
- The Replication and Promotion Wizard pauses a repository before moving it to another machine (see Chapter 17, “Using the Promotion and Replication Wizards to Copy an Entire Metadata Repository,” on page 135).

Use PROC METADATA to Check the Access State of a Repository

Sample PROC METADATA Code to Check the Access State of a Repository

If one or more repositories become unavailable on the SAS Metadata Server, it is a good idea to check their access states to see if a process has the repositories paused.

To check the access state of your repositories, use PROC METADATA to issue a GetRepositories method call that sets the OMI_ALL flag, as shown in the following example:

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

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

The GetRepositories method is part of the SAS Open Metadata Interface. The OMI_ALL flag is set by specifying a value of 1 in the **<Flags>** element.

To run this program, you must have ReadMetadata permission to all repositories on the SAS Metadata Server. You do not need to be an administrative user to run the program.

Interpreting PROC METADATA Output to Determine a Repository’s Access State

For each repository that is registered to the SAS Metadata Repository, the GetRepositories method returns the values of the following attributes:

id	specifies the repository’s unique object identifier
Name	specifies the name assigned to the repository
Desc	contains a description of the repository
Access	indicates the access mode that was defined when the repository was created. Valid values are as follows:
OMS_FULL	The repository was defined to have full access.

OMS_READONLY The repository was defined to have read-only access.

A repository cannot be registered in offline mode.

PauseState indicates whether a repository is paused and indicates the repository's temporary access state. Valid values are as follows:

READONLY The repository is paused to a read-only state.

OFFLINE The repository is paused to an offline state.

An empty string means that the repository is not paused.

This information is displayed in the following format in the SAS Output window:

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

In the example, the output means:

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

Checking the Access State of the Repository Manager

About Repository Manager Access States

The repository manager is a library of tables that define the set of repositories that can be accessed by a SAS Metadata Server. PROC METAOPERATE enables you to change the access state of the repository manager in addition to that of repositories. An access state of read-only or offline for the repository manager is typically reflected in the following behavior:

read-only

Clients can read and write metadata in repositories, but they cannot create or delete repositories or change their registrations.

offline

The behavior can vary depending on whether a repository is available in memory. Assume that repositories are unavailable for reading or writing to all users.

CAUTION:

Do not change the access state of the repository manager to offline unless all repositories are taken offline as well. \triangle

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

Use PROC METADATA to Check the Access State of the Repository Manager

To check the access state of the repository manager, use PROC METADATA to issue a GetRepositories method call that sets the OMI_ALL flag. This method returns the access state of the repository manager as well as the access state of each repository. For details, see the following sections:

- “Sample PROC METADATA Code to Check the Access State of a Repository” on page 38
- “Interpreting PROC METADATA Output to Determine a Repository’s Access State” on page 38

Checking the Access State of a SAS Metadata Server

About Metadata Server Access States

When a SAS Metadata Server is in a paused state, this means that client activity in all repositories and the repository manager has been purposely paused by an *administrative user*.

When the repository manager is paused to an offline state, users might or might not be able to write to repositories, depending upon whether or not the repository files were open previously. The repositories and repository manager will remain paused until an *administrative user* issues a Resume action to restore client activity.

Use PROC METAOPERATE to Check the Access State of a SAS Metadata Server

You might want to check the access state of the SAS Metadata Server to determine whether it is running, paused, or stopped. To check the access state of the SAS Metadata Server, use PROC METAOPERATE to issue a STATUS action, as in the following example:

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

```
ACTION=STATUS;
```

```
RUN;
```

Any user who can log on to the SAS Metadata Server can issue a STATUS action.

Interpreting PROC METAOPERATE Output to Determine a Metadata Server's Access State

If the SAS Metadata Server is running normally, information similar to the following will be written to the SAS Output window when you use PROC METAOPERATE to issue a STATUS action:

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

If the SAS Metadata Server is paused, information similar to the following will be written to the SAS Output window when you use PROC METAOPERATE to issue a STATUS action:

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

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

Checking the Status of the Object Spawner, SAS Workspace Servers, and SAS Stored Process Servers

The SAS Object Spawner is a program that listens for incoming client requests for SAS Workspace Servers and SAS Stored Process Servers and then launches instances of these servers to fulfill the request. The object spawner needs to be running in order for these server instances to be launched.

To make sure that the spawner is working and that connections can be successfully made to your workspace servers and stored process servers, follow these steps:

- 1 In SAS Management Console, expand the **Server Manager** node. Then expand the node for the application server (for example, **SASMain**) and the logical workspace server or logical stored process server that you want to test.
- 2 Highlight the server that you want to test.

The connections that are associated with the server appear in the right pane.

- 3 Right-click the connection that you want to test, and select **Test Connection**.

- 4 For workspace servers, if you are logged on to SAS Management Console as an *administrative user*, you will be prompted to enter the credentials of a user that can start a workspace server.
- 5 SAS Management Console attempts to connect to the server.

If the connection is successful, the following message appears:

```
Test Connection Successful
```

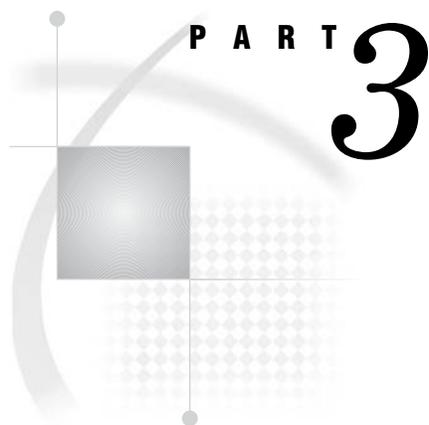
If the connection is not successful, an error message appears.

If you are unable to connect to both the workspace server and the stored process server, then your object spawner setup is probably incorrect. For assistance in resolving the problem, see *SAS Intelligence Platform: Troubleshooting Your Initial Installation*.

Check the Status of a SAS OLAP Server

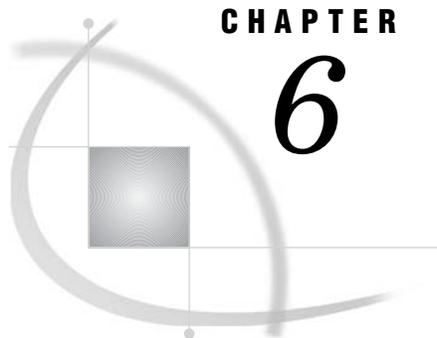
To make sure that connections can be successfully made to a SAS OLAP Server, follow these steps:

- 1 In SAS Management Console, expand the **Monitoring** node and click **SAS OLAP Server Monitor**.
- 2 In the right pane, right-click the name of a logical OLAP server and select **Connect**.
- 3 When you are connected, you can right-click the logical server name and select a monitoring action. For details, see “Using the SAS OLAP Server Monitor” on page 99.



Backing Up and Restoring the SAS Metadata Server

- Chapter 6* **Planning SAS Metadata Server Backups and Restores** 45
- Chapter 7* **Using the %OMABAKUP Macro to Perform Backups and Restores** 51
- Chapter 8* **Manually Restoring the SAS Metadata Server** 63



CHAPTER

6

Planning SAS Metadata Server Backups and Restores

<i>About %OMABAKUP</i>	45
<i>Why Use %OMABAKUP?</i>	46
<i>Best Practices for Using %OMABAKUP to Perform Backups</i>	46
<i>Scheduling Full Backups</i>	46
<i>Backing Up Configuration Files</i>	47
<i>Backing Up Associated Data</i>	47
<i>Using Alternating Backup Locations</i>	47
<i>Using the REORG Option to Reclaim Unused Disk Space</i>	48
<i>Using the RUNANALYSIS Option to Optimize Metadata Server Memory Usage</i>	48
<i>Best Practices for Using %OMABAKUP to Perform Restores</i>	48
<i>Performing Full Restores</i>	48
<i>Restoring the Repository Manager</i>	49
<i>Restoring a Running Metadata Server If Service Pack 4 Is Not Installed</i>	49
<i>Restoring an Unresponsive Metadata Server</i>	49
<i>Using PROC COMPARE To Compare Backup Datasets to Production Datasets</i>	50

About %OMABAKUP

The %OMABAKUP macro is a batch interface that enables you to perform a complete backup or restore of the SAS metadata server. The macro is part of the SAS autocall libraries and runs in a dedicated SAS session.

In its default mode, %OMABAKUP does the following:

- 1 pauses the repository manager, changing the access state to read-only status so that repositories cannot be added or deleted.
- 2 pauses each metadata repository. The pause operation does the following:
 - retains all active client connections
 - temporarily changes the repository's access state to offline status
 - temporarily stops receiving client requests to read or write metadata. If a client application submits a new request, the client displays a message indicating that the server is paused and that the user should try the request later
 - flushes updated metadata from the server's memory and writes the updated metadata to the metadata repository
 - closes the SAS data sets that serve as repository containers
- 3 uses PROC COPY to copy the metadata server's repository manager and all of its registered metadata repositories to a network path that you specify. Your regular system backups can then access these copies.
- 4 creates control files that contain information about the backup

5 resumes the handling of client requests when the backup operation is completed

Beginning in SAS 9.1.3 Service Pack 4, %OMABAKUP includes the RESTORE option, which enables you to restore repositories to a running metadata server. The macro restores metadata and repository relationships to the exact state they were in when the backup copy was made.

Because they ensure consistency, regular full backups and full restores are recommended as the best approach for backing up and restoring the SAS Metadata Server.

Why Use %OMABAKUP?

Instead of using %OMABAKUP, you can choose to manually pause the SAS Metadata Server, use operating system commands to copy the repository files and the repository manager, and then resume the server. However, the %OMABAKUP macro provides the following advantages:

- The macro automatically pauses the repository manager and the repositories that are being backed up or restored. The macro then quickly resumes the repository manager and the repositories after the backup or restore operation is completed. A manual pause and resume operation might result in longer down time.
- When performing backups, you can use the REORG option to reclaim unused disk space in metadata repositories.
- When performing backups, you can use the RUNANALYSIS option to optimize the memory usage of metadata records that are retained in memory on the metadata server.

Best Practices for Using %OMABAKUP to Perform Backups

Scheduling Full Backups

To safeguard the integrity of your metadata, make regular full backups of the metadata server. Full backups include the Foundation repository and all custom and project repositories. By performing full backups, you will ensure that dependencies and cross-references between repositories are correctly maintained and that the metadata server can be restored to a consistent state.

Full backups also include the repository manager, which contains information about the registered repositories. It is important to have a current backup copy of the active repository manager, because restoring a repository to an out-of-date repository manager can cause serious problems if the dependencies do not match.

If your metadata is continually being updated (for example, if information maps, reports, or user identities are created or modified frequently), then you should perform full backups on a nightly basis.

In addition to your regular full backup schedule, you should also perform a full backup at the following times:

- after you add a repository
- after you change the dependencies between two repositories
- before you delete a repository

The %OMABAKUP macro pauses the metadata server and changes repository access to offline status. To minimize the impact of repository downtime on clients, you should execute %OMABAKUP during periods when demand for the metadata server is low.

Backing Up Configuration Files

In order for the metadata server to function properly, the following configuration files must be present: **omaconfig.xml**, **adminUsers.txt**, and **trustedUsers.txt**. You should include these files as part of a full backup or full restore. Since %OMABAKUP does not back up or restore these files, you must use operating system commands or third-party tools.

Backing Up Associated Data

When you back up the metadata server, it is important to also back up the data that is associated with the metadata objects that are contained in the repositories. The metadata cannot be used without the associated data. If you need to recover from a failure, the metadata that you restore must be synchronized correctly with the associated data.

Your backup strategy should include the following data:

- content that is stored in WebDAV repositories, including portal content, report definition files, and other supporting files for reports (for example, PDF files and image files). Depending on your installation, the report files might be stored in the file system instead of WebDAV.
- other portal content, including custom portlets, custom themes, and applications
- source files for stored processes
- packages that have been published to a channel's persistent store (archive) location
- stored process results that are stored in permanent packages
- data sources, including the following:
 - SAS tables
 - OLAP data
 - SAS Scalable Performance Data Server data
 - RDMS data sets
- custom code, processes, and postprocesses that are associated with SAS Data Integration Studio jobs
- XML files and XML maps that are contained in libraries
- source files for SAS Data Integration Studio jobs that have been scheduled or deployed for scheduling

Since %OMABAKUP does not back up or restore these types of data, you must use operating system commands or third-party tools.

Using Alternating Backup Locations

To help safeguard the integrity of your backups, consider alternating backups to multiple backup locations. For example, you might alternate backups to different network-accessible drives, depending on the day of the week. Then if backup files become corrupted, or if a backup host is damaged, you can restore the SAS Metadata Server from another location. For a sample backup script that uses this method, see “Example: Backup Program on Windows, Using Alternating Destinations” on page 54.

Using the REORG Option to Reclaim Unused Disk Space

When metadata is deleted from a SAS Metadata Repository, the record is removed from both memory and disk. However, the disk space that had been allocated for the record remains in the data set after the record is removed.

When you use %OMABAKUP to perform a backup, you can use the REORG parameter to reclaim unused disk space that is left from previously deleted metadata objects. If this parameter is set to **YES**, %OMABAKUP does the following:

- re-creates the repository data sets as it copies them to the backup destination, eliminating unused space in the process
- copies the re-created data sets back to their original locations

Because of the overhead that is associated with re-creating the data sets, you might not want to use the REORG option for every backup. Consider using this option once a month and after large amounts of metadata have been deleted.

Using the RUNANALYSIS Option to Optimize Metadata Server Memory Usage

When you use %OMABAKUP to perform a backup, you can use the RUNANALYSIS option to check the repository backup files for unnecessary indexes, long character variables that can be stored as variable-length strings, duplicate strings, and other issues that can affect memory. The next time you run a backup with %OMABAKUP, the backup program uses the results of the analysis to perform memory optimization tasks. These tasks optimize the amount of space that is used when repository records are read into the metadata server's memory.

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

The %OMARUNAN macro performs the same functions as the RUNANALYSIS option of %OMABAKUP, except that %OMARUNAN runs independently of other programs and the SAS Metadata Server. To minimize the impact of the character variable analyses on the SAS Metadata Server, you might want to direct the backup copies that you want to analyze to a network-accessible drive, and execute %OMARUNAN on that computer. Note that %OMARUNAN must be executed in the same operating environment that was used to create the backups.

For detailed information about using the %OMARUNAN and the RUNANALYSIS option of %OMABAKUP to perform memory optimization, see “Using %OMARUNAN (or the RUNANALYSIS Option of %OMABAKUP) to Optimize Memory Usage” on page 113.

Best Practices for Using %OMABAKUP to Perform Restores

Performing Full Restores

A full restore is necessary when the SAS Metadata Server host computer is destroyed or damaged, or when one or more metadata repositories have been corrupted. A full restore ensures that the relationships and dependencies between repositories are correctly maintained. Like full backups, full restores are generally the best practice to follow to ensure that the metadata server returns to a consistent state.

If you are performing the restore because of a hardware failure, then you should also restore the following configuration files: **omaconfig.xml**, **adminUsers.txt**, and **trustedUsers.txt**. Since %OMABAKUP does not back up or restore these files, you must use operating system commands or third-party tools.

Note: Do not restore repository audit trails. After a restore, auditing should be resumed on emptied or new audit path locations (see “Administering the SAS Metadata Server Audit Trail” on page 74). △

Restoring the Repository Manager

When you specify the RESTORE option, %OMABAKUP reads the repository manager on the metadata server to determine which metadata repositories are registered. The macro cannot restore a repository unless the repository manager is aware that the repository exists.

As a default, the RESTORE option of %OMABAKUP does not copy the repository manager. However, in some cases it is necessary to restore the repository manager. Use these questions to determine whether this step is necessary:

- Is the purpose of the restore to recover a repository that was deleted by using the Delete action in SAS Management Console? Since the Delete action unregisters the repository, you must take one of the following actions before you can restore the deleted repository:
 - restore the repository manager
 - register a new repository, using the same name that had been assigned to the deleted repository

You can then run %OMABAKUP to restore your repositories.

- Have you added any repositories since the last full backup? If so, you should not restore the repository manager, because doing so would overwrite the new repository’s registration. You can re-register the repository, but you must use the same engine and options.
- Is the purpose of the restore to recover a repository that was accidentally deleted by using operating-system commands? If so, you should not restore the repository manager, since the registration will still be present in the repository manager. You need to restore only the repository files.

If you determine that you need to restore the repository manager, you should always restore it as a separate step before you restore the repositories, as follows:

- 1 First, run %OMABAKUP to restore only the repository manager, which you can specify by using the following option: **RepositoryList="REPOSMGR"**.
- 2 Then run %OMABAKUP again to restore the repositories.

Restoring a Running Metadata Server If Service Pack 4 Is Not Installed

If you have not installed SAS 9.1.3 Service Pack 4, you must use operating system commands instead of %OMABAKUP to perform a restore. For instructions, see “Manually Restore a Running SAS Metadata Server” on page 64.

Restoring an Unresponsive Metadata Server

The RESTORE option of %OMABAKUP can restore repositories only on a running metadata server. If your metadata server is not responding to client requests, then you

must use operating system commands to restore the server. For instructions, see “Manually Restoring an Unresponsive SAS Metadata Server” on page 63.

Using PROC COMPARE To Compare Backup Datasets to Production Datasets

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

Here are sample statements:

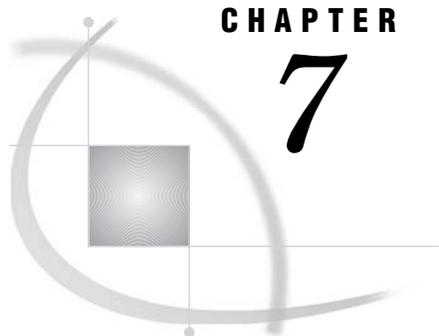
```
options
metaserver="localhost"
metaport=8562
metauser="administrative-userID"
metapass="encoded-password"
metaprotocol="bridge"
;
/* Pause the REPOSMGR for READONLY */
proc metaoperate
action=pause
options='<Repository id="REPOSMGR" Status="READONLY"/>'
noautopause;
run;

libname origrpos "c:\sas\metadata\lev1\SASMain\MetadataServer\rposmgr";
libname backup "c:\sas\metadata\lev1\SASBackup\REPOSMGR";

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

/* Resume REPOSMGR */
proc metaoperate
action=resume
options='<Repository id="REPOSMGR" Status="READONLY"/>'
noautopause;
run;
```

The MDASSOC data set contains repository dependencies, and the RPOSCTRL data set contains repository registrations.



CHAPTER

7

Using the %OMABAKUP Macro to Perform Backups and Restores

<i>Requirements for Using %OMABAKUP</i>	51
<i>Using %OMABAKUP on Windows or UNIX</i>	52
<i>Write and Execute an %OMABAKUP Program for Windows or UNIX</i>	52
<i>Error Logging on Windows or UNIX</i>	53
<i>Example: Program for Full Backup on Windows</i>	53
<i>Example: Backup Program on Windows, Using Alternating Destinations</i>	54
<i>Example: Program on Windows to Restore Only the Repository Manager</i>	55
<i>Example: Program on Windows to Restore All Repositories</i>	55
<i>Using %OMABAKUP on z/OS</i>	56
<i>How the %OMABAKUP Backup Process Works</i>	57
<i>Warning Messages</i>	58
<i>%OMABAKUP Macro: Reference</i>	59
<i>Server Connection Statement: Reference</i>	59
<i>%OMABAKUP Statement: Reference</i>	59

Requirements for Using %OMABAKUP

Before using %OMABAKUP to perform a backup or a restore, read the overview and best practice information that is provided in Chapter 6, “Planning SAS Metadata Server Backups and Restores,” on page 45.

The following requirements apply to the use of %OMABAKUP:

- %OMABAKUP must be executed on the computer that hosts the SAS Metadata Server.
- In a Windows environment, the %OMABAKUP macro must be executed either by a user who is logged in as a member of the Windows Administrators group, or by the same user ID that was used to start the SAS Metadata Server.
On UNIX and z/OS, the %OMABAKUP macro must be executed using the same user ID that was used to start the SAS Metadata Server.
- In all host environments, the user ID that executes %OMABAKUP must have *administrative user* status on the SAS Metadata Server, as well as full operating system access to the **MetadataServer** directory, the **rposmgr** directory, and all repository and backup directories. For information about *administrative user* status, see “About Administrative Users” on page 17.
- Repository auditing must be disabled on all repositories before you use %OMABAKUP to perform a restore. When you re-enable repository auditing following the restore, be sure to specify an empty audit path location.

Using %OMABAKUP on Windows or UNIX

Write and Execute an %OMABAKUP Program for Windows or UNIX

Use the following steps to execute %OMABAKUP on Windows or UNIX:

- 1 If you are performing a backup, first create and secure a directory in the path where the backup files will be stored. For Windows, the following path is recommended:

C:\SAS\config-dir-name\Lev1\SASBackup

For Unix, the following path is recommended:

installer's-home-dir/SAS/config-dir-name/Lev1/SASBackup

Note: The full pathname of a backup repository cannot exceed 200 characters. △

Use operating system permissions to secure the directory. The permissions should deny access to anyone other than the user who will run the backup or restore programs.

- 2 Create a SAS program with a server connection statement and the %OMABAKUP macro, as follows:

- a Use the following syntax to code a server connection statement:

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

For an explanation of each parameter, see “Server Connection Statement: Reference” on page 59.

- b Use the following syntax to code a statement that contains the %OMABAKUP macro and its parameters:

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

For an explanation of each parameter, see “%OMABAKUP Statement: Reference” on page 59.

- 3 Name the program **BackupRepositories.sas** or **RestoreRepositories.sas**, and save it to the directory where the SAS Metadata Server is started. For default Windows installations, the path is as follows:

C:\SAS\config-dir-name\Lev1\SASMain

For default Unix installations, the path is as follows:

installer's-home-dir/SAS/config-dir-name/Lev1/SASMain

Then use operating system permissions to secure the program. The permissions should deny access to anyone other than the person who will run the program.

- 4 If you specified **Restore="Yes"**, and if repository auditing is enabled on any of your repositories, follow these steps before you run the program:

- a Disable repository auditing on each repository. For instructions, see “Administering the SAS Metadata Server Audit Trail” on page 74.
 - b Prepare an empty audit path location that will be used after the restore is complete. You can do either of the following:
 - If you want to resume using the same audit directory after the restore is complete, then move the current audit files to a different location.
 - If you want to specify a different audit directory after the restore is complete, then create the new directory.
- 5 To execute the program on Windows, submit one of the following commands:

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

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

To execute the program on Unix, submit one of the following commands:

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

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

Alternatively, you can set up the program to be executed by a scheduling utility that is supplied by the operating system.

- 6 If you specified **Restore="Yes"**, you can re-enable repository auditing for each repository after the restore is completed. Be sure to specify an empty directory as the audit path location (see step 4).

Error Logging on Windows or UNIX

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

Example: Program for Full Backup on Windows

The following example program creates a full backup of the metadata server, including the repository manager and each repository that is registered in the repository manager.

```
/* These system options establish a connection to the metadata server */
options metaserver="D1234"
      metaport=8561
      metaprotocol="bridge"
      metauser="administrative-userID"
      metapass="encoded-password"
      metarepository="Foundation";
/* This %OMABAKUP command creates a backup on the D1234 machine in
C:\SAS\config-dir-name\Lev1\SASBackup */

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

Example: Backup Program on Windows, Using Alternating Destinations

This example program creates one backup image on machine D1234 and seven days of backup images on machine D2345. As a result, multiple restoration options are available if the backups become corrupted. The example includes two executions of %OMABAKUP:

- The first execution stores the backup files on the metadata server host (D1234). These backup files are overwritten every day, and they represent the most recent backup image.
- The second execution stores the backup files on a different machine (D2345). The name of the backup directory varies depending on the day of the week. The backup directory is specified as **Dmt01 Repos Backups\&day**, where **&day** is a directory named **Sun, Mon, Tue, Wed, Thu, Fri, or Sat**.

```

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

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

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

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

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

```

Example: Program on Windows to Restore Only the Repository Manager

The following example program restores only the repository manager.

If the repository manager on your metadata server does not contain active registrations for all of the repositories that you want to restore (for example, if a repository was accidentally deleted through a SAS Management Console command since the last backup was taken), you must run this program first before you can successfully restore your repositories. For more information, see “Restoring the Repository Manager” on page 49.

```
/* These system options establish a connection to the metadata server */
options metaserver="D1234"
      metaport=8561
      metaprotocol="bridge"
      metauser="administrative-userID"
      metapass="encoded-password"
      metarepository="Foundation";
/* This %OMABAKUP command restores the repository manager from the backup
files on the D1234 machine in
C:\SAS\config-dir-name\Lev1\SASBackup */

%omabakup(DestinationPath="C:\SAS\config-dir-name\Lev1\SASBackup",
          ServerStartPath="C:\SAS\config-dir-name\Lev1\SASMain",
          RposmgrPath="metadataServer\rposmgr",
          Restore=Y,
          RepositoryList="REPOSMGR")
```

Example: Program on Windows to Restore All Repositories

The following example program restores all registered repositories, but it does not restore the repository manager.

Use this program if you are certain that the server’s repository manager contains active registrations for all of the repositories that you want to restore. If it does not, then you must restore the repository manager first as a separate step. For detailed guidance, see “Restoring the Repository Manager” on page 49.

```
/* These system options establish a connection to the metadata server */
options metaserver="D1234"
      metaport=8561
      metaprotocol="bridge"
      metauser="administrative-userID"
      metapass="encoded-password"
      metarepository="Foundation";
/* This %OMABAKUP command restores the repositories from the backup
files on the D1234 machine in
C:\SAS\config-dir-name\Lev1\SASBackup */

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

Using %OMABAKUP on z/OS

Use the following steps to execute %OMABAKUP on z/OS:

- 1 First, create a directory in the path where the backup files will be stored. For z/OS, the following path is recommended:

```
/usr/lpp/SAS/config-dir-name/Lev1/SASBackup
```

Note: The full pathname of a backup repository cannot exceed 200 characters. Δ

Use operating system permissions to secure the directory. The permissions should deny access to anyone other than the user who will run the backup program.

- 2 Create a CONFIG data set that contains a server connection statement. The syntax for the server connection statement is as follows:

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

For an explanation of the parameters, see “Server Connection Statement: Reference” on page 59.

Save the file using the name *&prefix*.WO.SRVCFG(BACKUP).

- 3 Next, create a batch job that does the following:
 - starts SAS.
 - references the CONFIG data set that contains the server connection statement.
 - executes the %OMABAKUP macro, with appropriate options. The syntax for an %OMABAKUP statement is as follows:

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

For an explanation of the parameters, see “%OMABAKUP Statement: Reference” on page 59.

- Include job control language (JCL) that specifies where to write the log in the batch job that invokes %OMABAKUP—for example:

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

Here is an example of a batch job that runs on z/OS and stores the backup files on a UNIX machine:

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

// MSGCLASS=A,
// TIME=(5)
```

```

*-----
  RUN BACKUP
*-----
//SAS91 EXEC SAS91,

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

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

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

/*

//

```

4 Save the job in `&prefix.WO.SRVCNTL(BACKUP)`.

Restrict access to `&prefix.WO.SRVCNTL`. Access should be granted only to the person who will run the program.

- 5** If you specified `Restore="Yes"`, and if repository auditing is enabled on any of your repositories, follow these steps before you run the program:
- a** Disable repository auditing on each repository. For instructions, see “Administering the SAS Metadata Server Audit Trail” on page 74.
 - b** Prepare an empty audit path location that will be used after the restore is complete. You can do either of the following:
 - If you want to resume using the same audit directory after the restore is complete, then move the current audit files to a different location.
 - If you want to specify a different audit directory after the restore is complete, then create the new directory.
- 6** Use operating system commands to submit the batch job or to automate execution of the macro program.
- 7** If you specified `Restore="Yes"`, you can re-enable repository auditing for each repository after the restore is completed. Be sure to specify an empty directory as the audit path location (see step 5).

How the %OMABAKUP Backup Process Works

When you use the %OMABAKUP macro with the RESTORE parameter set to **No** (this is the default setting), the macro does the following:

- 1** It pauses the metadata server. The Pause action does the following:
 - temporarily changes the access state of each repository from full access to OFFLINE
 - instructs the SAS Metadata Server to stop accepting new client requests for each repository that is being backed up
 - closes the repository containers
 - retains client connections and causes the metadata server to temporarily stop accepting requests

- 2 In the location that is specified in the `DESTINATIONPATH` parameter, it creates a subdirectory called **REPOSMGR**. It then copies the repository manager to this subdirectory.

Note: `%OMABAKUP` assumes that the directory that is specified in `DESTINATIONPATH` has already been created. \triangle

- 3 For each repository that is being backed up, `%OMABAKUP` performs the following steps:

- a It creates a subdirectory in the location that is specified in the `DESTINATIONPATH` parameter. The subdirectory is named with the value of the repository's **Name** attribute.

For example, if the name of the repository is `Foundation` and the directory specified in `DESTINATIONPATH` is `SASBackup`, then the macro creates a subdirectory called `SASBackup/Foundation`.

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

Note: `%OMABAKUP` supports repository **Name** values of up to 32 characters. \triangle

- b It copies the repository's data sets to the subdirectory.
 - c It creates a `DS_DEFN` data set for each repository in its registered location. `DS_DEFN` is a control data set that specifies how character variables in repository data sets are to be treated in memory. This data set is used to load repositories back into memory when client activity is resumed on the metadata server.
 - d If the `RUNANALYSIS` option is specified, it creates a `DS_VCHAR` data set in the subdirectory. For information about `DS_VCHAR` data sets, see "Using `%OMARUNAN` (or the `RUNANALYSIS` Option of `%OMABAKUP`) to Optimize Memory Usage" on page 113.
- 4 Beginning in SAS 9.1.3 Service Pack 4, `%OMABAKUP` creates or updates the following catalogs in the destination directory:
 - `OMA_backup.sas7bcatalog`, which stores information about repository source and target destinations
 - `OMA_varan.sas7bcatalog`, which contains a job that SAS processes can use to analyze repository data sets in the backup subdirectories.

Warning Messages

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

WARNING: Apparent invocation of macro Directory not resolved.

This warning is generated if you have a repository name that contains a percent sign (`%`)— for example, `my%directory`. `%OMABAKUP` ignores the warning and creates the backup subdirectory anyway.

WARNING: No matching members in directory.

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

WARNING: Repository not found, using the first repository on the server.

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

%OMABAKUP Macro: Reference

Server Connection Statement: Reference

Before you can execute the %OMABAKUP macro, you must connect to a SAS Metadata Server. Use the following server connection statement to connect to the metadata server:

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

Set the options on the server connection statement as follows:

METASERVER

Specify **localhost** to connect to the metadata server on the same machine.

METAPORT

Specify the unique port number that identifies your metadata server. The default port number is 8561.

METAUSER

Specify the user ID and password of a user that meets these requirements:

- has *administrative user* status on the SAS Metadata Server
- has full operating system access to the MetadataServer directory, the rposmgr directory, and all repository and backup directories

METAPASS

Specify the password for the METAUSER. The password should be encrypted using SAS proprietary 32-bit encryption. To obtain the encrypted password, use PROC PWENCODE.

For example, to encrypt the password "SASmeta1," you would submit this code in the SAS Program Editor:

```
proc pwencode in="SASmeta1" method=sasenc;
run;
```

METAREPOSITORY

Specify the name of the default repository. The name that is recommended for all installations is Foundation.

%OMABAKUP Statement: Reference

The %OMABAKUP macro copies all of the repositories that are registered on a metadata server and the server's repository manager to a backup location. The macro can also be used to restore repositories from the backup location to the metadata server.

The syntax of a %OMABAKUP macro statement is as follows:

```
%omabakup(DestinationPath="pathname",
           ServerStartPath="pathname",
           RposmgrPath="pathname",
           <Reorg="Yes|No">,
           <RepositoryList="repository-name<,...repository-name-n">>,
           <Restore="Yes|No">,
           <RunAnalysis="Yes|No">)
```

Set the options for the %OMABAKUP macro as follows:

DESTINATIONPATH="pathname"

is a required parameter for a backup or a restore.

For a backup, specify the pathname of the directory where backup copies are to be stored. Specify the absolute pathname of an existing directory; %OMABAKUP will not create the directory for you.

For a restore, specify the pathname of the directory where backups are stored.

The following table provides recommended values for DESTINATIONPATH for each host environment:

Table 7.1 Recommended Values for DESTINATIONPATH

Host Environment	Recommended Value for DESTINATIONPATH
Windows	"C:\SAS\config-dir-name\Lev1\SASBackup"
UNIX	"installer's-home-dir/SAS/config-dir-name/Lev1/SASBackup"
z/OS	"/usr/lpp/SAS/config-dir-name/Lev1/SASBackup"

SERVERSTARTPATH="pathname"

is a required parameter for a backup and a restore. Specify the full path to the directory in which the metadata server process was started.

The following table provides the value of SERVERSTARTPATH for default installations in each host environment:

Table 7.2 Value of SERVERSTARTPATH for Default Installations

Host Environment	Value in Default Installation
Windows	"C:\SAS\config-dir-name\Lev1\SASMain"
UNIX	"installer's-home-dir/SAS/config-dir-name/Lev1/SASMain"
z/OS	"/usr/lpp/SAS/config-dir-name/Lev1/SASMain"

RPOSMGRPATH="pathname"

is a required parameter for a backup and a restore. Specify the relative location of the repository manager as defined in the **omaconfig.xml** file.

The value of RPOSMGRPATH for default installations in Windows, UNIX, and z/OS host environments is as follows:

MetadataServer/rposmgr

REORG="YES|NO"

is an optional parameter for a backup that enables you to reclaim unused disk space left from previously deleted metadata objects from SAS metadata repositories.

When metadata is deleted from a SAS metadata repository, the record is removed from both memory and disk. However, the disk space that had been allocated for the record remains in the data set after the record is removed.

Specify one of the following values for this parameter:

- YES** (or **Y**) if you want to reclaim disk space during the backup operation. (This option is not valid when %OMABAKUP is executed in restore mode.) If you specify this option, %OMABAKUP does the following:
 - re-creates the repository data sets as it copies them to the backup destination, eliminating unused space in the process
 - copies the re-created data sets back to their original locations

Note: Because of the overhead that is associated with re-creating the data sets, you might not want to use the REORG option for every backup. Consider using this option once a month and after large amounts of metadata have been deleted. Δ

- NO** (or **N**) if you want to copy the repositories as they are. This is the default value.

REPOSITORYLIST="REPOSMGR"

is an optional parameter that you can use if you need to restore the repository manager. The repository manager must be identified as REPOSMGR.

As a default, the RESTORE option of %OMABAKUP does not copy the repository manager. However, in some cases it might be necessary to restore it. For important guidance in using this parameter, see “Restoring the Repository Manager” on page 49.

RESTORE="YES | NO"

is an optional parameter that enables you to execute %OMABAKUP in restore mode. Specify one of the following values:

- YES** (or **Y**) to run the %OMABAKUP macro in restore mode. The macro will restore all repositories that are registered in the SAS Metadata Server’s repository manager to their registered locations.
- NO** (or **N**) to run the macro in backup mode. This is the default value.

RUNANALYSIS="YES | NO"

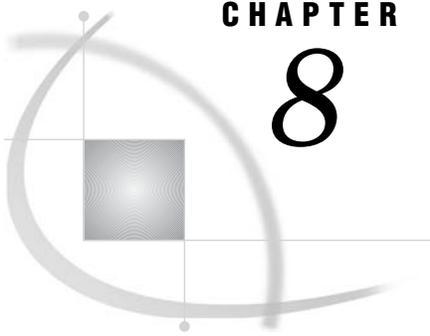
is an optional parameter for a backup that specifies whether to analyze backup copies of repository data sets for information that can be used to optimize the data sets’ memory footprint on the metadata server. The analysis is run immediately after client activity is resumed on the SAS Metadata Server.

Specify one of the following values:

- YES** (or **Y**) to analyze the backup repositories. This option is not valid when %OMABAKUP is executed in restore mode.
- NO** (or **N**) if you do not want to analyze the backup repositories. This is the default value.

For more information about this option, see “Using %OMARUNAN (or the RUNANALYSIS Option of %OMABAKUP) to Optimize Memory Usage” on page 113.

Note: The pathnames that are used in the DESTINATIONPATH, SERVERSTARTPATH, and RPOSMGRPATH parameters cannot contain embedded asterisks (*), backslashes (\), question marks (?), brackets ([]), or single or double quotation marks (with the exception that the backslash is supported as a directory delimiter on hosts that support the backslash as a delimiter). Δ



CHAPTER

8

Manually Restoring the SAS Metadata Server

Manually Restoring an Unresponsive SAS Metadata Server 63

Manually Restore a Running SAS Metadata Server 64

Manually Restoring an Unresponsive SAS Metadata Server

If the SAS Metadata Server is not responding to client requests, then you cannot use %OMABAKUP to perform a restore. Instead, you must stop the server and then use operating system commands to perform the restore.

This procedure assumes that metadata server components such as the **omaconfig.xml** file, **adminUsers.txt** file, and **trustedUsers.txt** file exist and are intact. If this is not the case, then you must re-install the metadata server before using this procedure.

Follow these steps to restore a metadata server that is not responding to client requests:

- 1 Follow the steps in “Use SAS Management Console to Stop the SAS Metadata Server” on page 33 to ensure that the server is stopped.
- 2 If the SAS Metadata Server was running with auditing enabled in one or more repositories, locate the audit trails and move all audit files to other directories. The active audit directories need to be empty so that when the repositories are restored, the metadata server does not write new audit records with the old records, which are no longer valid.
- 3 Open the directory where the SAS Metadata Server is started. In a planned SAS installation on Windows, this is usually *SAS-config-dir\Lev1\SASMain\MetadataServer*.
- 4 Use the appropriate operating system commands to delete the contents of the current **rposmgr** and repository directories.

Windows commands:

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

UNIX commands:

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

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

Windows commands:

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

UNIX commands:

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

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

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

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

Windows example:

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

UNIX example:

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

- 6 Use the instructions in “Use SAS Management Console to Stop the SAS Metadata Server” on page 33 to start the metadata server.
- 7 Follow the instructions in “Administering the SAS Metadata Server Audit Trail” on page 74 to verify that auditing is enabled with the appropriate settings.

Manually Restore a Running SAS Metadata Server

Use this procedure to restore the repository manager and repositories if your SAS Metadata Server is responsive, but you are using a SAS software version that is earlier than SAS 9.1.3 Service Pack 4. If the metadata server does not respond, use the procedure in “Manually Restoring an Unresponsive SAS Metadata Server” on page 63.

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

- The Pause action retains client connections and causes the metadata server to temporarily stop accepting requests.

See Chapter 4, “Starting, Stopping, and Pausing Servers,” on page 29 for instructions on how to stop the server.

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

```
/* Example of PROC METAOPERATE PAUSE action */
proc metaoperate
  server="host-name"
  port=8561
  userid="administrative-userID"
  password="password"
  protocol=bridge

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

- 2 If the SAS Metadata Server was running with auditing enabled in one or more repositories, locate the audit trails and move all audit files to other directories. The active audit directories need to be empty so that when the repositories are restored, the metadata server does not write new audit records with the old records, which are no longer valid.
- 3 Delete the current **rposmgr** and repository directories and copy the backup copies of the repository manager and repositories by following the instructions in “Manually Restoring an Unresponsive SAS Metadata Server” on page 63.
- 4 Resume or restart the metadata server.

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

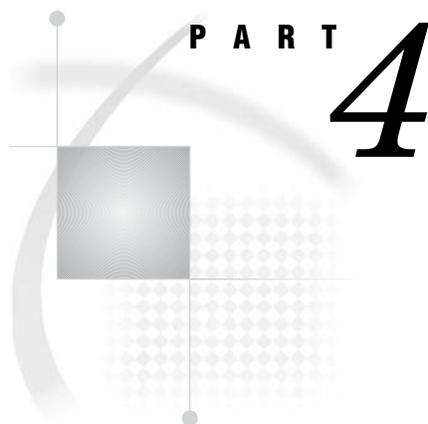
```
/* example of PROC METAOPERATE RESUME action */

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

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

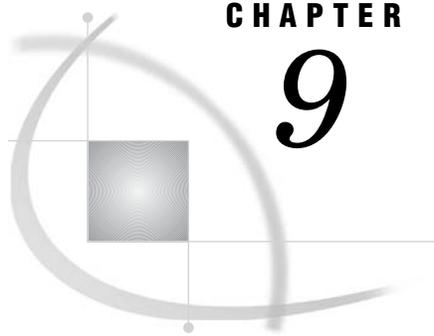
To restart the metadata server, use the instructions in Chapter 4, “Starting, Stopping, and Pausing Servers,” on page 29.

- 5 Follow the instructions in “Administering the SAS Metadata Server Audit Trail” on page 74 to view the audit settings for each repository and verify that auditing is enabled with the appropriate settings.



System Monitoring and Logging

- Chapter 9* **Administering the SAS Metadata Server Log and Audit Trail** 69
- Chapter 10* **Using the ARM_OMA Subsystem to Obtain Raw Performance Data for the SAS Metadata Server** 77
- Chapter 11* **Administering Logging for the SAS Object Spawner, SAS Workspace Server, SAS Stored Process Server, and SAS/CONNECT Server** 89
- Chapter 12* **SAS OLAP Server Monitoring and Logging** 97



CHAPTER

9

Administering the SAS Metadata Server Log and Audit Trail

<i>Administering the SAS Metadata Server Log</i>	69
<i>About SAS Metadata Server Logging</i>	69
<i>The Default Metadata Server Logging Configuration</i>	70
<i>Enable the Logging of SAS Metadata Server Authentication Events</i>	70
<i>Capture XML Information in the SAS Metadata Server Log</i>	71
<i>About Capturing XML Information</i>	71
<i>Configure the Logging of XML Information</i>	72
<i>Examples of XML Log Entries</i>	72
<i>Capture Debugging Information in the SAS Metadata Server Log</i>	74
<i>Administering the SAS Metadata Server Audit Trail</i>	74
<i>About Repository Auditing</i>	74
<i>Enable Repository Auditing</i>	74
<i>Disable Repository Auditing</i>	75

Administering the SAS Metadata Server Log

About SAS Metadata Server Logging

The SAS Metadata Server creates log entries that record server events, errors, and traces. These entries are directed to the destination specified by the LOG parameter on the SAS Metadata Server start command. The default location is as follows:

```
SAS-config-dir\Lev1\SASMain\MetadataServer\logs\MetadataServer#d#b#y.log
```

The log entries are useful to anyone who needs to know the status of the SAS Metadata Server and whether a given request was successful. The log entries also contain information that is useful for diagnosing server start-up problems, diagnosing authorization failures, and debugging method calls.

Depending on which SAS Metadata Server invocation options are used, the SAS Metadata Server can write seven categories of information to the SAS log:

- SAS Metadata Server start and stop information, including the invoking user ID, SAS long version number, SAS Metadata Model version, the directory where the metadata server was started, and configuration file options
- user connection and disconnection events
- repository creation/deletion, open/close, pause/resume, and refresh events
- errors, including task/thread exceptions, memory allocation errors, I/O errors, application logic errors, authentication errors, and authorization failures
- authentication events
- XML input and output strings

- traces invoked by `omaconfig.xml` debugging options

The Default Metadata Server Logging Configuration

The default configuration enables logging of the first four categories of information that are listed in the preceding topic. To enable these logging levels, the SAS Configuration Wizard specifies the following invocation options in the file

`SAS-config-dir\Lev1\SASMain\MetadataServer\sasv9_MetadataServer.cfg`:

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

The following table explains each option:

Table 9.1 Log-File-Related Options in `sasv9_MetadataServer.cfg`

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

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

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

- “Enable the Logging of SAS Metadata Server Authentication Events” on page 70
- “Capture XML Information in the SAS Metadata Server Log” on page 71
- “Capture Debugging Information in the SAS Metadata Server Log” on page 74

Enable the Logging of SAS Metadata Server Authentication Events

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

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

- 1 Open the file

`SAS-config-dir\Lev1\SASMain\MetadataServer\sasv9_MetadataServer.cfg`.

This file contains a line similar to the following:

```
-objectserverparms "cel=credentials protocol=bridge port=8561
classfactory=2887E7D7-4780-11D4-879F-00C04F38F0DB
```

```
trustsaspeer='MetadataServer\trustedPeers.xml'
```

- 2 In the list of object server parameters (the string that follows **-objectserverparms**), add one of the following options:

- APPLEVEL=2** enables IOM server logging with the default log entries, and also logs authentication requests by clients.
- APPLEVEL=3** enables IOM server logging with all of the information provided at **APPLEVEL=2**, and also logs requests by unregistered user IDs that are mapped as **PUBLIC**. You can use information about these user IDs to identify candidates for registration.

- 3 Save your changes to **sasv9_MetadataServer.cfg**.

- 4 Restart the metadata server.

The **APPLEVEL=2** option provides log entries similar to the following example:

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

Note:

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

△

Capture XML Information in the SAS Metadata Server Log

About Capturing XML Information

When clients communicate with the SAS Metadata Server to create, update, delete, or query a metadata object, information about metadata objects is represented in XML. The metadata server also uses XML entries to return information about metadata objects to clients. You can choose to capture these XML entries in the metadata server log.

For example, to add a metadata object to a repository, clients use the **AddMetadata** method of the SAS Open Metadata Interface. One of the parameters for this method is an XML description of the object to be added. For example, if the client is requesting that a table object be added, then the request includes an XML description of the table, as shown in this example:

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

The metadata server returns an XML string that includes the input description as well as the object ID of the new metadata object.

Note: For more information about the call interfaces, see the *SAS Open Metadata Interface: Reference*. △

Configure the Logging of XML Information

To configure the logging of XML information, complete these steps:

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

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

- 2 In the list of object server parameters (the string that follows `-objectserverparms`), add the following parameters:

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

For the second and third parameters, you can specify different values from those that are shown in the example. The parameters are explained in the following table:

Table 9.2 Parameters Required to Capture XML in the Log File

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

Examples of XML Log Entries

The format of the XML that is written to the log depends on whether the request was made using the standard SAS Open Metadata Interface or using the DoRequest method.

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

```
2003043:15.07.35.21: 00000019: IOM CALL {compRef:3cbbdd0}->OMI::DoRequest():
inString=
<multiple_requests>
  <AddMetadata>
    <Metadata>
      <RepositoryBase Name="scratch2" Desc="Repository 2" Path="scratch2">
        <DependencyUses>
          <RepositoryBase Name="scratch1" Desc="Repository 1" Path="scratch1"/>
        </DependencyUses>
        <DependencyUsedBy>
          <RepositoryBase Name="scratch3" Desc="Repository 3" Path="scratch3"/>
        </DependencyUsedBy>
      </RepositoryBase>
    </Metadata>
  </AddMetadata>
</multiple_requests>
```

```

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

2003043:15.07.35.38: 00000019: 2:SASTST@DOMAIN]: RegisterRepository
Id=A0000001.A5M9N7FJ, Name=scratch2, Path=scratch2, Engine=., Options=.,
AuditPath=..
2003043:15.07.35.56: 000000019: 2:SASTST@DOMAIN]: RegisterRepository
Id=A0000001.A53X8NYG, Name=scratch1, Path=scratch1, Engine=., Options=.,
AuditPath=..
2003043:15.07.35.77: 00000019: 2:SASTST@DOMAIN]: RegisterRepository
Id=A0000001.A5IOGZ01, Name=scratch3, Path=scratch3, Engine=., Options=.,
AuditPath=..
2003043:15.07.35.90: 00000019: IOM RETURN 0={compRef:3cbbdd0}->OMI::DoRequest():
outString=<multiple_requests><AddMetadata><Metadata><RepositoryBase
Name="scratch2" Desc="Repository 2" Path="scratch2" Id="A0000001.A5M9N7FJ"
Access="0" <RepositoryType=""><DependencyUses><RepositoryBase Name="scratch1"
Desc="Repository 1" Path="scratch1" Id="A0000001.A53X8NYG" Access="0"
RepositoryType=""/></DependencyUses><DependencyUsedBy><RepositoryBase
Name="scratch3" Desc="Repository 3" Path="scratch3" Id="A0000001.A5IOGZ01"
Access="0" RepositoryType=""/> </DependencyUsedBy></RepositoryBase></Metadata>
<Reposid>A0000001.A0000001</Reposid><Ns>REPOS</Ns><Flags>268435456</Flags>
<Options/></AddMetadata></multiple_requests>  retval=0

```

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

```

2003043:15.20.04.12: 00000019: IOM CALL {compRef:3cbbdd0}->OMI::AddMetadata():
inMetadata=
    <RepositoryBase Name="scratch2" Desc="Repository 2" Path="scratch2">
        <DependencyUses>
            <RepositoryBase Name="scratch1" Desc="Repository 1" Path="scratch1"/>
        </DependencyUses>
        <DependencyUsedBy>
            <RepositoryBase Name="scratch3" Desc="Repository 3" Path="scratch3"/>
        </DependencyUsedBy>
    </RepositoryBase>
    reposid=A0000001.A0000001 ns=REPOS flags=268435456
options=
2003043:15.20.04.25: 00000019: 2:SASTST@DOMAIN]: RegisterRepository
Id=A0000001.A55LJU3G, Name=scratch2, Path=scratch2, Engine=., Options=.,
AuditPath=..
2003043:15.20.04.37: 00000019: 2:SASTST@DOMAIN]: RegisterRepository
Id=A0000001.A5PE8JLK, Name=scratch1, Path=scratch1, Engine=., Options=.,
AuditPath=..
2003043:15.20.04.53: 00000019: 2:SASTST@DOMAIN]: RegisterRepository
Id=A0000001.A5ZJIVYZ, Name=scratch3, Path=scratch3, Engine=., Options=.,
AuditPath=..
2003043:15.20.04.62: 00000019: IOM RETURN 0={compRef:3cbbdd0}->OMI::AddMetadata():

```

```

outMetadata=<RepositoryBase Name="scratch2" Desc="Repository 2" Path="scratch2"
Id="A0000001.A55LJU3G" Access="0" RepositoryType=""><DependencyUses>
<RepositoryBaseName="scratch1" Desc="Repository 1" Path="scratch1"
Id="A0000001.A5PE8JLK" Access="0" RepositoryType="" /></DependencyUses>
<DependencyUsedBy><RepositoryBase Name="scratch3" Desc="Repository 3"
Path="scratch3" Id="A0000001.A5ZJIVYZ" Access="0" RepositoryType="" />
</DependencyUsedBy></RepositoryBase>  retval=0

```

Capture Debugging Information in the SAS Metadata Server Log

The SAS Metadata Server supports a number of reserved `omaconfig.xml` options that you should not specify unless you have been instructed to include them by SAS Technical Support. There are two categories of reserved options: SAS Metadata Server options and debugging options.

The debugging options can increase the processing overhead of the metadata server; therefore, they are not recommended for normal use. These options are specified in an `<OPTIONS>` XML element in the `omaconfig.xml` file. For information about these options, see “Reserved `omaconfig.xml` Options” on page 257.

Administering the SAS Metadata Server Audit Trail

About Repository Auditing

The SAS Metadata Server enables you to save information about changes to a repository in a set of audit files, referred to as an *audit trail*. Auditing is performed on individual repositories and is invoked in SAS Management Console after a repository is created. You can choose any of the following categories of metadata records to audit:

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

The availability of the audit trail is limited as follows:

- You can view audit records from Windows and UNIX hosts, but you cannot view audit records on z/OS hosts.
- Audit records are created by the Base SAS engine, whether the repositories themselves were created in SAS data sets, Oracle, or DB2.

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

The audit files are not intended for use in restoring repositories. To perform backups and restores, use the `%OMABAKUP` autocall macro as described in Chapter 7, “Using the `%OMABAKUP` Macro to Perform Backups and Restores,” on page 51. Note the instructions for handling audit files before and after a restore.

Enable Repository Auditing

To begin storing records of a repository’s metadata transactions in an audit trail, follow these steps:

- 1 In SAS Management Console, expand the **Metadata Manager** and the **Active Server** nodes.

All repositories that are registered on the SAS Metadata Server are displayed in the navigation tree.

- 2 Right-click the repository name and select **Audit**.

- 3 In the Audit Properties dialog box, follow these steps:

- a Select the **Enable Auditing** check box.
- b Select a type from the **Audit Type** drop-down list. To see a description of each type, click **Help**.
- c In the **Audit Path** field, specify the path where the audit file is to be saved (or click **Browse** to browse to the location).
- d Click **OK** .

Disable Repository Auditing

To disable repository auditing, follow these steps:

- 1 In SAS Management Console, expand the **Metadata Manager** and the **Active Server** nodes.

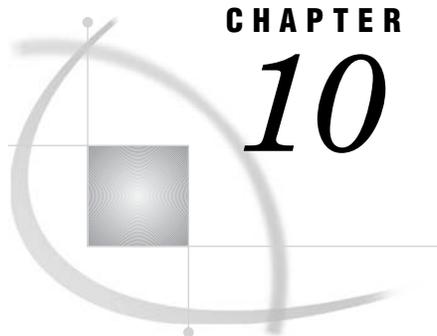
All repositories that are registered on the SAS Metadata Server are displayed in the navigation tree.

- 2 Right-click the repository name and select **Audit**.

- 3 In the Audit Properties dialog box, follow these steps:

- a Clear the **Enable Auditing** check box.
- b Clear the **Audit Path** field.
- c Click **OK** .

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



CHAPTER

10

Using the ARM_OMA Subsystem to Obtain Raw Performance Data for the SAS Metadata Server

<i>About the ARM_OMA Subsystem</i>	77
<i>Enabling ARM Logging</i>	78
<i>About Enabling ARM Logging</i>	78
<i>Start the ARM_OMA Subsystem at Server Start-up</i>	78
<i>Enable Other ARM Subsystems</i>	78
<i>Start or Stop ARM Logging on a Running Server</i>	79
<i>Understanding the Records Written by the ARM_OMA Subsystem</i>	79
<i>ARM_OMA Log Examples</i>	82
<i>About the ARM_OMA Log Examples</i>	82
<i>Example 1: ARM Records Created When Registering a Repository</i>	82
<i>Example 2: ARM Records Created When Adding a Table and Columns</i>	83
<i>Post-processing the ARM_OMA Log</i>	88

About the ARM_OMA Subsystem

Application Response Measurement (ARM) is a standard for managing the performance of distributed applications. The ARM_OMA subsystem of the SAS Metadata Server uses ARM API calls to return detailed log entries when the following events occur:

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

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

Enabling ARM Logging

About Enabling ARM Logging

You can use the following methods to enable ARM logging for the SAS Metadata Server:

- Provide continuous ARM logging by starting the ARM_OMA subsystem at server start-up. If you use this method, you can also enable other ARM subsystems.
- Enable periodic ARM logging on a running server by submitting ARM logging system options via PROC METAOPERATE.

See the following topics for instructions.

Start the ARM_OMA Subsystem at Server Start-up

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

ARMSUBSYS	enables and disables ARM logging. To enable ARM logging on the metadata server, specify the value ARM_OMA . To disable ARM logging, specify the value ARM_NONE .
ARMLOC	specifies a location to which to write the ARM log file.
ARMAGENT	specifies an ARM agent, which is an executable module that contains a vendor's implementation of the ARM API. If an agent is not specified, the default collector for the SAS Metadata Server is sasarmmg.dll .

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

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

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

Enable Other ARM Subsystems

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

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

For information about the ARM_IOM and ARM_DSIO subsystems, see the descriptions in the ARMSUBSYS= documentation.

Start or Stop ARM Logging on a Running Server

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

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

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

RUN;
```

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

These statements specify that the ARM log is to be called **armfile.log** and is to be written to the **logs** subdirectory of the **MetadataServer** directory.

To disable ARM logging, execute a similar request that does the following:

- specifies "**(ARM_NONE)**" in the ARMSUBSYS= system option
- omits the ARMLOC system option

Understanding the Records Written by the ARM_OMA Subsystem

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

Notes:

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

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

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

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

The following records are written to the ARM log:

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

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

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

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

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

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

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

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

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

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

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

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

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

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

```
U,1326479508.508000,1,1,2,1.612318,2.443513,sastst ,,
Column::AddMetadata,ObjRef=A9000001.A7EAD04N
```

- when a list of objects is retrieved by the GetMetadataObjects method. Additional fields are a user ID, the return code after processing the request, and the object type and function name **ResumePeersAll**, written as *metadata-type::ResumePeersAll*. Here is an example.

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

P (stop) record

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

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

E (end) record

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

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

ARM_OMA Log Examples

About the ARM_OMA Log Examples

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

Example 1: ARM Records Created When Registering a Repository

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

```
<AddMetadata>
  <Metadata>
    <RepositoryBase Name="scratch2" Desc="Repository 2" Path="scratch2">
      <DependencyUses>
        <RepositoryBase Name="scratch1" Desc="Repository 1" Path="scratch1"/>
      </DependencyUses>
      <DependencyUsedBy>
        <RepositoryBase Name="scratch3" Desc="Repository 3" Path="scratch3"/>
      </DependencyUsedBy>
    </RepositoryBase>
  </Metadata>
  <!--Reposid specifies Repository manager -->
  <Reposid>A0000001.A0000001</Reposid>
  <Ns>REPOS</Ns>
  <!-- OMI_TRUSTED_CLIENT flag -->
  <Flags>268435456</Flags>
  <Options/>
</AddMetadata>
```

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

```
S,1346212863.241000,1,1,2,1.502160,2.153096,sastst ,,request::AddMetadata
C,1346212863.262000,1,1,3,1,2,1.512174,2.153096,sastst ,,RepositoryBase::
  AddMetadata
C,1346212863.572000,1,1,4,1,3,1.592289,2.313326,sastst ,,RepositoryBase::
  omiPersistElement
C,1346212863.572000,1,1,5,1,4,1.592289,2.313326,sastst ,,RepositoryBase::
  AddMetadata
P,1346212863.953000,1,1,5,1.632347,2.433499,0,sastst , 0,RepositoryBase::
  AddMetadata
P,1346212863.953000,1,1,4,1.632347,2.433499,0,sastst , 0,RepositoryBase::
  omiPersistElement
```

```

U,1346212863.983000,1,1,3,1.652376,2.443513,1,sastst ,      0,DependencyUses::
  add
C,1346212863.983000,1,1,6,1,3,1.652376,2.443513,sastst ,,RepositoryBase::
  omiPersistElement
C,1346212863.983000,1,1,7,1,6,1.652376,2.443513,sastst ,,RepositoryBase::
  AddMetadata
P,1346212864.303000,1,1,7,1.772548,2.533643,0,sastst ,      0,RepositoryBase::
  AddMetadata
P,1346212864.303000,1,1,6,1.772548,2.533643,0,sastst ,      0,RepositoryBase::
  omiPersistElement
U,1346212864.323000,1,1,3,1.782563,2.533643,1,sastst ,      0,
  DependencyUsedBy::add
P,1346212864.323000,1,1,3,1.782563,2.533643,0,sastst ,      0,RepositoryBase::
  AddMetadata
P,1346212864.433000,1,1,2,1.802592,2.563686,0,sastst ,      0,request::
  AddMetadata

```

Here are the records in the log:

- Record 1: Identifies an AddMetadata request.
- Record 2: Logs adding a RepositoryBase object (**scratch2** in the XML).
- Record 3: Logs a call to process an associated element (**scratch1** in the XML).
- Record 4: Logs AddMetadata of a second RepositoryBase object (**scratch1** in the XML).
- Record 5: AddMetadata for scratch1 got a successful return code.
- Record 6: Processing associated element got a successful return code.
- Record 7: Logs adding DependencyUses dependency for scratch1.
- Record 8: Logs call to process an associated element (**scratch3** in the XML).
- Record 9: Logs AddMetadata of a third RepositoryBase object (**scratch3** in the XML).
- Record 10: AddMetadata for scratch3 got a successful return code.
- Record 11: Processing associated element got a successful return code.
- Record 12: Logs adding DependencyUsedBy dependency for scratch3.
- Record 13: AddMetadata for scratch2 got a successful return code.
- Record 14: AddMetadata request got a successful return code.

In the output, note the following:

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

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

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

```

<AddMetadata>
  <Metadata>
    <PhysicalTable Name="Sales Offices" Desc="Sales offices in NW region">
      <TablePackage>
        <SASLibrary ObjRef="$CLIENT.ADDSASLibrary$" Name="NW Sales"/>

```

```

</TablePackage>
<Columns>
  <Column
    Name="City"
    Desc="City of Sales Office"
    ColumnName="City"
    SASColumnName="City"
    ColumnType="12"
    SASColumnType="C"
    ColumnLength="32"
    SASColumnLength="32"
    SASFormat="$Char32."
    SASInformat="$32."/>
  <Column
    Name="Address"
    Desc="Street Address of Sales Office"
    ColumnName="Address"
    SASColumnName="Street_Address"
    ColumnType="12"
    SASColumnType="C"
    ColumnLength="32"
    SASColumnLength="32"
    SASFormat="$Char32."
    SASInformat="$32."/>
  <Column
    Name="Manager"
    Desc="Name of Operations Manager"
    ColumnName="Manager"
    SASColumnName="Manager"
    ColumnType="12"
    SASColumnType="C"
    ColumnLength="32"
    SASColumnLength="32"
    SASFormat="$Char32."
    SASInformat="$32."/>
  <Column
    Name="Employees"
    Desc="Number of employees"
    ColumnName="Employees"
    SASColumnName="Employees"
    ColumnType="6"
    SASColumnType="N"
    ColumnLength="3"
    SASColumnLength="3"
    SASFormat="3.2"
    SASInformat="3.2"/>
</Columns>
</PhysicalTable>
</Metadata>
<Reposid>$CLIENT.ADDRepositoryBase$ </Reposid>
<NS>SAS </NS>
<Flags>268435456</Flags>
<Options/>
</AddMetadata>

```

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

```

S,1346701949.827000,1,1,10,8.632412,2.193153,sastst ,,request::AddMetadata
C,1346701949.877000,1,1,11,1,10,8.662456,2.193153,sastst ,,PhysicalTable::
  AddMetadata
C,1346701949.917000,1,1,12,1,11,8.692499,2.203168,sastst ,,SASLibrary::
  omiPersistElement
P,1346701949.937000,1,1,12,8.702513,2.203168,0,sastst , 0,SASLibrary::
  omiPersistElement
U,1346701950.7000,1,1,11,8.772614,2.203168,1,sastst , 0,TablePackage::set
C,1346701950.7000,1,1,13,1,11,8.772614,2.203168,sastst ,,Column::
  omiPersistElement
C,1346701950.37000,1,1,14,1,13,8.782628,2.203168,sastst ,,Column::AddMetadata
C,1346701950.97000,1,1,15,1,14,8.832700,2.203168,sastst ,,PhysicalTable::
  omiPersistElement
C,1346701950.117000,1,1,16,1,15,8.852729,2.203168,sastst ,,*::
  omiVerifyAssociation
P,1346701950.127000,1,1,16,8.862744,2.203168,0,sastst , 2,*::
  omiVerifyAssociation
P,1346701950.127000,1,1,15,8.862744,2.203168,0,sastst , 0,PhysicalTable::
  omiPersistElement
U,1346701950.177000,1,1,14,8.912816,2.203168,1,sastst , 0,Table::set
P,1346701950.177000,1,1,14,8.912816,2.203168,0,sastst , 0,Column::
  AddMetadata
P,1346701950.177000,1,1,13,8.912816,2.203168,0,sastst , 0,Column::
  omiPersistElement
C,1346701950.177000,1,1,17,1,11,8.912816,2.203168,sastst ,,Column::
  omiPersistElement
C,1346701950.187000,1,1,18,1,17,8.922830,2.203168,sastst ,,Column::AddMetadata
C,1346701950.207000,1,1,19,1,18,8.942859,2.203168,sastst ,,PhysicalTable::
  omiPersistElement
C,1346701950.227000,1,1,20,1,19,8.962888,2.203168,sastst ,,*::
  omiVerifyAssociation
P,1346701950.238000,1,1,20,8.972902,2.203168,0,sastst , 2,*::
  omiVerifyAssociation
P,1346701950.238000,1,1,19,8.972902,2.203168,0,sastst , 0,PhysicalTable::
  omiPersistElement
U,1346701950.288000,1,1,18,9.22974,2.203168,1,sastst , 0,Table::set
P,1346701950.288000,1,1,18,9.22974,2.203168,0,sastst , 0,Column::
  AddMetadata
P,1346701950.288000,1,1,17,9.22974,2.203168,0,sastst , 0,Column::
  omiPersistElement
C,1346701950.288000,1,1,21,1,11,9.22974,2.203168,sastst ,,Column::
  omiPersistElement
C,1346701950.298000,1,1,22,1,21,9.32988,2.203168,sastst ,,Column::AddMetadata
C,1346701950.318000,1,1,23,1,22,9.53017,2.203168,sastst ,,PhysicalTable::
  omiPersistElement
C,1346701950.338000,1,1,24,1,23,9.73046,2.203168,sastst ,,*::
  omiVerifyAssociation
P,1346701950.348000,1,1,24,9.83060,2.203168,0,sastst , 2,*::
  omiVerifyAssociation
P,1346701950.348000,1,1,23,9.83060,2.203168,0,sastst , 0,PhysicalTable::
  omiPersistElement
U,1346701950.408000,1,1,22,9.143147,2.203168,1,sastst , 0,Table::set
P,1346701950.408000,1,1,22,9.143147,2.203168,0,sastst , 0,Column::

```

```

AddMetadata
P,1346701950.408000,1,1,21,9.143147,2.203168,0,sastst , 0,Column::
  omiPersistElement
C,1346701950.408000,1,1,25,1,11,9.143147,2.203168,sastst ,,Column::
  omiPersistElement
C,1346701950.418000,1,1,26,1,25,9.153161,2.203168,sastst ,,Column::AddMetadata
C,1346701950.438000,1,1,27,1,26,9.173190,2.203168,sastst ,,PhysicalTable::
  omiPersistElement
C,1346701950.458000,1,1,28,1,27,9.183204,2.213182,sastst ,,*::
  omiVerifyAssociation
P,1346701950.468000,1,1,28,9.193219,2.213182,0,sastst , 2,*::
  omiVerifyAssociation
P,1346701950.468000,1,1,27,9.193219,2.213182,0,sastst , 0,PhysicalTable::
  omiPersistElement
U,1346701950.528000,1,1,26,9.243291,2.223196,1,sastst , 0,Table::set
P,1346701950.528000,1,1,26,9.243291,2.223196,0,sastst , 0,Column::
  AddMetadata
P,1346701950.528000,1,1,25,9.243291,2.223196,0,sastst , 0,Column::
  omiPersistElement
P,1346701950.528000,1,1,11,9.243291,2.223196,0,sastst , 0,PhysicalTable::
  AddMetadata
P,1346701950.999000,1,1,10,9.493651,2.283283,0,sastst , 0,request::
  AddMetadata

```

Here are the records in the log:

- Record 1: Identifies an AddMetadata request.
- Record 2: Logs adding a PhysicalTable object (**Sales Offices** in the XML).
- Record 3: Logs call to persist associated SASLibrary element (**NW SALES** in the XML).
- Record 4: Persist of associated SASLibrary element got a successful return code **0**.
- Record 5: Setting TablePackage association got a successful return code **0**.
- Record 6: Logs call to persist associated Column element (**City** in the XML).
- Record 7: Logs AddMetadata of a new Column object (**City** in the XML).
- Record 8: Logs call to persist associated PhysicalTable for required association.
- Record 9: Logs call to omiVerifyAssociation to check for required association.
- Record 10: Return code (**2**) from omiVerifyAssociation indicates no existing association.
- Record 11: Persist of associated PhysicalTable element got a successful return code **0**.
- Record 12: Setting required Table association got a successful return code **0**.
- Record 13: AddMetadata for Column **City** got a successful return code **0**.
- Record 14: Persist of associated Column **City** got a successful return code **0**.
- Record 15-23: Same as records 6-14 for Column **Address**.
- Record 24-32: Same as records 6-14 for Column **Manager**.
- Record 33-41: Same as records 6-14 for Column **Employees**.
- Record 42: AddMetadata for PhysicalTable **Sales Office** got a successful return code **0**.
- Record 43: AddMetadata request got a successful return code **0**.

In a multi-threaded environment, the ARM records describing one OMA transaction might be interleaved with records describing other OMA transactions in the ARM_OMA

log. When this occurs, use the application identifier, ARM transaction class identifier, and ARM transaction identifiers to keep track of transactions. As an example, here are the first 14 ARM records from the previous example:

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

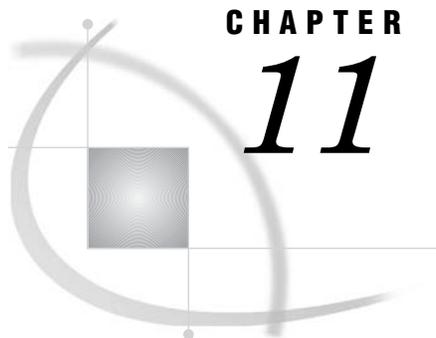
The following is a description of how the ARM records are nested:

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

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

Post-processing the ARM_OMA Log

SAS provides macros for post-processing an ARM log. See "SAS ARM Macros" in the *SAS Language Reference: Dictionary*.



CHAPTER

11

Administering Logging for the SAS Object Spawner, SAS Workspace Server, SAS Stored Process Server, and SAS/CONNECT Server

<i>Enabling Logging for the SAS Object Spawner</i>	89
<i>About Object Spawner Logging</i>	89
<i>Use SAS Management Console to Enable Object Spawner Logging</i>	90
<i>Use a telnet Client to Enable Object Spawner Logging</i>	90
<i>Adding Options to the Object Spawner Script to Enable Logging</i>	90
<i>About Logging Options for the Object Spawner Script</i>	90
<i>Add Logging Options When the Object Spawner Runs as a Windows Service</i>	91
<i>Add Logging Options When the Object Spawner Is Started From a Script</i>	91
<i>Enabling Logging for SAS Workspace Servers</i>	92
<i>About SAS Workspace Server Logging</i>	92
<i>Use SAS Management Console to Enable SAS Workspace Server Logging</i>	92
<i>Create a Configuration File in Which to Specify the Workspace Server Logging Options</i>	93
<i>Administering Logging for SAS Stored Process Servers</i>	93
<i>Enabling Increased Logging for the SAS/CONNECT Spawner</i>	94
<i>About SAS/CONNECT Spawner Logging</i>	94
<i>Enable Increased Logging When the SAS/CONNECT Spawner Runs as a Windows Service</i>	94
<i>Enable Increased Logging When the SAS/CONNECT Spawner Is Started From a Script</i>	94

Enabling Logging for the SAS Object Spawner

About Object Spawner Logging

Effective with SAS 9.1.3 Service Pack 4, logging for the SAS Object Spawner is disabled by default because of the amount of disk space that is required and because some of the logging options are designed primarily for debugging.

If you need to view log information for the spawner, you can enable logging by using various methods, depending on the purpose for the logs.

- If you want to enable logging for a purpose other than debugging (for example, to track which clients are requesting server connections and when the requests are occurring), use one of the following methods:
 - “Use SAS Management Console to Enable Object Spawner Logging” on page 90
 - “Use a telnet Client to Enable Object Spawner Logging” on page 90
- To troubleshoot problems with starting the object spawner, connecting to the object spawner, or connecting to a workspace server or stored process server, see “Adding Options to the Object Spawner Script to Enable Logging” on page 90. You should use this logging method if technical support requests that you provide a log.

The following path and filename are recommended for object spawner logs:

SAS-config-dir\Lev1\SASMain\ObjectSpawner\logs\objspawn.log

Note: For information about starting and stopping the object spawner, see Chapter 4, “Starting, Stopping, and Pausing Servers,” on page 29. For additional reference information, see “Spawner Invocation Options” and “Spawner Error Messages” in the *SAS Intelligence Platform: Application Server Administration Guide*. Δ

Use SAS Management Console to Enable Object Spawner Logging

If you want to enable object spawner logging in order to track which clients are requesting server connections and when the requests are occurring, follow these steps:

- 1 In SAS Management Console, expand the **Server Manager** node.
- 2 Right-click the spawner object (for example, **SAS Main - Spawner**) and select **Properties**.
- 3 Do the following on the **Initialization** tab:
 - a Select the **Verbose** check box if you want to enable detailed logging.
 - b In the **Log file** box, enter a path and filename for the log file.
 - c Click **OK**.
- 4 Restart the object spawner.

Use a telnet Client to Enable Object Spawner Logging

If you want to enable object spawner logging in order to track which clients are requesting server connections and when the requests are occurring, you can use a telnet client to connect to the spawner and then issue commands to start and end logging. This method does not require you to restart the spawner.

Follow these steps:

- 1 While the object spawner is running, use a telnet client to connect to the spawner’s operator interface port.

Note: The default operator port number is 8581. To determine the operator port number that has been defined for your object spawner, select the spawner object in SAS Management Console (for example, **SAS Main - Spawner**). The port number for the Operator protocol connection is displayed in the right pane. Δ

- 2 When the connection is established, issue the **btrace** (begin trace) command by using the following syntax:

```
btrace path-filename
```

where *path-filename* is the fully qualified path to the file that is to be used for logging object spawner activity.

- 3 When you want to discontinue logging, issue the command **etrace** (end trace).

Adding Options to the Object Spawner Script to Enable Logging

About Logging Options for the Object Spawner Script

To troubleshoot problems with starting the object spawner, connecting to the object spawner, or connecting to a workspace server or stored process server, you need to log metadata information. To do so, you must add the following options to the script that is used to start the object spawner:

-saslogfile filename

Specifies the fully qualified path to the file where the log is to be stored.

-sasVerbose

Specifies that the object spawner is to record more detail in the log file.

Use one of the following procedures, depending on how your spawner is started:

- “Add Logging Options When the Object Spawner Is Started From a Script” on page 91
- “Add Logging Options When the Object Spawner Runs as a Windows Service” on page 91

You should use one of these logging methods if technical support requests that you provide a log.

Add Logging Options When the Object Spawner Runs as a Windows Service

If your object spawner is installed as a Windows service, then follow these steps to add logging options to the spawner start-up script:

- 1 Stop the service in the Windows Services manager by using the command **ObjectSpawner.bat -stop**. The default location for the **ObjectSpawner.bat** file is as follows:

```
SAS-config-dir\Lev1\SASMain\ObjectSpawner
```

- 2 Remove the spawner service by using the command **ObjectSpawner.bat -remove**.
- 3 Open the **ObjectSpawner.bat** file in a text editor.
- 4 In the file, locate the line that contains the following string:

```
:install
```

The next line contains a **start** command. Add the following options to that command:

- saslogfile filename**
- sasVerbose**

- 5 Save the file.
- 6 Reinstall the service by using the command **ObjectSpawner.bat -install**.

Add Logging Options When the Object Spawner Is Started From a Script

If your object spawner is started from a script, then follow these steps to add logging options to the spawner start-up script:

- 1 Use the appropriate script to stop the object spawner.
- 2 Open the spawner invocation script (**ObjectSpawner.bat** or **ObjectSpawner.sh**) in an editor. The default location for the file is as follows:

```
SAS-config-dir\Lev1\SASMain\ObjectSpawner
```

- 3 In the file, locate the section that begins with the following string:

```
:start2
```

Locate the **start** command in that section, and add the following options to the command:

- saslogfile filename**
- sasVerbose**

- 4 Save the file, and restart the object spawner.

Enabling Logging for SAS Workspace Servers

About SAS Workspace Server Logging

Logging for the SAS Workspace Server is disabled by default because of the amount of disk space that is required and because the logs are primarily designed to be used for debugging.

If you need to view log information for the workspace server, follow these recommendations:

- Do not enable logging by updating `sasv9.cfg`, because this file is used by other servers.
- To avoid performance degradation, configure logging so that a unique log file is generated for each SAS session. To do so, specify a log filename that includes a unique session ID or a unique time and date stamp, and specify the option `rollover=session`.
- The log files are not removed automatically, so you might want to delete or archive them periodically to conserve disk space.

You can use either of the following procedures to enable SAS Workspace Server logging:

- “Use SAS Management Console to Enable SAS Workspace Server Logging” on page 92
- “Create a Configuration File in Which to Specify the Workspace Server Logging Options” on page 93

Note: For detailed information about the LOG and LOGPARM options, see the *SAS Language Reference: Dictionary*. For detailed information about object server parameters, see the *SAS Intelligence Platform: Application Server Administration Guide*. \triangle

Use SAS Management Console to Enable SAS Workspace Server Logging

If you want to use SAS Management Console to enable SAS Workspace Server logging, follow these steps:

- 1 In SAS Management Console, expand the **Server Manager** node, the application server node (for example, **SASMain**), and the logical workspace server (for example, **SASMain - Logical Workspace Server**).
- 2 Right-click the workspace server object (for example, **SAS Main - Workspace Server**), and select **Properties**.
- 3 On the **Options** tab, under **Launch Commands**, add the appropriate LOG and LOGPARM options to the end of the **Command** line. The following options are recommended:
 - `log "SAS-config-dir\Lev1\SASMain\WorkspaceServer\logs\WorkspaceServer_%v.log"`
 - `logparm "rollover=session open=replaceold write=immediate"`
 Be sure to include the quotation marks as shown.
- 4 Click **OK**.
- 5 Restart the object spawner.

Create a Configuration File in Which to Specify the Workspace Server Logging Options

If you want to specify the logging options in a configuration file instead of in the workspace server definition, follow these steps:

- 1 Use a text editor to create a file similar to the following. As you create the file, substitute the appropriate path names for your installation.

```

/*
 *   This config file contains the specific options needed for the SAS
 *   Workspace Server.
 *
 *   Variables Used:
 *
 *   $APPSERVER_DIR=c:\SAS\EntBIServer\Lev1\SASMain
 */
-config "C:\SAS\EntBIServer\Lev1\SASMain\sasv9.cfg"
-nosplash
-noterminal
-log "C:\SAS\EntBIServer\Lev1\SASMain\WorkspaceServer\logs\WorkspaceServer_%v.log"
-logparm "rollover=session open=replaceold write=immediate"
-pagesize max
-linesize max
-sasuser "C:\SAS\EntBIServer\Lev1\SASMain\WorkspaceServer\sasuser"
-rsasuser

```

- 2 Name the file **sasv9_WorkspaceServer.cfg**, and save it in the path *SAS-config-dir\Lev1\SASMain*.
- 3 In SAS Management Console, expand the **Server Manager** node, the application server node (for example, **SASMain**), and the logical workspace server (for example, **SASMain - Logical Workspace Server**).
- 4 Right-click the workspace server object (for example, **SAS Main - Workspace Server**) and select **Properties**.
- 5 On the **Options** tab, under **Launch Commands**, update the **Command** box so that it points to the new configuration file, as follows:


```
sas -config "SAS-config-dir\Lev1\SASMain\sasv9_WorkspaceServer.cfg".
```
- 6 Click **OK**.
- 7 Restart the spawner.

Administering Logging for SAS Stored Process Servers

Logging for SAS Stored Process Servers is enabled by default. The default location for the log is as follows:

```

SAS-config-dir\Lev1\SASMain\StoredProcessServer\logs\
Stored_Process_Server_%v.log

```

The Configuration Wizard sets the following default logging options for this server:

```
- logparm "rollover=session open=replaceold write=immediate"
```

If you want to change the log location or the logging options, you can do so by modifying the following file:

```
sas-config-dir\Lev1\SASMain\StoredProcessServer\sasv9_StorProcSrv.cfg
```

A new log file is created for each server session. The log files are not removed automatically, so you might want to delete or archive them periodically to conserve disk space.

Enabling Increased Logging for the SAS/CONNECT Spawner

About SAS/CONNECT Spawner Logging

By default, only minimal logging is enabled for the SAS/CONNECT Spawner. To increase the amount of information that is written to the log, you can add the **-verbose** logging option. To add this option, use one of the following methods, depending on how your spawner is started:

- “Enable Increased Logging When the SAS/CONNECT Spawner Runs as a Windows Service” on page 94
- “Enable Increased Logging When the SAS/CONNECT Spawner Is Started From a Script” on page 94

The following path and filename are recommended for SAS/CONNECT Spawner logs:
SAS-config-dir\Lev1\SASMain\ConnectServer\logs\spawner.log

Note: For information about starting and stopping the SAS/CONNECT Spawner, see Chapter 4, “Starting, Stopping, and Pausing Servers,” on page 29. For detailed information about SAS/CONNECT, see the *SAS/CONNECT User’s Guide*. Δ

Enable Increased Logging When the SAS/CONNECT Spawner Runs as a Windows Service

If your SAS/CONNECT Spawner is installed as a Windows service, then follow these steps to enable increased logging:

- 1 Stop the service by using the command **ConnectServer.bat -stop**. The default location for the **ConnectServer.bat** file is as follows:
SAS-config-dir\Lev1\SASMain\ConnectServer
- 2 Remove the SAS/CONNECT Spawner service by using the command **ConnectServer.bat -remove**.
- 3 Open the **ConnectServer.bat** file in a text editor.
- 4 In the file, locate the line that contains the following string:

```
:install
```

The next line contains a **start** command. Add the following option to that command:

```
-verbose
```

Note: The command should already include the following option:

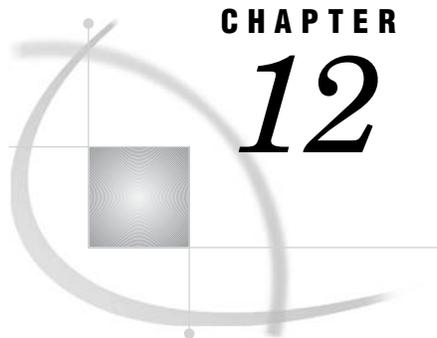
```
-saslogfile filename  $\Delta$ 
```

- 5 Save the file.
- 6 Reinstall the service by using the command **InstallConnectServer.bat**.

Enable Increased Logging When the SAS/CONNECT Spawner Is Started From a Script

If your SAS/CONNECT Spawner is started from a script, then follow these steps to enable increased logging:

- 1 Use the appropriate script to stop the SAS/CONNECT Spawner.
- 2 Open the spawner invocation script (**ConnectServer.bat** or **ConnectServer.sh**) in an editor. The default location for the file is as follows:
`SAS-config-dir\Lev1\SASMain\ConnectServer`
- 3 In the file, locate the section that begins with the following string:
:start2
Locate the **start** command in that section, and add the following option to the command:
-verbose
Note: The command should already include the following option:
-saslogfile filename Δ
- 4 Save the file, and restart the spawner.



CHAPTER

12

SAS OLAP Server Monitoring and Logging

<i>Overview of SAS OLAP Server Monitoring and Logging</i>	97
<i>Administering the OLAP Server Log</i>	97
<i>About SAS OLAP Server Logging</i>	97
<i>Increasing the SAS OLAP Server Logging Level</i>	98
<i>Using ARM to Monitor SAS OLAP Server Performance</i>	98
<i>Using the SAS OLAP Server Monitor</i>	99
<i>Start the SAS OLAP Server Monitor</i>	99
<i>View OLAP Server Sessions and Queries</i>	100
<i>Refresh the List of OLAP Servers or Server Connections</i>	100

Overview of SAS OLAP Server Monitoring and Logging

You can monitor SAS OLAP Servers by using the following tools:

- the SAS OLAP Server log
- the Application Response Measurement (ARM) interface
- SAS OLAP Server Monitor in SAS Management Console

For instructions, see the following topics.

Administering the OLAP Server Log

About SAS OLAP Server Logging

Logging for SAS OLAP Servers is enabled by default. The default location for the log is as follows:

sas-config-dir\Lev1\SASMain\OLAPServer\logs\OLAPServer#d#b#y.log

The Configuration Wizard sets the following default logging options for this server:

- **logparm "rollover=auto open=replaceold write=immediate"**

If you want to change the log location or the logging options, you can do so by modifying the following file:

SAS-config-dir\Lev1\SASMain\OLAPServer\sasv9_OLAPServer.cfg

New log files are created each day. The log files are not removed automatically, so you might want to delete or archive them periodically to conserve disk space.

Increasing the SAS OLAP Server Logging Level

If you need to research server errors, you can change the logging levels by following these steps.

CAUTION:

Increasing the logging levels can degrade performance. Be sure to restore the default log settings when the research is completed. Δ

- 1 Open the following file in a text editor:

SAS-config-dir\Lev1\SASMain\OLAPServer\sasv9_OLAPServer.cfg

- 2 Update the OBJECTSERVERPARMS statement to include one or more of the following options:

IOMLEVEL=0 | 1 | 2 | 3

traces client interactions when the level is greater than the default level 0. Levels 2 and 3 are reserved for use with SAS support representatives. Long queries may require you to increase the values of JNLSTRMAX= and JNLLINEMAX=.

PELEVEL=0 | 1 | 2 | 3

traces packets through the protocol engines BRIDGE and COM when the level is greater than 0. Level 3 specifies the most verbose log entries.

APPLEVEL=0 | 1 | 2 | 3

traces application logic. The default level is 1. Higher levels produce may more verbose log entries.

LOGFILE=*filename*

specifies the location of the log file.

JNLSTRMAX=*max-line-length*

truncates lines of MDX queries after they reach a specified maximum number of characters. The suggested value is 4096.

JNLLINEMAX=*max-lines*

truncates MDX queries after they reach a specified maximum number of lines. The suggested value is 4096.

- 3 Save the file.

Using ARM to Monitor SAS OLAP Server Performance

You can use the Application Response Measurement (ARM) interface to monitor the following information about SAS OLAP Servers:

- application behavior
- user behavior and usage
- server loads
- cube optimization (query response time)
- cube metrics, including counts of connections and queries

You can also use the ARM interface to optimize aggregations for commonly submitted queries. The Advanced Aggregation Tuning feature in SAS OLAP Cube Studio reads a specified ARM log file and suggests new aggregations.

CAUTION:

Enabling the ARM interface can degrade server performance. Enable the ARM interface only when you need to analyze server performance. △

The ARM interface is enabled by asserting system options on the SAS OLAP Server. Follow these steps to enable the ARM interface on a SAS OLAP Server:

- 1 Close sessions and stop the SAS OLAP Server by using the SAS OLAP Server Monitor in SAS Management Console. For instructions, see “Using the SAS OLAP Server Monitor” on page 99.
- 2 In a text editor, open the SAS OLAP Server configuration file at the following location:
`SAS-config-dir\Lev1\SASMain\OLAPServer\sasv9_OLAPServer.cfg.`
- 3 Add the following system options to the configuration file :
`ARM_SUBSYS=(ARM_OLAP_SESSION)`
 records OLAP transactions.
`ARMLOC=filename`
 specifies the location of the ARM log file. The default location is `ARMLOG.LOG`.
- 4 Save and close the configuration file.
- 5 Restart the SAS OLAP Server.
- 6 Use a single cube viewer to submit the queries that you are interested in.
- 7 Stop the SAS OLAP Server.
- 8 Remove the ARM system options from the SAS OLAP Server configuration file, save the file, and close the file.
- 9 Restart the SAS OLAP Server and resume normal operation.

For information about the contents of the ARM log file, see “Understanding the Records Written by the ARM_OMA Subsystem” on page 79.

For information about how to use the ARM interface, see “Monitoring Performance Using Application Response Measurement (ARM)” in *SAS Language Reference: Concepts*.

Using the SAS OLAP Server Monitor

Start the SAS OLAP Server Monitor

The SAS OLAP Server Monitor in SAS Management Console enables you to perform the following tasks:

- connect to SAS OLAP Servers
- refresh the server list, the connections list, and cube metadata
- monitor sessions and queries
- close sessions
- stop SAS OLAP Servers

Follow these steps to access the SAS OLAP Server Monitor:

- 1 Start SAS Management Console. If necessary, choose a metadata profile that has the Administrator permission.
- 2 In the navigation tree, expand the **Monitoring** node and the **SAS OLAP Server Monitor** node.

- 3 Right-click the name of the logical OLAP server, and select **Connect**. Enter your user ID and password at the prompt.
- 4 When you are connected, right-click the SAS Logical Server name and select an action.
- 5 When you are finished using the monitor, right-click the name of the logical OLAP server and select **Disconnect**.

Note: Online help becomes available when you select SAS OLAP Server Monitor in the navigation tree. Δ

View OLAP Server Sessions and Queries

To view the active SAS OLAP Server sessions and queries, follow these steps:

- 1 Start the OLAP Server Monitor, as described in the preceding topic.
- 2 Click the plus sign (+) next to the SAS Logical Server name to see a list of active sessions.

If the server has active connections, then the following information for each active session is displayed in a table:

- session name.
- description.
- session owner (the ID of the user who initiated the session). If the server is not secured, then the session owner is displayed as **<unknown>**.
- time (in seconds) that the server has been inactive.
- number of open results sets.

If there are no active sessions, then an error message is displayed in the table.

- 3 To list the active queries for a session, click the session entry in the table.

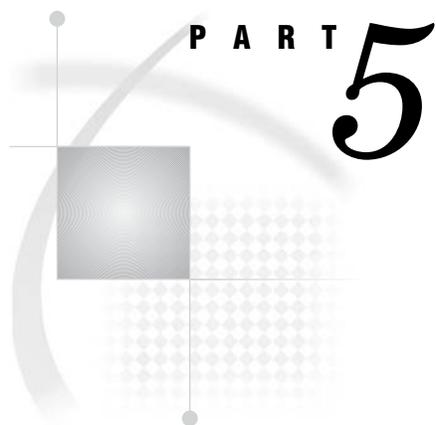
For each active query, SAS Management Console displays the type of query (unspecified, multidimensional, or flattened) and the MDX expression.

Refresh the List of OLAP Servers or Server Connections

After you have started the SAS OLAP Server Monitor, you can do the following:

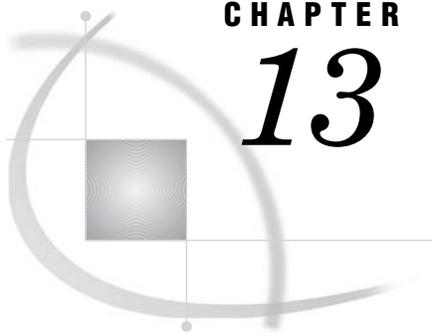
- To refresh the list of SAS OLAP Servers, right-click **SAS OLAP Server Monitor** and select **Refresh**.
- To refresh the list of connections to a SAS OLAP Server, expand the **Server Manager** node, the SAS Application Server node (for example **SASMain**), and the **Logical OLAP Server** node in the navigation tree. Then right-click the OLAP server name, and select **Refresh**.

Note: For information about refreshing the cube metadata for one or more cubes in the SAS OLAP Server's cube cache, see the *SAS Intelligence Platform: Application Server Administration Guide*. Δ



Optimizing the Performance of the SAS Metadata Server

- Chapter 13* **Using Configuration Settings to Optimize Metadata Server Performance** 103
- Chapter 14* **Managing Memory for Performance** 109



CHAPTER

13

Using Configuration Settings to Optimize Metadata Server Performance

<i>Configuring the Number of Threads Used by the Metadata Server</i>	103
<i>About the MAXACTIVETHREADS, THREADSMIN, and THREADSMAX Options</i>	103
<i>Change the MAXACTIVETHREADS, THREADSMIN, and THREADSMAX Options</i>	104
<i>Using Workunit Journaling to Optimize Performance on Multiprocessor Hosts</i>	104
<i>What Is Workunit Journaling?</i>	104
<i>Configure Workunit Journaling</i>	105
<i>Avoiding Changes to the Repository That Adversely Affect Performance</i>	107
<i>Determining Whether the COMPRESS or REUSE Attribute Has Been Set</i>	107
<i>Repairing Problematic Data Sets</i>	107

Configuring the Number of Threads Used by the Metadata Server

About the MAXACTIVETHREADS, THREADSMIN, and THREADSMAX Options

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

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

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

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

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

Change the MAXACTIVETHREADS, THREADSMIN, and THREADSMAX Options

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

- 1 Open the file **omaconfig.xml** in a text editor. This file is located in the directory *SAS-config-dir\Lev1\SASMain*.
- 2 Edit the value of the **MAXACTIVETHREADS** attribute of the **<OMA>** element. Change the value of the attribute from *number-of-processors* to a lower value.
- 3 Save your changes to **omaconfig.xml**.
- 4 Open the file **sasv9_MetadataServer.cfg** in a text editor. This file is located in the directory *SAS-config-dir\Lev1\SASMain\MetadataServer*.
- 5 Change the values of the **THREADSMIN** and **THREADSMAX** options in the string that serves as the argument to the **OBJECTSERVERPARMS** option, as shown below:


```
THREADSMIN=(MAXACTIVETHREADS*2)+1 THREADSMAX=(MAXACTIVETHREADS*2)+1
```
- 6 Save your changes to **sasv9_MetadataServer.cfg**.
- 7 Restart the metadata server.

Using Workunit Journaling to Optimize Performance on Multiprocessor Hosts

What Is Workunit Journaling?

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

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

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

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

- Updated records are written to two additional areas: the workunit journal file and a workunit buffer.

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

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

The progress of the repository update process is tracked in the metadata server log. If an error occurs when the update process is writing to repositories, an alert e-mail message is sent to the metadata server administrator so that the administrator can respond to the problem.

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

Configure Workunit Journaling

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

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

WORKUNITJOURNALPATH="*filename*"

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

WORKUNITJOURNALSIZE="*size-in-bytes*"

The **WORKUNITJOURNALSIZE** attribute specifies the size of the workunit journal file in bytes. We recommend that you start with a file size of 20MB (2000000). This space is reserved in the file system and is unavailable to other processes.

ALERTEMAIL="*email-address | (email-address-1, email-address-2 ...)*"

The **ALERTEMAIL** attribute specifies the e-mail address to which the metadata server should send mail if an error occurs while it is updating the workunit journal file. The address that you specify will appear in the **To** field of the e-mail message. To specify more than one address (for example, work and home

e-mail addresses) enclose the addresses in parentheses and separate them with a comma.

ALERTEMAILATTACHMENT="*filename*"

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

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

```
<OMA ADMINUSERS="MetadataServer/adminUsers.txt"
  TRUSTEDUSERS="MetadataServer/trustedUsers.txt"
  GROUPSFORMEMBERASSOCDELETE="DESKTOP_PORTALPAGES_GROUP,
    Portlet Template Group,
    OrderingGroup,
    DESKTOP_PAGEHISTORY_GROUP,
    Portal Collection"
  WORKUNITJOURNALPATH="WorkunitJournal.dat"
  WORKUNITJOURNALSIZE="2000000"
  ALERTEMAIL="sastest@company.com"
  ALERTEMAILATTACHMENT="omaconfig.xml"/>
```

There are two additional attributes that you might need to add to the **<OMA>** element: **WORKUNITJOURNALYIELDCOUNT** and **ALERTEMAILTEST**. The first specifies how many updates the metadata server can make to the workunit journal file before it must yield to another internal task. (You do not usually change the value of this option.) The second is a temporary option that enables you to determine whether the options previously discussed for the alert e-mail have valid values. For further information on this subject, see “Reference Information for omaconfig.xml” on page 254. This section also includes more in-depth discussions of the options shown in the preceding code snippet.

- 3 Save the file **omaconfig.xml**.
- 4 Open the file **sasv9_MetadataServer.cfg** in the directory *SAS-config-dir\Lev1\SASMain\MetadataServer*.
- 5 Add the following invocation parameters to the configuration file. The metadata server needs this information in order to send the alert e-mails mentioned previously.

-emailsys SMTP

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

-emailhost mail-server

Specify the full name of your mail-server host.

-emailid account-of-sender

Specify the e-mail address that you want to appear in the **From** field of an alert e-mail. The **EMAILID** option supports e-mail addresses in two forms:

"server-name<user-ID@domain>" and *"<user-ID@domain>"*. If you have multiple metadata servers (for example, development, test, and production servers), you can include a server name to indicate which server is sending the e-mail.

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

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

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

Avoiding Changes to the Repository That Adversely Affect Performance

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

Determining Whether the COMPRESS or REUSE Attribute Has Been Set

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

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

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

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

Repairing Problematic Data Sets

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

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

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

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

```

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

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

```

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

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

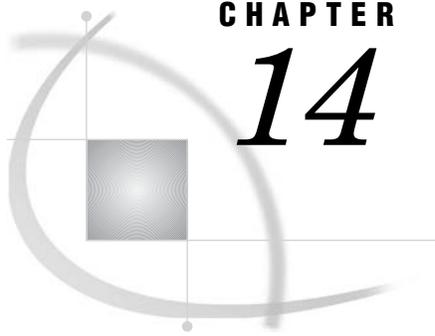
```

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

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

```

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



CHAPTER

14

Managing Memory for Performance

<i>Ensuring Adequate Memory</i>	109
<i>About Metadata Server Memory Usage</i>	109
<i>Input/Output Balancing</i>	109
<i>Concurrent Queries/Clients</i>	110
<i>Estimating Metadata Storage Requirements</i>	110
<i>Estimating Memory for Metadata Queries</i>	111
<i>Estimating Total Memory: Example</i>	111
<i>Note to Windows Users</i>	112
<i>Using %OMARUNAN (or the RUNANALYSIS Option of %OMABAKUP) to Optimize Memory Usage</i>	113
<i>About %OMARUNAN and the RUNANALYSIS Option of %OMABAKUP</i>	113
<i>When to Optimize Memory Usage</i>	113
<i>How Memory Optimization Works</i>	113
<i>Choosing Whether to Use %OMARUNAN or the RUNANALYSIS Option of %OMABAKUP</i>	114
<i>Using the %OMARUNAN Macro</i>	114
<i>%OMARUNAN Autocall Macro: Reference</i>	115
<i>Syntax</i>	115
<i>Details</i>	115
<i>Recovering Memory on the SAS Metadata Server</i>	115

Ensuring Adequate Memory

About Metadata Server Memory Usage

The SAS Metadata Server is an “in-memory” server. That is, all queried and updated metadata is held in memory by the server, so there should be more RAM than metadata to avoid excessive paging.

To ensure that sufficient memory is available on Windows and UNIX systems, set a MEMSIZE value in the start server command that allows the operating system to set the memory limits on the server. This can be accomplished by setting a value of 0 (zero), MIN, or MAX.

Input/Output Balancing

Because all metadata is held in memory, and because all updates to the metadata are single-threaded, you do not have to configure input/output contentions. Queries cause I/O activity only when the metadata is initially being loaded into memory based on type. For example, if one user requests metadata for a single Column, all Column

metadata is loaded into memory, and no I/O for Column metadata queries should happen again until the server is paused and resumed or stopped and restarted.

Concurrent Queries/Clients

The number of processors on the server determines the number of concurrent queries that can run on the server.

If you expect a small volume of concurrent clients, you can probably use a single-processor machine. If you are expecting 1,000 concurrent clients, you will want an eight-processor machine.

When configuring the metadata server, set the **MAXACTIVETHREADS** option in the **omaconfig.xml** server configuration file to the number of processors. Also be sure to set the THREADSMIN (TMIN) and THREADSMAX (TMAX) OBJECTSERVERPARMS options in the metadata server start-up command such that TMIN=TMAX and TMAX=(#_processors * 2) + 1. For more information about how the metadata server handles concurrent queries and threads, see “Configuring the Number of Threads Used by the Metadata Server” on page 103.

Estimating Metadata Storage Requirements

The metadata storage requirement is largely fixed and can be roughly estimated by using the following formula:

$$50\text{MB} + \text{size_of_all_repositories}$$

In this formula, 50MB is the approximate amount of memory that the server uses to register the first repository, and **size_of_all_repositories** is roughly equal to the size of all of the metadata type containers in all repositories on the metadata server.

A metadata type container is a SAS data set that stores the attributes of metadata object instances as observations. A metadata type container is created when the first object instance of that type is created in a repository. Therefore, the amount of required RAM increases as containers are added, until each repository on the server has containers for all of the metadata types in the SAS Metadata Model.

The server also creates several containers in each repository that do not map to a metadata type. For example:

- an MDASSOC container stores information about all metadata object associations
- a TXTPAGE stores overflow information from metadata type containers
- an MRRGSTRY container stores information about the object IDs that are used in the repository
- a CNTAINER container maintains a list of other containers that have been created

The memory requirement for stored metadata is also affected by:

- the number of observations created in each container
- the value of the PHYSICAL_DELETE server configuration option. When the PHYSICAL_DELETE option is on (the default), then the memory used by an observation will be recovered when a metadata object instance is deleted. When PHYSICAL_DELETE is turned off, then an observation for every metadata object ever created will be held in memory until the deleted records are purged from the repository.

A downloadable program called **ContainerStatsExport.sas** measures the size of the containers in a repository. The program also reports statistics about the containers that can be used to manage memory, such as the observation length, and the number of logical observations versus the number of physical observations. If you create a

repository that contains at least one instance of every metadata type that you intend to store, you can get a minimum estimate of the container storage requirement for a repository and use the observation length to extrapolate observation storage requirements. The formula for extrapolating observation storage requirements is as follows:

$$\text{observation_memory} = \text{number_of_physical_observations} * \text{observation_length}$$

To download **ContainerStatsExport.sas**, go to the following URL: <http://support.sas.com/rnd/eai/ContainerStatsExport/ContainerStatsExport.zip>. The file **ContainerStatsExport.zip** contains the program and a document called **usetool.html**, which contains instructions for using the program.

Estimating Memory for Metadata Queries

The amount of memory required for queries will depend on the number of concurrent clients on the server, the number of threads that are allocated to handle the requests, and the type of queries that are made.

- A *client* is a program or application that is connected to the metadata server, such as SAS ETL Studio, SAS Management Console, or a SAS session initiated via a METABROWSE command, PROC METADATA, PROC METAOPERATE, or the SAS metadata LIBNAME engine.
- A *thread* is a workspace on the server for handling requests. The number of active threads is configured by using the **MAXACTIVETHREADS** server configuration option. The **MAXACTIVETHREADS** option should be set to the number of processors times two (**#_processors * 2**). For example, the default value of 8 expects to allocate requests across 4 processors.

Each client uses approximately 256K of queueing space as it waits for an active thread. Therefore, the amount of memory required for a given set of concurrent clients is estimated as follows:

$$256K * \text{number_of_concurrent_clients}$$

The amount of memory used by a thread varies based on the type and the frequency of the requests that are launched in that thread. For example, a query that gets all of the attributes and associations of an OLAP cube will be more intensive than a query that gets information about the keys defined for a table. Also, a query that traverses repositories will be more memory-intensive than a query that reads objects within a single repository. In terms of frequency, a trusted user connection that makes metadata requests on behalf of multiple users might make more requests overall than a client connection established by a single user. However, memory used for queries is temporary – memory is held only for the duration of a given query and then it is released. In the absence of definitive information, it is recommended that you allocate a minimum of 20M per thread, as shown in this formula:

$$20M * \text{value_of_MAXACTIVETHREADS}$$

Estimating Total Memory: Example

Suppose that a metadata server runs on a single-processor host that contains five repositories and serves two clients. Using the **ContainerStatsExport.sas** tool, we have determined that the repositories will require a total of 61MB on disk.

Based on the information we have, we can determine the total memory requirement using the following formula:

```
50MB + size_of_all_repositories + (256K * number_of_concurrent_clients)
+ (20M * value_of_MAXACTIVETHREADS)
```

This formula provides the following results:

```
50MB overhead + 61MB = 111MB metadata storage requirement
256KB queuing space * 2 clients = 512KB client requirement
20MB thread memory * (1 processor * 2) = 40MB thread requirement
```

```
512KB + 40MB =
  512 * 1024 bytes = 524,288 bytes
  40 * 1048576 bytes = 41,943,040 bytes
41,943,040 + 524,288 = 42,467,328 bytes
42,467,328/1048576 = 40.5MB query memory requirement
```

```
111MB metadata storage requirement + 40.5MB query memory requirement= 151.5MB total RAM
```

Note to Windows Users

If you have determined that your memory requirements will approach 2GB, you should be aware of the 2GB per-process address limit on Windows hosts. The following Windows hosts support a 4GT RAM Tuning feature that will enable you to increase the per-process address limit to 3GB:

- Microsoft Windows NT 4.0 Server, Enterprise Edition
- Microsoft Windows 2000 Advanced Server
- Microsoft Windows 2000 Datacenter Server
- Microsoft Windows XP Professional Edition
- Microsoft Windows Server 2003 Standard Edition
- Microsoft Windows Server 2003 Enterprise Edition
- Microsoft Windows Server 2003 Datacenter Edition

If your memory requirements exceed 3GB, then you should consider running your metadata server in a host environment that has a higher per-process address space. For more information about the 4GT RAM Tuning feature, see "4GT RAM Tuning" in the Microsoft MSDN Library at <http://msdn.microsoft.com/library>.

The following is an example of a **boot.ini** file that specifies a 3GB address limit:

```
[boot loader]
timeout=5 default=multi(0)disk(0)rdisk(0)partition(1)\WINNT
[operating systems]
multi(0)disk(0)rdisk(0)partition(1)\WINNT="Microsoft Windows XP
Professional" /fastdetect /3GB
```

Using %OMARUNAN (or the RUNANALYSIS Option of %OMABAKUP) to Optimize Memory Usage

About %OMARUNAN and the RUNANALYSIS Option of %OMABAKUP

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

To optimize the usage of memory on the metadata server, you can do either of the following:

- Set the RUNANALYSIS parameter to “YES” when you use %OMABAKUP to perform a backup. This causes the %OMARUNAN macro to be executed after the backup is complete and after client activity has resumed on the metadata server.
- Execute the %OMARUNAN macro after you have performed a backup.

In each data set, the RUNANALYSIS option (or the %OMARUNAN macro) analyzes the handling of character variables. The analysis checks for such items as unnecessary indexes, long character variables that can be stored as variable-length strings, and duplicate strings.

The next time %OMABAKUP is used to perform a backup, it uses the results of the analysis to perform memory optimization tasks. To take advantage of the analysis, consider executing a second backup anytime you use the RUNANALYSIS option of %OMABAKUP (or anytime you use the %OMARUNAN macro).

CAUTION:

If you run %OMABAKUP in restore mode after having used %OMARUNAN or the RUNANALYSIS %OMABAKUP option, the optimizations will be lost. The RESTORE option deletes the DS_VCHAR data sets from the backup directories before restoring repositories to their registered locations. Δ

When to Optimize Memory Usage

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

How Memory Optimization Works

The following processing occurs when you run the %OMARUNAN macro, or when you use %OMABAKUP with the RUNANALYSIS option:

- The subprogram accesses the repository data sets in the backup destination. In each data set, it analyzes the handling of character variables. The analysis checks for such items as unnecessary indexes, long character variables that can be stored as variable-length strings, and duplicate strings.
- For each repository, the subprogram stores the results of the analysis in a data set called DS_VCHAR. This data set is stored in the same directory as the backup repository data sets.

CAUTION:

The **RESTORE** option of %OMABAKUP deletes the DS_VCHAR data sets from the backup directories before it restores the repositories to their registered locations.

Therefore, if you run %OMABAKUP in restore mode after having used %OMARUNAN or the RUNANALYSIS %OMABAKUP option, the optimizations will be lost. Δ

- The next time %OMABAKUP is used to perform a backup, it merges the information in each repository's DS_VCHAR data set with the repository's DS_DEFN data set. This merge operation occurs in the repository's registered location after the backup operation has been completed.
- The next time the repository's records are loaded into the metadata server's memory, the updated information in the DS_DEFN data set is used to optimize the memory usage. The optimizations are not applied to repository data sets on disk.

Choosing Whether to Use %OMARUNAN or the RUNANALYSIS Option of %OMABAKUP

The RUNANALYSIS option executes on the metadata server host immediately after a backup, which can consume host resources at a time when client requests have resumed.

You can execute %OMARUNAN at any time. In addition, you can execute %OMARUNAN from any computer that has SAS licensed and that has a network connection to the computer where the backup files are stored.

To minimize the impact of %OMARUNAN on the metadata server computer, you can choose to do either of the following:

- Direct the backups to a drive that the metadata server can access over the network, and run %OMARUNAN on that computer.
- If you do not have another networked computer that has a SAS license, run %OMARUNAN on the metadata server during a period of low activity.

Using the %OMARUNAN Macro

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

%OMABAKUP also creates the OMA_Varansas7bcatalog that is used by %OMARUNAN. The creation of this catalog has two implications for running %OMARUNAN:

- The catalog is created to the specifications of the host operating environment; therefore, you need to execute %OMARUNAN in the same operating environment in order to be able to read the catalog. That is, if the backups are created on a UNIX host, then you need to execute %OMARUNAN from another UNIX host.
- The information in OMA_Varansas7bcatalog reflects the last backup that was made. That is, if a full backup was made, then OMA_Varansas7bcatalog contains information to analyze all repositories that are registered on the SAS Metadata Server. Keep this in mind when you choose backups that you want to analyze.

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

%OMARUNAN Autocall Macro: Reference

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

Syntax

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

DESTINATIONPATH

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

Details

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

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

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

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

Recovering Memory on the SAS Metadata Server

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

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

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

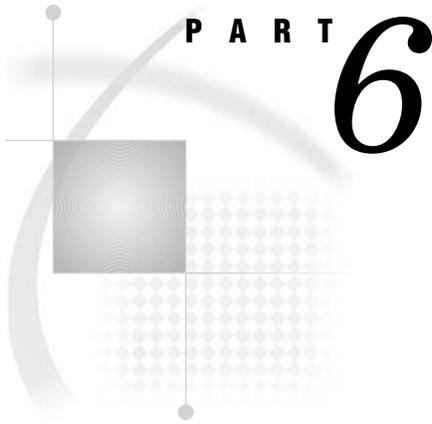
- The Pause action instructs the SAS Metadata Server to stop accepting new client requests for the specified repositories, or for the entire metadata server, and closes repository containers.
- The Resume action resumes client activity in the specified repositories or on the entire SAS Metadata Server.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

- The Refresh action pauses and resumes the specified repositories or the entire SAS Metadata Server in one step.

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

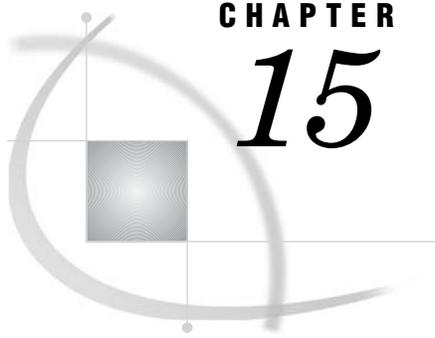
See the documentation for the METAOPERATE procedure in the *SAS Open Metadata Interface: Reference* for examples of procedure statements that issue Pause, Resume, and Refresh actions on specific repositories.

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



Managing SAS Metadata Repositories

- Chapter 15*..... **Creating, Moving, Copying, and Deleting Metadata Repositories** 119
- Chapter 16*..... **Creating a Metadata Repository in a Database Management System** 127
- Chapter 17*..... **Using the Promotion and Replication Wizards to Copy an Entire Metadata Repository** 135
- Chapter 18*..... **Promoting Individual BI Objects** 169



CHAPTER

15

Creating, Moving, Copying, and Deleting Metadata Repositories

<i>Creating Repositories</i>	119
<i>Copying, Promoting, Importing, and Exporting Your Metadata</i>	120
<i>Tools for Copying, Promoting, Importing, and Exporting Metadata</i>	120
<i>About the Promotion Wizard and the Replication Wizard</i>	121
<i>About the BI Manager Import and Export Wizards</i>	121
<i>About the Metadata Manager Import and Export Wizards</i>	121
<i>About the SAS Data Integration Studio Import and Export Wizards</i>	122
<i>Moving a Repository to a New Location on the Same SAS Metadata Server</i>	122
<i>Porting a Repository to a Different Host Environment</i>	123
<i>Using the %OMAPORT Macro</i>	123
<i>%OMAPORT Autocall Macro: Reference</i>	124
<i>Syntax</i>	124
<i>Details</i>	125
<i>Deleting a Metadata Repository</i>	125

Creating Repositories

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

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

By default, repositories are created using the SAS Base engine. If you have the appropriate SAS/ACCESS software installed, you can also create repositories in Oracle and IBM DB2 databases. SAS/ACCESS and database software requirements are described in Chapter 16, "Creating a Metadata Repository in a Database Management System," on page 127.

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

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

include the path to the repository manager. That is, a repository directory should not be nested within the **rposmgr** directory.

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

- When registering a repository, use a relative pathname for the repository location. The use of a relative pathname makes a repository easier to copy and move. A SAS Metadata Server that was set up by the planned SAS installation must have its repositories identified relative to the **SASMain** directory—for example, **MetadataServer/MetadataRepositories/repository-name**.
- Repository names must be unique.
- A repository directory should not contain any files that are not part of the repository.
- Do not overwrite an old repository with a new one.

Note:

- The metadata server must add, update, and delete repositories when no other activity is taking place on the server, so it automatically delays other client requests until the repository changes are complete. This delay might have a small effect on server performance.
- Custom repositories are supported only by SAS Data Integration Studio, SAS OLAP Cube Studio, and solutions that use SAS Solutions Services (for example, SAS Financial Management and SAS Strategic Performance Management). Project repositories are supported only by SAS Data Integration Studio.

Δ

Copying, Promoting, Importing, and Exporting Your Metadata

Tools for Copying, Promoting, Importing, and Exporting Metadata

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

The tools that are available include the following:

- Metadata Manager Promotion Wizard and Replication Wizard (to copy an entire metadata repository)
- BI Manager Import and Export Wizards (to copy selected objects from one repository to another)
- Metadata Manager Import and Export Wizards (to copy metadata for relational data and DBMSs to and from external sources)
- SAS Data Integration Studio Import and Export Wizards (to copy metadata for relational data and DBMSs to and from external sources with change analysis)
- %OMAPORT Macro (to port a metadata repository from one operating system environment to another)

The wizards are described briefly in the following paragraphs. For information about the %OMAPORT macro, see “Porting a Repository to a Different Host Environment” on page 123.

About the Promotion Wizard and the Replication Wizard

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

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

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

The Promotion and Replication wizards are located in the Metadata Manager **Job Definitions** folder. For detailed information, see Chapter 17, “Using the Promotion and Replication Wizards to Copy an Entire Metadata Repository,” on page 135.

About the BI Manager Import and Export Wizards

BI Manager is a feature of SAS Management Console that provides a common interface for administering SAS BI objects. BI Manager contains an Export Wizard and an Import Wizard that enable you to selectively promote individual BI objects (including stored processes, information maps, reports, data explorations, and folders) from one metadata repository to another.

The BI Manager wizards export and import both the content and the metadata for the selected objects. For example, when you export metadata for reports, BI Manager also exports the associated XML (.srx) files.

The BI Manager wizards are located in the BI Manager node of SAS Management console. For detailed information, see Chapter 18, “Promoting Individual BI Objects,” on page 169.

About the Metadata Manager Import and Export Wizards

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

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

To access this feature, navigate to the **Metadata Manager** node in SAS Management Console. Expand the **Active Server** node under **Metadata Manager**, and then

right-click a repository name. On the menu that appears, you can select **Import Metadata** or **Export Metadata**.

By default, Metadata Manager imports and exports metadata in the Common Warehouse Metamodel (CWM) format. Additional import and export formats are supported if you install SAS Metadata Bridges (MITI Bridges). For information about this software, contact your SAS account executive.

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

Table 15.1 Metadata Type Conversions

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

Notes:

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

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

About the SAS Data Integration Studio Import and Export Wizards

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

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

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

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

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

- 1 Pause client activity in the repository that you want to move to a read-only state. (In the current release, the pausing of individual repositories is supported only in PROC METAOPERATE.)
- 2 Use host commands to copy the contents of the repository to a new location.
- 3 When the files are in the new location, use the tools in the SAS Management Console Metadata Manager to delete the repository in the old location and to register it in the new location.
 - To delete a repository, highlight it in the server's navigation tree and select **Actions ► Delete**. The Delete action deletes the contents of a repository and removes the repository's registration from the repository manager.
 - To register a repository, highlight the active-server icon, and select **Actions ► Add Repository**. When you are prompted for the repository path, specify the repository's new location. The SAS Metadata Server will read the existing files in the specified location to create the new repository's registration.

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

- 1 In the Metadata Manager, expand the **Active Server** node.
- 2 Right-click the repository that you want to move, and select **Unregister** from the pop-up menu.
- 3 Use host commands to move the contents of the repository to a new location.
- 4 When the files are in the new location, re-register the repository using the Add Repository Wizard.

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

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

Δ

Porting a Repository to a Different Host Environment

Using the %OMAPORT Macro

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

- NFS or some other network convention

- binary FTP
- PROC CPORT/CIMPORT
- PROC DOWNLOAD/UPLOAD

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

Notes:

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

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

%OMAPORT Autocall Macro: Reference

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

Syntax

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

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

OLDRPOS specifies the pathname of the source repository. If the source and target hosts are connected in a network such as an NFS network, and you specify an absolute pathname for the *source-repository-path*, then %OMAPORT references the source repository on the source host. Otherwise, you must use one of the copy methods listed in “Details” on page 125 to move the source repository to a working directory on the target host and specify this working directory as the *source-repository-path*.

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

For example, if you are porting a foundation repository to the **MetadataRepositories** directory of a planned SAS installation location, you should specify a *target-repository-path* similar to this:

SAS-config-dir\Lev1\SASMain\MetadataServer\
MetadataRepositories\repository-name.

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

Details

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

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

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

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

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

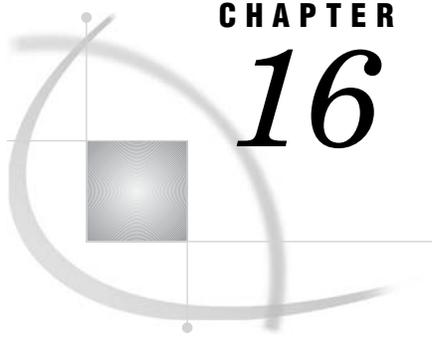
Deleting a Metadata Repository

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

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

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

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



CHAPTER

16

Creating a Metadata Repository in a Database Management System

<i>About Creating a Metadata Repository in a Database Management System</i>	127
<i>Prerequisites for Setting Up a DBMS Repository</i>	127
<i>Create the Repository Manager and a Metadata Repository in an Oracle or a DB2 Database</i>	128
<i><RPOSMGR> Attributes Used to Connect to an Oracle Database</i>	130
<i><RPOSMGR> Attributes Used to Connect to a DB2 Database</i>	131

About Creating a Metadata Repository in a Database Management System

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

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

CAUTION:

For performance reasons, we strongly recommend that you use the default SAS data sets for the repository manager and for the foundation metadata repository. However, we recognize that your company might require you to store the repository manager tables and repository tables in a third-party relational database. If you are going to make this change, you should make it as soon as possible after installation to minimize the amount of metadata that you need to re-create. △

Prerequisites for Setting Up a DBMS Repository

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

- Obviously, you must have an Oracle or DB2 server that you can access from your metadata server host. A database administrator must have created at least two schemas for you: one for the repository manager tables and one for the foundation repository tables. We recommend that you name the schema for the repository manager SASRMGR. If you plan to create additional repositories (custom or project repositories), you will also need a schema for each such repository. The

database administrator must also have supplied you with the information that you will need to connect to the database and access these schemas.

- You must have installed the SAS/ACCESS Interface to Oracle or the SAS/ACCESS Interface to DB2 on your metadata server host machine. If your database server is running on a UNIX machine, you must also have configured the SAS/ACCESS product as explained in the *Configuration Guide for SAS Foundation for UNIX Environments*. This document is included in your SAS Installation Kit.
- You must also have installed the database client software on your metadata server host.

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

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

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

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

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

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

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

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

<i>path</i>	an alias (service name) that represents information about the protocol to be used in communicating with the database, the name of the server hosting the database server, and the port on which the server is listening. You can define this alias on the metadata server host by using an Oracle configuration tool, or you can have your database administrator create it.
-------------	--

<i>user</i>	the user ID that you will use to connect to the database server.
<i>password</i>	the corresponding password.
<i>schema</i>	the name of the schema in which you plan to create the repository manager tables.

For further information about the attributes that appear in the **<RPOSMGR>** element when you connect to an Oracle database, see “<RPOSMGR> Attributes Used to Connect to an Oracle Database” on page 130.

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

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

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

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

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

For further information about the attributes that appear in the **<RPOSMGR>** element when you connect to a DB2 database, see “<RPOSMGR> Attributes Used to Connect to a DB2 Database” on page 131.

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

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

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

- 5 Start SAS Management Console, and connect to the metadata server as the SAS Administrator (sasadm).
- 6 You see a dialog that displays the message “No Repositories are present on the requested server. Would you like to add a repository now?” Click **Yes**. A wizard that enable you to define a new repository starts.
- 7 On the Select Repository Type page, select **Foundation** and click **Next**.
- 8 On the General Information page, leave **Foundation** in the **Name** field, and click **Next**.
- 9 On the Definition of Data Source page, perform the following steps:
 - a Select **Oracle** or **DB2** from the **Engine** drop-down list. (This will cause the **Path** field to be dimmed and the **Options** text area to be populated.)

- b In the **Options** text area, supply values for the **PATH** (Oracle only), **DB** (DB2 only), **USER**, and **PASSWORD** options. Also, add a **SCHEMA** option if the name of the schema to be used for the foundation repository does not match the value of the **USER** option.
- c Click **Next**.
- d On the Current Settings page, click **Finish**.
- e You see a dialog that displays the message “You must pause the server in order to apply the new authorization inheritance scheme. Pause the server now?” Click **Yes**.
- f You see a dialog that display the message “Repository initialization was successful.” Click **OK**.

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

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

<RPOSMGR> Attributes Used to Connect to an Oracle Database

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

LIBREF

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

ENGINE

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

PATH

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

OPTIONS

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

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

reread_exposure=yes

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

preserve_names=yes

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

- dbchar_constant_is_spoofed=yes**
required and is valid only when used for access to SAS Metadata Repositories by the Oracle engine.
- connection=shared**
specifies that all tables that are opened for reading by the LIBNAME or libref share this connection.
- readbuff=1**
specifies the number of rows in a single Oracle fetch. A value of 1 is required.
- insertbuff=1**
specifies the number of rows in a single Oracle insert operation. A value of 1 is required.
- spool=no**
prevents spooling in the Oracle engine. This spooling is unnecessary because the SAS Metadata Server uses the SAS In Memory Database for the repository tables.
- utilconn_transient=no**
specifies that a utility connection is maintained for the lifetime of the libref.
- dbmax_text=4096**
(optional) on 64-bit platforms, specifies the length of any very long DBMS character data that is read into SAS or written from SAS via the SAS/ACCESS engine.
- PATH=path**
specifies an alias representing the database server to which you want to connect. The alias points to information such as a hostname, a port number, and a transmission protocol. If you are not sure what path to use, consult with your database administrator.
- USER=user**
specifies an Oracle user name. If the user name contains blanks or national characters, enclose the name in quotation marks. If you omit an Oracle user name and password, then the default Oracle user OPS\$sysid is used, if it is enabled. The **USER** option must be used with the **PASSWORD** option.
- PASSWORD=password**
specifies an Oracle password that is associated with the Oracle user name. If you omit the **PASSWORD** option, then the password for the default Oracle user ID OPS\$sysid is used, if it is enabled.
- SCHEMA=schema**
(optional) enables you to register multiple repositories using the same Oracle user account. When you omit the **SCHEMA** option, the software uses the default schema, which is the requesting user ID, to create the repository. A schema name such as SASRMGR is recommended when you are creating the repository manager. The schema name must be entered in uppercase letters and be a valid SAS name that is unique for the **USER**, **PASSWORD**, and **PATH** options.

<RPOSMGR> Attributes Used to Connect to a DB2 Database

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

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

ENGINE

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

PATH

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

OPTIONS

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

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

reread_exposure=yes

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

preserve_names=yes

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

dbchar_constant_is_spoofed=yes

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

connection=shared

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

readbuff=0

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

insertbuff=1

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

spool=no

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

utilconn_transient=no

specifies that a utility connection is maintained for the lifetime of the libref.

dbmax_text=4096

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

DB=database

specifies the DB2 data source or database to which you want to connect.

USER=user

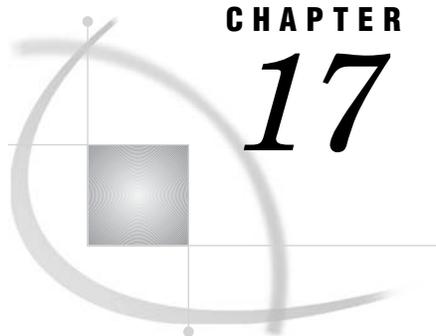
enables you to connect to a DB2 database with a user ID that is different from the one you used to log on to your computer. The **USER** and **PASSWORD** connection options are optional for DB2. If you specify **USER**, then you must also specify **PASSWORD**. If you omit **USER**, then your operating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

PASSWORD=*password*

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

SCHEMA=*schema*

(optional) enables you to register multiple repositories using the same DB2 user account. When you omit the **SCHEMA** option, the software uses the default schema, whose name matches the requesting user ID, to create the repository. A schema name such as SASRMGR is recommended when you are creating the repository manager. You must enter the schema name in uppercase, and the name must be a valid SAS name that is unique for the **USER**, **PASSWORD**, and **DB** options.



CHAPTER

17

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

<i>When to Use the Promotion and Replication Wizards</i>	136
<i>Features and Capabilities of the Promotion and Replication Wizards</i>	137
<i>Special Considerations for Using the Promotion and Replication Wizards</i>	137
<i>Overview of the Promotion or Replication Process</i>	139
<i>Promoting Dependent Repositories</i>	139
<i>Preparing Your Site for a Promotion Job</i>	140
<i>Overview of Preparing for a Promotion Job</i>	140
<i>Setting Up the Source Environment for a Promotion Job</i>	142
<i>Summary of Steps for Setting Up the Source Environment</i>	142
<i>Install and Configure the Necessary SAS Software</i>	142
<i>Set Up an Administrative User (Windows Only)</i>	143
<i>Make the SAS User an Administrative User (UNIX Only)</i>	143
<i>Create a Metadata Access File</i>	143
<i>Setting Up the Target Environment for a Promotion Job</i>	144
<i>Summary of Steps for Setting Up the Target Environment</i>	144
<i>Install the Platform Components Called for by Your Plan</i>	144
<i>Install and Configure the SAS/CONNECT Software</i>	145
<i>Start the SAS/CONNECT Spawner</i>	145
<i>Give the Target Administrator the Required Directory Permissions</i>	146
<i>Setting Up the Administration Environment for a Promotion Job</i>	147
<i>Summary of Steps for Setting Up the Administration Environment</i>	147
<i>Install and Configure the Necessary Software</i>	147
<i>Create an Operating System Account for the SAS Replication Administrator</i>	148
<i>Define the Source Metadata Server in the Metadata</i>	148
<i>Define the Source Workspace Server in the Metadata</i>	149
<i>Define the Source Object Spawner in the Metadata</i>	149
<i>Define the Target Metadata Server in the Metadata</i>	150
<i>Define the Target SAS/CONNECT Server in the Metadata</i>	150
<i>Define the SAS Replication Administrator in the Metadata</i>	151
<i>Creating and Running a Promotion Job</i>	152
<i>Overview of Creating and Running a Promotion Job</i>	152
<i>Building a Promotion Job</i>	152
<i>Running a Promotion Job</i>	158
<i>Making Modifications after Running a Promotion Job</i>	159
<i>Modifications to the SAS/CONNECT Server</i>	159
<i>Modifications to Server Definitions</i>	159
<i>Modifications to Users</i>	161
<i>Modifications to Content Mapping for Reports</i>	161
<i>Modifications to Foundation Services</i>	162
<i>Modifications to Cubes</i>	162
<i>What Happens When You Run a Promotion Job?</i>	163

<i>Moving Associated BI Content Following a Promotion Job</i>	165
<i>About Moving BI Content</i>	165
<i>Moving Xythos WebFile Server Content</i>	165
<i>Moving Portal Content</i>	167
<i>Redeploying Jobs from SAS Data Integration Studio</i>	167
<i>Performing a Demotion</i>	167

When to Use the Promotion and Replication Wizards

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

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

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

- You have implemented the SAS Intelligence Platform in a production environment, and you want to use that environment as the basis for creating a test environment. In the process, you need to make modifications such as changing your server definitions so that they refer to hosts in the test environment.
- You have developed a completely new implementation of the SAS Intelligence Platform in a test environment, and you want to move the new implementation to a production environment. In the process, you need to make modifications such as changing your server definitions so that they refer to hosts in the production environment.
- After having promoted your metadata from a test environment to a production environment, you want to refresh the test environment with metadata from the most current production environment. To do this, you would use the Promotion Wizard to demote metadata from the target environment back to the source environment (see “Performing a Demotion” on page 167).
- You have created a metadata repository for a particular organizational unit, and you want to create another instance of the repository for another organizational unit.

CAUTION:

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

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

To promote metadata for only selected BI objects (including selected information maps, reports, stored processes, and data explorations), use BI Manager’s Export and Import Wizards. For details, see Chapter 18, “Promoting Individual BI Objects,” on page 169.

Features and Capabilities of the Promotion and Replication Wizards

Both the Promotion Wizard and the Replication Wizard provide a series of dialog boxes that guide you through the process of creating a job definition. The dialog boxes prompt for the following information:

- the name of the source metadata server and the repository that you want to promote.
- the name of the target metadata server.
- credentials to log on to the source metadata server and the target metadata server.
- for promotion jobs, the values for any metadata attribute substitutions that should be made, including values for port numbers, host machine names, schema names, and paths. For each of these metadata attributes, the wizard displays the values that currently exist in the source metadata repository. For each value, you can enter a corresponding substitution. The wizard checks the validity of each substitution value that you enter.
- paths for work directories on both the source metadata server and the target metadata server.
- the path to a repository backup location on the target metadata server. The promotion or replication job will back up the existing metadata repository and place it in this location before replacing it with the promoted or replicated repository.
- a name for the job definition.
- the action to take with the job definition. You can choose to save the job and run it later, or you can go ahead and run the job and also save the job definition.

Based on the information that you provide, the wizard generates the SAS code that is necessary to promote or replicate your metadata repository. The code is saved using the job definition name that you specify. The substitution values that you enter are saved with the job.

You can create as many different promotion and replication jobs as necessary to handle different scenarios. The jobs can be rerun at any time, provided that you first modify the job properties to reflect any changes to the current environment.

Special Considerations for Using the Promotion and Replication Wizards

Before choosing to use the Promotion Wizard or the Replication Wizard, you should be aware of the following special considerations:

- The Promotion and Replication wizards will completely replace the metadata repository in the target environment. This means that all of the metadata in the target environment will be overwritten, including the following:
 - server definitions
 - library definitions
 - user definitions
 - access controls
 - personal folders that users have created to store reports
 - preferences that users have selected in the SAS Information Delivery Portal
 - portal content (including pages, page templates, portlets, links, syndication channels, and applications) that users have created or modified

- The Promotion and Replication wizards copy all of the metadata in the repository. To promote metadata for only selected BI objects (including selected information maps, reports, stored processes, and data explorations), use BI Manager's Export and Import Wizards. For details, see Chapter 18, "Promoting Individual BI Objects," on page 169.
- Although the Promotion Wizard automatically replaces some metadata values based on information that you provide, you will need to manually enter some metadata changes after the promotion job has completed. Manual changes will be required for server definitions, user logins, content mapping for reports, Foundation Services deployment definitions, and cube definitions. For details, see "Making Modifications after Running a Promotion Job" on page 159.
- The Promotion and Replication wizards promote only your metadata. They do not promote associated content, such as the following:
 - WebDAV folders and their contents (for example, report definition files and portal content)
 - other portal content, including custom portlets, custom themes, and applications
 - source files for stored processes
 - physical tables and databases
 - deployed jobs

For information about moving this content, see "Moving Associated BI Content Following a Promotion Job" on page 165.

- Before you can promote or replicate a dependent repository to a target machine, you must first promote or replicate its set of parent repositories to that same host. For details, see "Promoting Dependent Repositories" on page 139.
- When you use the Promotion and Replication wizards, the source environment and the target environment must have the same operating system. For example, you can promote or replicate a metadata repository from a Windows host to another Windows host, or from a UNIX host to another UNIX host.

If you want to promote or replicate metadata from one operating system to another, you must copy the repository to the target environment and then run a SAS macro that converts the repository data sets to the appropriate internal format. For details, see "Porting a Repository to a Different Host Environment" on page 123.

- SAS/CONNECT software must be installed in both the source environment and the target environment.
- Before you use the Promotion or Replication wizard, you must prepare the source environment, the target environment, and a separate administration environment with the appropriate software and server definitions. In both the source environment and the administration environment, you will need to set up special administrative users. For details, see "Preparing Your Site for a Promotion Job" on page 140.
- Before you use the Promotion or Replication wizard, you should back up the existing target repository if there is any chance that you might need to recover the metadata. For details, see Chapter 6, "Planning SAS Metadata Server Backups and Restores," on page 45.
- If you have previously run a promotion job, and you now want to copy a metadata repository from the target environment back to the source environment, follow the setup procedures in "Performing a Demotion" on page 167.

Overview of the Promotion or Replication Process

The remainder of this chapter explains the following procedures:

- how to set up your source and target environments (for example, production and test environments) for a promotion or replication operation
- how to set up an administration environment from which you can create and run a promotion or replication job
- how to use SAS Management Console to create and run a promotion or replication job
- how to customize a promotion or replication job
- how to troubleshoot a failed promotion or replication job
- how to move content that is associated with the metadata following a promotion or replication job
- how to perform a demotion, which is the process of copying a metadata repository from the target environment back to the source environment after a promotion has been performed (see “Performing a Demotion” on page 167)

Note: Throughout the rest of this chapter—except where we explicitly state otherwise—we use the term *promotion* to mean promotion or replication. △

The topics listed above and others are discussed in the following sections:

- “Promoting Dependent Repositories” on page 139
- “Preparing Your Site for a Promotion Job” on page 140
- “Creating and Running a Promotion Job” on page 152
- “What Happens When You Run a Promotion Job?” on page 163
- “Moving Associated BI Content Following a Promotion Job” on page 165

If you want to promote a dependent repository, you should first read “Promoting Dependent Repositories” on page 139.

Promoting Dependent Repositories

Promoting a dependent repository is a special case. Before you can promote a dependent repository to a target machine, you must promote its set of parent repositories to that same host. The promotion process uses the permissions that are defined in the server’s foundation repository and inherited by the dependent repository.

So the general procedure for promoting a dependent repository is as follows:

- 1 Promote all parent repositories of the dependent repository from the source server to the target server.
- 2 Promote the dependent repository from the source server to the target server.
- 3 On the target server, define the dependency relationships between the dependent repository and its parents so that they are the same as the dependencies between the corresponding repositories on the source server.

It is only necessary to create the dependencies between the parent and child repositories on the target server the first time that you promote the repositories. The dependency relationships will remain in place for subsequent promotions of the repositories. However, if the dependency structure changes on the source server, you must establish an equivalent dependency structure on the target server.

Preparing Your Site for a Promotion Job

Overview of Preparing for a Promotion Job

Before you can run a promotion job, you must correctly set up three *environments*, each consisting of a metadata server, a metadata repository, and some related software components:

- a *source environment*, which includes the following:
 - a metadata server in an existing installation of the SAS Intelligence Platform.
 - the metadata repository that you want to promote to the target environment.
 - SAS/CONNECT software on the metadata server machine. The promotion process uses this software to move the repository data sets to the target environment.
 - SAS Integration Technologies software, an IOM object spawner, and a workspace server on the metadata server machine (these components are required if you want to be able to run the promotion job from the Promotion Wizard).

Note:

- The workspace server must be configured on the same machine that hosts the metadata server. The promotion wizard cannot access a workspace server that is located on another machine.
- If you plan to run the promotion job in batch mode, you do not need the workspace server. However, the remainder of the chapter assumes that you want to be able to run the promotion job from the Promotion Wizard.

△

- a *target environment*, which is the environment to which you want to promote the source environment's metadata repository.

For the most part, you should install and configure in this environment the same SAS software that you originally installed in the source environment. An important exception to this rule is that you should not create a metadata repository in the target environment during configuration. Basically, this means not following any of the instructions in the `instructions.html` files produced by the SAS Configuration Wizard.

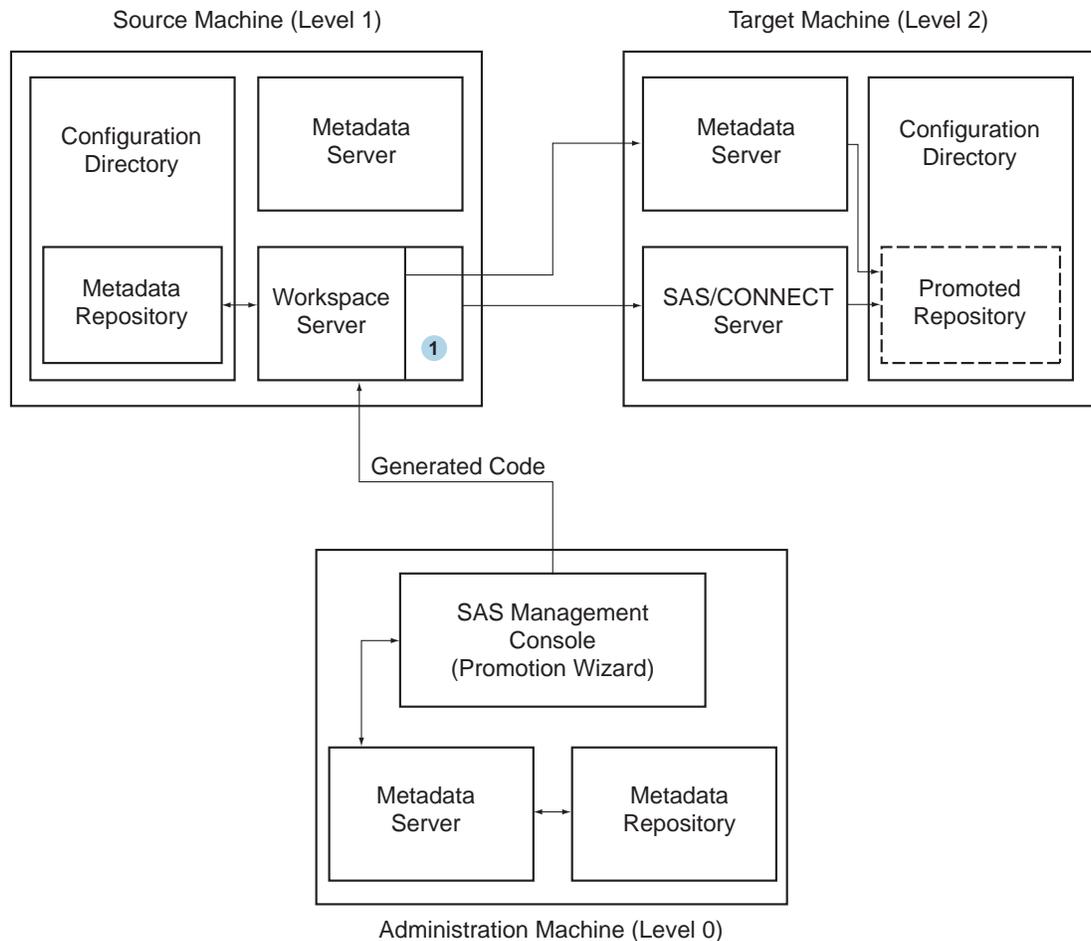
For the reasons mentioned previously, the target environment must also include a SAS/CONNECT server that has been installed on the metadata server machine.

- an *administration environment*, which is the environment from which you run the Promotion Wizard to create and execute a promotion job, and which includes the following:
 - Base SAS software and SAS Management Console
 - a metadata server
 - a metadata repository that contains information about the SAS servers and spawners in the source and target environments

Note: Throughout the rest of this chapter, it is assumed that the source environment is Level 1, the target environment is Level 2, and the administration environment is Level 0. When the SAS Configuration Wizard creates a directory called **Lev1** in the path to your configuration directory, the 1 in the name means “production.” A test environment would use **Lev2** in the path, and an administration environment would use **Lev0** in the path. △

The following diagram illustrates the primary components of the promotion infrastructure.

Figure 17.1 Required Infrastructure for a Promotion Job



1 SAS/CONNECT Client

Generally, the source environment is installed on one or more computers, and the target environment is installed on a different set of one or more computers. The administration environment is typically installed on a machine by itself or on the metadata server host in the source environment.

The Promotion Wizard generates code that performs the promotion. The wizard then submits this code to a workspace server running in the source environment. The code uses SAS/CONNECT software to copy the source environment's metadata repository to the target environment.

Note: For a much more detailed explanation of the promotion process, see “What Happens When You Run a Promotion Job?” on page 163. You may need this extra detail to customize the promotion process or to troubleshoot a problem with promotion. Δ

For detailed information about how to set up the source, target, and administration environments prior to creating and running a promotion job, see the following subsections.

Note: If you have previously run a promotion job, and you now want to copy a metadata repository from the target environment back to the source environment, follow the setup procedures in “Performing a Demotion” on page 167. For example, you might want to refresh your development environment or your test environment by copying metadata from your current production environment. Δ

Setting Up the Source Environment for a Promotion Job

Summary of Steps for Setting Up the Source Environment

On the source metadata server machine, perform the following setup:

- Make sure that an IOM object spawner and a workspace server have been configured on the system and that SAS/CONNECT software has been installed. The SAS code that performs the promotion runs on the workspace server, and the SAS/CONNECT software enables the workspace server to move the repository data sets to the target machine.

You can determine whether SAS/CONNECT software is installed on the source metadata server machine by looking in the SAS installation area and seeing whether the folder **SAS 9.1** contains a folder named **connect**. To determine whether an object spawner and workspace server have been configured on the machine, look in the folder *SAS-config-dir\Lev1\SASMain*, and see whether it contains directories named **ObjectSpawner** and **WorkspaceServer**.

- Create an *administrative user* on Windows, or make the SAS user (sas) an *administrative user* on UNIX. Make sure that this user has full access to the necessary parts of the configuration directory on the source machine. The user must have administrative access in order to pause and resume the metadata server.
- Create a metadata access file that is used to establish communication between the administration metadata server and the source metadata server.

The following subsections explain how to perform these tasks.

Install and Configure the Necessary SAS Software

If SAS/CONNECT software is installed and a workspace server has been configured on the source metadata server machine, you can skip this section. Otherwise, perform these tasks on the source metadata server machine:

- 1 Install SAS/CONNECT software (if it is not already installed) and SAS Integration Technologies software (if it is not already installed). Both of these components are part of the SAS 9.1 Foundation, so you can install them by performing a Software Index installation of the SAS Foundation. (For information on how to perform a Software Index installation, see the *SAS Intelligence Platform: Installation Guide*.) During the installation of the SAS Foundation, you will have an opportunity to specify which components the installer should install.
- 2 If you installed Integration Technologies in step 1, use the SAS Configuration Wizard to configure an object spawner and a workspace server. (For information about how to run the SAS Configuration Wizard outside the context of a planned installation, see the *SAS Intelligence Platform: Installation Guide*.)

Note: The workspace server must be configured on the same machine that hosts the metadata server. The promotion wizard cannot access a workspace server that is located on another machine. Δ

Set Up an Administrative User (Windows Only)

Before you can run a promotion job on a Windows system, you must set up an *administrative user* on the source metadata server machine. This user will have to be authenticated by the source machine's authentication provider and will need full access to the **MetadataRepositories** and **ReplicationWorkArea** directories.

To set up such a user on a Windows system, follow these steps:

- 1 Create an operating-system user account for a user with the ID `srcadmin`.
- 2 Make this user an *administrative user* by adding the user's ID to the file **adminUsers.txt**, which is located in the directory `SAS-config-dir\Lev1\SASMain\MetadataServer`. This file should already contain an entry for the SAS Administrator (`sasadm`). Add a similar entry for `srcadmin`, but do *not* place an asterisk before the user ID.

Note: For detailed instructions, see "User and Group Management" in the *SAS Intelligence Platform: Security Administration Guide*. △

You will have to restart the metadata server for this new setting to take effect.

- 3 Give the user `srcadmin` full control of the following directories, either explicitly or by adding the user to the Administrators group:
 - the directory `SAS-config-dir\Lev1\SASMain\MetadataServer\ReplicationWorkArea` and its subdirectories
 - the directory that holds your repository data sets, typically `SAS-config-dir\Lev1\SASMain\MetadataServer\MetadataRepositories\Foundation`

Make the SAS User an Administrative User (UNIX Only)

If your source metadata server machine is running under UNIX, you should make the SAS user (`sas`) an *administrative user* by adding this user to the file **adminUsers.txt**. Do not place an asterisk before the user ID. This user should already be the owner of the **MetadataRepositories** and **ReplicationWorkArea** directories.

Create a Metadata Access File

The metadata access file is used to establish communication between the administration metadata server and the source metadata server. To create the file, follow these directions:

- 1 First decide on the following values, which you will use when you set up the administration machine:
 - The number of the port on which the metadata server on this machine will listen. By default, this will be port 8560.
 - The user ID that you will use for the SAS Replication Administrator account, by convention, `sasrpadm`. This user will be an *administrative user* on the administration machine.
 - The password that you will assign to the SAS Replication Administrator.
- 2 Create a file named **mdrep1js.sas** in the directory `SAS-config-dir\Lev1\SASMain` on the source metadata server machine. The content of this file should look like this:

```
options metaserver="host-name"
        metaport=port-number
        metaprotocol="bridge"
        metauser="user-ID"
```

```
metapass="encoded-password";
```

You need to substitute actual values for the variable values shown in italics. The table below explains how to replace the variables in this text.

Table 17.1 Creating the File `mdrepljs.sas`

Variable	Replace With
<i>host-name</i>	The host name of the administration machine.
<i>port-number</i>	The number of the port on which the metadata server on the administration machine will listen (8560).
<i>user-ID</i>	The user ID of the SAS Replication Administrator. By convention, this user is named <code>sasrpadm</code> . If the administration machine is a Windows system, you must qualify this ID with a domain or machine name (for example, <code>D1234\sasrpadm</code>).
<i>encoded-password</i>	The encoded password for the SAS Replication Administrator. To determine the encoded form of the password, start an interactive SAS session, and submit the following code in the Editor: <pre>proc pwencode in='unencoded-pwd'; run;</pre> <p>The encoded password will appear in the Log window.</p>

Setting Up the Target Environment for a Promotion Job

Summary of Steps for Setting Up the Target Environment

On the target metadata server machine, perform the following setup:

- Install the same components of the SAS Intelligence Platform that you installed on the source machine when you first created your system. (The target server is typically a mirror image of the source machine.) But do not create a metadata repository. The target metadata repository is created when you run your promotion job.
- Make sure that SAS/CONNECT software has been installed and that a SAS/CONNECT server has been configured. A SAS/CONNECT spawner must be running on the target machine in order for the workspace server on the source machine to move the repository data sets to the target machine.
- Manually update the script that starts the SAS/CONNECT spawner so that you can run it without registering the SAS/CONNECT server in the metadata. (You do not have a metadata repository on the target machine yet.) Then, start the spawner.
- Set up an *unrestricted user*, who will be the process owner for the SAS/CONNECT server.

The following subsections explain how to perform these tasks.

Install the Platform Components Called for by Your Plan

When you originally installed the SAS Intelligence Platform, you installed and configured a set of products on the source machine. Under normal circumstances, you want to install and configure the same set of products on your target machine. To do this, you follow basically the same procedure that you used to install the platform on your source machine:

- 1 Perform the pre-installation tasks required for your platform and the products that you are installing on your target machine.
- 2 Use the SAS Software Navigator to install the set of products listed in a deployment plan. This may be the same deployment plan that you used when you installed software on the source machine, or a similar one that contains different host names.
- 3 Use the SAS Configuration Wizard to configure the products that you installed, but *do not* use SAS Management Console to create a metadata repository, and do not follow any of the instructions in the file **instructions.html**. You create the metadata repository on the target machine when you run your promotion job.

When you run the SAS Configuration Wizard, be sure to do the following things:

- a When you are prompted for the location of the configuration directory, specify the same value that you used when you configured the source machine.
- b When you are prompted for the port on which the metadata server will listen, enter the value **8562**. On your source machine, the name of the Level directory (**Lev1**) and many of the port numbers end with the number 1. This number indicates that these items are part of your production environment. Similarly, the number 2 indicates that an element is part of your first derived configuration, which is typically a test environment.
- c When you reach the Advanced Properties Editor screen, open the properties file and make the following changes:
 - Change **Lev1** to **Lev2**.
 - Change the port number **8591** to **8592**, if necessary. This is the workspace server port.
 - Change the port number **7551** to **7552**, if necessary. This is the SAS/CONNECT server port.

Install and Configure the SAS/CONNECT Software

To install SAS/CONNECT software on and create a script that you can use to start a SAS/CONNECT spawner, follow these directions on the target metadata server machine:

- 1 Install SAS/CONNECT software by performing a Software Index installation of the SAS Foundation. (For information on how to perform a Software Index installation, see the *SAS Intelligence Platform: Installation Guide*.) During the installation of the SAS Foundation, you will have an opportunity to specify that you want to install only the SAS/CONNECT component of the SAS Foundation.
- 2 Use the SAS Configuration Wizard to configure the SAS/CONNECT software. For information about how to run the SAS Configuration Wizard outside the context of a planned installation, see the *SAS Intelligence Platform: Installation Guide*. *Do not* follow the directions in the **instructions.html** file generated by the configuration wizard.

Start the SAS/CONNECT Spawner

When you ran the SAS Configuration Wizard in the previous step, it created a script that you can use to start a SAS/CONNECT spawner. Edit this script as described in the following procedure, and then run the script as directed in step 5 to start the spawner.

On Windows systems, you edit and run the script `SAS-config-dir\Lev2\SASMain\ConnectServer\ConnectServer.bat` as follows:

- 1 In the file **ConnectServer.bat**, locate the line that sets the value of the **USEMETADATA** option, and change it so that it reads:

```
set USEMETADATA=0
```

- 2 Locate the line that sets the value of the **IsService** option, and change it so that it reads:

```
set IsService=0
```

- 3 Locate the line that begins **echo Manual start of Spawner**. Make sure that the **start** command on the next line contains the **-security** option. For example, the finished **start** command might look like this:

```
start \b "Connect Spawner" "%sasdir%\spawner" -security -service %CONNPORT%
-SASCMD %CONNCMD%
```

- 4 Save your changes to **ConnectServer.bat**.
- 5 Start the spawner by running the script **StartConnectServer.bat**, which is located in the same directory as the **ConnectServer.bat** script. To start a spawner by using a command that contains the **-security** option, you must have the following Windows user rights:
 - Act as part of the operating system**
 - Adjust memory quotas for a process (or Increase quotas)**
 - Bypass traverse checking**
 - Log on locally**
 - Replace a process level token**

On UNIX systems, you edit and then run the script *SAS-config-dir\Lev2\SASMain\ConnectServer\ConnectServer.sh*.

- 1 In the file **ConnectServer.sh**, locate the line that sets the value of the **USEMETADATA** option, and change it so that it reads:

```
set USEMETADATA=0
```

- 2 Save your changes to **ConnectServer.sh**.
- 3 Start the spawner by running the script **ConnectServer.sh**. Use the following command:

```
ConnectServer.sh start
```

Give the Target Administrator the Required Directory Permissions

In the promotion procedure discussed in this chapter, a special account on the target machine is used to start a SAS/CONNECT server process and to communicate with the metadata server on that machine. This user must be an *unrestricted user* and must have full access to certain directories.

On Windows systems, we recommend that you use the SAS Administrator account (sasadm) for this purpose. This user is already an *unrestricted user*, but must be granted full access to the following directories in order for the promotion to succeed:

- SAS-config-dir\Lev2\SASMain\MetaDataServer\ReplicationWorkArea* and its subdirectories
- SAS-config-dir\Lev2\SASMain\MetaDataServer\MetaDataRepositories\Foundation*

On UNIX systems, we recommend that you use the SAS User account (sas). This user already has access to the requisite directories; however, the user is not an *unrestricted user* of the metadata. To make sas an *unrestricted user*, enter the following line in the file **adminUsers.txt**:

```
*sas
```

Setting Up the Administration Environment for a Promotion Job

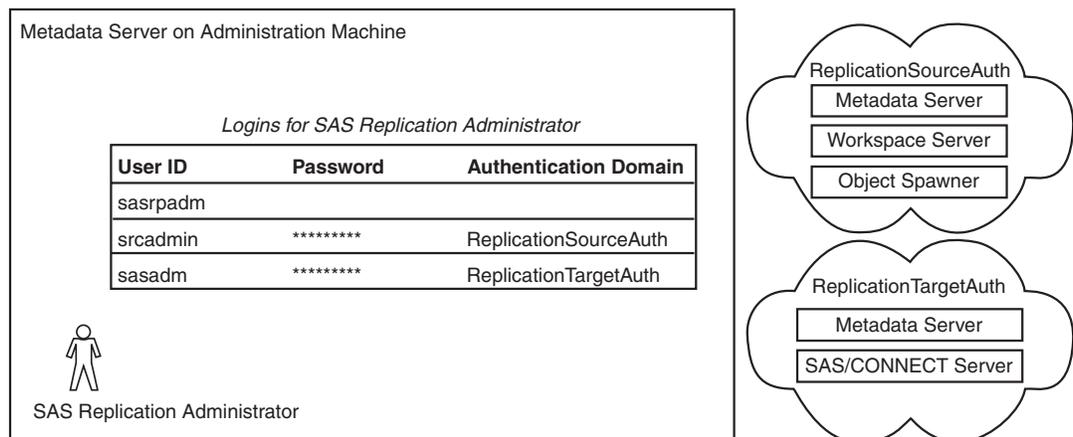
Summary of Steps for Setting Up the Administration Environment

Setting up the administration machine involves performing the following tasks:

- installing Base SAS software and SAS Management Console
- configuring a metadata server
- creating an operating-system user account for a SAS Replication Administrator
- creating metadata objects for the SAS Replication Administrator and the SAS servers that are running on the source and target machines

In order to run the Promotion Wizard, you log into SAS Management Console as the SAS Replication Administrator. This same user later needs to be able to start a workspace server on the source machine and a SAS/CONNECT server on the target machine. To make sure that the SAS Replication Administrator is authenticated when he or she attempts to start these servers, you need to place the SAS servers on the source machine in one authentication domain, place the SAS servers on the target machine in a different authentication domain, and create three logins for the replication administrator. This setup is shown below.

Figure 17.2 Logins for the SAS Replication Administrator on a Windows System



Note: The figure above shows the logins that the SAS Replication Administrator would have if you were promoting a repository from one Windows system to another. If you are promoting a repository on a UNIX machine, the second and third logins should contain the credentials of the SAS User (sas). Δ

You can find details about how to set up these authentication domains and logins—and all of the other tasks mentioned above—in the following sections.

Install and Configure the Necessary Software

To install Base SAS software and SAS Management Console on the administration machine and to configure a metadata server there, follow these directions:

- 1 Perform the set of pre-installation tasks required for your platform and the set of products that you plan to install.

Note: If your source and administration environments will reside on the same machine, you will already have performed this step. Δ

- 2 Use the SAS Software Navigator to perform a Software Index installation of the SAS Foundation and SAS Management Console. During the installation of the SAS Foundation, you will have an opportunity to specify which components of the foundation you want to install. You need to install only Base SAS software.

Note: If your source and administration environments will reside on the same machine, you will already have performed this step. Δ

- 3 Use the SAS Configuration Wizard to configure a metadata server on this host, following the directions in the *SAS Intelligence Platform: Installation Guide*. The only instructions specific to the current situation are listed below.
 - a When you are prompted for the metadata server's port number, enter **8560**. (The 0 in this port number indicates that this instance of the metadata server is part of the administration environment.)
 - b Use the Advanced Properties Editor to change the value of the **LEVEL** property from **Lev1** to **Lev0**.
 - c When the configuration wizard generates the instruction file **instructions.html**, follow only the instructions for creating a Foundation repository. Do not follow the instructions that have you enter metadata in the repository.

Create an Operating System Account for the SAS Replication Administrator

You need to set up a user account for a new user and make this user an *administrative user*.

- 1 Create an operating-system user account for a user with the ID `sasrpadm`.
- 2 Make this user an *administrative user* by adding the user's ID to the file **adminUsers.txt**, which is located in the directory `SAS-config-dir\Lev0\SASMain\MetadataServer`. This file should already contain an entry for the SAS Administrator (`sasadm`). Add a similar entry for `sasrpadm`, but do *not* place an asterisk before the user ID. You will have to restart the metadata server for this new setting to take effect.

Define the Source Metadata Server in the Metadata

The remaining instructions for setting up the administration machine involve creating metadata definitions for the SAS servers that are running on the source machine and the target machine and creating a definition for the SAS Replication Administrator. To perform each of these tasks, you should be logged in to SAS Management Console as the SAS Replication Administrator. You use SAS Management Console to create the server and user definitions.

To create a definition for the metadata server running on the source machine:

- 1 Right-click **Server Manager**, and select **New Server**. The New Server Wizard appears.
- 2 Run the New Server Wizard, supplying the following values:

Note: For any properties that are not listed, you should accept the default value. Δ

- Server type: **SAS Application Server**
- Name: **Lev 1 - application-server-name** (for example, **Lev 1 - SASMain**)
- Type of server to add to application server: **Metadata Server**

- Authentication Domain: **ReplicationSourceAuth**

Note: Click **New** to create this authentication domain, if necessary. Δ

- Host Name: Name of source machine
- Port Number: **8561**

When you click **Finish**, the following nodes are added to the Server Manager tree.

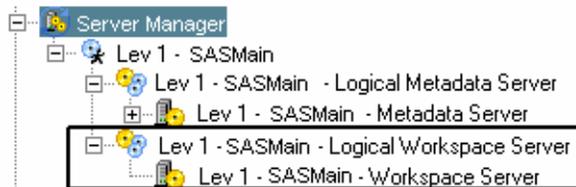


Define the Source Workspace Server in the Metadata

To create a definition for the workspace server running on the source machine:

- 1 Right-click the node for the application server you defined in the previous section (**Lev1 - application-server-name**), and select **Add Application Server Component**. The New Application Server Component Wizard appears.
- 2 Run the wizard, supplying the following values:
 - Type of server to add to application server: **Workspace Server**
 - Authentication Domain: **ReplicationSourceAuth**
 - Host Name: Name of source machine
 - Port Number: **8591**

When you click **Finish**, the following nodes are added to the Server Manager tree.



Define the Source Object Spawner in the Metadata

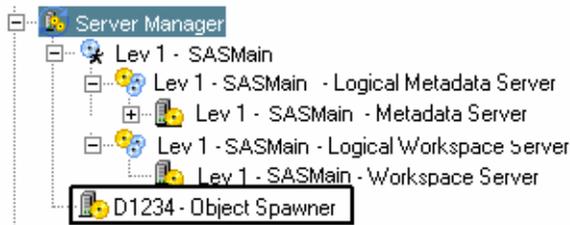
To create a definition for the object spawner running on the source machine:

- 1 Right-click **Server Manager**, and select **New Server**. The New Server Wizard appears.
- 2 Run the wizard, supplying the following values:

Note: For any properties that are not listed, you should accept the default value. Δ

- Server type: **Object Spawner**
- Name: *hostname* - **Object Spawner** (for example, **D1234 - Object Spawner**)
- Associated machine: Name of source machine
- Selected Servers: **Lev 1 - application-server-name - Workspace Server**
- Authentication Domain: **ReplicationSourceAuth**
- Host Name: Name of source machine
- Port Number: **8581**

When you click **Finish**, the following node is added to the Server Manager tree.



Define the Target Metadata Server in the Metadata

To create a definition for the metadata server running on the target machine:

- 1 Right-click **Server Manager**, and select **New Server**. The New Server Wizard appears.
- 2 Run the New Server Wizard, supplying the following values:

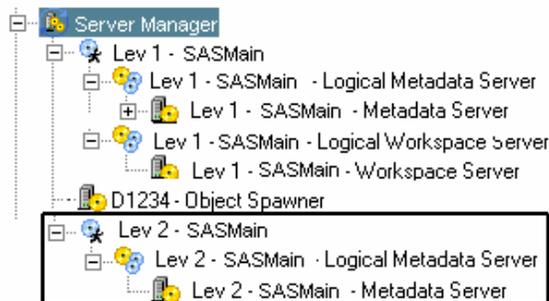
Note: For any properties that are not listed, you should accept the default value. Δ

- Server type: **SAS Application Server**
- Name: **Lev 2 - application-server-name** (for example, **Lev 2 - SASMain**)
- Type of server to add to application server: **Metadata Server**
- Authentication Domain: **ReplicationTargetAuth**

Note: Click **New** to create this authentication domain, if necessary. Δ

- Host Name: Name of target machine
- Port Number: **8562**

When you click **Finish**, the following nodes are added to the Server Manager tree.



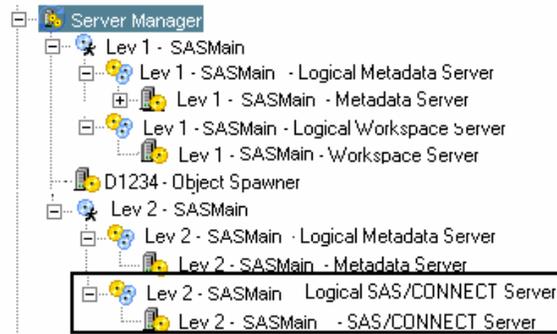
Define the Target SAS/CONNECT Server in the Metadata

To create a definition for the SAS/CONNECT Server running on the target machine:

- 1 Right-click the node for the application server you defined in the previous section (**Lev2 - application-server-name**), and select **Add Application Server Component**. The New Application Server Component Wizard appears.
- 2 Run the wizard, supplying the following values:
 - Type of server to add to application server: **SAS/CONNECT Server**
 - Authentication Domain: **ReplicationTargetAuth**
 - Host Name: Name of target machine

- Port Number: **7552**

When you click **Finish**, the following nodes are added to the Server Manager tree.



Define the SAS Replication Administrator in the Metadata

When you run a promotion job, you run it as the SAS Replication Administrator. Because the replication administrator must be able to start a workspace server on the source machine and a SAS/CONNECT server on the target machine, this user must have a total of three logins:

- one for the administration machine
- one for the source machine
- one for the target machine

To define this user and the user's three logins, follow these directions:

- 1 Right-click **User Manager** and select **New ► User**. The New User Properties dialog box appears.
- 2 On the **General** tab, enter **SAS Replication Administrator** in the **Name** text box.
- 3 Select the **Logins** tab.
- 4 Define a login for the user sasrpadm.
 - a Click **New** to bring up the New Login Properties dialog box.
 - b Enter the user ID for the sasrpadm account in the **User ID** field. On Windows systems, this ID must be qualified with a domain name or a machine name.
 - c Use the **Authentication Domain** list box to clear the associated text field.
 - d Click **OK**.
- 5 Define a login using the credentials for the srcadmin account (Windows) or the sas account (UNIX) on the source machine. Include a password, and set the **Authentication Domain** to **ReplicationSourceAuth**.
- 6 Define a login using the credentials for the sasadm account (Windows) or the sas account (UNIX) on the target machine. Include a password, and set the **Authentication Domain** to **ReplicationTargetAuth**. (This login needs to contain information about an unrestricted user.)
- 7 Click **OK** in the New User Properties dialog box.

At the end of this step, you have created the necessary infrastructure and are ready to run the promotion wizard to create the promotion job.

Creating and Running a Promotion Job

Overview of Creating and Running a Promotion Job

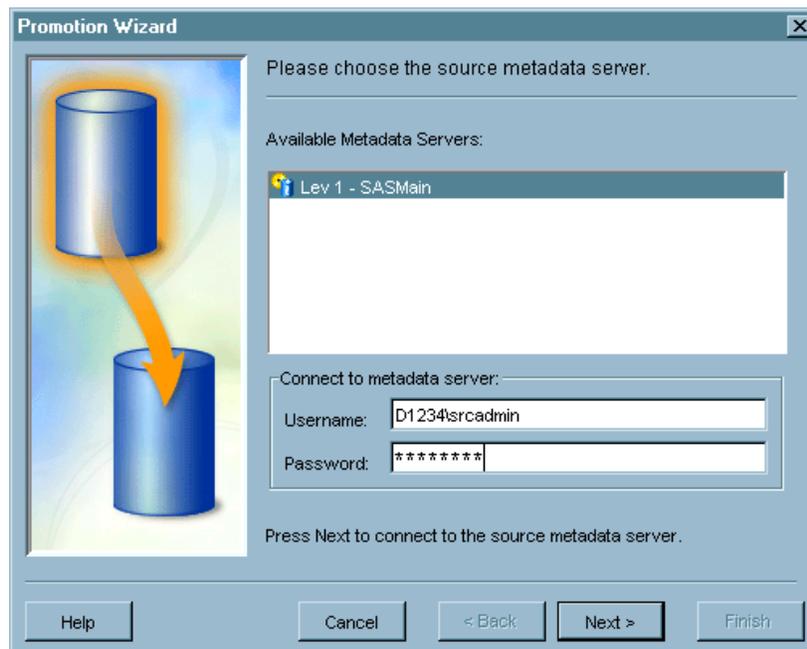
Once you have set up your source, target, and administration machines, you are ready to use SAS Management Console to create a promotion job and to run that job. After you run the promotion job, you complete the promotion manually by editing some of the metadata in the target repository. These topics are discussed in the following sections:

- “Building a Promotion Job” on page 152
- “Running a Promotion Job” on page 158
- “Making Modifications after Running a Promotion Job” on page 159

Building a Promotion Job

You create a promotion job—that is, you generate the SAS code required to promote a metadata repository—by running the SAS Management Console Promotion Wizard. To run this wizard, follow these directions.

- 1 On the administration machine, start SAS Management Console, and connect to the metadata server on that machine using a metadata profile that contains the credentials of the SAS Replication Administrator (sasrpadm).
- 2 Expand the **Metadata Manager** and **Job Definition** nodes in the SAS Management Console tree. Underneath the **Job Definition** node, you will see icons labeled **Promotion** and **Replication**.
- 3 Right-click the **Promotion** icon, and select **New Definition** from the pop-up menu that appears. The Promotion Wizard starts.
- 4 The wizard’s first page asks you to choose the source metadata server.

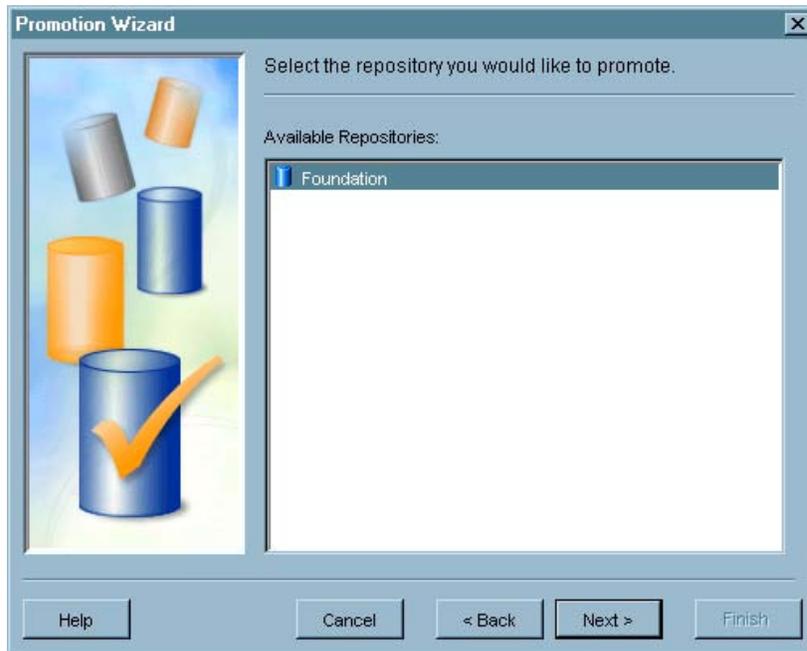


Provide the following information:

- a In the **Available Metadata Servers** list, select the SAS application server that contains the metadata server on the source machine.
- b In the **Connect to metadata server** area, supply the credentials of an *administrative user* on the source machine. Normally, you enter the user ID and password for the sreadmin or sas account. (“Set Up an Administrative User (Windows Only)” on page 143 explains how to give this user administrative rights.)

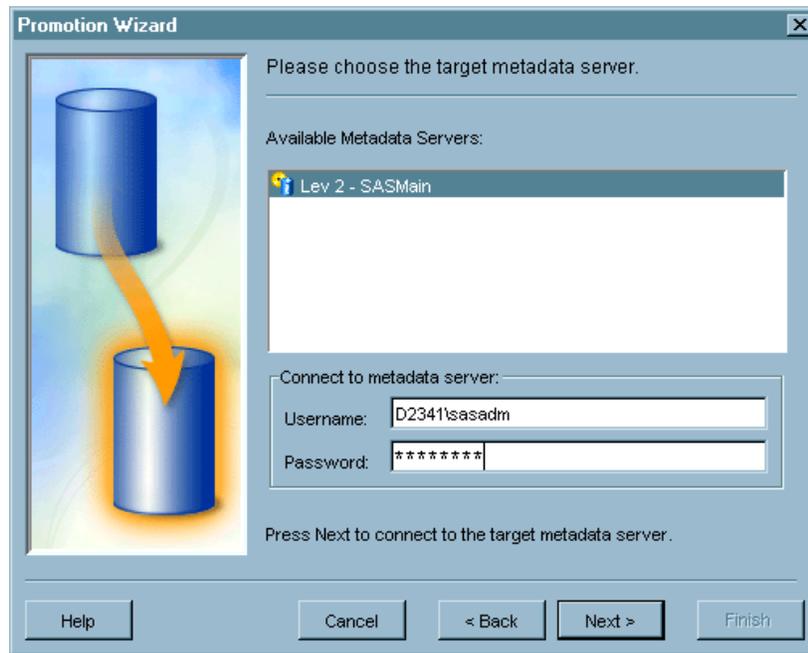
Then click **Next**.

- 5 The wizard’s next page asks you to select the repository that you want to promote.



From the list of metadata repositories on the source machine, select the repository that you want to promote, and then click **Next**.

- 6 The wizard’s next page asks you to choose the target metadata server.

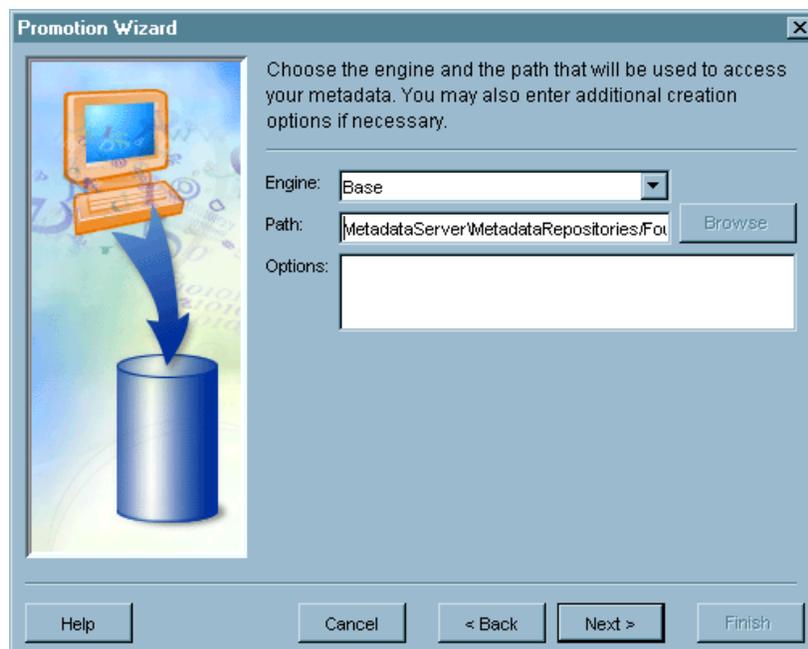


Provide the following information:

- a In the **Available Metadata Servers** list, select the SAS application server that contains the metadata server on the target machine.
- b In the **Connect to metadata server** area, supply the credentials of the SAS Administrator (sasadm) or SAS User (sas) on the target machine.

Then click **Next**.

- 7 If the repository that you selected has not been defined on the target machine—which will be the case the first time that you promote a repository—the wizard's next page asks you to specify the type and location of the new repository.



Provide the following information:

- a From the **Engine** list, select the engine that will be used to access the metadata repository. Because the repository normally consists of a group of SAS data sets, typically you should select **Base**.
- b In the **Path** field, enter the path from the **SASMain** directory in your configuration directory to the directory for the metadata repository. If you are promoting your foundation repository, this path will typically be **MetadataServer/MetadataRepositories/Foundation**.

Click **Next**.

- 8 If you are performing a promotion, the wizard's next page enables you to specify what substitutions the resulting promotion job should make in the metadata. (The Replication Wizard does not have this page.)

For each metadata attribute, specify the promotion substitution value.

Substitution Table

Metadata Attribute: Ports
Hosts
Schema Names

No substitutions for this metadata attribute

Metadata Type	From Value	To Value
TCPIPConnection	8531	<Enter Value>
TCPIPConnection	8601	<Enter Value>
TCPIPConnection	8611	<Enter Value>

Validate

Help Cancel < Back Next > Finish

The wizard makes substitutions in the following four areas:

- port numbers
- host names
- schema names
- file system paths

For each type of information, follow this procedure:

- a From the **Metadata Attribute** list, select an attribute such as **Ports**. If you select **Ports**, the table in the middle of the page is populated with a list of the port numbers in the source metadata repository. Typically, you want to specify a replacement value for each value in the **From Value** column. However, you can also select **No substitutions for this metadata attribute** to indicate that you do not want to make any changes for a particular attribute.

Note: Before you proceed to the next page in the wizard, you must supply substitution information for all four metadata attributes. △

- b Unless you have chosen the no-substitutions check box for an attribute, for each value in the **From Value** column of the substitutions table, specify a value in

the **To Value** column. For example, you might indicate that you want each occurrence of the port number **8591** in the source metadata repository to be changed to **8592** in the target repository. The following table provides some brief recommendations on how to choose a **To Value**.

Table 17.2 Substitutions to Make in the Target Metadata

Metadata Attribute	Change to Make in the Source Values
Ports	Add 1 to each port number that ends in 1. For example, if a port number is 4321 , change the value to 4322 .
Hosts	Change host names in the source metadata to names that will be valid in the target system. Use fully qualified host names. For example, if a source host name is D1234 , the target host name might be D1235.na.abc.com .
Schema Names	Change any schema names to ones that will be appropriate on the target server.
Paths	Change any paths that do not exist on the target system to paths that are valid there. Leave all relative paths unchanged.

- c** Click **Validate**. If you have entered an acceptable **To Value** in each row of the table (or if you chose not to make any substitutions), a green check appears to the right of the currently selected metadata attribute. If you see a red **x** instead, one or more of the values you entered is not a valid replacement value. Fix each problematic row—a red **x** appears in the rightmost column of each such row—and then click **validate** again. Repeat this set of steps as many times as necessary.

When all four metadata attributes have green check marks next to them, click **Next**.

- 9** The wizard's next page asks you to specify work directories on the source and target metadata server hosts and to specify a backup directory on the target host.

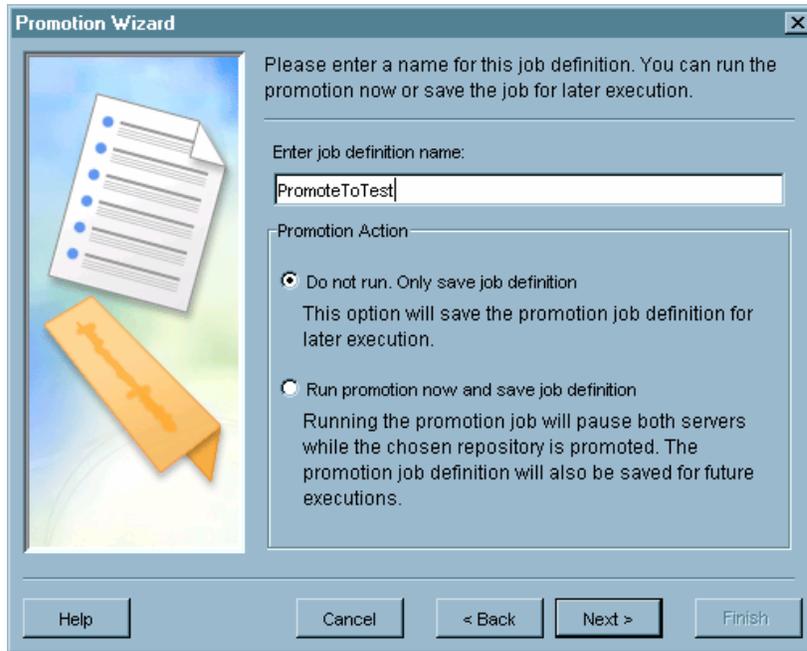
To ensure that the paths are valid, you might want to enter relative paths, as shown in the following examples. The paths are relative to either *sas-config-dir*\Levn\SASMain\ or *sas-config-dir*\Levn\SASApp\, depending on how the machine is configured.

Note: After the promotion job has been saved, you will not be able to modify these paths. Δ

- a work directory location for the source metadata server. For example:
MetadataServer\ReplicationWorkArea\work
- a work directory location for the target metadata server. For example:
MetadataServer\ReplicationWorkArea\work
- a location on the target metadata server to store the backup of the repository. For example:
MetadataServer\ReplicationWorkArea\tempBackup

Then click **Next**.

- 10** The wizard's next page prompts you for the name of your promotion job and asks whether you want to run the promotion job now.

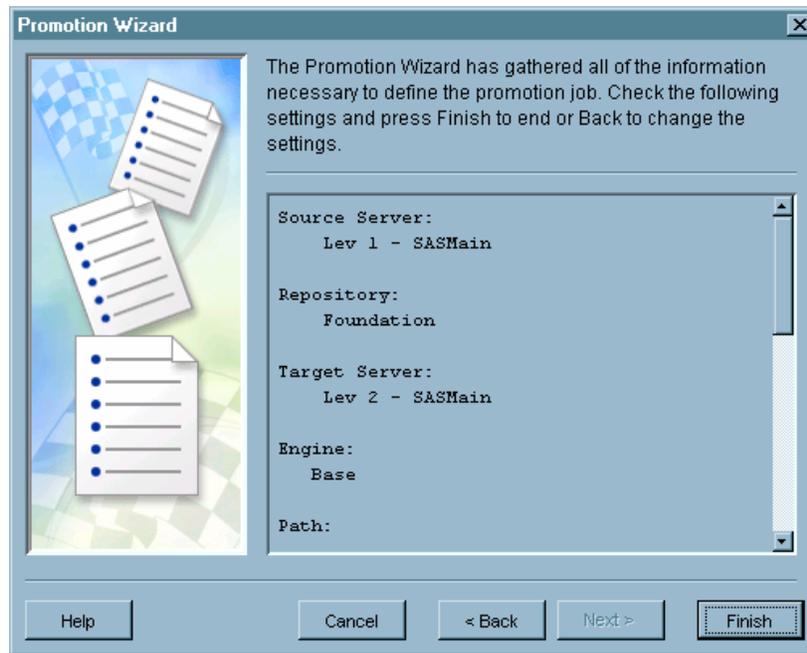


Supply the following values:

- a In the **Enter job definition name** field, enter a name for the replication job (for example, **Source1 Dev to Test - Foundation**). In this name, “Source1” is a description of the metadata server associated with the repository being promoted; this part of the name is optional if, as usual, the environment contains only one metadata server. The “Dev to Test” part of the name describes the source and target environments. The “Foundation” part of the name is the name of the repository being promoted.
- b Select one of the radio buttons in the **Promotion Action** area. If you select the top button, the job will not be run when the wizard exits; a definition of the job will be saved in the metadata, and you can run the job later. If you select the bottom button, the job will be run when the wizard exits *and* a definition of the job will be saved so that you can run it later.

Click **Next**.

- 11 The wizard’s last page displays all of the information that you have specified in the wizard.



To make changes, click **Back** until you reach the appropriate page. If all of the information is correct, then click **Finish** to create the replication job definition and to run the job, if you elected to do so.

Running a Promotion Job

To run a promotion or replication job that has already been created, follow these steps in SAS Management Console:

- 1 In the SAS Management Console tree, expand the **Metadata Manager** and **Job Definitions** nodes. Under the **Job Definitions** node, you see **Promotion** and **Replication** icons.
- 2 Select the **Promotion** or **Replication** icon. If you select the **Promotion** icon, you see a list of promotion jobs in the right pane, and if you select the **Replication** icon, you see a list of replication jobs.
- 3 Right-click a promotion or replication job in the right pane; then select **Run Job** from the pop-up menu.

Any errors that occur are written to the log file **errorLog_SAS_Management_Console.txt**, which is located in one of the following paths:

- On Windows systems: **C:\Documents and Settings\user**
- On UNIX systems: *user's-home-directory/*

Note: If you want to run the job on a batch basis (instead of running it from SAS Management Console), right-click the job name and select **Save to File**. The dialog box that appears enables you to save the job as a SAS program locally on the administration machine. Before running the job, you must copy it to the source metadata server machine in the path *SAS-config-dir\Lev1\SASMain*. You can then start an interactive SAS session on this machine, change the working directory (current folder) to *SAS-config-dir\Lev1\SASMain*, and submit the job. △

Making Modifications after Running a Promotion Job

After you run a promotion job, you must make some manual changes to the following:

- the script used to start the SAS/CONNECT spawner
- the promoted metadata

The following sections explain how to make these modifications.

Note: You do not need to modify the metadata after running a replication job. Δ

Modifications to the SAS/CONNECT Server

On the target machine, you must stop the SAS/CONNECT spawner, edit the **ConnectServer.ext** script, and then restart the spawner.

On Windows systems, perform the following steps:

- 1 Modify the **ConnectServer.bat** file, which is typically located in *SAS-config-dir\Lev2\SASMain\ConnectServer*.
 - a Locate the **USEMETADATA** option, and change the line containing it to read as follows:


```
USEMETADATA=1
```
 - b If you are going to install the SAS/CONNECT spawner as a service, locate the **IsService** option and change the line containing it to read as follows:


```
IsService=1
```
 - c Save your changes.
- 2 To install the SAS/CONNECT spawner as a service, run the script **InstallConnectServer.bat**. This script is located in the same directory as the **ConnectServer.bat** script.
- 3 Start the SAS/CONNECT spawner.
 - If you installed the SAS/CONNECT spawner as a service, start the service by using either the **Start** menu or the Control Panel.
 - If you did not install the spawner as a service, run the script **StartConnectServer.bat**.

On UNIX systems, perform the following steps:

- 1 In the file **ConnectServer.sh**, locate the line that sets the value of the **USEMETADATA** option, and change it so that it reads as follows:


```
set USEMETADATA=1
```
- 2 Save your changes to **ConnectServer.sh**.
- 3 Start the spawner by running the script **ConnectServer.sh**. Use the following command:

```
ConnectServer.sh start
```

Modifications to Server Definitions

Perform the following tasks on the metadata-server host in the target system:

- 1 In SAS Management Console, connect to the target metadata server using the credentials of the SAS Administrator (sasadm).
- 2 In the SAS Management Console tree, expand **Server Manager** and, under that node, any SAS Application Servers.

- 3 If a SAS Application Server contains a logical workspace server, expand the logical workspace server, and perform the following steps for the physical workspace server(s):

- a Right-click the icon for the workspace server, and select **Properties** from the pop-up menu. A Workspace Server Properties dialog box appears.
- b In the Workspace Server Properties dialog box, select the **Options** tab.
- c Update the command in the **Command** text field, as appropriate. On a Windows system, enter the following command:

```
sas -config "SAS-config-dir\Lev2\SASMain\sasv9.cfg"
```

On a UNIX system, enter the following command:

```
SAS-config-dir/Lev2/SASMain/sas.sh
```

- d Click **OK**.

- 4 If a SAS Application Server contains a logical stored process server, expand the logical stored process server, and perform the following steps for the physical stored process server(s):

- a Right-click the icon for the stored process server, and select **Properties** from the pop-up menu. A Stored Process Server Properties dialog box appears.
- b In the Stored Process Server Properties dialog box, select the **Options** tab.
- c Update the command in the **Command** text field, as appropriate. On a Windows system, enter the following command:

```
sas -config "SAS-config-dir\Lev2\SASMain\StoredProcessServer\
sasv9_StorProcSrv.cfg"
```

On a UNIX system, enter the following command:

```
SAS-config-dir/Lev2/SASMain/StoredProcessServer/sas_SPS.sh
```

- d Click **OK**.

- 5 If a SAS Application Server contains a logical OLAP server, expand the logical OLAP server, and perform the following steps for the physical SAS OLAP Server:

- a Right-click the icon for the OLAP server, and select **Properties** from the pop-up menu. A properties dialog box appears.
- b On the **Options** tab, click **Advanced Options** to open the Advanced Options dialog box.
- c On the **Performance** tab, edit the value of the **Path for temporary working files** field, if necessary. If the path in this field is a relative path, leave the value unchanged. If it contains a full path, change the value so that it points to a work directory on the target machine.
- d If you made changes on the **Performance** tab, click **OK** to save your changes; otherwise, click **Cancel**. You are returned to the properties dialog box.
- e Click **OK**.

- 6 If a SAS Application Server contains a logical SAS/CONNECT server, expand the logical SAS/CONNECT server, and perform the following steps for the physical SAS/CONNECT Server:

- a Right-click the icon for the SAS/CONNECT Server, and select **Properties** from the pop-up menu. A Connect Server Properties dialog box appears.
- b In the Connect Server Properties dialog box, select the **Options** tab.
- c Update the command in the **SASCMD Option** text field, as appropriate. On a Windows system, enter the following command:

```
SAS-config-dir\Lev2\SASMain\sasconnect.bat
```

On a UNIX system, enter the following command:

```
SAS-config-dir/Lev2/SASMain/sasconnect.sh
```

- d Click **OK**.
- 7 If a SAS Application Server contains a SAS/SHARE Server, perform the following steps for the this server:
 - a Select the icon for the SAS/SHARE Server. Information about a SAS/SHARE connection appears in the right pane.
 - b Right-click the connection name, and select **Properties** from the pop-up menu. A properties dialog box appears.
 - c In the properties dialog box, select the **Options** tab.
 - d On the **Options** tab, change the values for the **Server Host**, **Remote Session ID**, and the **Server ID** to values that are appropriate for the machine that is hosting the target metadata server.
 - e Click **Connection Information Options** and, if necessary, change the **Host Name** value to the machine that is hosting the target metadata server. Click **OK** to return to the properties dialog box.
 - f Click **OK** in the properties dialog box.
- 8 If a SAS Application Server contains a logical SAS Data Step Batch Server, expand the logical batch server, and perform the following steps for the physical batch server:
 - a Right-click the icon for the batch server, and select **Properties** from the pop-up menu. A SAS Data Step Batch Server Properties dialog box appears.
 - b In the SAS Data Step Batch Server Properties dialog box, select the **Options** tab.
 - c In the **Command Line** text field, enter the following command:


```
SAS-config-dir\Lev2\SASMain\BatchServer\sasbatch
```
 - d In the **Logs Directory** text field, enter the following command:


```
SAS-config-dir\Lev2\SASMain\BatchServer\logs
```
 - e Click **OK**.

Modifications to Users

If necessary, modify the logins for local users who were promoted to the target system.

- 1 In SAS Management Console, select **User Manager**. A list of users and groups appears in the right pane.
- 2 Right-click a user or group, and select **Properties** from the pop-up menu. A properties dialog box appears.
- 3 In the properties dialog box, select the **Logins** tab.
- 4 Modify any logins that contain a user ID that was valid on the source system, but is not valid on the target system. For example, you must change the user IDs for local Windows users so that they contain the correct hostname.

Modifications to Content Mapping for Reports

If your metadata repository includes report objects, then you must modify the content mapping properties of the root folder that contains the reports. Content mapping establishes the location of the XML (.srx) files that define the reports. The location can be either a path on a WebDAV server or a path on the file system.

To modify the content mapping properties:

- 1 Under BI Manager in SAS Management Console, identify the root folder that contains reports. In default installations, this root-level folder is called **BIP Tree**.
- 2 Right-click this folder, and then click **Properties**.
- 3 On the **Content Mapping** tab, enter information about the target system's report content. If you specify a WebDAV server location, you must enter the user ID and password for the SAS Web Administrator user in the appropriate fields.

Note: The Promotion Wizard does not copy your report content to the target system. For information about copying report content and other BI content from the source system to the target system, see “Moving Associated BI Content Following a Promotion Job” on page 165. Δ

Modifications to Foundation Services

If you are using SAS Web Report Studio and SAS Information Delivery Portal at your site, you must use the Foundation Services Manager in SAS Management Console to delete—and then reimport—the following service deployments:

- ID Portal Local Services
- Query and Reporting
- Remote Services

To update your service deployment metadata, follow these steps:

- 1 In SAS Management Console, expand the **Foundation Services Manager** node. A list of service deployments appears.
- 2 For each of the service deployments listed above, right-click the service deployment name, and select **Delete** from the pop-up menu.
- 3 Right-click the **Foundation Services Manager**, and select **Import Service Deployment** from the pop-up menu. An Import Service Deployment dialog box appears.
- 4 Click **Add** in this dialog box. A file browser (an Open dialog box) appears.
- 5 Use the file browser to display the contents of the directory `SAS-config-dir\Lev2\web\Deployments\Portal`.
- 6 Select the following files and click **Open**.
 - `sas_services_idp_remote_omr.xml`
 - `sas_services_idp_local_omr.xml`

The names of these files will be displayed in the Import Services Deployment dialog box.

- 7 In the same way, add the file `SAS-config-dir\Lev2\web\Deployments\WebReportStudio\hostname_sas_pfs_queryandreporting.xml` to the list.
- 8 Click **OK** in the Import Service Deployment dialog box.

Modifications to Cubes

If you promoted cube metadata from the source metadata server to the target metadata server, you must modify the work path, index path, and data path for each promoted cube. The following instructions explain how to use SAS Data Integration Studio to edit a cube's structure:

- 1 Use SAS Data Integration Studio to connect to the target metadata server. You can use the login information for any user who has WriteMetadata access to the cube metadata that must be modified.

- 2 In the SAS Data Integration Studio inventory tree, select the cube.
- 3 To launch the Cube Designer wizard, right-click the cube name and select **Edit Cube Structure**.
- 4 On the General page, change the **Work Path** to the correct path for the target machine.
- 5 Click **Next** until the Generated Aggregations page appears. On that page, click **Advanced** to open the Performance Options dialog box. In this dialog box, make these changes on the **Global** tab:
 - a Change the value of the **Location of index component files** to the correct path for the target machine.
 - b Change the value of the **Location of partitions in which to place aggregation table data** to the correct path for the target machine.
 - c Click **OK** to close the dialog box and return to the Cube Designer wizard.
- 6 Click **Next** until the Finish page appears.
- 7 Depending on whether you want to save the metadata and rebuild the cube or just save the metadata, select the appropriate radio button.
- 8 Click **Finish**.

What Happens When You Run a Promotion Job?

This in-depth view of what happens when you run a promotion job should be of interest if you are debugging a promotion problem or are interested in customizing such a job.

Note: To customize a promotion job, you can add code to the SAS macros that are mentioned in this section. These macros are located in the following paths:

- *sas-install-dir*\core\sasmacro on Windows machines
- !SASROOT\sasautos on UNIX machines
- the library that is defined by the SASAUTOS fileref on z/OS machines

△

When you run a promotion job, the Promotion Wizard submits generated code to the workspace server on the source machine. This code performs the following actions:

- 1 It establishes a connection to the metadata server on the source machine, which in turn establishes a connection to the metadata server on the target machine.

If the job fails at this point, check the following items:

- a Open the file *SAS-config-dir*\Lev1\SASMain sasv9.cfg, and make sure that the values of the METASERVER, METAPORT, and METAPROTOCOL options are correct.
- b Open the file *SAS-config-dir*\Lev1\SASMain\MetadataServer\MetadataServer.ext, and make sure that the OMAADMIN and OMAAPW variables hold the user name and password of the source administrator. On Windows, this is typically the SAS Administrator (sasadm), and on UNIX, it is typically the SAS installer account (sas).
- c Make sure that the file mdrepljs.ext exists in the directory *SAS-config-dir*\Lev1\SASMain.
- d Open the file mdrepljs.ext, and make sure that it contains the information that is needed to establish a connection to the metadata server on the administration machine.

- 2 It retrieves the three logins for the SAS Replication Administrator from the metadata repository on the administration machine.

If the job fails at this point, the most likely cause is that the users identified by these logins have not been defined as *administrative* or *unrestricted* users on the appropriate machine. (See Figure 17.2 on page 147.)

- The user referred to in the first login must be an *administrative user* on the administration machine.
 - The user referred to in the second login must be an *administrative user* on the source machine.
 - The user referred to in the third login must be an *unrestricted user* on the target machine.
- 3 It connects to the source metadata server to ensure that the repository to be promoted still exists.

If the repository does not exist and you are rerunning the job from outside the Promotion Wizard, the job will fail.

- 4 It connects to the target metadata server to determine whether a repository with the same object ID as the source repository exists.

If such a repository does not exist and you are rerunning your job outside of the Promotion Wizard, the job will fail.

- 5 It validates the substitution tables defined for the repository promotion. For all four substitution tables (Hosts, Ports, Schema, and Paths), the job retrieves the substitution values from the metadata on the administration server. It then gets the values that will be replaced from the source metadata server and matches or merges the substitutions with the current metadata values. Finally, the job creates an XML file that it will use later when it applies changes to the metadata on the target machine.

At the end of this step, a macro called **mduval.sas** is run. If you want to perform any custom validation, you should add your code here.

- 6 It performs any optional preprocessing.

By default, no preprocessing occurs. You can run custom preprocessing code on the source metadata server host by adding code to the macro **mdpresrc.sas**.

- 7 It runs PROC DATASETS to delete any data sets in the work directory on the source machine.

- 8 It pauses the source metadata server.

- 9 It uses PROC COPY to copy the repository from its location at *SAS-config-dir\Lev1\SASMain\MetadataServer\MetadataRepositories\repository-name* to *SAS-config-dir\Lev1\SASMain\MetadataServer\ReplicationWorkArea\work*.

A failure at this point indicates that the source administrator does not have the necessary file-system permissions to copy the repository.

- 10 It resumes the source metadata server.

- 11 It calls the macro **mdpstsrc.sas**. By default, this macro does nothing. However, you can add code to it to perform any desired postprocessing on the source metadata server host.

- 12 It performs any optional (user-written) preprocessing on the target server. You can specify such preprocessing by adding code to the macro **mdpretrg.sas** on the target metadata server.

- 13 It pauses the target metadata server.

- 14 It copies the metadata being promoted from the source machine to the target machine.

During this step, the job does the following:

- a uses PROC UPLOAD to copy the repository from the work directory on the source machine to a work directory on the target machine
- b cleans out the backup directory on the target machine and then moves the existing repository (if any) to the backup directory
- c cleans out the target repository location and then copies the repository being promoted from the work directory on the target machine to its final location

Any errors that occur at this stage in the process are typically the result of the target administrator not having the necessary file-system permissions.

- 15 It resumes the target metadata server.
- 16 It performs substitutions on the target metadata based on the contents of the XML file mentioned earlier.
- 17 It performs any optional postprocessing on the target machine by executing the macro `mdpsttrg.sas`. By default, this macro is just a stub; however, you can add code to it on the target metadata server to perform custom postprocessing.
- 18 It writes any errors that occur to the log file `errorLog_SAS_Management_Console.txt`, which is located in one of the following paths:
 - On Windows systems: `C:\Documents and Settings\user\`
 - On UNIX systems: `user's-home-directory/`

Moving Associated BI Content Following a Promotion Job

About Moving BI Content

Since the Promotion and Replication Wizards promote only your metadata, you must use other methods to move the BI content that is associated with your metadata. This content includes the following:

- WebDAV folders and their contents (for example, report definition files and portal content). If you are using Xythos WebFile Server (Xythos WFS), then you can use utilities that are delivered with Xythos WFS to promote this content. For instructions, see “Moving Xythos WebFile Server Content” on page 165.
- other portal content, including custom portlets, custom themes, and applications. For instructions, see “Moving Portal Content” on page 167.
- source files for stored processes. Using an appropriate utility or operating system command, move these files to the paths that you specified for stored process source code when you ran the Promotion Wizard.
- physical tables and databases. Using an appropriate utility or operating system command, move these files to the locations that you specified as your library paths when you ran the Promotion Wizard.
- jobs that have been deployed by SAS Data Integration Studio (see “Redeploying Jobs from SAS Data Integration Studio” on page 167).

Moving Xythos WebFile Server Content

If you are using Xythos WFS, then you can use the `WFSDump` and `WFSRestore` utilities to move your WebDAV content. These utilities copy both the WebDAV content and the associated WebDAV properties, including quotas and permissions.

Follow these steps:

- 1 On the source environment's Xythos WFS machine, navigate to the path where the utilities are installed:
 - For Xythos WFS 4.2.34, the path is *drive:\xythos\custom\bin* (Windows) or *install-location/xythos/custom/bin* (UNIX).
 - For Xythos WFS 4.0.48, the path is *drive:\xythos\wfs-4.0.48\utils* (Windows) or *install-location/xythos/wfs-4.0.48/utils* (UNIX).
- 2 Use a text editor to open the file **WFSDump.bat** (Windows) or **WFSDump.sh** (UNIX), and edit the file as follows:
 - For Xythos WFS 4.2.34, change the value of XYTHOS_HOME to the appropriate value for your environment.
 - For Xythos WFS 4.0.48, change the values of XYTHOS_HOME and JDBC_DRIVER to the appropriate values for your environment.

Then save the file.

- 3 Create an empty directory in your file system for the utility to store the output files.
- 4 At an operating system prompt, enter the following command:

```
WFSDump path
```

where *path* is the path for the directory that you created in the previous step.

In this directory, WFSDump writes a file for each unique content resource that it finds on Xythos WFS. It also creates a file called **WFSDump.xml**, which contains metadata for the content resources.

- 5 If the directory is not accessible from the target machine, copy the directory and its contents to a location that is accessible from the target machine.
- 6 On the target machine, use a text editor or XML editor to open **WFSDump.xml**. To this file, make any changes that are necessary to reflect the target environment's Xythos WFS installation. At a minimum, you will need to edit the machine names that are included in domain-qualified user IDs. Do not, however, edit machine names that represent the WebDAV server, because WFSRestore will modify those. For example, in the expression

```
principal='machine-name\sasdemo@machine-name'
```

you would edit the first instance of *machine-name*, but not the second instance.

- 7 On the target machine, navigate to the path where the utilities are installed, as described in step 1. Use a text editor to open the file **WFSRestore.bat** (Windows) or **WFSRestore.sh** (UNIX), and edit the file as follows:
 - For Xythos WFS 4.2.34, change the value of XYTHOS_HOME to the appropriate value for your environment.
 - For Xythos WFS 4.0.48, change the values of XYTHOS_HOME and JDBC_DRIVER to the appropriate values for your environment.

- 8 Enter the following command:

```
WFSRestore path
```

where *path* is the path for the directory that contains **WFSDump.xml** and the content resource files.

WFSRestore writes the folders and files to the target Xythos WebFile Server. For each resource, the utility assigns the properties that are specified in **WFSDump.xml**.

Moving Portal Content

To finish moving portal content following a promotion job, you might need to perform the following tasks on the middle tier of the target environment:

- deploy custom themes
- deploy PAR files for custom portlets
- deploy Web applications that are associated with custom remote portlets
- deploy other applications that are called by the portal
- move WebDAV content (see “Moving Xythos WebFile Server Content” on page 165)

For instructions on deploying portal themes, portlets, and applications, see the *SAS Intelligence Platform: Web Application Administration Guide*.

Redeploying Jobs from SAS Data Integration Studio

To redeploy SAS Data Integration Studio jobs after promoting a repository, follow these steps:

- 1 Use Server Manager in SAS Management Console to make sure that the command for the associated batch server is correct in the target environment.
 - 2 In SAS Data Integration Studio, redeploy all of your jobs by using the **Redeploy Jobs for Scheduling** option on the **Tools** menu.
 - 3 Use Schedule Manager in SAS Management Console to perform one of the following actions:
 - Add the redeployed jobs to existing flows, and reschedule those flows.
 - Create new flows that contain the redeployed jobs, and schedule the new flows.
-

Performing a Demotion

Demotion is the process of copying a metadata repository from the target environment back to the source environment. For example, you might want to refresh your development environment or your test environment by copying metadata from your current production environment.

The preparation steps that are described in this section assume that you have already performed the following tasks:

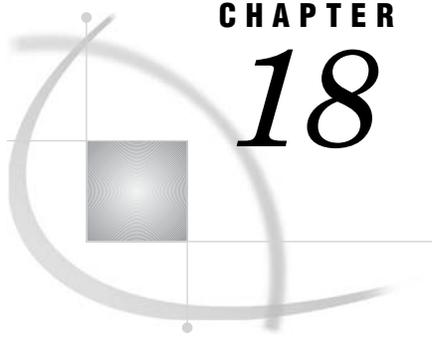
- You have set up your environment for a promotion job as described in “Preparing Your Site for a Promotion Job” on page 140. In this setup, the administration server is Level 0, the source metadata server is Level 1, and the target metadata server is Level 2.
- You have created and run a promotion job as described in “Creating and Running a Promotion Job” on page 152. This job promoted metadata from Level 1 to Level 2.

To prepare your site for a demotion job that copies the metadata repository from Level 2 back to Level 1, follow these steps:

Note: For details about any of these steps, see “Preparing Your Site for a Promotion Job” on page 140. △

- 1 On the Level 2 metadata server machine, install SAS Integration Technologies software (if it is not already installed).

- 2 If you installed Integration Technologies in step 1, use the SAS Configuration Wizard to configure an object spawner and a workspace server on the Level 2 metadata server machine. (An object spawner and workspace server should already be configured on the Level 1 metadata server machine.)
- 3 Make sure that SAS/CONNECT software has been installed on the Level 2 metadata server machine. (This software should have been installed as a prerequisite for performing the original promotion.)
- 4 On the Level 1 machine, configure and start the SAS/CONNECT Spawner. Follow the instructions that are provided in the following topics:
 - “Install and Configure the SAS/CONNECT Software” on page 145
 - “Start the SAS/CONNECT Spawner” on page 145
 - “Give the Target Administrator the Required Directory Permissions” on page 146
- 5 Log on to SAS Management Console as the SAS Replication Administrator, and connect to the Level 0 (administration) metadata server. Then follow these steps:
 - Create a metadata definition for the workspace server which is configured on the Level 2 metadata server machine.
 - Add the Level 2 workspace server to the list of servers that the object spawner can start.
 - Create a metadata definition for a SAS/Connect Server on the Level 1 metadata server machine.
- 6 On the Level 2 metadata server machine, create a file named **mdrepljs.sas** in the directory *SAS-config-dir*\Lev2\SASMain.
- 7 On the Level 0 (administration) metadata server machine, define the user ID that will be performing the demotion as an *administrative user* of the Level 1 server and as an *unrestricted user* of the Level 2 server. If the same user ID will be used to promote and demote repositories, you can define this user ID as an *unrestricted user* of both metadata servers.
- 8 Follow these steps to create the demotion job:
 - a Log on to SAS Management Console as the SAS Replication Administrator, and connect to the Level 0 (administration) metadata server.
 - b Use the Promotion Wizard to create the demotion job. In the substitution table, specify values appropriate to the new target and source machines. For further details, see “Creating and Running a Promotion Job” on page 152.
- 9 Execute the demotion job. After the demotion is complete, update the appropriate metadata in the Level 1 environment, as described in “Making Modifications after Running a Promotion Job” on page 159, and move associated content as described in “Moving Associated BI Content Following a Promotion Job” on page 165.



CHAPTER

18

Promoting Individual BI Objects

<i>Overview of Promoting Individual BI Objects</i>	169
<i>Wizards for Promoting Individual BI Objects</i>	169
<i>Overview of the Promotion Process</i>	170
<i>Selective Promotion</i>	171
<i>Promotion of Content Along With Metadata</i>	172
<i>Restoration of Metadata Associations</i>	173
<i>Optional Promotion of Access Controls</i>	174
<i>How to Access the Export and Import Wizards</i>	174
<i>Special Considerations for Promoting Individual BI Objects</i>	175
<i>Preparing to Use the Export and Import Wizards</i>	176
<i>Set Up Servers and Start Spawners</i>	176
<i>In SAS Management Console, Collect Tables and Libraries under BI Manager</i>	176
<i>Set Up Folders in the Destination Metadata Repository</i>	177
<i>For Reports, Set Up Content Mapping in the Destination Repository</i>	177
<i>Set Up Security for the Folders or Objects</i>	177
<i>For Stored Processes, Set Up a Source Code Repository and an Archive Path (If Needed) in the Target Environment</i>	178
<i>Using the Export and Import Wizards: Example Usage Scenario</i>	179
<i>Description of Scenario</i>	179
<i>Prerequisite Actions</i>	182
<i>Step 1: Export the Folders and Objects</i>	182
<i>Step 2: Import the SPK File</i>	186
<i>Troubleshooting the Export and Import Wizards</i>	188

Overview of Promoting Individual BI Objects

Wizards for Promoting Individual BI Objects

SAS Data Integration Studio 3.4 and SAS Management Console provide a common set of wizards for importing and exporting metadata in SAS Open Metadata Architecture format. You can use the Export Wizard and the Import Wizard in these applications to selectively promote the following types of metadata objects:

- data explorations
- documents
- external files
- folders
- generated transformation

- information maps
- jobs
- libraries
- mining results
- notes
- reports
- stored processes
- tables

With the Export Wizard and Import Wizard, you can promote individual objects or groups of objects from one repository to another, either on the same host machine or on different host machines. The wizards are useful in the following situations:

- You want to move newly developed (or newly modified) metadata objects and their associated content from a test environment to a production environment.
- You want to distribute metadata objects among repositories that belong to different organizational entities.

The source metadata server and the destination metadata server can run on host machines that have different operating systems. For example, you can import objects into a metadata repository that is running on a UNIX system even though those objects were exported from a Windows system.

The Export Wizard and Import Wizard work with dependent repositories as well as foundation repositories. For example:

- You can export objects from a foundation repository, and then import them to a custom repository that is dependent on the foundation repository.
- You can export objects from a dependent repository, and then import them to the foundation repository.

Note: SAS Web Report Studio cannot access metadata that is stored in a dependent repository. Δ

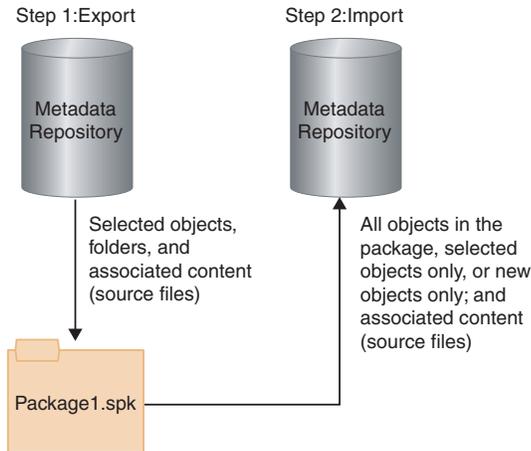
Overview of the Promotion Process

Promotion is a two-step process:

- First, you export the selected objects or folders from the source metadata repository.

The Export Wizard places the exported objects in a SAS Package (SPK) file. If you want the SPK file to be available remotely, then you can instruct the wizard to place the package in a shared location within your file system. You can also e-mail the package after it has been created.

- Second, you import the SPK file to another metadata repository, which can be located either on the same host machine or a different host machine. You can use either the same instance of SAS Management Console or SAS Data Integration Studio, or a different instance to perform the import.

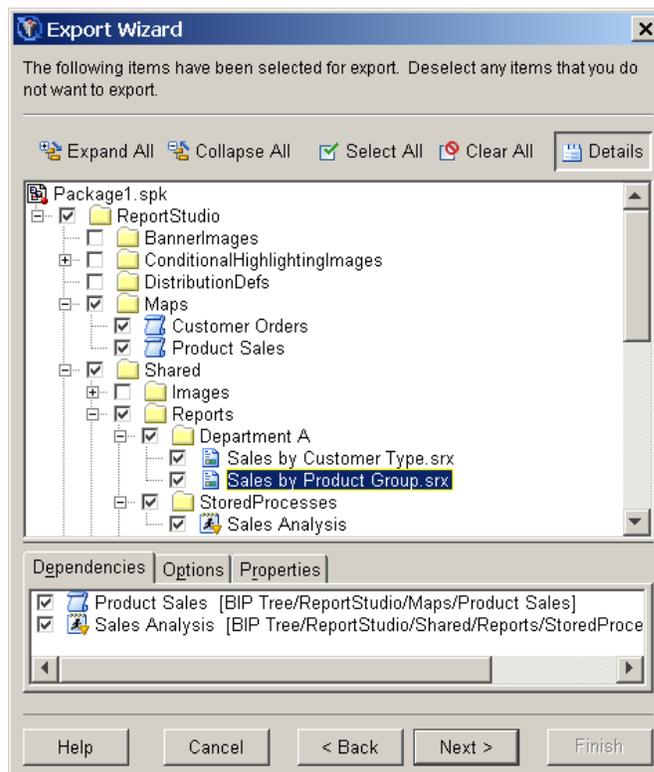


The package format is the same regardless of whether the package is created by SAS Management Console or SAS Data Integration Studio. A package that is exported using SAS Management Console can be imported using SAS Data Integration Studio, and vice versa.

Selective Promotion

The wizards provide the following options for selecting objects to export or import:

- You can select multiple nested folders.
- In SAS Data Integration Studio, you can select an entire repository.
- You can include all of the objects that are contained in a folder (or repository), or just selected objects, as shown in the following example:



- As shown in the preceding example, you can highlight one or more objects in the Export Wizard and view their dependencies. The wizard identifies the following types of dependent objects and enables you to select which ones you want to include in the export:

Table 18.1 Object Dependencies Displayed by the Export Wizard

Object	Dependencies
Information map	Libraries, tables, and stored processes that are used by the information map
Library	Tables that are contained in the library
Job	Nested jobs, mining results, tables, and external files that are used by the job
Report	Information maps and stored processes that are used by the report

- When importing a package, you can import all objects in the package or just new objects (that is, objects that do not already exist in the destination metadata repository).

Promotion of Content Along With Metadata

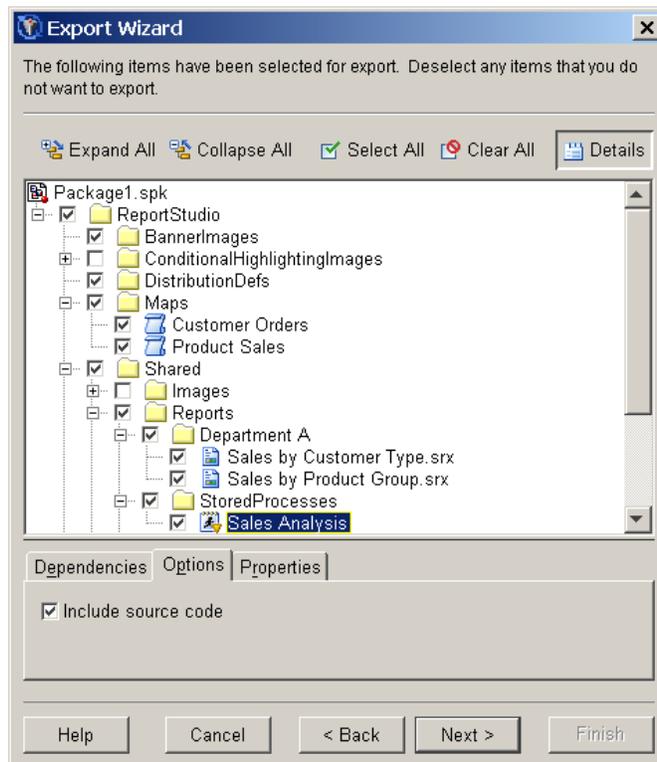
For most object types, the wizards export and import both the content and the metadata. Some content is promoted automatically with the metadata, and other content is promoted optionally, as described in the following table.

Table 18.2 Promotion of Content with Metadata Objects

Object	Associated Content	Promotion Details
External file	Physical file	Promoted optionally
	External format file (EFF)	Promoted optionally
Job	Custom code, preprocesses, and postprocesses	Promoted automatically
Library	XML files and XML maps that are contained in the library	Promoted optionally
Report	XML (.srx) file that defines the report	Promoted automatically to a target WebDAV server or file system path that you designate

Object	Associated Content	Promotion Details
Stored process	Source code (.sas) file	Promoted optionally to the source code repository that you designate. A metadata definition for the source code repository must be present on the target system.
Table	Physical table	Promoted optionally. Use this option with small data sets only. Since data sets are packaged in the .spk file, large data sets could require lengthy processing times.

Content that is promoted optionally is displayed in the wizards' **Options** tab. In the following example, a stored process object is highlighted, and the **Options** tab displays a check box for including the source code with the stored process. The check box is selected by default.



When you export or import an external file, a library, or a table, the option to include the content is not selected by default.

Restoration of Metadata Associations

The Import Wizard displays prompts that assist you in associating imported objects with the appropriate objects and entities in the target environment. Depending on the

types of objects that you are importing, you might be prompted to establish associations with the following objects and entities on the target system:

- a SAS Application Server
- a WebDAV server
- base path on a WebDAV server
- libraries
- tables
- external files
- physical locations for external files or for libraries
- OLAP schemas
- mining results
- a source code repository and an archive path (for stored processes)

You can choose to restore the same metadata associations that were established in the source environment, or you can establish different associations.

Optional Promotion of Access Controls

You can include direct assignments of access control entries (ACEs) and access control templates (ACTs) in the promotion process. The following limitations apply to the promotion of access controls:

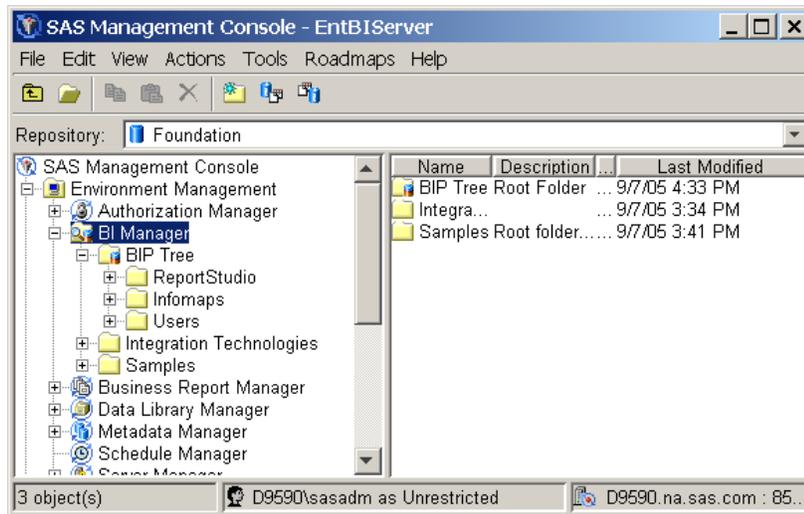
- The wizards cannot promote ACTs, but they can promote direct associations between ACTs and objects. In order for an object's ACT association to be promoted, an ACT of the same name must exist in the destination metadata repository.
- Access control entries and ACT associations that are inherited from a folder are not promoted. For inheritance to take effect in the destination metadata repository, you must also promote the folder that contains the direct access control entries or ACT association.
- Access control entries are not promoted if they refer to users or groups that do not exist in the destination metadata repository (or in the parent Foundation repository, if the destination is a dependent repository).

If you do not specify that you want to include access controls in the promotion process, then access controls are applied as follows:

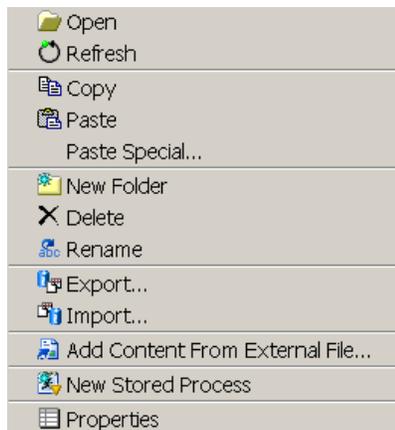
- If you import an object that already exists in the destination metadata repository, then the permissions that have been applied to the existing object are preserved when you overwrite the object.
- If you import an object for the first time, then the object inherits permissions from the folder into which the object is imported.

How to Access the Export and Import Wizards

To access the Export Wizard and Import Wizard in SAS Management Console, navigate to the **BI Manager** node, as shown here:



Right-click any folder or object under the **BI Manager** node (or highlight a folder or object, and then click **Actions** in the menu bar). The applicable BI Manager actions are displayed in a menu, as shown here:



To access the Export Wizard and Import Wizard in SAS Data Integration Studio, right-click any repository, folder, or object in either the Custom tree or the Inventory tree. Or you can highlight a repository, folder, or object in the Custom tree or the Inventory tree, and then click **File** in the menu bar. The menu that appears will include the **Export** and **Import** options, if they are applicable to the object that you have selected.

Special Considerations for Promoting Individual BI Objects

Before choosing to use the Export Wizard and Import Wizard to promote metadata, you should be aware of the following important considerations:

- The wizards can be used to promote only the following metadata objects: data explorations, documents, external files, folders, generated transformations, information maps, jobs, libraries, mining results, notes, reports, tables, and stored processes.

- In order for objects to function properly in the target environment, you must import the resources that objects depend on, unless those resources already exist in the target environment. For example, if you want reports to function properly, you must import the information maps that the reports depend on. If a report has stored processes or images associated with it, then you must import those objects in order for the report to function properly.
- The wizards do not export or import the servers, database schemas, source code repositories, or archive paths that are associated with the BI objects. During the import process, the wizard prompts you to associate the imported objects with the appropriate servers, database schemas, source code repositories, and archive paths in the target environment.
- If you choose to export physical tables along with metadata, you should be aware that large data sets could require lengthy processing times (since data sets are packaged in the SPK file along with the metadata).
- When you export physical tables along with DBMS table objects, do not log on as an *unrestricted* user (for example, sasadm). *Unrestricted users* cannot access the servers that are necessary to export physical DBMS tables.

Preparing to Use the Export and Import Wizards

Set Up Servers and Start Spawners

Before using the Export Wizard and Import Wizard in BI Manager or SAS Data Integration Studio, make sure that the appropriate servers are set up and running, as follows:

- The object spawner must be running in the source environment.
- If you are promoting stored processes, then a stored process server or workspace server (depending on which type of server is used to run the stored process) must exist in the target environment.
- If you are promoting tables, then a workspace server must exist and an object spawner must be running in the target environment.
- If your stored processes create packages that are stored permanently in a WebDAV location, then a WebDAV server must exist in the target environment.
- If you are promoting reports and want to store report content on a WebDAV server, then a WebDAV server must exist and must be running in the target environment.
- The user who launches the Export Wizard must have ReadMetadata access to the objects that are being exported. The user who launches the Import Wizard must have WriteMetadata access to the objects that are being imported. The user must also have the appropriate access to associated content that is being exported and imported.

In SAS Management Console, Collect Tables and Libraries under BI Manager

If you want to use SAS Management Console to export libraries and tables, the tables must appear in the BI Manager tree. To make them appear under the BI Manager tree, follow these steps:

- 1 Right-click the **BI Manager** node.
- 2 On the menu that appears, click **Collect Ungrouped Library Metadata**.

BI Manager places pointers to your libraries and tables in a folder called **Shared Data**. From this folder, you can export and import library and table objects.

You only need to complete this procedure once. From this point forward, new library and table objects that you create in Data Library Manager will automatically appear in the BI Manager tree after you click **View ► Refresh**. The library and table objects will also continue to appear under the **Data Library Manager** node. If you delete any of the objects from the **Shared Data** folder, the objects will also be deleted from the Data Library Manager.

Set Up Folders in the Destination Metadata Repository

If you will be using folders to control access to metadata, then you might want to set up folders in the destination metadata repository before you import your metadata. If you use a folder structure that corresponds to the folder structure in the source metadata repository, then you should import objects at the same level in the folder hierarchy that you exported them from.

For Reports, Set Up Content Mapping in the Destination Repository

Make sure that content mapping has been specified in the destination metadata repository if you are planning to promote reports. Content mapping establishes the location of the XML (.srx) files that define the reports. The location can be either a path on a WebDAV server or a path on the file system.

To specify content mapping:

- 1 Under BI Manager in SAS Management Console, identify the root folder that contains reports. In default installations, this root-level folder is called **BIP Tree**.
- 2 Right-click this folder, and then click **Properties**.
- 3 On the **Content Mapping** tab, enter information about the content location. If you specify a WebDAV server location, you must enter the user ID and password for the SAS Web Administrator user (saswbadm) in the appropriate fields.

Set Up Security for the Folders or Objects

To implement security for promoted objects, you can choose from a variety of options, including the following:

Option 1: Apply ACTs and ACEs to folders and objects in the source environment, and promote the access controls along with the folders and objects.

This option assumes that you have done the following in the source environment:

- 1 You have set up user groups that mirror the groups that will exist in the target environment, and you have defined a small number of users in each group for testing purposes.
- 2 You have created ACTs that define the permission levels for each user group, applied the ACTs to objects and folders, and tested the ACTs to ensure that the appropriate level of security is applied.
- 3 Instead of (or in addition to) ACTs, you have applied ACEs to folders and objects to define the permission levels for each user group, and you have tested the ACEs to ensure that the appropriate level of security is applied.

If you use this option, then your promotion process needs to include the following steps:

- 1 Create copies of the ACTs in the target environment. Be sure to assign the ACTs the same names in both environments.
- 2 When you export and import the objects, select the **Include access controls** option. The ACT associations and ACEs will be promoted and will be re-established in the target environment.

Note: In order for the ACT associations and the ACEs to be promoted, you must promote the objects or folders to which the ACTs were directly applied. Inherited ACT associations and ACEs are not promoted. △

Option 2: Apply ACTs or ACEs to folders and objects directly in the target environment.

For this option, you need to define user groups, ACTs, and ACEs only in the target environment instead of in both environments. You do not select **Include access controls** in the Export Wizard and Import Wizard.

This option involves fewer steps, but it does not provide the opportunity to test your security settings before promotion.

For Stored Processes, Set Up a Source Code Repository and an Archive Path (If Needed) in the Target Environment

If you are promoting stored processes, then you must ensure that a location for the source code files (called a source code repository) exists in the target environment. A metadata definition for the source code repository must also exist in the destination metadata repository.

If the stored processes will create packages that are stored permanently in the file system, then you must ensure that an archive path for the packages exists in the target environment. A metadata definition for the archive path must also exist in the destination metadata repository.

Follow these steps to set up the source code repository and the archive path:

- 1 On the file system of the target application server, identify (or create) the physical directory where the source code files for stored processes will be stored. Ensure that the person who will run the Import Wizard has Write permissions for the source code directory.
- 2 If the stored processes will create packages that are stored permanently in the file system, then identify (or create) the physical directory for the archive path on the target application server.
- 3 *If the destination metadata repository already contains one or more stored process definitions*, then follow these steps to determine whether metadata definitions already exist for the source code repository and the archive path:
 - a In SAS Management Console, right-click any stored process object, and choose **Properties**.
 - b Click the **Execution** tab.
 - c Click the **Source code repository** list to see the list of source code repositories that have been defined.

If a definition already exists for the source code repository that you plan to use, then you do not need to define one. If the definition does not exist, then you must add it as described in step 5.

- d If your stored processes require an archive path, then complete these steps:
 - i On the **Execution** tab, select **Permanent package** from the **Output** list, and click **Details**.
 - ii On the Permanent Package Details dialog box, click the **Publish to server file system** radio button, and then click the **Archive path** list.

If a definition already exists for the archive path that you plan to use, then you do not need to define one. If the definition does not exist, then you must add it as described in step 6.

- 4 *If the destination metadata repository does not have any stored process definitions*, then you must create a stored process strictly for the purpose of creating the source code repository and archive path definitions. If you click **Cancel** at the end of this procedure, the source code repository definition and the archive path definition will be saved, but the stored process definition will not be saved. Follow these steps:
 - a In SAS Management Console, right-click any folder under BI Manager. In the menu that appears, click **New Stored Process**.
 - b In the **Name** field of the New Stored Process Wizard, type any name (for example, **Test**). Then click **Next**.
 - c Select the SAS Server that is associated with the intended location of the source code files.
 - d Use the procedure in step 5 to create the source code repository definition. Use the procedure in step 6 (if needed) to create the archive path definition.
- 5 To add a new source code repository definition, complete these steps either on the **Execution** tab of the Properties dialog box or on the second page of the New Stored Process Wizard:
 - a Click **Manage**.
 - b In the Manage Source Code Repositories dialog box, click **Add**.
 - c In the Add Source Code Repository dialog box, type the fully qualified path for the directory. The path should point to the folder that you created in step 1. Click **OK**.
 - d In the Manage Source Code Repositories dialog box, click **OK**.
- 6 To add a new archive path definition, complete these steps either on the **Execution** tab of the Properties dialog box or on the second page of the New Stored Process Wizard:
 - a If you have not already done so, select **Permanent package** from the **Output** list, and click **Details**.
 - b On the Permanent Package Details dialog box, click the **Publish to server file system** radio button, and then click **Manage**.
 - c In the Manage Archive Paths dialog box, click **Add**.
 - d In the Add Archive Path dialog box, enter the fully qualified path for the folder that is to be used to store packages. The path should point to the folder that you created in step 2. You can also enter a description. Then click **OK**.
 - e Click **OK** in the Manage Archive Paths dialog box and in the Permanent Package Details dialog box.
- 7 Click **Cancel** in the Properties dialog box or the New Stored Process Wizard.

Using the Export and Import Wizards: Example Usage Scenario

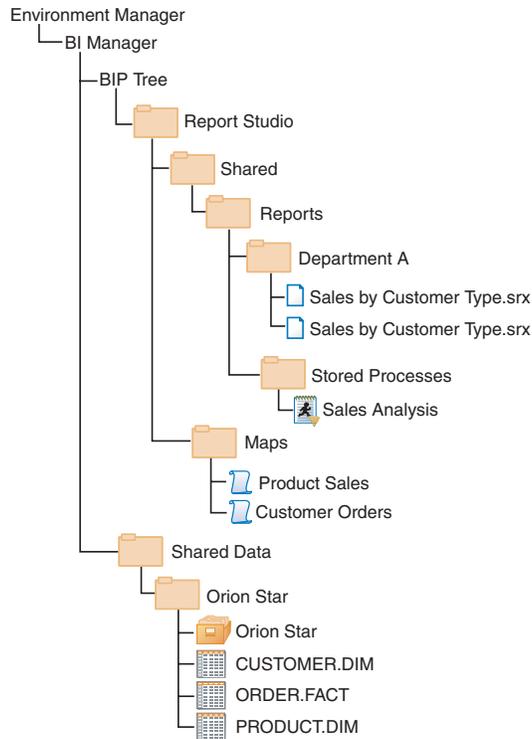
Description of Scenario

In this example, a set of objects has been developed using a SAS Metadata Repository in Environment 1. The objects will be promoted to a different SAS Metadata Repository in Environment 2. The objects to be promoted are as follows:

- a report called Sales by Customer Type

- a report called Sales by Product Type
- a stored process called Sales Analysis
- an information map called Customer Orders
- an information map called Product Sales
- a library called Orion Data
- three SAS tables: CUSTOMER_DIM, PRODUCT_DIM, and ORDERS_FACT

The metadata for these objects is stored in the **BIP Tree** in SAS Management Console, as shown in this figure:



Note: This example scenario uses SAS Management Console to perform the promotion. If you use SAS Data Integration Studio, the procedure is the same except where noted. Δ

Before promoting these objects, it is important to understand their associations with one another, as well as their associations with other objects and entities that do not appear in the tree. The following table describes these associations. It also describes the preparatory actions that must be taken in the target environment to ensure that the associations are restored following the promotion.

Table 18.3 Example Scenario: Objects To Be Promoted and Their Associations

Metadata Object or Folder Name	Associated Objects or Content in the Source Environment	Action Needed to Prepare the Target Environment
Department A folder	Reports_ A ACT	Create the Reports_A ACT. Set up the user groups that the ACT refers to.
Sales by Customer Type report	Sales by Customer Type.srx file (on WebDAV server)	Ensure that a WebDAV server exists.
	Customer Orders information map	None
Sales by Product Group report	Sales by Product Group.srx file	Ensure that a WebDAV server exists.
	Product Sales information map	None
	Sales Analysis stored process	None
Stored Processes folder	Processes_ A ACT	Create the Processes_A ACT. Set up the user groups that the ACT refers to.
Sales Analysis stored process	Analysis.sas source file	Set up a source code directory in the file system, and apply appropriate permissions.
	Source file repository definition	Ensure that a workspace server exists. Create a source file repository definition that corresponds to the source code directory and the workspace server.
	Stored process server	Ensure that a stored process server exists.
Orion Star library	CUSTOMER_DIM, PRODUCT_DIM, and ORDER_FACT tables	Identify a physical location for the library.
CUSTOMER_DIM, PRODUCT_DIM, and ORDER_FACT tables	Physical tables	None

These associations include the following:

- The **Department A** folder and the **Stored Process** folder are each associated with an ACT. If the export and import operations include access controls, then BI Manager will expect to find the ACTs already present in the destination metadata repository.
- Each of the reports is associated with an XML file (with the suffix .srx) on the WebDAV server. For these to be promoted, BI Manager will expect a WebDAV server to exist in the target environment.
- Each of the reports is associated with a different information map. The information maps must be promoted along with the reports in order for the reports to function properly.

- Each of the information maps is associated with specific tables. The table metadata must be promoted along with the information maps in order for the information maps to function properly. The physical tables must also be promoted, or must already exist in the target environment.
- One of the reports is associated with a stored process. The stored process must be promoted in order for the report to function properly.
- The stored process is associated with a source code file that is stored on the file system. For this to be promoted, BI Manager will expect a source code repository to exist on the target application server.
- The stored process is associated with a SAS application server and a source code repository definition. BI Manager will prompt you to establish an association between the stored process and both an application server and a source code repository definition in the target environment.

Prerequisite Actions

In the example steps that follow, it is assumed that the following preparatory actions have already been completed:

- The following physical items must be created (or already present) in the target environment:
 - a WebDAV server
 - a file path for the source code file for the stored process
 - physical tables that correspond to the tables in the source environment (or file paths for these tables, if they are being exported along with the metadata)
- In SAS Management Console, the library and table metadata must appear in folders under BI Manager (see “In SAS Management Console, Collect Tables and Libraries under BI Manager” on page 176). When you use SAS Data Integration Studio, this requirement does not apply.
- The following metadata must be created (or already present) in the destination metadata repository:
 - ACTs that correspond to, and have the same names as, the ACTs in the source metadata repository
 - user group definitions for the groups that the ACTs refer to
 - a server definition for the WebDAV server
 - an application server definition that includes a stored process server and a workspace server
 - a source code repository definition

Step 1: Export the Folders and Objects

Follow these steps to export the folders and objects from the source environment. These steps assume that you have already completed all of the preparatory actions that are described in the preceding section.

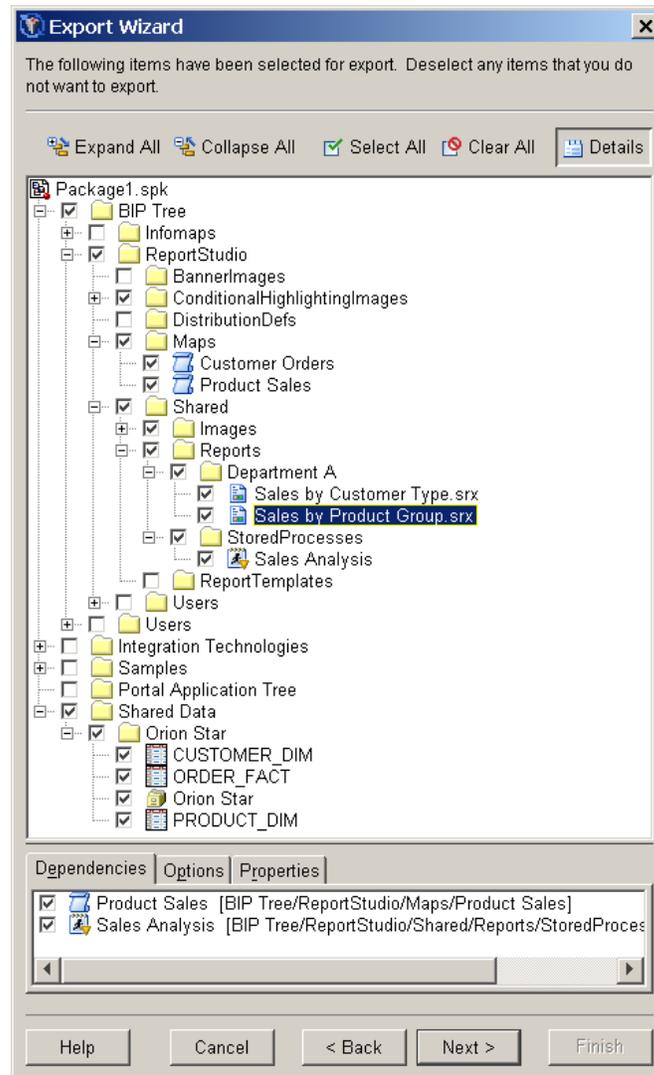
- 1 Open SAS Management Console, and connect to the source metadata repository (that is, the repository from which you want to promote the folders and objects).
- 2 Select the **BI Manager** node.

Note: In SAS Data Integration Studio, you would select the repository that contains the objects that you want to export. △

- 3 Right-click to display the menu, and click **Export**.
- 4 On the first Export Wizard page, complete the following steps:
 - a Enter a name for the SPK file that will contain the exported objects. The default name is **Package1.spk**.
 - b Click **Browse**, and navigate to the location on the file system where you want the SPK file to be saved. For convenience, you might want to choose a location that will be accessible from the target environment.
 - c Click **Include access controls**. This option will retain the associations that have been established between folders and ACTS.



- d Click **Next**.
- 5 On the next Export Wizard page, do the following:
 - a Select and unselect the checkboxes to specify which folders and objects are to be exported, as shown here. Be sure to select the folders whose access controls you want to promote.



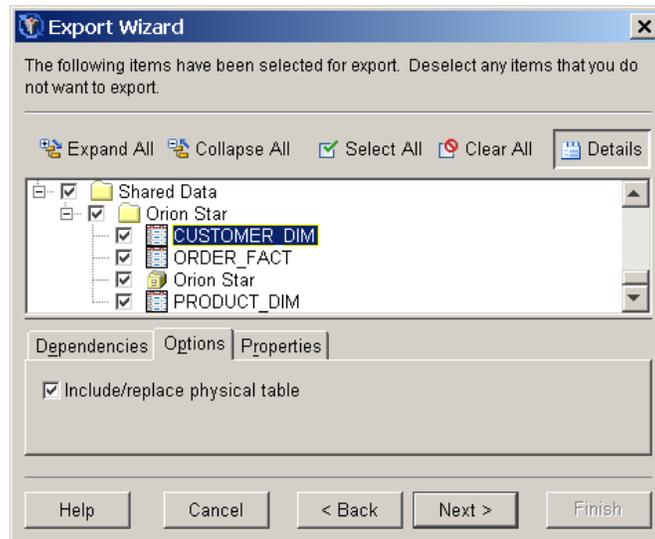
- b Highlight each information map, report, and library object that you have selected. As you highlight each object, view the **Dependencies** tab to ensure that you have included the objects that the highlighted object depends on.

In the preceding example, the report **Sales by Product Group** is highlighted. The **Dependencies** tab lists the **Product Sales** information map and the **Sales Analysis** stored process, which are used by this report. Both objects have been selected to export.

Note: If you press the CTRL key and highlight multiple objects, the **Dependencies** tab displays dependencies for all of the highlighted objects. △

- c Highlight each of the table objects, and use the **Options** tab to specify whether the associated physical table is to be exported with the metadata.

In the following example, the **CUSTOMER_DIM** table is highlighted. On the **Options** tab, the **Include/replace physical table** check box has been selected to indicate that the physical table is to be included in the export.



Note:

- If you choose the option to export physical tables, you should be aware that large data sets could require lengthy processing times (since data sets are packaged in the SPK file along with the metadata).
- If you press the CTRL key and highlight multiple objects of the same type, the **Options** tab enables you to select options for all of the highlighted objects.

△

- d After you have selected all of the appropriate metadata objects, dependencies, and options, click **Next**.
- 6 The next Export Wizard page displays a list of the objects and folders that you selected. If the list is correct, click **Export**.
- 7 As the export process begins, BI Manager might display a prompt asking you to log on to the application server. Enter your SAS Administrator's user ID and password at the prompt.
- 8 When the export process finishes, a log with a date and time stamp is saved in your user directory and a message like the following is displayed:



If errors or warnings are present, you can click **View Log** to view explanatory messages. For assistance in resolving errors, see “Troubleshooting the Export and Import Wizards” on page 188.

Note: For complete step-by-step instructions for using the Export Wizard, see the product Help. △

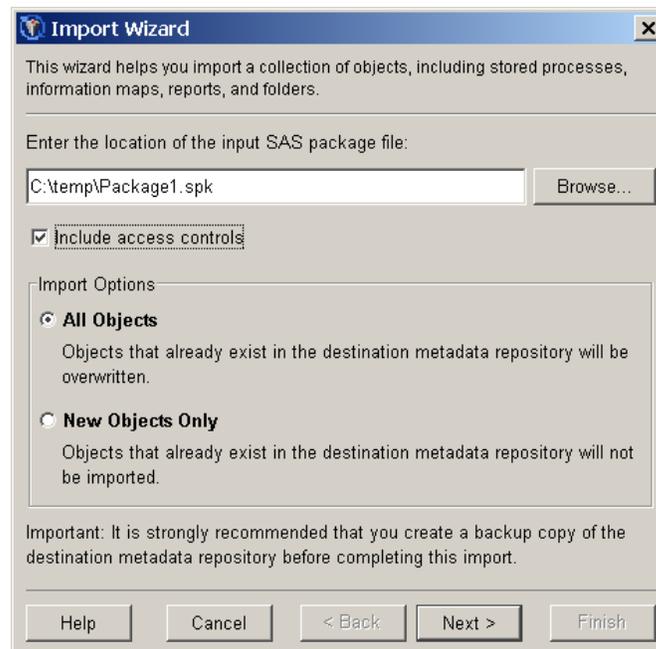
Step 2: Import the SPK File

Follow these steps to import the folders and objects from the SPK file into the target environment. These steps assume that you have completed step 1 and that no errors occurred. These steps also assume that you have completed all of the preparatory actions.

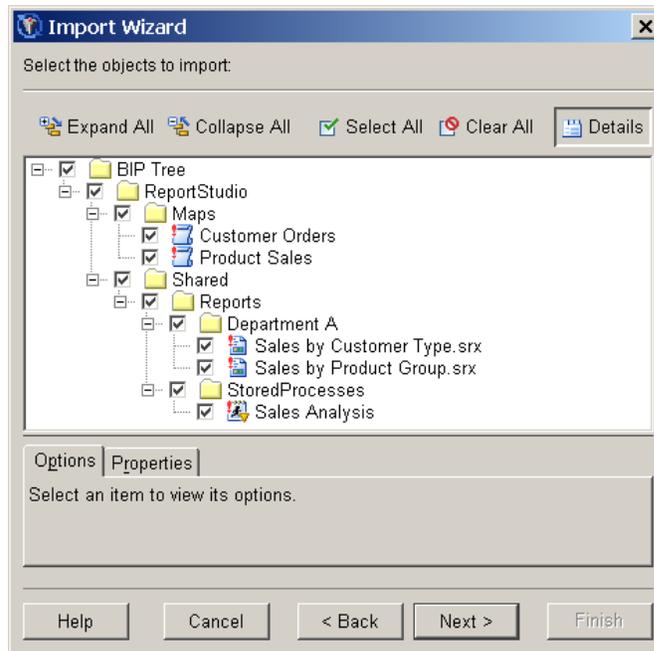
- 1 Locate the SPK file that was created by the Export Wizard. If necessary, copy it, move it, or e-mail it to a location that is accessible from the target environment.
- 2 Open SAS Management Console, and connect to the destination metadata repository (that is, the repository to which you want to promote the folders and objects).
- 3 Select the **BI Manager** node.

Note: In SAS Data Integration Studio, you would select the repository to which you want to import the objects. △

- 4 Right-click to display the BI Manager menu, and click **Import**.
- 5 On the first Import Wizard page, complete the following steps:
 - a Click **Browse**, and navigate to the SPK file that contains the folders and objects.
 - b Click **Include access controls**. This option will retain the associations that have been established between folders and ACTS.
 - c Click the **All Objects** radio button.



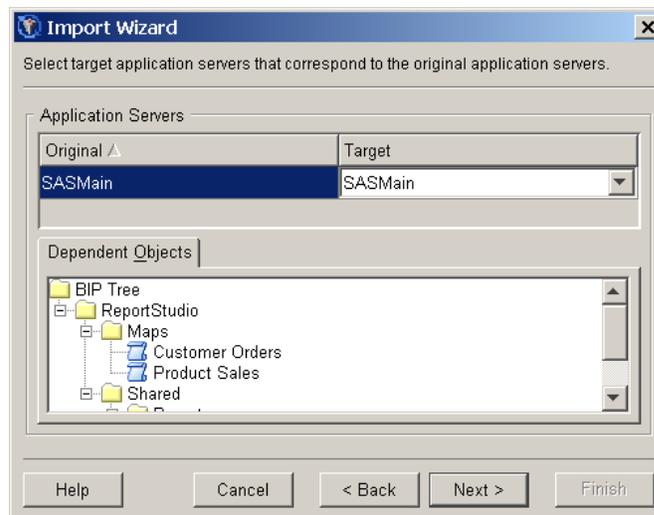
- d Click **Next**.
- 6 The next Import Wizard page displays a tree which shows the objects that are included in the package.



On this page, you can highlight the table objects, and then use the **Options** tab to indicate whether the physical tables are to be imported along with the metadata. You can also edit the table names. (These options are available only if you exported the physical tables.)

Then click **Next**.

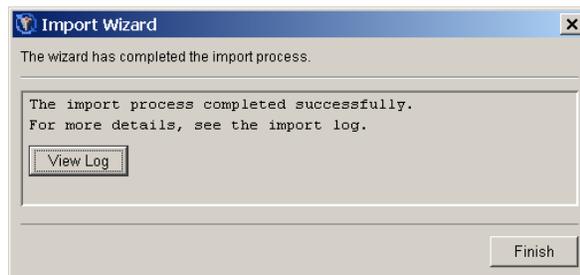
- 7 The next Import Wizard page provides advance information about the metadata values that you will need to select on subsequent windows. For this scenario, these include values for application servers, library locations, and source code repositories. Click **Next**.
- 8 In the **Target** column of the next page, select the application server that the imported objects will be associated with in the target environment.



Click **Next**.

- 9 In the **Target** column of the next page, enter (or browse to) the path where the library is to be located in the target environment. Then click **Next**.

- 10 In the **Target** column of the next page, select the path where the source code file for the stored process should be stored in the target environment. This path is defined in the source code repository definition on the destination metadata repository. Then click **Next**.
- 11 The next page displays a list of the objects that will be imported. The metadata associations for these objects, both in the source environment and the target environment, are also listed. If the information is correct, click **Import**.
- 12 As the import process begins, the wizard might display a prompt asking you to log on to the application server in the target environment. Enter your SAS Administrator's user ID and password at the prompt.
- 13 When the import process finishes, a log with a date and time stamp is saved in your user directory, and a message like the following is displayed:



If errors or warnings are present, you can click **View Log** to view explanatory messages. For assistance in resolving errors, see “Troubleshooting the Export and Import Wizards” on page 188.

Note: The preceding steps are for the example scenario. Additional wizard pages might appear, depending on the types of objects you are promoting. For complete step-by-step instructions for using the Import Wizard, see the product Help. Δ

Troubleshooting the Export and Import Wizards

The following error messages might appear when you run the Export Wizard or Import Wizard in SAS Management Console or in SAS Data Integration Studio:

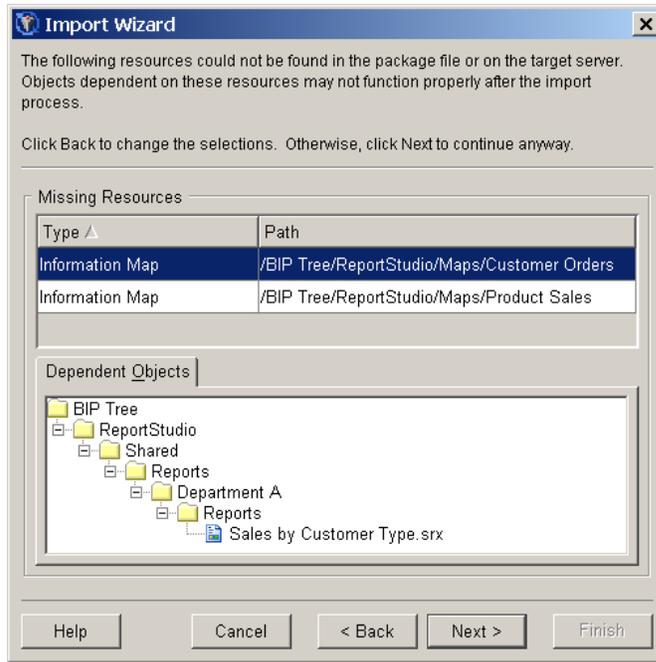
Destination folder has invalid or missing content mapping; cannot import reports.

If this message appears when you try to run the Import Wizard, click **OK** to close the message box and click **Cancel** to stop the import process.

Specify content mapping on the destination metadata repository (see “For Reports, Set Up Content Mapping in the Destination Repository” on page 177), and start the Import Wizard again.

The following resources could not be found in the package file or on the target server. Objects dependent on these resources may not function properly after the import process.

When you run the Import Wizard, this message could appear in a dialog box similar to the following:



This message means that the objects that you are importing depend on objects that are not present in the destination metadata repository.

You will need to import the resources that objects depend on, unless those resources already exist in the target environment. For example, in order for a report to function properly, you must import the information map that the report depends on, as well as any stored processes that are associated with the report.

Could not import "*filename.sas*" in "*path*" on application server "*server name*". Reason: Insufficient authorization to access *path*.

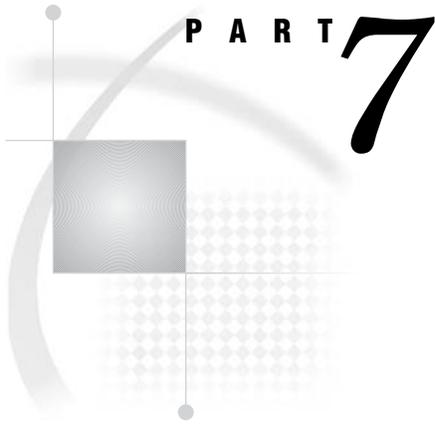
This message appears in the error log at the end of the import process. It means that the user who ran the wizard does not have Write permission for the specified directory in the file system of the target machine. Grant the user Write permission to the directory, and run the wizard again.

Stored Process "*stored process name*" (fqid: "*id*") does not have a Directory metadata object associated with its source File metadata object. This stored process was imported but needs to be fixed. No content was imported.

This message appears in the error log at the end of the import process. It means that the source code file for the stored process was not imported because a source code repository had not been defined. Define a stored process repository on the target system (see "For Stored Processes, Set Up a Source Code Repository and an Archive Path (If Needed) in the Target Environment" on page 178), and run the Import Wizard again.

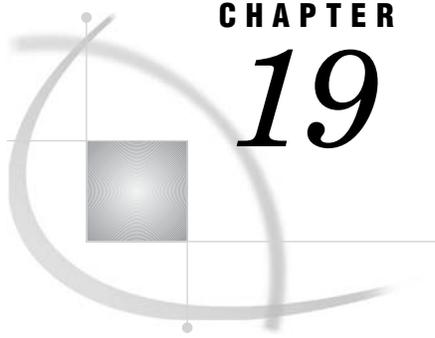
Libname *library name* is not assigned.

This message could appear in the error log if you try to export physical tables along with DBMS table objects while you are logged on as an *unrestricted user*. *Unrestricted users* (for example, sasadm) cannot access the servers that are necessary to export physical DBMS tables. Log on as a different user, and start the export process again.



Scheduling Jobs and Reports

- Chapter 19* **SAS Scheduling Overview** 193
- Chapter 20* **Installing and Configuring Platform Product Suite for SAS** 199
- Chapter 21* **Scheduling Jobs from SAS Data Integration Studio** 205
- Chapter 22* **Using an Operating System Scheduling Server** 211
- Chapter 23* **Enabling the Scheduling of Reports** 219



CHAPTER

19

SAS Scheduling Overview

<i>About SAS Scheduling</i>	193
<i>About Schedule Manager in SAS Management Console</i>	193
<i>About Scheduling Metadata</i>	194
<i>Overview of Scheduling Metadata</i>	194
<i>About Scheduling Server Definitions</i>	194
<i>About SAS Batch Server Definitions</i>	195
<i>About Deployment Directory Definitions</i>	195
<i>About Job and Flow Metadata</i>	196
<i>About Platform Product Suite for SAS</i>	196
<i>About Configuring and Using SAS Scheduling</i>	197

About SAS Scheduling

The SAS scheduling tools enable you to automate the scheduling and execution of SAS jobs across your enterprise computing environment. The scheduling functions are provided through the following tools:

- the software components of Platform Product Suite for SAS, including Platform Flow Manager, the Process Manager Server, and Platform Load Sharing Facility (LSF)
- the Schedule Manager plug-in for SAS Management Console. This tool provides an interface between SAS software and the Platform tools. It can also interact with your operating system scheduler.

The scheduling tools are tightly integrated with SAS applications, including SAS Data Integration Studio, SAS Web Report Studio, and SAS Marketing Automation. The tools are especially designed to handle the scheduling and execution of batch jobs that are generated by these applications. You can also use the tools to schedule SAS jobs that are independent of a specific SAS application.

About Schedule Manager in SAS Management Console

The Schedule Manager plug-in for SAS Management Console is a user interface that enables you to create flows, which consist of one or more SAS jobs. Each job within a flow can be triggered to run based on a date and time, the state of a file on the file system, or the status of another job within the flow.

The flows can include jobs that were created in SAS Data Integration Studio, SAS Marketing Automation, and SAS Web Report Studio. You can also use Schedule

Manager to deploy and schedule SAS jobs that are independent of a specific SAS application.

Schedule Manager is designed as a scheduler-neutral interface. When you create a flow, you specify which scheduling server the flow is to be associated with. Schedule Manager converts the flow information to the appropriate format and submits it to the scheduling server (either the Process Manager Server or an operating system scheduling server).

Reports can be scheduled either with or without the Schedule Manager, since SAS Web Report Studio contains a scheduling wizard that automatically creates flows and passes them to the Process Manager Server.

About Scheduling Metadata

Overview of Scheduling Metadata

The SAS Metadata Server provides central storage and management of metadata about scheduling resources. Before you can use SAS scheduling, metadata must exist for the following resources:

- a scheduling server
- a batch server
- one or more deployment directories (if you are scheduling jobs from SAS Data Integration Studio)
- one or more jobs that have been deployed for scheduling

These metadata types are described in the following topics.

About Scheduling Server Definitions

A scheduling server is a server that executes programs when a specified trigger occurs, such as a date and time. Before you can use SAS scheduling, a definition for a scheduling server must exist in the metadata repository. You can create a definition for a scheduling server by using Server Manager in SAS Management Console.

The scheduling server can be either a Process Manager Server or an operating system scheduling server, as described here:

Process Manager Server

is a server on which the Platform software has been installed.

Through a Java API, a Process Manager Server receives job and flow information from Schedule Manager in SAS Management Console. The Process Manager Server manages the dependencies among the jobs in a flow and controls the submission of jobs to Platform LSF.

The metadata for a Process Manager Server includes the following information:

- the network address of a machine
- the port number for the server

Operating system scheduling server

uses operating system services, such as the Windows AT command, to execute programs when a specified trigger condition occurs.

Operating system scheduling servers can execute jobs in a sequential flow that was created in SAS Management Console's Schedule Manager. However, you

cannot make the execution of a job dependent on error codes that are generated by the previous job. In addition, you cannot use an operating system scheduling server to schedule jobs that are generated by SAS Web Report Studio.

The metadata for an operating system scheduling server includes the following:

- the network address of a machine
- the port number for the server
- the directory on the server where scheduled flows should be stored (control directory)
- the command to start a SAS session on the server

When you create a flow in SAS Management Console's Schedule Manager, you specify which scheduling server the flow is to be associated with. Schedule Manager then passes the flow information to the appropriate scheduling server. (Reports that are scheduled in SAS Web Report Studio are automatically passed to the Process Manager Server. It is not necessary to use Schedule Manager for reports.)

About SAS Batch Server Definitions

A SAS batch server is a metadata object that associates a batch command with a particular machine. SAS Management Console's Schedule Manager uses the information in batch server objects to create the job scripts that are passed to the scheduling server.

Batch server metadata objects are components of the SAS Application Server (for example, SASMain), and can be created by using the Server Manager in SAS Management Console. Each batch server object includes the following attributes:

- the network address of the associated machine

Note: This address is for documentation purposes only. The batch job executes on the machine that is specified in the scheduling server's metadata. △

- a command that will run in batch mode on the scheduling server

Types of SAS batch server objects include the following:

SAS Data Step Batch Server

is used for scheduling SAS programs that contain a DATA step and procedure statements. These include jobs that are created in and deployed from SAS Data Integration Studio. If you have installed SAS Data Integration Studio, then a SAS Data Step Batch Server object should have been defined as part of your initial installation and configuration.

The metadata definition for a SAS Data Step Batch Server must include a SAS start command that will run in batch mode.

SAS Java Batch Server

is used for scheduling Java applications that are supplied by SAS and that are invoked and launched by the standard `java.exe` shell. This server definition is used for reports that are created and scheduled through SAS Web Report Studio, and for jobs that are created in SAS Marketing Automation.

SAS Generic Batch Server

is used for scheduling stand-alone commands or executables that are supplied by SAS. This server can be used to execute scheduled jobs for SAS applications that do not have automated scheduling interfaces.

About Deployment Directory Definitions

A deployment directory is a location on the file system of your scheduling server. When a SAS Data Integration Studio user deploys a job for scheduling, a DATA step job

is generated and placed in this directory. When the job is executed, the job information is read from this directory.

In order for SAS Data Integration Studio users to deploy jobs, a metadata definition for the deployment directory must exist in the metadata repository. The metadata definition includes the following information:

- the name of the SAS Application Server with which the directory is associated
- a name for the directory
- the path to the directory

About Job and Flow Metadata

Job metadata becomes available to Schedule Manager when you use SAS Web Report Studio to schedule a report, when you use SAS Data Integration Studio to deploy a job, or when you use Schedule Manager to deploy a job manually. The job metadata includes the following information:

- the command that is to be used to execute the job
- the name of the SAS batch server that is associated with the job
- the deployment directory that is associated with the job
- the file name of the executable program

Flow metadata is created when you use Schedule Manager to create a flow and when you use SAS Web Report Studio to schedule a report. The flow metadata includes the following information:

- the name of the scheduling server that is to execute the flow
- the triggers and dependencies that have been assigned to each job in the flow

Depending on the scheduling server that the user specifies (either the Process Manager Server or an operating system scheduling server), Schedule Manager converts the flow metadata to the appropriate format and submits it to the scheduling server. Flows that are created by SAS Web Report Studio are automatically directed to the Process Manager Server.

About Platform Product Suite for SAS

Platform Product Suite for SAS is an integrated enterprise job scheduler from Platform Computing Corporation. Platform Product Suite is specifically designed to manage job flows that are generated by SAS software, and includes the following components:

Process Manager Server

controls the submission of jobs to Platform LSF and manages all dependencies among jobs.

Platform Flow Manager

provides a visual representation of flows that have been created for a Process Manager Server. These include flows that were created in SAS Management Console's Schedule Manager, as well as reports that have been scheduled through SAS Web Report Studio.

Platform Flow Manager provides information about each flow's status and associated dependencies. You can view or update the status of jobs within a flow, and you can run or rerun a single job regardless of whether the job failed or completed successfully.

Process Manager Client

enables you to monitor jobs as they run.

Platform Calendar Editor

is a scheduling client for a Process Manager Server. This client enables you to create new calendar entries for time dependencies for jobs that are scheduled to run on the server. You can use it to create custom versions of the calendars that are used to create time dependencies for jobs.

Platform Load Sharing Facility (LSF)

handles the load balancing and execution of batch commands (which represent jobs). Platform LSF manages and accelerates batch workload processing for computer- and data-intensive applications. Process Manager Servers use Platform LSF for the execution of the jobs within a flow.

Platform Grid Management Services

manages jobs that are scheduled in a grid environment. This software collects information on the jobs that are running on the grid and the nodes to which work has been distributed, and makes the information available to the Grid Manager plug-in for SAS Management Console. You can use Grid Manager to view and manage the grid workload information.

For more information, see “Supporting Grid Computing” in the *SAS Intelligence Platform: Application Server Administration Guide*.

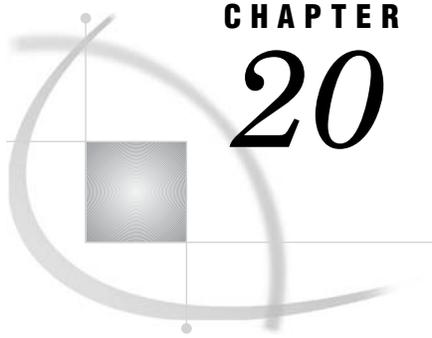
If you wish, you can use your operating system’s scheduling capabilities instead of Platform Product Suite for SAS. However, some scheduling features are not available. In addition, you cannot use an operating system scheduler for jobs that are generated by SAS Web Report studio.

About Configuring and Using SAS Scheduling

See the following topics for details about configuring and using SAS scheduling software:

- Chapter 20, “Installing and Configuring Platform Product Suite for SAS,” on page 199
- Chapter 21, “Scheduling Jobs from SAS Data Integration Studio,” on page 205
- Chapter 22, “Using an Operating System Scheduling Server,” on page 211
- Chapter 23, “Enabling the Scheduling of Reports,” on page 219

Also refer to the product Help for Schedule Manager in SAS Management Console.



CHAPTER

20

Installing and Configuring Platform Product Suite for SAS

<i>Summary of Platform Product Suite Installation and Configuration Tasks</i>	199
<i>Step 1: Create the Necessary Users, Groups, and Logins</i>	199
<i>Set Up the LSF Administrative User and Assign Appropriate Rights</i>	199
<i>Ensure That Metadata Logins Exist for Users Who Will Schedule Reports</i>	200
<i>Step 2: Install Platform Product Suite for SAS</i>	200
<i>Step 3: Give the LSF Administrative User Full Control of Scheduling Folders</i>	200
<i>Step 4: Define the Process Manager Server</i>	201
<i>Resolving Issues Related to Scheduling With Platform Product Suite for SAS</i>	201
<i>Check Error Messages in Platform Flow Manager</i>	201
<i>Resolving Windows-specific Issues Related to Scheduling With Platform Product Suite for SAS</i>	202
<i>Resolving AIX Issues Related to Scheduling With Platform Product Suite for SAS</i>	203
<i>Resolving SAS Exit Behavior When Using Platform Product Suite for SAS</i>	203
<i>Further Resources</i>	204

Summary of Platform Product Suite Installation and Configuration Tasks

To use the scheduling capabilities of Platform Product Suite for SAS, complete the following installation and configuration tasks:

- 1 Create the necessary users, groups, and logins.
- 2 Install Platform Product Suite for SAS.
- 3 Give the LSF administrative user full control of the scheduling folders.
- 4 Define the Process Manager Server (if it has not already been set up by the Configuration Wizard).

In this chapter, the machine where you install the Platform Product Suite software is referred to as the *scheduling server*.

Step 1: Create the Necessary Users, Groups, and Logins

Set Up the LSF Administrative User and Assign Appropriate Rights

The user who is the primary administrator for the Platform scheduling server must be added to the operating system of the scheduling server. The suggested name for this user is `lsfadmin`.

On Windows systems, this user owns the scheduling services. For Windows systems, perform the following tasks on the operating system of the scheduling server:

- 1 Assign rights to the lsfadmin user to **Act as part of the operating system** and **Log on as a batch job**.
- 2 Add the lsfadmin user to the Administrators Group.

This user does not need to be added to the metadata repository.

Ensure That Metadata Logins Exist for Users Who Will Schedule Reports

For each user who will use SAS Web Report Studio, SAS Data Integration Studio, or another application to schedule jobs, a login (including a password) must exist in the metadata repository. Use User Manager in SAS Management Console to add logins for these users.

Step 2: Install Platform Product Suite for SAS

Install Platform Product Suite for SAS, using the installation instructions that are provided. The following software will be installed:

- Process Manager Server
- Process Manager Client
- Platform LSF

Process Manager Server and Platform LSF must both be installed on the same machine. Platform LSF can also be installed on additional machines to support grid configurations. The Process Manager Client can be installed on any machine or machines.

When the Platform Process Manager Setup Wizard prompts for the owner of the Process Manager Server, specify the user that you created in “Set Up the LSF Administrative User and Assign Appropriate Rights” on page 199. For Windows installations, be sure to specify the user’s domain or machine name with the user ID.

When the installation process is complete, check to make sure that the following services are running on the server, with the LSF administrative user as owner:

- LSF *cluster name* LIM
- LSF *cluster name* RES
- LSF *cluster name* SBD
- Process Manager

Note: For a scheduled job to work, Platform LSF must be installed on the same machine as the command that will be used to execute the job. The command is provided by the batch server that is associated with the deployed job. Δ

Step 3: Give the LSF Administrative User Full Control of Scheduling Folders

Give the LSF administrative user full control of all of the LSF and Process Manager directories. On Windows systems, the default directories include the following:

- C:\LSF_6.0**
- C:\Program Files\Platform Computing\Platform Process Manager\3.1**

On UNIX systems, the default directories include the following:

- `/usr/local/LSF_6.0`
- `/usr/local/Platform Computing/Platform Process Manager/3.1`

Step 4: Define the Process Manager Server

A Process Manager Server must be defined in the metadata repository. This server definition might have been created during your initial installation, or you might have created it as part of the process of installing Platform Product Suite for SAS. If the server is not yet defined, define it as follows:

- 1 In SAS Management Console, right-click **Server Manager** and select **New Server**.
- 2 On the first page of the New Server Wizard, under **Scheduling Servers**, select **Platform Process Manager Server**. Click **Next**.
- 3 On the next wizard page, enter a name in the **Name** field (for example, **Process Manager Server**).
- 4 On the next wizard page, click **Next** to accept the default values.
- 5 On the next wizard page, select **DefaultAuth** as the Authentication Domain. In the **Host Name** field, enter the name of the machine that hosts the scheduling server. Accept the default port number of **1966**, or enter a different port number if this one is being used. Click **Next**.
- 6 On the next wizard page, review the settings, and click **Finish**.

After you have completed the installation and configuration of Platform Product Suite for SAS, you must complete additional configuration steps in order to schedule jobs from SAS applications. See the following topics:

- Chapter 21, “Scheduling Jobs from SAS Data Integration Studio,” on page 205
- Chapter 23, “Enabling the Scheduling of Reports,” on page 219

Resolving Issues Related to Scheduling With Platform Product Suite for SAS

Check Error Messages in Platform Flow Manager

Platform Flow Manager enables you to view status and other information about the jobs that have been scheduled on Platform scheduling servers. To view error messages in Platform Flow Manager, follow these steps:

- 1 Open Platform Flow Manager on the scheduling server. On Windows systems, click **Start ► All Programs ► Platform Process Manager ► Flow Manager**.
- 2 Log on as the scheduling user (lsfuser). You might need to include a domain name or machine name with the user ID.
- 3 On the left pane of Flow Manager, expand the folder for lsfuser. If you selected **Run Now** when you scheduled the report, the job will be in a folder called **adhoc**. In the folder, look for the most recently scheduled job. If the job failed, its name might include **(Exit)**.
- 4 Double-click on the job name (for example, **1(Exit)**). An icon for the job will appear in the right pane. A red icon indicates that errors occurred, and a blue icon indicates that the job was successful.
- 5 Right-click the icon in the right pane, and select **View History**.

- 6 If messages appear, check the following list for possible causes:

Could not load user profile for user domain\lsfuser.

On Windows systems, the user IDs lsfuser or lsfadmin might not have the appropriate local policy permissions. For information about the required permissions, see “Assign the Appropriate Rights to the Required Users (Windows Only)” on page 221. Be sure to reboot the system if you make any changes.

Unable to find user.

On Windows systems, the user ID lsfuser might not be defined correctly in the `passwd.lsfuser` file. For information about updating this file, see “Step 4: Update the LSF Password File (Windows Only)” on page 224.

Note: You can also examine the contents of logs in the LSF directory. Go to `C:\LSF_6.0\logs` on Windows, or `/usr/local/LSF_6.0/logs` on UNIX. Δ

Resolving Windows-specific Issues Related to Scheduling With Platform Product Suite for SAS

To properly install the Platform Computing software on Windows systems, you must supply a valid user ID and password to run and administer the services. The user ID that is used to run the installation program must have **Act as part of the operating system** privileges. The user ID that you specify to run the services must have **Log on as a batch job** privileges. Issues can occur if this user does not have the required privileges, if the user’s password has expired, or if the wrong domain name is provided.

One simple way to test the domain, user, and password is to do the following:

- 1 Bring up a DOS command prompt.
- 2 To bring up a new DOS command prompt running as the other user ID, enter the `runas` command using the following syntax:
`>runas /user:DOMAIN\userid cmd`
- 3 Enter the password. A new DOS command prompt appears.

From the new DOS command prompt, you can run the various scheduled commands that are failing. If the commands succeed when they are entered at the prompt, this indicates that there are some issues with the scheduling setup. If the commands fail, then additional information might appear in the DOS command window to help you determine the problem.

In a multi-user environment, in which more than one user is submitting and running flows, the appropriate permission settings must be in place on folders. The Platform scheduling server folders should already be set for scheduler service and administrator accounts and should need no further changes. Make sure that the following files installed by LSF have the following permissions:

Table 20.1 Permissions for Files Installed by LSF

Folder	User Group	Permissions
JSTOP\work	LSF service accounts	Full control (All)(All)
JSTOP\work	LSF administrators	Full control (All)(All)
JSTOP\work	Everyone	Special access (R)(R)

Folder	User Group	Permissions
JSTOP\logs	LSF service accounts	Full control (All)(All)
JSTOP\logs	LSF administrators	Full control (All)(All)
JSTOP\logs	Everyone	Special access (R)(R)
JSTOP\conf\lsfuser.passwd	JS service accounts	Special access (R)(R)

Make sure that the files installed by Platform scheduling servers have the following permissions:

Table 20.2 Permissions for Files Installed by the Platform Scheduling Server

Folder	User Group	Permissions
JSTOP\work	LSF service accounts	Full control (All)(All)
JSTOP\work	JS administrators	Full control (All)(All)
JSTOP\work	Everyone	Special access (R)(R)
JSTOP\log	JS service accounts	Full control (All)(All)
JSTOP\log	JS administrators	Full control (All)(All)
JSTOP\log	Everyone	Special access (R)(R)

Resolving AIX Issues Related to Scheduling With Platform Product Suite for SAS

The AIX environment is unique in that you can configure your kernel to be in either 32-bit mode or 64-bit mode, and your 64-bit applications will run with either configuration.

Platform Computing requires that the kernel be in 64-bit mode for its 64-bit applications. Therefore, your kernel must be configured in 64-bit mode to use Platform scheduling servers and Platform LSF. To determine if your kernel is in 64-bit mode, use the `lsconf` command. Here is an example of the command output that appears if your kernel is in 32-bit mode:

```
$ lsconf -k
Kernel Type: 32-bit
$ lslpp -l | grep bos | grep 64
bos.64bit          5.1.0.50 COMMITTED Base Operating System 64 bit
bos.mp64          5.1.0.50 COMMITTED Base Operating System 64-bit
bos.64bit          5.1.0.35 COMMITTED Base Operating System 64 bit
bos.mp64          5.1.0.35 COMMITTED Base Operating System 64-bit
```

To switch to 64-bit mode, use the `rtfm` command.

Resolving SAS Exit Behavior When Using Platform Product Suite for SAS

During the installation process, the SAS Configuration Wizard creates a script called `sasbatch` in the following path:

SAS-config-dir\Lev1\SASMain\BatchServer\sasbatch

As a default, jobs that are deployed from SAS Data Integration Studio are run using this script.

If you are not using this script, then the Platform scheduling server will report your SAS jobs as completing with a status of **Exit** instead of **Done**. This occurs because Platform scheduling servers treat any non-zero exit code as **Exit** instead of **Done**.

To resolve this issue, you can use the **sasbatch** script that was created by the SAS Configuration Wizard. If you are using a different script, you can modify the script so that it uses a job starter. A job starter is similar to an automatic job wrapper and is responsible for starting the task.

To use a job starter on Windows installations, modify your script as follows:

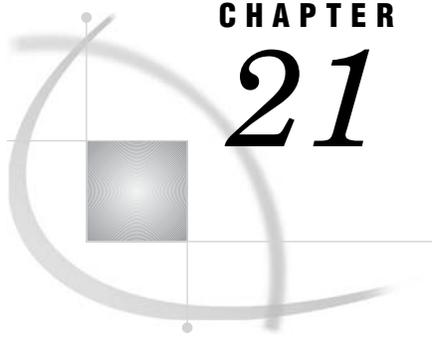
```
call sas.bat %*%
set rc=%ERRORLEVEL%
if %rc==1 goto makenormalexit
exit %rc
:makenormalexit
exit 0
```

To use a job starter on UNIX installations, modify your script as follow:

```
sas $*
rc=$?
if [ $rc -eq 1 ]; then
    exit 0
else
    exit $rc
fi
```

Further Resources

For additional information about managing job submissions, see the white paper *Implementing Site Policies for SAS® Scheduling with Platform JobScheduler* at <http://support.sas.com/documentation/whitepaper/technical/JobScheduler.pdf>.



CHAPTER

21

Scheduling Jobs from SAS Data Integration Studio

<i>Overview of Scheduling for SAS Data Integration Studio</i>	205
<i>Enabling the Scheduling of SAS Data Integration Studio Jobs</i>	206
<i>Requirements for Scheduling SAS Data Integration Studio Jobs</i>	206
<i>Define a Deployment Directory</i>	206
<i>Define a SAS Data Step Batch Server</i>	207
<i>Deploying Data Integration Studio Jobs for Scheduling</i>	208
<i>Scheduling Deployed SAS Data Integration Studio Jobs</i>	208
<i>Troubleshooting the Scheduling of SAS Data Integration Studio Jobs</i>	209

Overview of Scheduling for SAS Data Integration Studio

To schedule jobs that are created by SAS Data Integration Studio, you can use either Product Platform Suite for SAS or your operating system scheduler. If you use the operating system scheduler, fewer scheduling features are available.

Scheduling of SAS Data Integration Studio jobs works as follows:

- 1 Users create jobs by using SAS Data Integration Studio. The job definitions are stored in job objects in the metadata repository.
- 2 Users deploy jobs for scheduling by using SAS Data Integration Studio. The deployment process does the following:
 - a creates a SAS program based on the job metadata
 - b places the SAS program in the specified deployment directory on the scheduling server
 - c creates a metadata object for the deployed job
- 3 An administrator uses Schedule Manager in SAS Management Console to create and schedule a flow that includes the deployed job. Schedule Manager passes the schedule information and the associated commands to the designated scheduling server (either the Process Manager Server or an operating system scheduling server). To construct the commands, Schedule Manager uses the Command Line and Log Directory attributes of the Data Step Batch Server definition.
- 4 If you are using Product Platform Suite for SAS, then Platform Process Manager submits the scheduled job to Platform LSF at the designated time (or after the designated dependent event), and the scheduling server executes the job. Information about the job status is stored on the scheduling server. Administrators can use Platform Flow Manager to view the status of the job.

If you are using an operating system scheduling server, then the operating system submits the job at the designated time. Administrators can use operating system tools to view the status of the job.

Enabling the Scheduling of SAS Data Integration Studio Jobs

Requirements for Scheduling SAS Data Integration Studio Jobs

To enable the scheduling of SAS Data Integration Studio jobs, you must have the following items in place:

a scheduling server

A scheduling server is a server that executes programs when a specified trigger occurs, such as a date and time.

If you are using Platform Product Suite for SAS, then the scheduling server is the server on which you have installed Platform Product Suite for SAS. A corresponding definition for a Process Manager Server must exist in the metadata repository. For detailed setup instructions, see Chapter 20, “Installing and Configuring Platform Product Suite for SAS,” on page 199.

If you are using operating system scheduling, then the scheduling server is the server whose operating system scheduling services you are using. A corresponding definition for an operating system scheduling server must exist in the metadata repository. For detailed setup instructions, see Chapter 22, “Using an Operating System Scheduling Server,” on page 211.

a deployment directory

A deployment directory is a location on the file system of your scheduling server. A corresponding definition for the deployment directory must exist in the metadata repository. When a SAS Data Integration Studio user deploys a job for scheduling, a job is generated and placed in this directory. When the job is executed, the job information is read from this directory.

The initial configuration process should have created a physical deployment directory in the path

SAS-config-dir\Lev1\SASMain\SASEnvironment\SASCode\Jobs, and a corresponding deployment directory definition in the metadata repository.

If you need to set up a deployment directory manually, see “Define a Deployment Directory” on page 206.

a SAS Data Step Batch Server

A SAS Data Step Batch Server is a metadata object that specifies the command to execute SAS code that is generated by SAS Data Integration Studio jobs. The metadata specifies the location of the command on the scheduling server and where the resulting log files are to be stored on the scheduling server.

The initial configuration process should have created a definition for a server called SAS Data Step Batch Server, which is a component of your SAS Application Server (for example, SASMain). The default definition points to the following:

- a command at *SAS-config-dir*\Lev1\SASMain\BatchServer\sasbatch
- a logging directory at *SAS-config-dir*\Lev1\SASMain\BatchServer\logs

If you need to set up a SAS Data Step Batch Server manually, see “Define a SAS Data Step Batch Server” on page 207.

Define a Deployment Directory

The deployment directory definition and the corresponding physical directory should have been set up during your initial installation and configuration. If you need to set them up manually, follow these steps:

- 1 Create the physical deployment directory in the file system of the scheduling server machine. Assign Write permission to this directory to the SAS Data Integration Studio users who will be deploying jobs for scheduling.
- 2 In SAS Management Console, right-click the **Schedule Manager** node and select **Deployment Directories**.
- 3 In the Deployment Directories dialog box, select the application server (for example, **SASMain**) that is associated with the scheduling server machine. Then click **New**.
- 4 In the New Directory dialog box:
 - a Enter a name for the deployment directory (for example, **Batch Jobs**).
 - b Enter the path for the deployment directory, or click **Browse** to navigate to the path. You can enter either a fully qualified path, or a path that is relative to the application server configuration directory (*SAS-config-dir\Lev1\SASMain*).
 - c Click **OK**.
- 5 Click **OK** to exit the Deployment Directories dialog box.

Define a SAS Data Step Batch Server

A SAS Data Step Batch Server should have been set up during your initial installation and configuration. If you need to set it up manually, follow these steps:

- 1 In SAS Management Console, expand the **Server Manager** node.
- 2 Right-click the application server name with which the SAS Data Step Batch Server is associated (for example, **SASMain**), and select **Add Application Server Component**.
- 3 On the first page of the New SAS Application Server Component Wizard, select **SAS Data Step Batch Server** and click **Next**.
- 4 On the next wizard page, enter a name (for example, **SAS Data Step Batch Server**), and click **Next**.
- 5 On the next wizard page, do the following:
 - From the **Associated Machine** drop-down list, select the name of the machine that contains the SAS Application Server.

Note: The machine name attribute is for documentation purposes only. SAS scheduling does not use this machine name to set up or execute the job. △
 - From the **SubType** list, select the operating system of the associated machine.
 - In the **Command** box, enter the path to the command that starts a SAS session on the scheduling server. Your initial configuration provides a command in the following path: *SAS-config-dir\Lev1\SASMain\Batch Server\sasbatch*.
 - In the **Logs Directory** box, enter the path for the directory where the SAS log is to be stored. Your initial configuration provides a logs directory in the following path: *SAS-config-dir\Lev1\SASMain\BatchServer\logs*.
 - If you want to change the rolling log options, click **Advanced Options**.
 - Click **Next**.
- 6 On the next wizard page, review the information you have entered. If it is correct, click **Finish**.

Deploying Data Integration Studio Jobs for Scheduling

When you have completed the appropriate setup tasks, SAS Data Integration Studio users can deploy jobs for scheduling. The procedure for deploying jobs is as follows:

- 1 In SAS Data Integration Studio, right-click a job and select **Scheduling**.
- 2 In the Deploy for Scheduling dialog box, do the following:
 - a Select the application server (for example, **SASMain**) where the job is to be executed.
 - b Select the deployment directory where the generated job is to be stored.
 - c A default file name for the job is displayed. You can modify the file name.
 - d Click **OK**.
- 3 If you are prompted for a user ID and password for the application server, enter the appropriate credentials. The credentials are validated through the application server's workspace server connection.
- 4 Click **OK** in the Deploy for Scheduling dialog box.

SAS Data Integration Studio generates SAS code from the job metadata, transfers the file to the deployment directory location on the scheduling server, and creates job metadata for the Schedule Manager to use.

Scheduling Deployed SAS Data Integration Studio Jobs

To schedule a job that has been deployed from SAS Data Integration Studio, complete the following steps:

- 1 In SAS Management Console, right-click the **Schedule Manager** node and select **New Flow**.
- 2 In the New Flow dialog box, do the following:
 - a Enter a name for the flow. Make sure that the name does not contain spaces.
 - b In the **Scheduling Server** list, select either your Process Manager Server (if you have installed and configured Platform Product Suite for SAS) or your operating system scheduling server (if you have set up operating system scheduling as described in Chapter 22, "Using an Operating System Scheduling Server," on page 211).
- 3 From the **Available Jobs** list, highlight the jobs to be scheduled and move them to the right column. Then click **OK**. The flow appears in the navigation tree. If you expand the flow, the jobs that the flow contains are also displayed.
- 4 If you are using the Process Manager Server, then you can define dependencies among the jobs that are contained in the flow. To do so, right-click each job and select **Manage Dependencies**. You can assign each job a combination of time, job, and file dependencies. When defining time dependencies, you will be prompted to log on to the scheduling server in order to retrieve the default calendars that have been defined.
- 5 To schedule the flow, right-click the flow name and select **Schedule Flow**. Do the following in the Schedule Flow dialog box:
 - If you are using the Process Manager Server, then you can choose **Run Once** (which runs the job immediately), you can choose to run the job manually, or you can choose to run the job on a recurring basis based on a trigger event.

You can click **New** to define a new trigger event, and then select that event as the trigger; or you can choose one of the job dependencies that you defined as the trigger. For example, if you specified a time dependency of Mondays at 3:30 for one of the jobs, then you can schedule the entire flow to be started at this time. After you have scheduled your flow, you can view the flow in the Platform Flow Manager.

- If you are using an operating system scheduling server, then you can choose **Run Once** (which runs the job immediately), you can choose to run the job manually, or you can choose to run the job on a recurring basis based on a time event.

To create a new time event, select **New**. In the New Time Event dialog box, enter the start date and time and select the desired recurrence criteria. After you create the time event, you can select it as the trigger for the flow.

- 6 Click **OK** to exit the Schedule Flow dialog box.

Note: If you use both Foundation and custom repositories, then the following considerations apply:

- If you deploy a job while you are connected to a dependent custom repository in SAS Data Integration Studio, then you must log on to the custom repository in SAS Management Console in order to schedule the job. The job will not be available for scheduling if you are connected to the Foundation repository.
- If you deploy a job while you are connected to the Foundation repository in SAS Data Integration Studio, then you can log on to either the Foundation repository or a dependent custom repository in SAS Management Console in order to schedule the job.

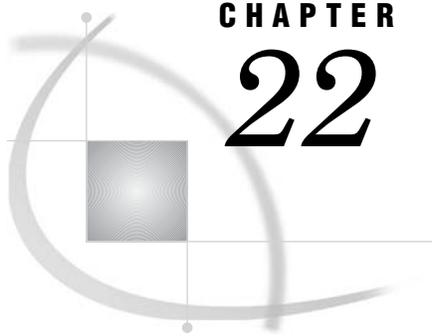
△

Troubleshooting the Scheduling of SAS Data Integration Studio Jobs

User credentials play an important role in the scheduling and execution of SAS Data Integration Studio jobs. When SAS Data Integration Studio deploys a job for scheduling, it generates a SAS file as the `-sysin` value for running SAS in batch mode. Generated SAS programs often contain LIBNAME statements that specify credentials for the data sources in the library. If an issue occurs in scheduling or executing a SAS Data Integration Studio job, check to see if one of the following situations has occurred:

- Has a user's password changed? If the password has changed, you will need to either redeploy the job to update the SAS file with the new credentials, or edit the generated SAS file to update the credentials.
- Are the specified credentials valid in the scheduling environment? The user who generated the job might have credentials that are valid in SAS Data Integration Studio but not in the scheduling environment. If this is true, you will need to either redeploy the job with the appropriate credentials, or edit the generated SAS file to update the credentials.

For additional troubleshooting information, see “Resolving Issues Related to Scheduling With Platform Product Suite for SAS” on page 201.



CHAPTER 22

Using an Operating System Scheduling Server

<i>Capabilities and Limitations of Operating System Scheduling Servers</i>	211
<i>Summary of Setup Tasks</i>	212
<i>Step 1: Assign Local Administration Rights to User (Windows Only)</i>	212
<i>Step 2: Define an Operating System Scheduling Server</i>	212
<i>Step 3: Assign a Spawner to the Operating System Scheduling Server</i>	213
<i>Using Operating System Scheduling on Windows Systems</i>	213
<i>Limitations of Windows Operating System Scheduling Servers</i>	213
<i>How Operating System Scheduling Works on Windows Systems</i>	214
<i>Manually Submitting a Flow for Execution on a Windows Operating System Scheduling Server</i>	214
<i>Canceling a Scheduled Flow on a Windows Operating System Scheduling Server</i>	215
<i>Using Operating System Scheduling on UNIX Systems</i>	215
<i>Limitations of UNIX Operating System Scheduling Servers</i>	215
<i>How Operating System Scheduling Works on UNIX Systems</i>	215
<i>Manually Submitting a Flow for Execution on a UNIX Operating System Scheduling Server</i>	216
<i>Canceling a Scheduled Flow on a UNIX Operating System Scheduling Server</i>	216
<i>Using Operating System Scheduling on z/OS Systems</i>	216
<i>Limitations of z/OS Operating System Scheduling Servers</i>	216
<i>How Operating System Scheduling Works on z/OS Systems</i>	217
<i>Manually Submitting a Flow for Execution on a z/OS Operating System Scheduling Server</i>	217
<i>Canceling a Scheduled Flow on a z/OS Operating System Scheduling Server</i>	217

Capabilities and Limitations of Operating System Scheduling Servers

If you wish, you can use your operating system's scheduling capabilities (instead of Platform Product Suite for SAS) to schedule jobs. However, some scheduling features are not available, as described in the following list:

- You cannot schedule jobs that are generated by SAS Web Report Studio.
- You can use Schedule Manager in SAS Management Console to create a list of jobs. However, the jobs will not necessarily execute in the order in which they are listed. In addition, you cannot make the execution of a job dependent on error codes that are generated by the previous job.
- For specific information about the scheduling capabilities and limitations for each operating system, see the following topics:
 - "Limitations of Windows Operating System Scheduling Servers" on page 213
 - "Limitations of UNIX Operating System Scheduling Servers" on page 215
 - "Limitations of z/OS Operating System Scheduling Servers" on page 216

Summary of Setup Tasks

To enable operating system scheduling, you must have the following:

- a metadata definition for a deployment directory, and a corresponding physical deployment directory. For details, see “Define a Deployment Directory” on page 206.
- a SAS Data Step Batch Server. For details, see “Define a SAS Data Step Batch Server” on page 207.
- a scheduling server. The setup tasks are as follows:
 - 1 Assign local administration rights to users (Windows only).
 - 2 Define an operating system scheduling server.
 - 3 Assign a spawner to the operating system scheduling server.

Step 1: Assign Local Administration Rights to User (Windows Only)

On Windows systems, the user ID under which the job will be run must have **Local Administration** rights on the scheduling server. To grant these rights, add the user to the Administrators Group.

Step 2: Define an Operating System Scheduling Server

Define an operating system scheduling server in the metadata repository, as follows:

- 1 In SAS Management Console, right-click **Server Manager** and select **New Server**.
- 2 On the first page of the New Server Wizard, under **Scheduling Servers**, select **Operating System Services**. Click **Next**.
- 3 On the next wizard page, enter a name (for example, **OS Scheduling Server**).
- 4 On the next wizard page, enter information in the following fields:

SubType

Select the appropriate operating system.

Control Directory

Enter the path for the directory that will store the script for executing a flow. For a default installation, the directory is *sas-config-dir\Lev1\SchedulingServer*.

Command

Enter the appropriate SAS command, making any changes that are necessary for your installation. Sample commands are as follows:

- For Windows:

```
"C:\Program Files\SAS\SAS 9.1\sas" -metaserver
"your.win.host" -metaport 8561
-metarepository "Foundation" -metaprotocol BRIDGE
-log sas-config-dir\Lev1\SchedulingServer\logs
\scheduling_server#Y.#m.#d_#H.#M.#s.log
-logparm "rollover=session"
```

It is recommended that you test the command line by pasting it into a DOS command window. The command should execute with no errors.

- For UNIX:

```
/usr/lib/SAS_9.1/sas -metaserver "your.unix.host"
-metaport 8561 -metarepository "Foundation" -metaprotocol
BRIDGE -log sas-config-dir/Levl/SchedulingServer/logs/
scheduling_server#Y.#m.#d_#H.#M.#s.log
-logparm "rollover=session"
```

- For z/OS:

```
sas-config-dir/Levl/SASMain/startsas.sh
o('log=" ../SchedulingServer/logs/
scheduling_server#Y.#m.#d_#H.#M.#s.log"
logparm="rollover=session"
METASERVER=YOUR.MVS.HOST METAPORT=8561
METAREPOSITORY=Foundation METAPROTOCOL=BRIDGE')--
```

Object Server Parameters

Enter **applelevel=2**.

- 5 On the next wizard page, select **DefaultAuth** as the Authentication Domain. In the **Host Name** field, enter the name of the machine that hosts the scheduling server. Accept the default port number of **8451**, or enter a different port number if this one is being used. Click **Next**.
- 6 On the next wizard page, review the settings, and click **Finish**.

Step 3: Assign a Spawner to the Operating System Scheduling Server

Associate the operating system scheduling server with a spawner definition, as follows:

- 1 In SAS Management Console, right-click the spawner that is associated with the application server (for example, **SASMain Spawner**), and select **Properties**.
- 2 In the Spawner Properties dialog box, select the **Servers** tab.
- 3 From the **Available Servers** list, select the operating system scheduling server and move it to the **Selected Servers** list.
- 4 Click **OK**.

You can now deploy jobs for scheduling (see “Deploying Data Integration Studio Jobs for Scheduling” on page 208). You can also use Schedule Manager in SAS Management Console to schedule flows. See the following topics:

- “Using Operating System Scheduling on Windows Systems” on page 213
- “Using Operating System Scheduling on UNIX Systems” on page 215
- “Using Operating System Scheduling on z/OS Systems” on page 216

Using Operating System Scheduling on Windows Systems

Limitations of Windows Operating System Scheduling Servers

On Windows operating system scheduling servers, SAS 9.1.3 Service Pack 3 supports the scheduling of flows that have a single time event as a trigger. Flows can contain

multiple jobs, but the individual jobs cannot have dependencies. The jobs in a flow are executed serially, and the order of execution is not guaranteed.

The following limitations apply to Windows operating system scheduling:

- If a scheduled job generates a Windows error message, the message is written to a location that is not visible in the user interface. The system will suspend execution of the script and wait for a response. A user with the appropriate rights will then need to stop the specific instance of the Command Shell (`cmd.exe`) that is executing the script.

CAUTION:

Because of this limitation, always test the command that is contained in a job before you schedule the job. Δ

- The user ID under which the job will be run must have **Local Administration** rights. To grant these rights, you must add the user to the Administrators Group.
- Windows scheduling executes jobs using the **at** command. As a default, the **at** command executes jobs using the SYSTEM user name. Therefore, scheduled jobs have full and open access to the machine.

If you do not want the jobs to have full and open access, the system administrator can change the default behavior so that the **at** command runs under a different user name. For instructions, see your operating system documentation.

- Start dates are ignored for every scheduling option except **Run Once**.
- The following types of trigger events are not supported on Windows. If you specify one of these triggers, then a script is created. However, the script is not submitted for execution, and an error message similar to the following is written to the log:

```
ERROR: trigger is currently not supported on Windows systems
ERROR: One or more flows were not successfully scheduled.
```

- **skipCount** if the value of **count** is greater than 1
- **Yearly interval**
- **WhichWeek** (for example, the third week of every month)
- **RunOnce (datelist)** for a date and time that has already passed
- **RunOnce (datelist)** for a date and time that is more than 30 days in the future
- **endTime**

How Operating System Scheduling Works on Windows Systems

When you schedule a flow using a Windows operating system scheduling server, Schedule Manager creates a script file called *flowname.bat*.

The script file and the job status information are written to the path *outdir\XMLFlowDef-name\flow-name*, where *outdir* is the value of the Control Directory attribute in the operating system scheduling server definition.

The path must be a valid Windows path. Valid Windows paths include relative paths, Universal Naming Convention (UNC) paths (for example `\\server\volume\directory`), and mapped paths. If the path is invalid, or if it points to a file instead of a directory, then an error message is written to the SAS log.

Manually Submitting a Flow for Execution on a Windows Operating System Scheduling Server

If you need to manually submit a flow that has been scheduled on a Windows operating system, you can do either of the following:

- Execute the generated script *flowname.bat*. The file is located in the following path:

```
outdir\XMLFlowDef-name\flow-name
```

Either double-click the filename in Windows Explorer, or execute it from a command prompt.

- Use the Windows Task Scheduler to manually schedule the script. To access the Task Scheduler, select **Scheduled Tasks** on the Control Panel. On XP and newer systems, you can also use the **schtasks** command line interface to surface many functions of the Task Scheduler.
- Use the **at** command to manually schedule the script. The capabilities of this command are more limited than those of the Task Scheduler. See your operating system documentation for details.

Canceling a Scheduled Flow on a Windows Operating System Scheduling Server

If you need to cancel a flow that has been scheduled on Windows, you can do either of the following:

- Use the Windows Task Scheduler to cancel the job.
- Use the **/del** option of the **at** command. You can use this method only if the job was submitted with the **at** command.

After a scheduled flow has been canceled, you can remove the script that was created for the flow.

Using Operating System Scheduling on UNIX Systems

Limitations of UNIX Operating System Scheduling Servers

On UNIX operating system scheduling servers, SAS 9.1.3 Service Pack 3 supports the scheduling of flows that have a single time event as a trigger. Flows can contain multiple jobs, but the individual jobs cannot have dependencies. The jobs in a flow are executed serially.

If you schedule a flow to run on a date and time that has already passed, Schedule Manager creates a script. However, the script is not submitted for execution, and an error message similar to the following is written to the log:

```
One or more flows were not successfully scheduled.
NOTE: The SAS System stopped processing this step
because of errors.
```

This behavior is consistent with UNIX operating system scheduling.

How Operating System Scheduling Works on UNIX Systems

When you schedule a flow using a UNIX operating system scheduling server, Schedule Manager creates the following files:

- a flow command script called *flowname.sh*
- a log for the submitted script, called *flowname.log*

The files are written to the path `/outdir/username/flowdefname`, where `outdir` is the value of the Control Directory attribute in the operating system scheduling server definition.

Since the scheduler does not delete either of these files, you must delete them manually when they are no longer needed.

Manually Submitting a Flow for Execution on a UNIX Operating System Scheduling Server

If you need to manually submit a flow that has been scheduled on a UNIX operating system scheduling server, follow these steps:

- 1 Change to the directory where the file `flowname.sh` is located.
- 2 Type `sh flowname.sh now` at the UNIX shell prompt.

Canceling a Scheduled Flow on a UNIX Operating System Scheduling Server

If you need to cancel a flow that has been scheduled on a UNIX operating system scheduling server, follow these steps:

- 1 Open the file `flowname.log`, and find the job identifier for the job that you want to cancel. The log will contain a separate job identifier for each date and time that the flow is scheduled. For example:

```
05/04/20 14:57:30 Flow DailyTestFlow starting commands will be executed
using /bin/ksh job 1114107300.a at Thu Apr 21 14:15:00 2005
```

- 2 At the UNIX shell prompt, type `at -/`
Scheduled jobs will be displayed in the following format:
1114107300.a Thu Apr 21 14:15:00 2005
- 3 Enter the following command at the UNIX shell prompt, where `job-id` is the identifier of the job that you want to cancel: `at -r job-id`.
For example, if the job identifier is 1114107300.a, enter `at -t 1114107300.a`.
Note: For Linux operating systems, the syntax is `atrm job-id`. Δ

Using Operating System Scheduling on z/OS Systems

Limitations of z/OS Operating System Scheduling Servers

On z/OS operating system scheduling servers, SAS Service Pack 3 supports the scheduling of flows that have a single time event as a trigger. Flows can contain multiple jobs, but the individual jobs cannot have dependencies. The jobs in a flow are submitted to the Job Entry Subsystem (JES) for execution in an asynchronous manner.

The following limitations apply to scheduling on a z/OS scheduling server:

- If you schedule a flow to run on a date and time that has already passed, an error is generated and the flow is not executed.
- If a recurrence is specified, then the flow is executed on the next recurrence date that is equal to or later than the current date.

- If the specified start date does not comply with the recurrence criteria, then the initial execution occurs on the first date following the start date that does comply with the recurrence criteria.

For example, suppose you create a schedule on Friday, April 1, for a recurring job that is to run only on weekdays, and you specify a start date of Sunday, April 3. The initial execution will occur on Monday, April 4.

How Operating System Scheduling Works on z/OS Systems

When you schedule a flow, Schedule Manager creates the following files:

- a Restructured Extended Executer (REXX) program called **RunFlow_yymmddhhmmsssss.rex**, which submits each batch job to run on the requested schedule.
- a file called *jobname.status*, which contains information about the status of the job.
- the following files, each of which invokes the REXX program to log the job status:
 - **logstart_***jobname*, which is executed as the first step of each job.
 - **logend_***jobname*, which is executed as the last step of the job if all previous steps have a return code of 0.
 - **logfail_***jobname*, which is executed as the last step of the job if any of the previous steps has a return code other than 0.
- *flowname.cmd*, which contains the statements that execute the program for the flow. This file is passed to the **at** command when the time event trigger occurs.
- *flowname.log*, which contains the log information for the flow.
- *flowname.status*, which contains the current status of the flow.

The files are written to a subdirectory called *username.flowdefname* in the Control Directory that was specified in the operating system scheduling server definition.

Manually Submitting a Flow for Execution on a z/OS Operating System Scheduling Server

Note: On z/OS, unexpected results can occur if you manually submit a flow without having selected **Manually Schedule** when you scheduled the flow. △

If you need to manually submit a flow that has been scheduled on a z/OS operating system scheduling server, follow these steps:

- 1 Change to the directory where the flow's files are located (*outdir/username/flowdefname*), where *outdir* is the value of the Control Directory attribute in the operating system scheduling server definition.
- 2 At the USS prompt, type a command with the following syntax:

```
RunFlow_yymmddhhmmsssss.rex runonce|start
```

Specify **runonce** if you want to run the flow just once, or specify **start** if you want to resume a recurrence that was specified for the flow.

Canceling a Scheduled Flow on a z/OS Operating System Scheduling Server

If you need to cancel a flow that has been scheduled on a z/OS operating system scheduling server, follow these steps:

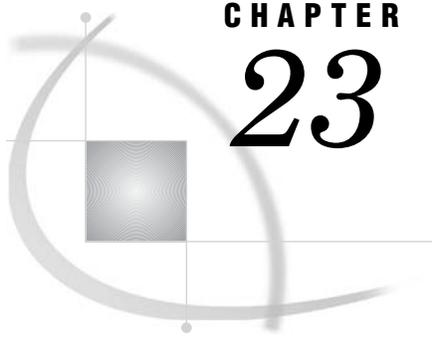
- 1 At the USS prompt, type **at -l**

Note: Type a lowercase L, not the numeral 1. Δ

A list of jobs appears, as in the following example:

```
1116622800.a  Fri May 20 17:00:00 2005
1116734400.a  Sun May 22 00:00:00 2005
1116820800.a  Mon May 23 00:00:00 2005
1116620801.a  Mon May 23 00:00:00 2005
1117029000.a  Wed May 25 09:50:00 2005
```

- 2 For each job that you want to cancel, type the following command at the USS prompt: **at -r at-jobid**.



CHAPTER 23

Enabling the Scheduling of Reports

<i>How Report Scheduling Works</i>	219
<i>Overview of Tasks for Enabling Report Scheduling</i>	220
<i>Step 1: Create the Necessary Users, Groups, and Logins</i>	220
<i>Ensure That the Required Operating System User Accounts Are Present</i>	220
<i>Assign the Appropriate Rights to the Required Users (Windows Only)</i>	221
<i>Define a Scheduling Admins Group in the Metadata</i>	221
<i>Ensure That Metadata Logins Exist for Users Who Will Schedule Reports</i>	221
<i>Step 2: Define a SAS Java Batch Server</i>	222
<i>Step 3: Update the SAS Web Report Studio Configuration File</i>	223
<i>Step 4: Update the LSF Password File (Windows Only)</i>	224
<i>Step 5: Create a Log Directory for the Output Generation Tool, and Apply Appropriate Directory Permissions</i>	224
<i>Step 6: Edit the Initialization File for the Output Generation Tool (Windows Only)</i>	225
<i>Step 7: Verify the Configuration File for the Output Generation Tool</i>	225
<i>Step 8: Define Archive Paths for Publishing Reports to Channels</i>	226
<i>Step 9: Test the Configuration</i>	227
<i>Troubleshooting the Configuration</i>	227
<i>Defining a SAS Java Batch Server to Run SAS Web Report Studio 2.1 and 3.1 Reports</i>	228

How Report Scheduling Works

Scheduling of SAS reports requires Platform Product Suite for SAS. The scheduling process for reports works as follows:

- 1 Users create reports using SAS Web Report Studio. The report definitions (.srx files) are stored in the WebDAV repository (or on a file system), and metadata about the reports (including permissions information) is stored in the metadata repository.
- 2 Users create schedules for particular reports using SAS Web Report Studio. Information about the scheduled reports jobs is stored in the metadata repository. The job information is also passed to the Process Manager Server, which is the default scheduling server for reports. The command for the job is created using information from the SAS Java Batch Server definition.
- 3 Administrators can use Schedule Manager in SAS Management Console to view metadata about scheduled report jobs. Optionally, administrators can use Schedule Manager to create flows that include the scheduled report jobs.
- 4 The scheduling server submits the scheduled job to Platform LSF at the designated time (or after the designated dependent event).

- 5 The Batch Generation Tool, running on the scheduling server, executes the report and places the output on the WebDAV server. Metadata about the report output is stored in the metadata repository, and information about the job status is stored on the Process Manager Server. Administrators can use Platform Flow Manager to view the status of the job.

If report distribution has been set up, then the report output is also sent to the appropriate destinations. For information about report distribution, see “Scheduling and Distributing Pre-generated Reports” in the *SAS Intelligence Platform: Web Application Administration Guide*.

- 6 Users can view the static output by using SAS Web Report Studio (or another application, such as the SAS Information Delivery Portal).

To schedule reports, you must have Platform Product Suite for SAS installed. In addition, the SAS software must be configured correctly. For details, see “Overview of Tasks for Enabling Report Scheduling” on page 220.

Note: An operating system scheduling server cannot be used to schedule reports. Δ

Overview of Tasks for Enabling Report Scheduling

To enable users to use the report scheduling capabilities of SAS Web Report Studio, you must first install and configure Platform Product Suite for SAS. For instructions, see Chapter 20, “Installing and Configuring Platform Product Suite for SAS,” on page 199.

Note: For report scheduling to work, Platform LSF must be installed on the same machine as the SAS Reporting Services component of SAS Query and Reporting Services. In this chapter, this machine is referred to as the *scheduling server*. Δ

When the preceding requirements are in place, complete the following additional setup tasks:

- 1 Create the necessary users, groups, and logins.
- 2 Define a SAS Java Batch Server.
- 3 Update the SAS Web Report Studio configuration file.
- 4 Update the LSF password file (Windows only).
- 5 Create a log directory for the Output Generation Tool, and apply appropriate directory permissions.
- 6 Edit the initialization file for the Output Generation Tool.
- 7 Verify the configuration file for the Output Generation Tool.
- 8 Define archive paths for publishing reports to channels.
- 9 Test the configuration.

Each of these tasks is described in the sections that follow.

Step 1: Create the Necessary Users, Groups, and Logins

Ensure That the Required Operating System User Accounts Are Present

The following user accounts must be added to the operating system of the scheduling server:

Table 23.1 User IDs Needed for Report Scheduling

User ID	Purpose
lsfadmin (suggested name)	Serves as the primary administrator for the Platform scheduling server. On Windows systems, this user owns the scheduling services. This user does not need to be added to the metadata repository. This user should have been added when you installed and configured Platform Product Suite for SAS.
lsfuser (suggested name)	Used by Web Report Studio to run the Output Generation Tool, which creates scheduled reports in batch mode. This user is sometimes referred to as the <i>scheduling user</i> . This user does not need to be added to the metadata repository.
sastrust	Used by the Output Generation Tool to obtain user credentials from the metadata server. In a default installation, this user already exists in the metadata repository.
saswbadm	Used by SAS Web Report Studio to perform the nightly cleanup of partially deleted jobs. In a default installation, this user already exists in the metadata repository.

Assign the Appropriate Rights to the Required Users (Windows Only)

For Windows systems, perform the following tasks on the operating system of the scheduling server:

- 1 Assign rights to the lsfadmin user to **Act as part of the operating system and Log on as a batch job**.
- 2 Add the lsfadmin user to the Administrators Group.
- 3 Add the lsfuser to the SAS Server Users Group.

Define a Scheduling Admins Group in the Metadata

To handle the metadata permissions that are required in order to schedule reports, a user group called Scheduling Admins must exist in the metadata repository. The group must include the sastrust and saswbadm users. To create the group:

- 1 Start SAS Management Console, and connect to the metadata server as the SAS Administrator (for example, sasadm).
- 2 Right-click the **User Manager** icon and select **New ► Group**.
- 3 In the New Group Properties dialog box, select the **General** tab. Enter **Scheduling Admins** in the **Name** box. Enter the text exactly as shown here, with the initial letters capitalized and a single space between the words.
- 4 Select the **Members** tab, and move SAS Trusted User and SAS Web Administrator from the **Available Members** list to the **Current Members** list.
- 5 Click **OK**.

Ensure That Metadata Logins Exist for Users Who Will Schedule Reports

For each user who will use SAS Web Report Studio to schedule reports, a login (including a password) must exist in the metadata repository. To add a login for a user:

- 1 While you are logged onto SAS Management Console as the SAS Administrator, select **User Manager**. A list of users and groups appears in the right pane.
- 2 Right-click the user to whose login you want to add a password, and select **Properties**.
- 3 In the Properties dialog box, select the **Logins** tab. Then right-click the login to which you want to add a password, and click **Modify**.
- 4 In the Edit Login Properties dialog box, enter the user's password in the **Password** and **Confirm Password** boxes. Then click **OK**.
- 5 Click **OK** in the Properties dialog box.

Step 2: Define a SAS Java Batch Server

You also need to define a SAS Java Batch Server as a component of your SAS Application Server. To do this, complete the following steps:

CAUTION:

If you have a SAS Java Batch Server that was defined for a release prior to SAS Web Report Studio 3.1, then delete that definition and create a new one as described in the following instructions. If you modify the existing definition, an error will occur when you schedule a report. △

- 1 In SAS Management Console, expand the **Server Manager** node.
- 2 Right-click the **SASMain** application server, and select **Add Application Server Component**.
- 3 On the first page of the New SAS Application Server Component Wizard, select **SAS Java Batch Server**, and click **Next**.
- 4 On the next wizard page, enter **BRM Java Batch Server** in the **Name** box, and click **Next**.
- 5 On the next wizard page, select the name of the machine that hosts the scheduling server from the **Associated Machine** drop-down list. Select **Business Report Manager** from the **SubType** list. The machine that you select must have Platform LSF installed, as well as the SAS Reporting Services component of SAS Query and Reporting Services.
- 6 In the **Command Line** box, enter a command that contains the following components:

"path-to-outputgen-executable"

The fully qualified path to the **outputgen** executable file, enclosed in quotation marks.

--config-file *configuration-file-location*

The fully qualified path, enclosed in quotation marks, to the configuration file for the Output Generation Tool. (This is the file that is referenced in "Step 7: Verify the Configuration File for the Output Generation Tool" on page 225.)

--repository *metadata-repository*

The name of the repository that contains the job definitions.

Here are examples of valid commands:

Windows example

```
"C:\Program Files\SAS\SASQueryandReportingServices\3.1\outputgen.exe"
--config-file "C:\Program Files\SAS\SASQueryAndReportingServices\3.1\
OutputManagementConfigTemplate.xml" --repository Foundation
```

Note: Enter the command with no line breaks, and be sure to put quotation marks around the arguments as shown in the example. If you copy and paste the command, be sure to include a space after **outputgen.exe**.

You can verify the command by pasting it into a DOS command window. The command should execute with no errors. (You can ignore **Missing option** messages, unless they pertain to the **--config-file** option or the **--repository** option.) △

UNIX example

```
"/usr/local/SAS/SASQueryandReportingServices/3.1/outputgen"
--config-file OutputManagementConfigTemplate.xml --repository Foundation
```

Note: Enter the command with no line breaks. If you copy and paste the command, be sure to include a space after **outputgen**.

You can verify the command by pasting or typing it into a Telnet window. The command should execute with no errors. (You can ignore **Missing option** messages, unless they pertain to the **--config-file** option or the **--repository** option.) △

- 7 On the next wizard page, review the information you have entered. If it is correct, click **Finish**.

Note: If you are upgrading from SAS Web Report Studio 2.1 to version 3.1, but you want to schedule reports for SAS Web Report Studio 2.1 and 3.1 on the same host machine, then complete the additional steps that are provided in “Defining a SAS Java Batch Server to Run SAS Web Report Studio 2.1 and 3.1 Reports” on page 228. △

Step 3: Update the SAS Web Report Studio Configuration File

The credentials of the user ID **lsfuser** must be present in the SAS Web Report Studio configuration file. You might have already provided these credentials during the installation. To verify that the credentials are present, and to update the credentials (if necessary), follow these steps:

- 1 On the middle-tier server (that is, the server on which SAS Web Report Studio is installed), start SAS Management Console and connect to the metadata server using the credentials of the user who installed the software on this host.

Note: If you use SAS Management Console on a different machine, you will not be able to access the SAS Web Report Studio configuration file. △

- 2 Expand the **Application Management** node. Under that node, expand the **Report Studio Configuration** node. The **Web Report Studio** node appears.
- 3 Right-click the **Web Report Studio** node, and select **Open Configuration File**.
- 4 In the Open Configuration File dialog box, navigate to the file **wrs.config** and open it.

Note: The **wrs.config** file is located in the installation directory for SAS Web Report Studio. On Windows systems, the default location is **C:\Program Files\SAS\SASWebReportStudio\3.1**. On UNIX systems, the default location is **SAS-install-dir/SASWebReportStudio/3.1**. △

- 5 In the right pane of SAS Management Console, right-click the value in the column labeled **Host** (on the far left). Then select **Modify Server Definition**.
- 6 In the Modify Server Definition dialog box, look at the value of the **Scheduling Server User ID**. If the field does not contain the user ID **lsfuser**, enter that user ID in the text box. On Windows systems, the ID must be qualified with a domain or machine name.

- 7 In the **Scheduling Server Password** text box, enter the password for lsfuser (if the password is not already present).
- 8 Click **OK** in the Modify Server Definition dialog box.
- 9 Back in the main window of SAS Management Console, right-click the **Web Report Studio** node, and select **Save Configuration File**.
- 10 In the Save Configuration File dialog box, navigate to the file **wrs.config**. (This file name should appear in the **File name** box.) Then click **Save**.
- 11 If you have edited the configuration file, you must redeploy SAS Web Report Studio. For instructions, see “Redeploy SAS Web Report Studio” in the *SAS Intelligence Platform: Web Application Administration Guide*. See also the document **deployment.html**, which is located in your SAS Web Report Studio installation directory.

Step 4: Update the LSF Password File (Windows Only)

If your scheduling server is running on a Windows machine, then you must supply the scheduling server with the credentials for the user ID lsfuser.

The user name and password that you provide in this step must match the ones that you provided in the `wrs.config` file, as described in “Step 3: Update the SAS Web Report Studio Configuration File” on page 223.

Follow these steps:

- 1 On the scheduling server, open a Command Prompt window.
- 2 Use the program **lspasswd** to register the credentials for the user ID lsfuser, as follows:

- a Issue a command similar to this:

```
C:\LSF_60\bin\lspasswd.exe -u .\lsfuser
```

The program prompts you for this user’s password.

- b Enter the user’s password and then confirm it. This completes the registration process.

Step 5: Create a Log Directory for the Output Generation Tool, and Apply Appropriate Directory Permissions

SAS Web Report Studio uses the Output Generation Tool to create sets of reports in batch mode. When a scheduled report is executed on Windows, the tool records information in a logs directory. (On UNIX systems, the log statements are directed to the console when a scheduled report is executed.)

On Windows systems, follow these steps to set up the log directory:

- 1 On the scheduling server, create a directory called **logs** in the installation directory for SAS Query and Reporting Services. For a default Windows installation, the path is as follows:

```
C:\Program Files\SAS\SASQueryandReportingServices\3.1\logs
```

For a default UNIX installation, the path is as follows:

```
SAS-install-dirSASQueryandReportingServices/3.1/logs
```

- 2 Update the permissions on the **logs** directory to give Write access to the SAS Server Users group. (This group should contain the user ID lsfuser.)

Step 6: Edit the Initialization File for the Output Generation Tool (Windows Only)

On Windows systems, you must edit the initialization file for the Output Generation Tool to specify the directory location of the logs that are created when a scheduled report is executed. Complete the following steps:

- 1 Open the **outputgen.ini** file in a text editor. The file is located in the installation directory for SAS Query and Reporting Services.
- 2 Locate the line that starts with:


```
;LogDirectory=
```
- 3 Delete the semicolon at the beginning of the line (which marks the line as a comment).
- 4 After the equal sign, enter the path to the directory in which you want the tool to create the log file. This should be the same directory that you created in “Step 5: Create a Log Directory for the Output Generation Tool, and Apply Appropriate Directory Permissions” on page 224.
- 5 Verify that the **CommandToExecute** property is set to **C:\j2sdk1.4.2_05\bin\java.exe**. This is the path to the Java Runtime Environment that the Output Generation Tool will use. Be sure that the reference to the executable file includes the .exe extension. This version of the Java SDK must be installed on the system.
- 6 Save your changes.

Step 7: Verify the Configuration File for the Output Generation Tool

When you installed SAS Query and Reporting Services, the installation process created a configuration file for the Output Generation Tool. To verify that the configuration file is correct, follow these steps:

- 1 Open the file called **OutputManagementConfigTemplate.xml**, which is located in the installation directory for SAS Query and Reporting Services (on the scheduling server). For a default Windows installation, the file is located in the following path:


```
C:\Program Files\SAS\SASQueryandReportingServices\3.1
```

 For a default UNIX installation, the file is located in the following path:


```
SAS-install-dir/SASQueryandReportingServices/3.1/
```
- 2 In this file, verify that the user ID and password for the SAS Trusted User (sastrust) are specified correctly, as shown here:

```
<User>
  <Name>sastrust</Name>
  <Password>encodedPassword</Password>
  <Trusted>true</Trusted>
</User>
```

If the **<Name>** element includes a domain, then either remove the domain or make sure that the domain is followed by a forward slash (/) rather than a backward slash (\).

The *encodedPassword* will be replaced with the actual password in encoded format.

- 3 Verify that the mail server is specified correctly for your environment. The default value is the local host machine, as shown in this example:

```
<MailServer>
  <!-- Default: localhost -->
  <Name>mailhost</Name>
  <!-- Default: 25 -->
  <Port>25</Port>
</MailServer>
```

If the mail server is not the local host machine, then make sure the host name is specified.

- 4 Edit the values of the following elements, if necessary:

```
<Activated>${RENDERER_OPTIMIZER_SERVER_ENABLED}</Activated>
<Level>${RENDERER_OPTIMIZER_SERVER_LOGLEVEL}</Level>
<File>${OUTPUTGEN_DISTLIST_SERVER_LOGFILE}</File>
```

Depending on your installation, the variables might have already been replaced with the correct values. If the variables do not contain the correct values, then you must edit them as follows:

- Replace `${RENDERER_OPTIMIZER_SERVER_ENABLED}` with `true`.
- Replace `${RENDERER_OPTIMIZER_SERVER_LOGLEVEL}` with `error`.
- Replace `${OUTPUTGEN_DISTLIST_SERVER_LOGFILE}` with the filename and path that will be used for logs that are created when a scheduled report is distributed.

For example, on Windows you might specify the following:

```
C:\SAS\<configuration_dir>\Lev1\SASMain\ReportingServices\logs\wrsdist.log
```

On UNIX, you might specify the following:

```
/users/<user ID>/SAS/<configuration_dir>/Lev1/SASMain/
ReportingServices/logs/wrsdist.log
```

- 5 Update the permissions on the `wrsdist.log` directory to give Write access to the SAS Server Users group. (This group should contain the `lsfuser`.)

Step 8: Define Archive Paths for Publishing Reports to Channels

Scheduled reports can be distributed to channels that are defined using the SAS Publishing Framework. If users will be distributing reports to channels that use archive locations (persistent stores), then the archive location must be accessible to both SAS Web Report Studio and the scheduling software. If the scheduling software and SAS Web Report Studio reside on different machines, then the archive location should be specified as a network path rather than a local path.

To specify the appropriate archive path, follow these steps:

- 1 In SAS Management Console, expand the **Publishing Framework** node.
- 2 Expand the appropriate metadata repository node.
- 3 Expand the **Channels** node.
- 4 Right-click the channel to which you will be publishing reports, and select **Properties**.
- 5 In the Properties dialog box, select the **Persistent Store** tab.

- 6 In the **Path** box, specify the path to a network location that all users who will be creating content can write to.
- 7 Click **OK**.

You might also need to update permissions on the physical folder to allow access for the users who will be publishing reports. For information about setting up publication channels, see “Adding SAS Publication Channels” in the *SAS Intelligence Platform: Web Application Administration Guide*.

Step 9: Test the Configuration

After you have completed steps 1 through 8, use the following procedure to test your configuration:

- 1 On a client machine, use SAS Web Report Studio to create and schedule a report as follows:
 - a Create and save a simple report.
 - b Click **View Report** to display the report.
 - c Select **Report ► Schedule**.
 - d On the Schedule Report page, select **Now**, and click **Next**.
 - e On the next page, click **Finish**. If the word **Running** appears in the status field, wait for the job to finish.
- 2 In SAS Management Console, check the properties of the scheduled report as follows:
 - a Expand the **BI Manager** node, expand the **BIP Tree**, and then expand the folders until you locate the report that you scheduled.
 - b Right-click the report, and choose **Properties**.
 - c Select the **Content** tab. A folder called **Batch** should be displayed on this tab. The folder contains the output of the scheduled report.
- 3 If the **Batch** folder is present on the report’s **Content** tab, as described in the previous step, then return to SAS Web Report Studio and complete these steps:
 - a Log out, and then log back in.
 - b Open the report. The following message should appear at the top of the report:

To interact with this report, you must first refresh the data.

If this message appears, then the configuration is correct.

If the configuration is not correct, see “Troubleshooting the Configuration” on page 227.

Troubleshooting the Configuration

If you experience problems when you test the configuration, first use Platform Flow Manager to check for error messages on the Platform scheduling server. For instructions, see “Check Error Messages in Platform Flow Manager” on page 201.

Then, check to see if any logs were created by the Output Generation Tool. For a default Windows installation, the logs are located in the following path:

C:\Program Files\SAS\SASQueryandReportingServices\3.1\logs

For a default UNIX installation, the path is as follows:

SAS-install-dir/SASQueryandReportingServices/3.1/logs

If no logs are present in this path, check to make sure that you have completed these tasks:

- created a logs directory, and assigned the appropriate user rights to the directory. See “Step 5: Create a Log Directory for the Output Generation Tool, and Apply Appropriate Directory Permissions” on page 224.
- verified that the **LogDirectory** and **CommandtoExecute** statements in the **outputgen.ini** file are correct. See “Step 6: Edit the Initialization File for the Output Generation Tool (Windows Only)” on page 225.
- accurately entered the command in the **Command Line** field when you defined the SAS Java Batch Server in SAS Management Console. See “Step 2: Define a SAS Java Batch Server” on page 222.

Make sure that quotation marks appear around the arguments, as shown in the example. If you copied and pasted the command, make sure that a space appears after **outputgen.exe**.

On Windows systems, you can verify the command by pasting it into a DOS command window. The command should execute with no errors.

If logs are present in the path, then check for the following error messages:

An error occurred: Unable to login user. Cause: Access denied. or Caused by: com.sas.iom.SASIOdefs.GenericError: Error authenticating user SAS Trusted User in function LogonUser.

Make sure that the user ID and password for the SAS Trusted User (sastrust) are specified correctly in the **OutputManagementConfigTemplate.xml** file. You might need to remove the domain, or make sure that the domain is followed by a forward slash (/) rather than a backward slash (\). See “Step 7: Verify the Configuration File for the Output Generation Tool” on page 225.

An error occurred: Exception while rendering report in batch. Cause: cxp.002.ex.msg: The connection factory requires an identity for the authentication domain "DefaultAuth", but the user context provided for user "SAS Demo User" does not have any identities for that domain. Therefore, the user cannot be authorized to use the connection factory.

Make sure that the SAS Web Report Studio user who is trying to schedule the report has a login (including a password) in the metadata repository. See “Ensure That Metadata Logins Exist for Users Who Will Schedule Reports” on page 221.

For additional troubleshooting information, see “Resolving Issues Related to Scheduling With Platform Product Suite for SAS” on page 201.

Defining a SAS Java Batch Server to Run SAS Web Report Studio 2.1 and 3.1 Reports

If you are upgrading from SAS Web Report Studio 2.1 to version 3.1, but want to schedule reports for SAS Web Report Studio 2.1 and 3.1 on the same host machine, then define the SAS Java Batch Server for version 3.1 as described in the previous section. Then, add the following lines to the SAS Web Report Studio **LocalProperties.xml** file. (For instructions on creating a **LocalProperties.xml** file, see “Administrative Files for SAS Web Report Studio” in the *SAS Intelligence Platform: Web Application Administration Guide*.)

```

<wrs.scheduling.batchServerComponentName>
  SAS Java Batch Server Name
</wrs.scheduling.batchServerComponentName>
<wrs.scheduling.schedulingServerComponentName>
  Platform Job Scheduler
</wrs.scheduling.schedulingServerComponentName>

```

The name that you provide for the SAS Java Batch Server should match the name that is shown for the 3.1 server in the Server Manager plug-in. By default, the name is **BRM Java Batch Server**. If the tags are not implemented, then 3.1 uses the first server definition.

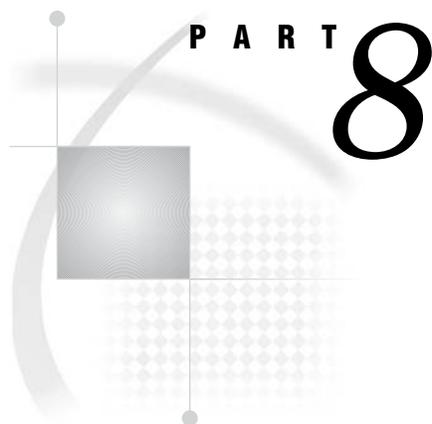
To obtain a list of server names, add the following entry to the **DefaultLoggerProperties.xml** file.

```

<LoggingContext name = "com.sas.apps.citation.model.scheduler"
  priority = "INFO"
  chained = "false">
  <OutputRef outputID = "WRS" />
</LoggingContext>

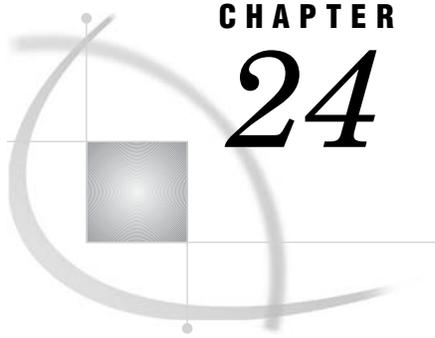
```

The above logging context will generate numerous messages and is intended to be used temporarily only to obtain the names of Java batch servers. After you have defined the server for release 3.1, you should remove or comment out this logging context.



Troubleshooting

Chapter 24 **Troubleshooting the SAS Metadata Server** 233



CHAPTER

24

Troubleshooting the SAS Metadata Server

<i>Overview of Troubleshooting the SAS Metadata Server</i>	233
<i>How to Respond When the SAS Metadata Server Will Not Start</i>	233
<i>Check for a SAS Log File</i>	233
<i>Check for SAS Initialization Failure</i>	234
<i>How to Respond When a Client Cannot Connect to SAS Metadata Server</i>	235

Overview of Troubleshooting the SAS Metadata Server

The following are two common problems that you might encounter when using a SAS Metadata Server:

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

This chapter provides troubleshooting recommendations for these problems.

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

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

How to Respond When the SAS Metadata Server Will Not Start

Check for a SAS Log File

When the metadata server fails to initialize, first check to see if a SAS log file exists. The default path and filename are as follows:

```
sas-config-dir\Lev1\SASMain\MetadataServer\logs\MetadataServer#d#b#y.log
```

- If the SAS log does not exist, then one of the following statements is true:
 - The LOG parameter was omitted from the metadata server start command, or the destination filename does not match the name of the destination directory.

- APPLELEVEL=0 was specified in the metadata server start command. For a description of APPLELEVEL values, see “Administering the SAS Metadata Server Log” on page 69.
- SAS could not initialize. See “Check for SAS Initialization Failure” on page 234.
- If the SAS log file exists, open it and check the text for a problem description. The SAS log file contains error messages that you can use to diagnose and correct initialization problems. Successful initialization is indicated by a message similar to the following:

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

A failed initialization includes this message in the log:

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

Follow the instructions in the error messages. If the error messages indicate a problem in the **omaconfig.xml** file, open the file in an Internet browser. If you see the contents of the file with no other messages, then no syntax errors were found.

- If the file exists but is empty, or if you do not understand the error messages, contact SAS Technical Support for help in diagnosing the problem.

Check for SAS Initialization Failure

Answer the following questions to identify problems with initialization:

Were the start-up parameters entered correctly?

To be sure that the start-up parameters were entered correctly, check the parameters in your start-up command against the information in Appendix 5, “Options for the Metadata Server Invocation Command,” on page 259.

As a second verification step, re-execute the start-up command, omitting the NOTERMINAL system option. When the log file cannot be created, SAS diagnostic messages are written to the SAS Console Log. The NOTERMINAL option suppresses the SAS Console Log so that it does not interfere with server processing. The SAS Console Log will report any problems with server start-up parameters.

If an error is found, edit the metadata server start-up command to correct the problem and to add back the NOTERMINAL option. Then restart the server. For more information about the SAS Console Log, see the companion documentation for your host environment .

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

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

```
proc options;run;
```

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

Are you able to initialize SAS?

At an operating system prompt, issue this command:

```
sas -sysin verify
```

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

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

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

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

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

If the server is running and an attempt by a client to connect fails, answer the following questions to identify the problem:

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

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

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

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

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

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

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

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

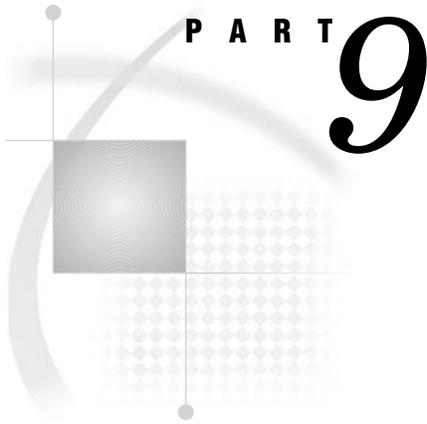
If you are connecting to a server under UNIX, was SAS set up to run as root?

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

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

For user accounts that are authenticated by a domain controller, the server must see a domain qualifier on the user ID. A network domain can be specified as

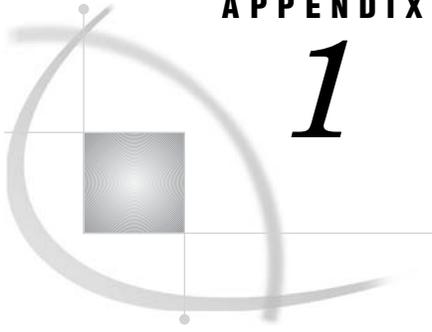
domain\userID or *userID@domain*. For a local user ID, use the machine name as the domain qualifier—for example, *machine-name\userID*. Depending on the interfaces being used, you might have to enter more than one backslash in order for the server to receive a single backslash. A double-backslash typically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



Appendixes

<i>Appendix 1</i>	Default Operating System Users and Groups	239
<i>Appendix 2</i>	Default SAS Ports	243
<i>Appendix 3</i>	Default Log Locations	247
<i>Appendix 4</i>	Configuration Files	251
<i>Appendix 5</i>	Options for the Metadata Server Invocation Command	259
<i>Appendix 6</i>	Configuration Directories	265
<i>Appendix 7</i>	Managing an Environment	275
<i>Appendix 8</i>	Recommended Reading	281

APPENDIX

1

Default Operating System Users and Groups

Users 239

Users

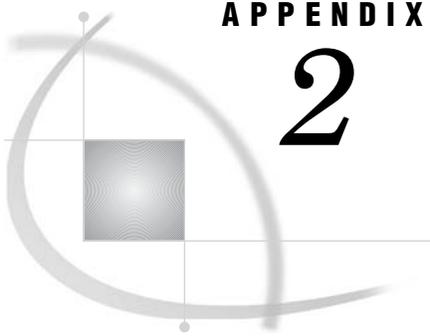
A number of operating system user accounts are needed to run SAS software. The user accounts that are created depend on the software that is installed and the platforms on which that software is installed. For information on these user accounts, see “Setting Up Required Users and Groups” in the *SAS Intelligence Platform: Installation Guide*.

Table A1.1 Special Platform Users

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

User	Role
SAS Guest (sasguest)	The SAS Guest account is used to provide general access to your system's metadata. For example, if you have installed the SAS Information Delivery Portal, this user configures the Public Kiosk for the portal Web application.
SAS Installer (sas)	The installer uses the SAS Installer account when he or she installs and configures software on UNIX and z/OS systems. In addition, the SAS Installer is the owner of configuration directories and their contents and is the process owner for items such as the metadata server, the OLAP server, and the object spawner.
LSF Administrator (lsfadmin)	Necessary if you are installing the Platform Suite for SAS in support of either scheduling or grid computing, this user runs Platform LSF services. In addition, this user is the owner of the LSF configuration and log files and has permission to perform cluster-wide operations, edit configuration files, and reconfigure a cluster.
LSF User (lsfuser)	Necessary if you will be scheduling reports, this user runs command line programs such as the Batch Report Generation Tool.
PostgreSQL User (postgres)	Necessary if you will be using a PostgreSQL database with Xythos WebFile Server.
Pool Administrator (rpooladm)	Necessary only if you are implementing row-level security using workspace server pooling, this user handles requests for processes in the workspace server pool. The password for this account must be unique.
Puddle Log-in Account (rpoolsrv)	Necessary only if you are implementing row-level security using workspace server pooling, this user ID and its associated password must be present in the metadata group containing the SAS Web Report Studio users who will have access to the pool.
SAS Stored Process User (sasspusr)	This user is created on the stored process physical server and requires logon as batch job user rights. It should not have any access to data.
Solutions Installer (slninstl)	Necessary only if you are using SAS Solution Services, this user must exist as a user account on the data-tier machine and must belong to the machine's Administrators group and SAS Server Users group. This identity is used for loading data during the installation and can be removed (from the metadata and from the machine) once installation is complete. For more information, see the <i>SAS Solutions Services: System Administration Guide</i> .
Solutions Role Administrator (slnadm)	Necessary only if you are using SAS Solution Services, the Solutions Role Administrator is a system user that should always be a member of all solutions-created roles. It is used for cases in which a user must perform a query as a part of a larger process, but the query requires a role that the user does not generally need. For more information, see the <i>SAS Solutions Services: System Administration Guide</i> .

User	Role
Administrative User (srcadmin)	Necessary only if you will be promoting metadata from a source to a target machine, this user must set up on the source machine. It needs to be authenticated by the source machine's authentication provider and needs full access to the MetadataRepositories and ReplicationWorkArea directories.
SAS Replication Administrator (sasrpadm)	Necessary only if you will be promoting metadata from a source to a target machine, this user must be set up on the administration machine, for which it will be the administrator.
SAS Forecast Server Administrative User (fsadm)	Necessary only if you are using SAS Forecast Server, this user must be able to authenticate to the SAS Metadata Server machine and the SAS Workspace Server machine. (These could be either the same machine or different machines.) The user is needed to perform post-installation tasks. For more information, see the <i>SAS Forecast Server Administrator's Guide</i> .



APPENDIX

2

Default SAS Ports

Overview **243**

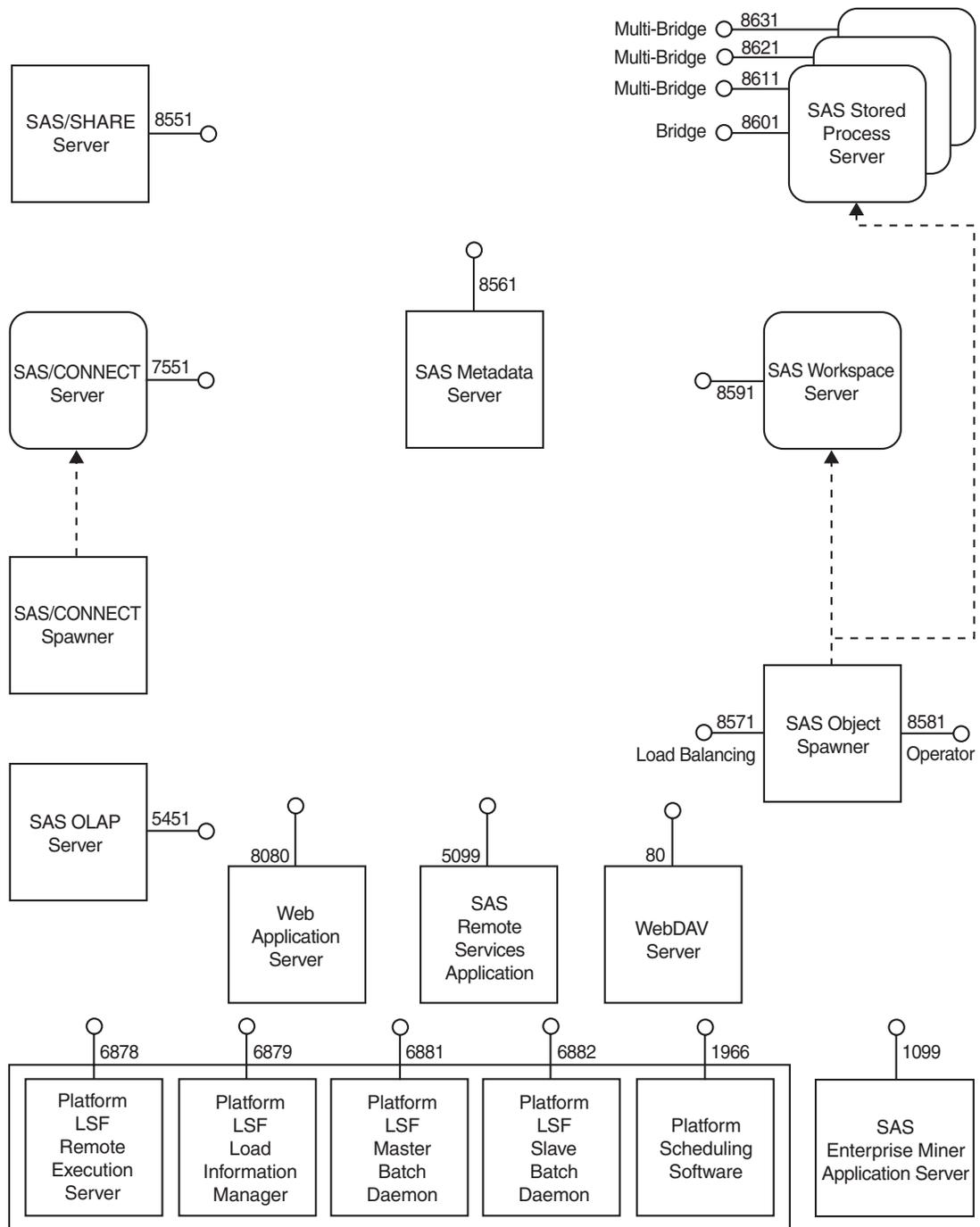
Port Numbering in a Multiple-Level Environment **246**

Overview

The servers in the SAS Intelligence Platform communicate with clients and other servers using TCP/IP. Thus, each server listens on a particular port or ports for incoming requests. The SAS Configuration Wizard configures the servers to use a standard set of ports—unless you tell it to do otherwise.

The default port assignments are shown in the following figure.

Figure A2.1 Default Ports



Note: The port number listed for “Web Application Server” above assumes you are using Apache Tomcat as your servlet container. If you are using a different Java application server, see Table A2.1 on page 245. Δ

If you will be using some nonstandard ports, you also need to know that the SAS Configuration Wizard reads a set of properties to get these port numbers. For example, the port number on which the metadata server will listen is stored in the property OMAPORT. The following table presents a complete list of port-number properties.

(Although there are exceptions, you generally change the value of port-number properties using the SAS Configuration Wizard's Advanced Properties Editor.)

Table A2.1 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
GRID_GMS_PORT	1976	Grid Monitoring Server
DAV_PORT	80	WebDAV Server port (Apache HTTP Server)
DAV_PORT	8300	WebDAV Server port (Xythos WFS)
postgres.port	5432	PostgreSQL database with Xythos WebFile Server
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*

Property Name	Default Port Value	Description
LSB_MDB_PORT	6881	LSF Master Batch Daemon (MBD) port*
LSB_SBD_PORT	6882	LSF Slave Batch Daemon (SBD) port*
PPM_PORT	1966	Platform Process Manager port**
OSSRVSCHEM_PORT	8451	Operating System Services scheduler
EM_APPSRV_PORT	1099	SAS Enterprise Miner application server port***
DBMS_PORT	3306	DBMS port
ANT_SERVER_LISTEN_PORT	17000	ANT server listen port
SERVICES_EVENT_BROKER_ADMIN_PORT	5098	Services Event Broker administration port
SERVICES_EVENT_BROKER_HTTP_PORT	8118	Services Event Broker HTTP port
BESPORT	9221	SAS E-mail Marketing

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

** For information about how to change this value, see the Platform Process Manager documentation.

***For information about how to change this value, see the SAS Enterprise Miner documentation.

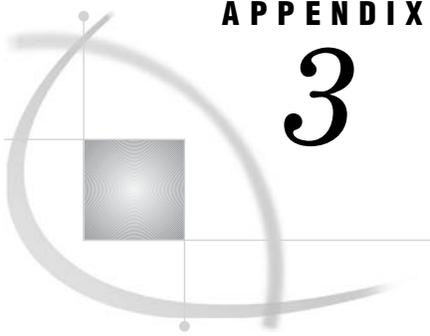
Port Numbering in a Multiple-Level Environment

If you are setting up a multiple-level environment (for example, if your environment consists of development, test, and production levels), you will need to increment each of your port numbers by 1 for each level. For example, in setting up Lev2, use incremented port numbers as shown in the following table.

Table A2.2 Incremented Port Numbers in Level 2 of Multiple-Level Environment

Server	Port Number
SAS Metadata Server	8562
SAS Object Spawner load balancing port	8572
SAS Object Spawner operator port	8582
SAS OLAP Server port	5452
SAS/CONNECT Server port	7552
SAS/SHARE Server port	8552
SAS Workspace Server port	8592
SAS Stored Process Server port	8602, 8612, 8622, 8632

In addition, you will need to set up an administration environment. By convention, port numbers in the administration environment end in 0 (zero). For example, the port number of the metadata server in the administration environment should be 8560.



APPENDIX

3

Default Log Locations

Default Locations for Server Logs 247

Default Locations for Application Logs 248

Default Locations for Server Logs

The following table provides the default file names and locations for server logs.

Note: If your application server has a name other than **SASMain**, replace **SASMain** in the directory path with the appropriate name. For example, your application server might be called **SASApp** if it is installed on a machine other than the metadata server machine. △

Names and Locations of Server Log Files

Object Spawner (listener for SAS Workspace Server and SAS Stored Process Server):

sas-config-dir\Lev1\SASMain\ObjectSpawner\logs\objspawn.log

This log is not enabled by default. To enable logging, see “Enabling Logging for the SAS Object Spawner” on page 89.

SAS/CONNECT Spawner:

sas-config-dir\Lev1\SASMain\ConnectServer\logs\spawner.log

SAS Data Step Batch Server:

sas-config-dir\Lev1\SASMain\BatchServer\logs

SAS Metadata Server:

sas-config-dir\Lev1\SASMain\MetadataServer\logs\MetadataServer#d#b#y.log

This file records information about users who connect to the server. Refer to this file if a client is unable to access a repository.

SAS OLAP Server:

sas-config-dir\Lev1\SASMain\OLAPServer\logs\OLAPServer#d#b#y.log

SAS/SHARE Server:

sas-config-dir\Lev1\SASMain\ShareServer\shareServer#d#b#y.log

SAS Stored Process Server:

*sas-config-dir\Lev1\SASMain\StoredProcessServer\logs\
Stored_Process_Server_%v.log*

SAS Workspace Server:

sas-config-dir\Lev1\SASMain\WorkspaceServer\logs

This log is not enabled by default. To enable logging, use SAS Management Console. Do not enable logging directly in **sasv9.cfg**, since this file is used by other servers.

To set logging options for SAS servers, see:

- Chapter 9, “Administering the SAS Metadata Server Log and Audit Trail,” on page 69
- Chapter 11, “Administering Logging for the SAS Object Spawner, SAS Workspace Server, SAS Stored Process Server, and SAS/CONNECT Server,” on page 89
- Chapter 12, “SAS OLAP Server Monitoring and Logging,” on page 97

Default Locations for Application Logs

The following table provides the default log file names and locations for logs for SAS Intelligence Platform applications.

Names and Locations of Application Log Files

Platform Product Suite for SAS:

C:\Program Files\Platform Computing\Platform Process Manager\log
C:\Program Files\Platform Computing\Platform Process Manager\work\history
C:\LSF6.0\logs

SAS Data Integration Studio:

On Windows systems:

C:\Documents and Settings\user\errorLog_SAS_Data_Integration_Studio.txt

On UNIX systems:

user's-home-directory/errorLog_SAS_Data_Integration_Studio.txt

SAS Information Delivery Portal:

sas-config-dir\Lev1\web\Deployments\Portal\portal.log

sas-config-dir\Lev1\web\Deployments\Portal\prefs.log

sas-config-dir\Lev1\web\Deployments\Portal\stp.log

SAS Information Map Studio:

On Windows systems:

C:\Documents and Settings\user\IMS_log.log

On UNIX systems:

user's-home-directory/IMS_log.log

By default, logging is not enabled. To enable logging, see “Administering SAS Information Map Studio” in the *SAS Intelligence Platform: Desktop Application Administration Guide*.

SAS Management Console:

On Windows systems:

C:\Documents and Settings\user\errorLog_SAS_Management_Console.txt

On UNIX systems:

user's-home-directory/errorLog_SAS_Management_Console.txt

SAS Management Console (BI Manager):

On Windows systems:

C:\Documents and Settings\user\My Documents\Export_date-and-time.log

C:\Documents and Settings\user\My Documents\Import_date-and-time.log

C:\Documents and Settings\user\My Documents\CopyPaste.log

On UNIX systems:

user's-home-directory/Export_date-and-time.log

user's-home-directory/Import_date-and-time.log

user's-home-directory/CopyPaste.log

SAS OLAP Cube Studio:

On Windows systems:

C:\Documents and Settings\user\errorLog_SAS_OLAP_Cube_Studio.txt

On UNIX systems:

user's-home-directory/errorLog_SAS_OLAP_Cube_Studio.txt

SAS Services Application (Remote Services):

sas-config-dir\Lev1\web\Deployments\RemoteServices\logs\services.log

SAS Web OLAP Viewer for Java:

sas-config-dir\Lev1\web\Deployments\SASWebOlapViewerforJava\logs

SAS Web Report Studio:

The following log records events such as application use, failed attempts to log on, report access, and batch report activities:

sas-config-

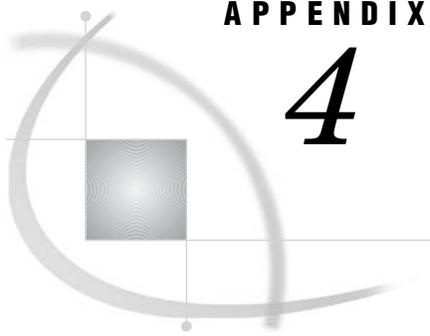
dir\Lev1\web\Deployments\WebReportStudio\logs\WebReportStudio_KeyActions.log

The following log records events such as serious errors, application restarts, and users logging on:

sas-config-dir\Lev1\web\Deployments\logs\WebReportStudio\WebReportStudio.log

To set logging options for applications, see the following documentation:

- *SAS Intelligence Platform: Desktop Application Administration Guide*
- *SAS Intelligence Platform: Web Application Administration Guide*



APPENDIX

4

Configuration Files

<i>Location of Configuration Files for SAS Servers</i>	251
<i>Using adminUsers.txt to Configure New Privileged Users of the SAS Metadata Server</i>	253
<i>Using trustedUsers.txt to Configure New Trusted Users</i>	253
<i>Reference Information for omaconfig.xml</i>	254
<i>About the omaconfig.xml File</i>	254
<OMA> Options	254
<RPOSMGR> Options	257
Reserved omaconfig.xml Options	257
Reserved <OMA> Options	258
Debugging Options	258

Location of Configuration Files for SAS Servers

On server-tier machines that host the SAS Metadata Server, the configuration files for SAS servers are located in the following path:

sas-config-dir\Lev1\SASMain\

On server-tier machines that do not host the SAS Metadata Server, the configuration files for SAS servers are located in the following path:

sas-config-dir\Lev1\SASApp\

Within this path, the configuration files for specific SAS servers are contained in subdirectories that correspond to each server.

The **SASMain** or **SASApp** directory also contains a file called **sasv9.cfg**, which contains SAS system options that are shared by the SAS Metadata Server, SAS Workspace Servers, SAS OLAP Servers, and SAS Stored Process Servers. To incorporate these shared options, the configuration files for the metadata server, OLAP servers, and stored process servers call **sasv9.cfg**. In turn, **sasv9.cfg** calls a configuration file that is located in the SAS installation directory.

By default, the SAS Workspace Server, the SAS/CONNECT Server, and the SAS Data Step Batch Server do not have their own configuration files. Instead, they directly use the options contained in **sasv9.cfg**.

The following table provides the names and locations of configuration files for servers.

Note: For servers that are installed on a machine other than the machine that hosts the SAS Metadata Server, replace **SASMain** with **SASApp** in the directory paths. Δ

Names and Locations of SAS Server Configuration Files

Object Spawner (listener for SAS Workspace Server and SAS Stored Process Server):

sas-config-dir\Lev1\SASMain\ObjectSpawner\OMRConfig.xml

This is the main configuration file for the object spawner. It contains login credentials for the Trusted User and the name of the metadata server machine. The object spawner must be stopped and restarted to pick up changes to this file.

SAS Data Step Batch Servers:

sas-config-dir\Lev1\SASMain\sasv9.cfg

This file provides configuration information to the following file:

sas-config-dir\Lev1\SASMain\sas_internal.bat

The file **sas_internal.bat** is called by the following file:

sas-config-dir\Lev1\SASMain\sas.bat

The file **sas.bat** is called by the following file:

sas-config-dir\Lev1\SASMain\BatchServer\sasbatch.bat

SAS/CONNECT Server:

sas-config-dir\Lev1\SASMain\ConnectServer\OMRConfig.xml

This file contains login credentials for the Trusted User and the name of the metadata server machine. The SAS/CONNECT spawner must be restarted to pick up changes to this file.

sas-config-dir\Lev1\SASMain\sasv9.cfg

This file provides configuration information to the following batch file:

sas-config-dir\Lev1\SASMain\sas_internal.bat

Additional files are called, in the following order, to start the SAS/CONNECT Server:

sas-config-dir\Lev1\SASMain\sas.bat

sas-config-dir\Lev1\SASMain\sasconnect.bat

sas-config-dir\Lev1\SASMain\ConnectServer\Connect Server.bat

sas-config-dir\Lev1\SASMain\ConnectServer\StartConnectServer.bat

SAS OLAP Server

sas-config-dir\Lev1\SASMain\OLAPServer\sasv9_OLAPServer.cfg

This is the main configuration file for the SAS OLAP server. This file calls the following file:

sas-config-dir\Lev1\SASMain\sasv9.cfg

SAS Metadata Server (the metadata server must be restarted if the following files are changed):

sas-config-dir\Lev1\SASMain\omaconfig.xml

sas-config-dir\Lev1\SASMain\MetadataServer\AdminUsers.txt

sas-config-dir\Lev1\SASMain\MetadataServer\trustedUsers.txt

sas-config-dir\Lev1\SASMain\MetadataServer\trustedPeers.xml

sas-config-dir\Lev1\SASMain\MetadataServer\sasv9_MetadataServer.cfg

This file calls the following file:

sas-config-dir\Lev1\SASMain\sasv9.cfg

SAS/SHARE Server:

```
sas-config-dir\Lev1\SASMain\ShareServer\sasv9_ShareServer.cfg
```

This file calls the following file:

```
sas-config-dir\Lev1\SASMain\sasv9.cfg
```

```
sas-config-dir\Lev1\SASMain\ShareServer\libraries.sas
```

This file pre-allocates SAS libraries for the SAS/SHARE Server. The server must be restarted to pick up changes.

SAS Stored Process Server:

```
sas-config-dir\Lev1\SASMain\StoredProcessServer\sasv9_StorProcSrv.cfg
```

This file calls the following file:

```
sas-config-dir\Lev1\SASMain\sasv9.cfg
```

SAS Workspace Server:

```
sas-config-dir\Lev1\SASMain\sasv9.cfg
```

Using adminUsers.txt to Configure New Privileged Users of the SAS Metadata Server

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

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

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

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

```
D1234\adminUser
```

- 3 Save your changes to the file.
- 4 Restart the metadata server.

Using trustedUsers.txt to Configure New Trusted Users

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

- 1 Open the file **trustedUsers.txt** in the directory *SAS-config-dir\Lev1\SASMain\MetadataServer*.
- 2 This file contains an entry for the SAS Trusted User (usually sastrust). Add analogous entries for other users whom you want to make *trusted users*. On Windows systems, you must qualify domain user IDs with a domain name and local user IDs with a hostname. For example, you might use the following entry for a local user:

```
D1234\trustedUser
```

- 3 Save your changes to the file.
- 4 Restart the metadata server.

Reference Information for omaconfig.xml

About the omaconfig.xml File

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

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

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

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

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

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

<OMA> Options

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

ADMINUSERS="*filename*"

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

ALERTEMAIL="*email-address*"

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

ALERTEMAILATTACHMENT="*filename*"

specifies an existing file to attach to the **ALERTEMAIL** message. Specify a relative pathname to identify the file. The path should be relative to the directory from which the metadata server is started. For example, if you want to attach the **omaconfig.xml** file, and the file is located in the same directory as the script that

starts the server, you can simply enter the filename **omaconfig.xml**. If the **omaconfig.xml** file is located in a **config** directory that is parallel to the directory from which the metadata server is started, you can specify **..\config\omaconfig.xml**.

ALERTEMAILTEST

enables you to verify that the **ALERTEMAIL** and **ALERTEMAILATTACHMENT** options are set correctly. **ALERTEMAILTEST** is a *temporary omaconfig.xml* option that causes a test message to be sent at server startup in order to verify that the e-mail address and filename specified in the **ALERTEMAIL** and **ALERTEMAILATTACHMENT** options are valid. An error will cause the e-mail not to be sent and the metadata server to be stopped, so that the administrator can investigate which option has a problem. The metadata server is also stopped when the e-mail is sent correctly so that the administrator can remove the **ALERTEMAILTEST** option from the **omaconfig.xml** file. The SAS Metadata Server will not run with the **ALERTEMAILTEST** option set. Be sure to specify a string value for the **ALERTEMAILTEST** options. Using the option without a value can also prevent the e-mail from being sent.

GROUPSFORMEMBERASSOCDELETE="*comma-separated-list-of-groups*"

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

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

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

MAXACTIVETHREADS="*number-of-threads*"

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

Beginning in SAS 9.1.3 Service Pack 4, the planned SAS installation automatically sets the value of the **MAXACTIVETHREADS** option to the number of processors on the metadata server host. It also sets the values of the related **THREADSMIN** and **THREADSMAX** metadata server invocation options. For information about how the metadata server uses threads, and recommendations for setting the **THREADSMIN** and **THREADSMAX** options, see “Configuring the Number of Threads Used by the Metadata Server” on page 103.

TRUSTEDUSERS="*filename*"

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

WORKUNITJOURNALPATH="*filename*"

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

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

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

WORKUNITJOURNALSIZE="*number-of-bytes*"

specifies a size, in bytes, for the workunit journal file. This space is reserved in the file system and is unavailable to other processes, whether the workunit journal file is full or not.

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

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

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

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

WORKUNITJOURNALYIELDCOUNT="*number-of-observations*"

is designed to prevent large repository updates (for example, requests that add hundreds of objects) from monopolizing the server. It specifies the number of

records that the update process can apply before yielding to allow other internal processing tasks, such as writing to the metadata server log, to occur and then resuming operation. For example, if **WORKUNITJOURNALYIELDCOUNT** were set to 20 and a workunit consisted of 100 records, the update process would apply 20 records, stop to allow internal processes to function, apply 20 more records, and so on, until all 100 records were applied to repositories.

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

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

<RPOSMGR> Options

The <RPOSMGR> options include:

LIBREF="*name*"

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

ENGINE="*libname-engine*"

specifies the LIBNAME engine that is to be used to create the repository manager. The default value is an empty string; it defaults to base SAS. Other supported values are **ORACLE** and **DB2**, representing the SAS/ACCESS to Oracle and SAS/ACCESS to DB2 LIBNAME engines. When changing the default value, note that engine values must be specified in uppercase letters. For information about using a repository manager that exists in an external DBMS, see Chapter 16, "Creating a Metadata Repository in a Database Management System," on page 127.

PATH="*pathname*"

specifies an alternate location for the repository manager. The planned installation creates the repository manager in

SAS-config-dir\Lev1\SASMain\MetadataServer\rposmgr. *Pathname* can be an absolute reference or a relative reference.

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

OPTIONS="*libname-options*"

specifies connection options for the SAS/ACCESS to Oracle or SAS/ACCESS to DB2 LIBNAME engines. The required options are described in "<RPOSMGR> Attributes Used to Connect to an Oracle Database" on page 130 and "<RPOSMGR> Attributes Used to Connect to a DB2 Database" on page 131.

Reserved omaconfig.xml Options

The SAS Metadata Server supports a number of reserved **omaconfig.xml** options that you should not specify in the **omaconfig.xml** file unless you have been instructed

to include them by SAS Technical Support. There are two categories of reserved options: SAS Metadata Server options and debugging options. The SAS Metadata Server options control features that are considered important for the optimized operation of the metadata server, but can be turned off if necessary. Because they are turned on by default, the only reason to include them in the **omaconfig.xml** file is to turn them off.

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

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

Reserved <OMA> Options

BLOCK_ADDS="0 | 1"

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

LOCK_DEPENDENT_REPOS="0 | 1"

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

PHYSICAL_DELETE="0 | 1"

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

WORKUNIT_INDEXES="0 | 1"

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

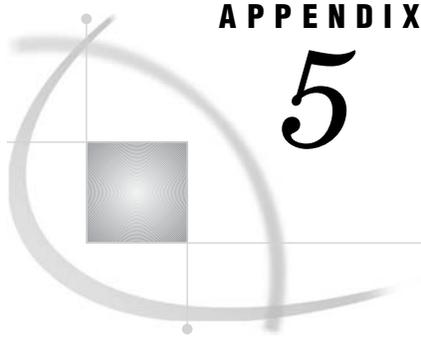
Debugging Options

DEBUG

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

GENERIC="0 | 1"

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



APPENDIX

5

Options for the Metadata Server Invocation Command

<i>Recommended Parameters</i>	259
<i>Recommended Parameters for All Host Environments</i>	259
<i>Additional Windows Parameters</i>	261
<i>Additional UNIX Parameters</i>	261
<i>Additional z/OS Parameters</i>	262
<i>Optional Parameters</i>	262

Recommended Parameters

Recommended Parameters for All Host Environments

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

- the pathname of the non-interactive SAS executable at your site.
- the environment control parameters NOAUTOEXEC, NOTERMINAL, SASUSER, and RSASUSER:
 - NOAUTOEXEC
 - specifies not to process the SAS autoexec file, even if one exists. Processing of the file can result in unpredictable server behavior.
 - NOTERMINAL
 - specifies to run the server in batch mode.
 - SASUSER="*library-specification*"
 - specifies the SAS data library to contain the server's profile catalog. This library must be dedicated to the metadata server and is typically a directory named **sasusrms** or **sasuser** under the **MetadataServer** directory.
 - RSASUSER
 - limits access to the SASUSER library to read-only mode.
- the log output parameters LOG, LOGPARM, PAGESIZE, and LINESIZE:
 - LOG="*file-specification*"
 - enables metadata server logging and specifies a location to write the SAS log. The SAS Metadata Server is a non-interactive SAS session; therefore, this parameter is required to enable logging. If the LOG option is omitted, then SAS log entries display in the SAS Message Log window, which is closed

during a non-interactive SAS session. The default location for the log file is `SAS-config-dir\Lev1\SASMain\MetadataServer\logs`.

`LOGPARAM="WRITE=value ROLLOVER=value OPEN=value"`

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

`PAGESIZE=value`

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

`LINESIZE=value`

specifies the line size of SAS output. A setting of **MAX** is recommended to reduce log entry wrap-around.

- the encryption parameter NETENCRYPTALGORITHM:

`NETENCRYPTALGORITHM="algorithm1" | ("algorithm1", "algorithm2", ...)`

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

- OBJECTSERVER, to invoke an Integrated Object Model (IOM) server.
- OBJECTSERVERPARMS, to pass IOM parameters to the server:

`CLIENTENCRYPTIONLEVEL=NONE | CREDENTIALS | EVERYTHING`

set in conjunction with the NETENCRYPTALGORITHM option, this option specifies the degree of encryption that the server uses when making outbound calls.

NONE Nothing is encrypted. Clients transmit user IDs and passwords as plain text.

CREDENTIALS Clients encrypt user IDs and passwords.

EVERYTHING Clients encrypt all client-server communications.

CEL is an alias for CLIENTENCRYPTIONLEVEL.

`PROTOCOL=value`

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

`PORT=port-number`

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

`CLASSFACTORY=factory-number`

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

TRUSTSASPEER=trustedPeers.xml

enables peer servers used by a SAS metadata client to connect to the metadata server using the credentials of the connecting client. SAS applications that require this support will state the requirement in their documentation. For more information, see “Setting up Trusted Peer Connections for SAS Sessions” in the *SAS Integration Technologies: Server Administrator’s Guide* at

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

THREADSMIN=(number-of-processors * 2) + 1

specifies the minimum number of threads in the server’s thread pool. When the number of threads falls below this number, IOM mechanisms automatically create new threads. TMIN is an alias for THREADSMIN.

Note: It is recommended that TMIN=TMAX. Δ

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

THREADSMAX=(number-of-processors * 2) + 1

specifies the maximum number of threads in the server’s thread pool. When the number of threads exceeds this number, IOM mechanisms automatically delete threads. TMAX is an alias for THREADSMAX.

Note: It is recommended that TMIN=TMAX. Δ

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

Additional Windows Parameters

A Windows server also requires the following parameters:

the system administration parameter **MEMSIZE=value**

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

the display parameter **NOSPLASH**

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

Additional UNIX Parameters

A UNIX server requires the following parameter in addition to those described in “Recommended Parameters for All Host Environments” on page 259:

the system administration parameter **MEMSIZE=value**

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

Additional z/OS Parameters

A z/OS server requires the following parameters in addition to those described in “Recommended Parameters for All Host Environments” on page 259:

- the file control parameter FILESYSTEM=HFS
specifies to run the server in a hierarchical file system.
- the OBJECTSERVERPARMS parameter THREADSSTACKSIZE=1048576
specifies the stack size to give each thread.

Optional Parameters

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

- \square The access control parameter AUTHPROVIDERDOMAIN:

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

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

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

\triangle

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

`EMAILSYS SMTP`

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

`EMAILHOST server-network-address`

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

`EMAILID "server-email-address"`

specifies the e-mail address in the **From** field for ALERTEMAIL messages.

The e-mail address can be entered in the form

"server-name<user-account@domain>" or simply as *"<user-account@domain>"*.

Note: These EMAIL parameters should not be used unless workunit journaling is configured. For more information, see “Using Workunit Journaling to Optimize Performance on Multiprocessor Hosts” on page 104. \triangle

- The OBJECTSERVERPARMS parameters:

APPLEVEL=0 | 1 | 2 | 3

controls IOM server logging. The default value (1) enables logging; therefore, this parameter is optional. APPLEVEL=0 disables IOM server logging.

Setting APPLEVEL=0 is highly discouraged because the server will not have a log. Other APPLEVEL options invoke additional logging functionality. For more information about logging levels and the possible values of this option, see “Administering the SAS Metadata Server Log” on page 69.

IOMLEVEL= 0 | 1

specifies whether input and output XML strings are captured in the SAS log.

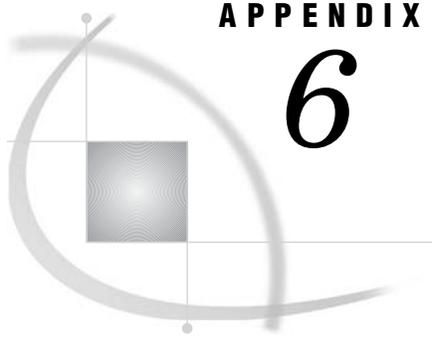
The default value of 0 disables capturing of input and output XML strings. A value of 1 enables capturing of input and output XML strings. For more information, see “Capture XML Information in the SAS Metadata Server Log” on page 71.

JNLSTRMAX=*number*

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

JNLLINEMAX=*number*

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



APPENDIX

6

Configuration Directories

<i>Overview of the SAS Configuration Environment</i>	265
<i>Basic Concepts of the SAS Configuration Environment</i>	265
<i>The Main Directory Structure</i>	267
<i>The Lev1 Directory Structure</i>	268
<i>SASMain Contents</i>	269
<i>The SASMain Subdirectories</i>	270
<i>Web Contents</i>	273
<i>Default Directory Permissions</i>	273

Overview of the SAS Configuration Environment

The SAS deployment process is based on a set of best practices for setting up a planned configuration environment. The environment has these characteristics:

- It provides a context under which all SAS servers can execute.
- It is easily extended to add development, test, and production levels (SAS Data Integration Studio only).
- It enables multiple users to work together and yet keep changes isolated.
- It eases integration and troubleshooting between multiple machines.

Basic Concepts of the SAS Configuration Environment

Best practices for the default configuration were developed around these basic concepts:

Environment identifies an entire set of related information such as levels, SAS Application Servers, scripts, utilities, and documentation. The root of the environment is the configuration directory.

Level identifies the production status of the information contained within a specific area in the environment. Level 1 (Lev1) is the production level.

Note: User-written scripts can take advantage of the consistent format of the level directory names. For example, a script that searches for a particular piece of SAS code could search the lowest level (or the level where the request originates) and then continue to increment through the other levels, moving towards production. △

Change management a general term for administering modifications to the information in the environment, such as moving information from one level to

another (replication and promotion), as well as handling multiple users of a single level.

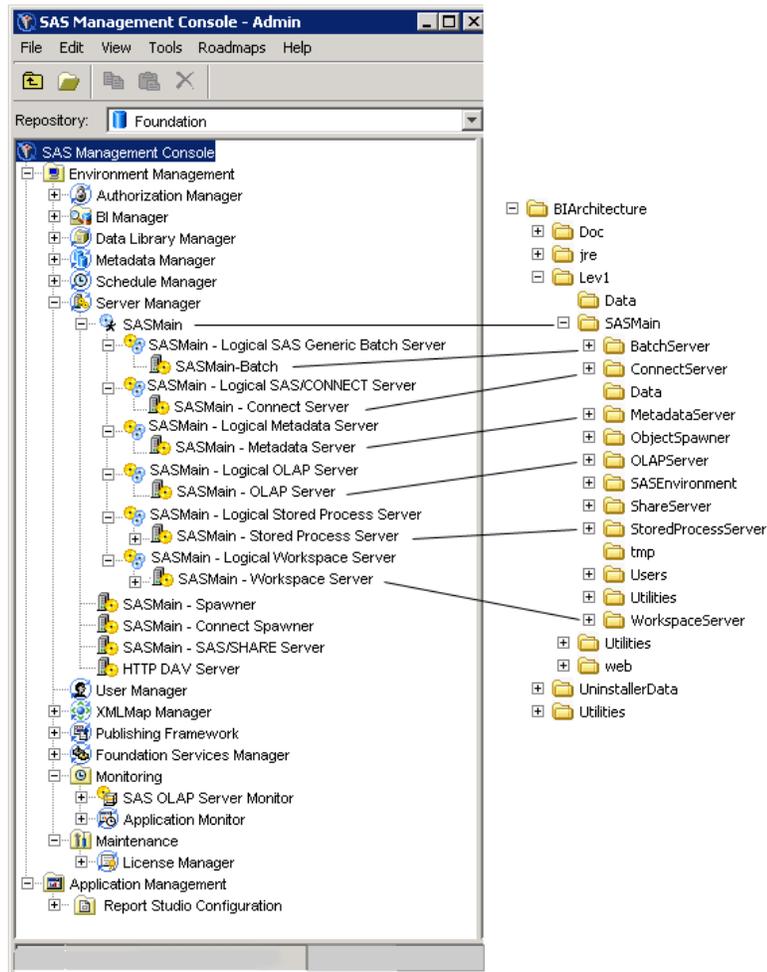
For more information, see the following:

- Chapter 17, “Using the Promotion and Replication Wizards to Copy an Entire Metadata Repository,” on page 135
- “Setting Up Change Management” in the *SAS Intelligence Platform: Desktop Application Administration Guide*

<i>SAS Application Server</i>	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.
<i>Logical servers</i>	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.
<i>Servers</i>	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 A6.1 The Metadata View of the Environment in SAS Management Console and the Corresponding Physical View on Disk



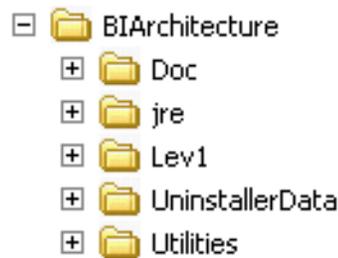
For an explanation of the **SASMain** directory structure, see “SASMain Contents” on page 269.

Note: **SASMain** is the name of the application server on the machine that hosts the SAS Metadata Server. Machines that are not hosting a SAS Metadata Server can have user-defined application server names; the default name is **SASApp**. This appendix only discusses **SASMain**. △

The Main Directory Structure

The first time that you run the SAS Configuration Wizard on a machine, it prompts you for the name of a configuration directory. The configuration directory is the root of the current environment. In this example, the configuration directory is named **BIArchitecture**.

Note: Depending on your operating system and your installed products, your site might also have subdirectories that are not shown in the following display. △



Here are descriptions of the directories that are shown in the illustration:

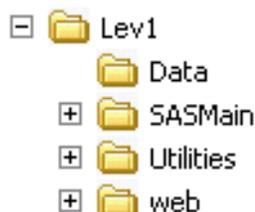
Doc	provides a location for you to save your own documents, which is related to the current environment.
jre	contains the Java Runtime Environment, which interprets Java code.
Lev1	identifies information that is at production status. Beneath a level, you can create any number of directories that are 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.

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 A6.2 Example of a Lev1 Directory Structure on a Windows System



Here are descriptions of the directories that are shown in the illustration:

- 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 269.

Note: On Web-tier only machines, this directory is empty. △

- Utilities** contains utilities used in deploying Web applications. Among them is a sample SAS application named **LoadPlannedStoredProcessSamples.sas** that can be used to load stored process samples into the metadata repository.

- web** for machines with installed Web components, this directory contains these items:

- a servlet container (or J2EE) start-up script named **startServletContainer**
- a **Deployments** subdirectory that contains information related to deploying Web applications, including policy files and service definition files
- a **Utilities** subdirectory for any utilities relevant to installed Web applications.
- a **webapps** subdirectory that contains the Web applications archive (WAR) files for your Web applications such as the SAS Information Delivery Portal.

For more information about the contents of **web**, see “Web Contents” on page 273.

SASMain Contents

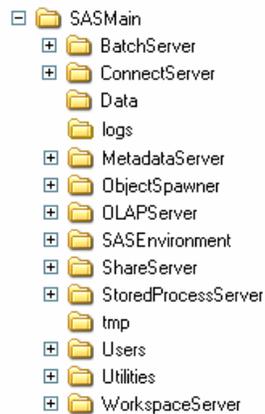
On machines with an installed SAS Metadata Server (and perhaps other SAS servers), this directory represents a SAS Application Server that contains configuration files, a variety of subdirectories, and utilities that support the SAS Application Server. On Web-tier only machines, this directory is empty.

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 A6.3 Example of a SASMain Directory Structure on a Windows System



Here are descriptions of the directories that are shown in the illustration:

- | | |
|-----------------------|--|
| BatchServer | contains log files and SAS code that are associated with the current SAS Application Server running in batch mode. |
| ConnectServer | the physical directory that corresponds to a server definition in a SAS Metadata Repository. This server provides single-user server functionality for Remote Library Services. (Remote Library Services provides 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"> □ one or more start-up scripts that are appropriate for the operating system □ the logs subdirectory that stores any log files generated by this server invocation □ any utility files that are used to manage the server |
| Data | contains data that is specific to the current SAS Application Server but that is shared across all logical servers defined within the SAS Application Server. |
| logs | contains any log files that are specific to the current SAS Application Server. |
| MetadataServer | the physical directory that corresponds to a server definition in a SAS Metadata Repository. This multi-user server enables users to read metadata from or write metadata to one or more SAS metadata repositories. This directory contains these items: <ul style="list-style-type: none"> □ one or more start-up scripts that are appropriate for the operating system. Typically, the server is started as a service on Windows. Before the start command is executed, the script |

queries the metadata server for information, such as a port number, that is required for the launch.

- three security-related files: **adminUsers.txt**, **trustedUsers.txt**, and **trustedPeers.xml** (see “About the Security-Related Configuration Files” on page 13).
- 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, which 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, which is a SAS library that contains SAS catalogs that enable you to tailor features of SAS for your needs. SAS assigns the SASUSER library at invocation.
- the **logs** subdirectory that stores any log files generated by this server invocation.
- the **work** subdirectory that stores any other files generated by this server invocation.
- any other utility files that are used to manage the server.

SASEnvironment contains the elements that make up 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 Other SAS Server Configuration Files” on page 13):

- The **qltyKb** subdirectory contains one or more locales. Each locale contains definitions and other information that are supplied with SAS Data Quality Server software. This software uses these definitions when it performs data analysis and data cleansing.
- The **SASCode** subdirectory is intended to be used as the deployment directory for SAS Data Integration Studio jobs that you deploy for scheduling. This subdirectory contains a **Jobs** directory that stores all the SAS code for each job in the environment, and a **Steps** directory that stores all the SAS code for each step inside a specific job. By sorting the SAS code in this way, developers on a project can share source code. You can set operating-level permissions on these directories.
- The **SASFormats** subdirectory is provided for you to use to store 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 then available to applications such as SAS Data Integration Studio and SAS Information Map Studio, regardless of which SAS invocation technologies deploy the SAS code.
- The **SASMacro** subdirectory is provided for you to use to store user-created 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 then available regardless of which SAS invocation technologies deploy the SAS code.

ShareServer the physical directory that corresponds to a server definition in a SAS Metadata Repository. This multi-user data server enables two or more clients to write to the same SAS file at the same time. This directory contains these items:

- one or more start-up scripts that are appropriate for the operating system. Typically, the server is started as a service on Windows.
- the **logs** subdirectory that stores any log files generated by this server invocation.
- any other utility files that are used to manage the server.

StoredProcess Server the physical directory that corresponds to a server definition in a SAS Metadata Repository. This server executes stored processes. A stored process is a SAS program that is stored on a server and can be executed as required by requesting applications. This directory contains these items:

- the **logs** subdirectory that stores any log files generated by this server invocation
- the **sasuser** subdirectory that stores any files relevant to the sasuser
- the **data** subdirectory that stores any data files generated by this server invocation
- any other utility files that are used to manage the server

tmp contains temporary working files.

- Users** contains a **TemplateUser** subdirectory that should be copied and renamed for each developer who will be working in the current SAS Application Server. Each user-specific directory contains these subdirectories:
- the **Data** subdirectory that contains temporary output as the result of testing changes in the user directory.
 - the **SASEnvironment** subdirectory that contains a structure that matches the **SASEnvironment** subdirectory located under the **SASMain** directory, so that users can test changes to the SAS Application Server.
- Utilities** contains utilities that are used at the SAS Application Server level. For example, this directory might contain scripts to backup the SAS Application Server and import users into the metadata.
- WorkspaceServer** the physical directory that corresponds to a server definition in a SAS Metadata Repository. This server fulfills client requests for specific SAS sessions. This directory contains these items:
- the **logs** subdirectory that stores any log files generated by this server invocation.
 - any other utility files that are used to manage the server.

Web Contents

Depending on your operating system and your installed products, the **web** directory might contain the following subdirectories:

- Deployments** contains information related to deploying Web applications, including policy files, service configuration files, and service deployment files.
- webapps** contains the Web applications archive (WAR) files for your Web applications such as the SAS Information Delivery Portal. These WAR files are actually JAR files that contain all of the files that make up the Web application, including servlets, JavaServer Pages, and HTML documents.
- Note:* If you are using the Tomcat servlet container to execute your Web applications, the SAS Configuration wizard has already copied these WAR files to Tomcat's **webapps** directory. If you are using a J2EE application server for this purpose, you must manually deploy these files to your server's **webapps** directory. Follow the instructions in your vendor's documentation for deploying an application. △
- Utilities** contains any utilities related to deploying Web applications.

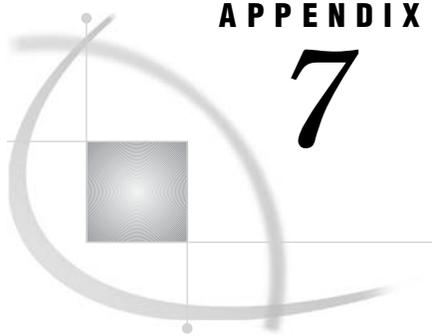
Default Directory Permissions

For UNIX and z/OS systems, here are the default permissions for the directories, files, and scripts that are created in a planned installation.

Note: There are no default permissions set on Windows systems. All directories, files, and scripts are owned by the user who performs the installation. △

Table A6.1 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

7

Managing an Environment

<i>Overview of Managing an Environment</i>	275
<i>Customizing the Properties of a New Environment</i>	275
<i>What Is a Configuration Property?</i>	275
<i>Ways to Modify the Configuration Properties File for a New Environment</i>	276
<i>Using a Command-Line Option to Specify a Custom Properties File</i>	276
<i>Editing the Configuration File from Within the SAS Configuration Wizard</i>	277
<i>Re-Creating an Existing Environment</i>	277
<i>Adding to an Environment</i>	278
<i>Preparing to Use the SAS Configuration Wizard to Modify an Environment</i>	278
<i>Adding a SAS Application Server to an Existing Environment</i>	278
<i>Including the Same Servers</i>	279
<i>Including Different Servers</i>	279
<i>Adding an Application Server Component to an Existing SAS Application Server</i>	279
<i>Uninstalling an Environment</i>	279

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 appendix explains how to perform these tasks:

- customize the properties of a new environment
- add to an existing environment
- re-create an existing environment
- uninstall an environment

Customizing the Properties of a New Environment

What Is a Configuration Property?

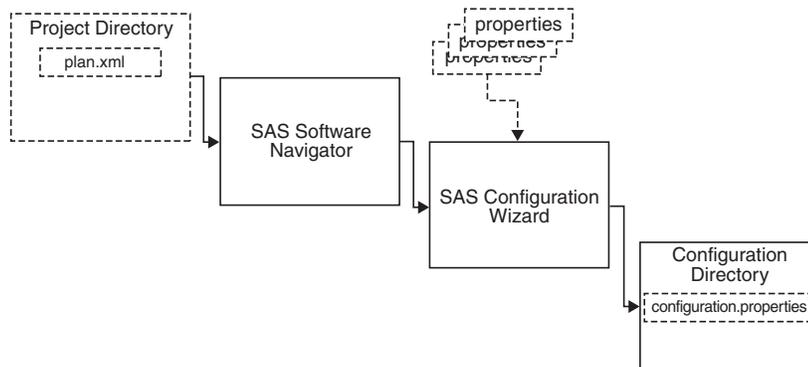
A configuration property is a simple name/value pair. The pairs are specified in a properties file, which contains one property pair (**name=value**) per line. Throughout the configuration process, property values are substituted for property names.

Note: The properties file is one of two data inputs to the SAS Configuration Wizard. The other input is the planning file or project directory. △

The following figure provides an overview of how properties are used during the configuration process. The properties that are shown as input to the SAS Configuration

Wizard represent the default properties files that exist on the media as shipped from SAS. These default properties are distributed as a group of files in order to reduce duplication between generic properties and operating-specific properties.

Figure A7.1 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.

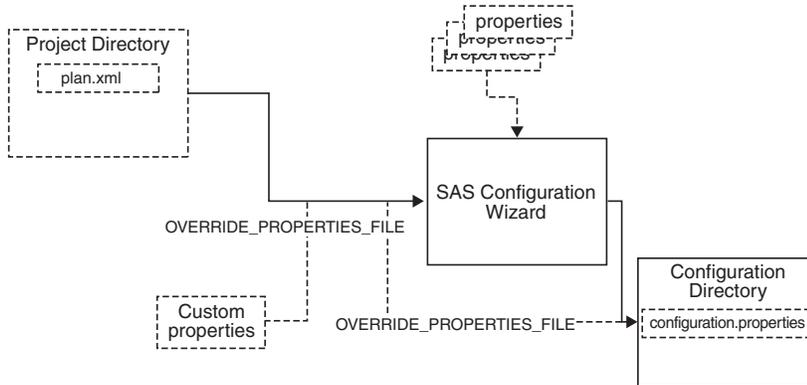
To use this option, you launch the SAS Configuration Wizard by using a command such as the following:

```
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. \triangle

The following figure shows at which point the custom properties are introduced into the configuration process.

Figure A7.2 Using the `OVERWRITE_PROPERTIES_FILE` Command to Specify a Custom Property File

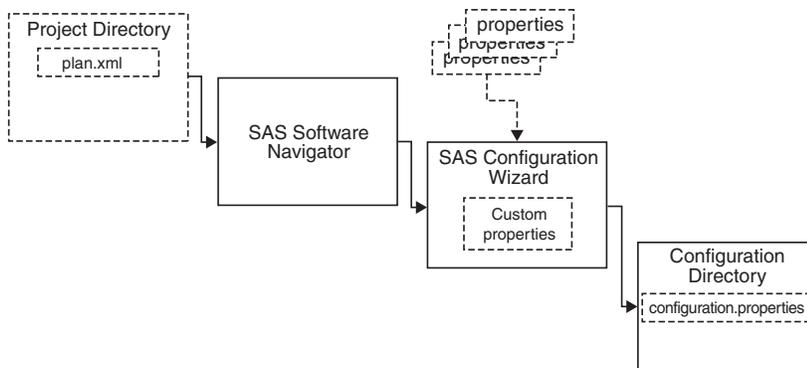


Editing the Configuration File from Within the SAS Configuration Wizard

You also can use the Advanced Properties Editor window in the SAS Configuration Wizard to modify the contents of the `configuration.properties` file. 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 A7.3 Using the Advanced Properties Editor Window from Within the SAS Configuration Wizard to Specify Custom Properties



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 the following:

```
setup -DOVERRIDE_PROPERTIES_FILE='C:\location of copied configuration file
\configuration.properties -i silent
```

Adding to an Environment

Preparing to Use the SAS Configuration Wizard to Modify an Environment

When you rerun the SAS Configuration Wizard in order to modify an environment, you might need to perform these tasks:

- Because the wizard will replace any existing files without notification, you should make copies of the files that you do not want to be overwritten.
- On Windows, the wizard will attempt to re-create services. To prevent this action, edit the **configuration.properties** file by setting the **_SERVICE** properties to **0**. For example, the service property name for the metadata server is **METADATA_SERVICE**.

Adding a SAS Application Server to an Existing Environment

The steps to add a SAS Application Server to an existing environment depend on whether you want to include the same servers that were used in the default SAS Application Server or you want to include different servers.

Including the Same Servers

If you want to use the same servers that exist in the default SAS Application Server, then you complete the SAS Configuration Wizard as you did for the original installation. When the Advanced Properties Editor window appears, click **Edit Properties**. In the **configuration.properties** file that opens, make these changes:

- 1 Change **AppName** to the name of the new SAS Application Server.
- 2 Change **MDAPDIR** to the name of the new SAS Application Server directory. This is the same value as **AppName**.

Including Different Servers

If you want the new SAS Application Server to contain different servers, then use one of these two methods:

- Run the SAS Configuration Wizard as you did for the initial deployment but specify a planning file that contains just the new servers.
- On the Select Install Set wizard window, select **Custom** and then choose to install just the servers that you want the new SAS Application Server to contain.

If you use this method, then remember to also enter the name of the new application server and its directory. When the Advanced Properties Editor window appears, click **Edit Properties**. In the **configuration.properties** file that opens, make these changes:

- 1 Change **AppName** to the name of the new SAS Application Server.
- 2 Change **MDAPDIR** to the name of the new SAS Application Server directory. This is the same value as **AppName**.

Adding an Application Server Component to an Existing SAS Application Server

To add an application server component to an existing SAS Application Server, you complete the SAS Configuration Wizard by entering basically the same information that you entered to create the environment, including using same user IDs and passwords, the same configuration name, and the same paths. The modifications are listed in these steps:

- 1 When prompted, indicate that you are not using a planning file.
- 2 On the Select Install Set wizard window, select **Custom**.
- 3 When prompted, select one or more server components that you want to configure.

Note: Do not select any servers that already exist in the SAS Application Server. If you do, the original server information will be overwritten. △

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, or services 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.

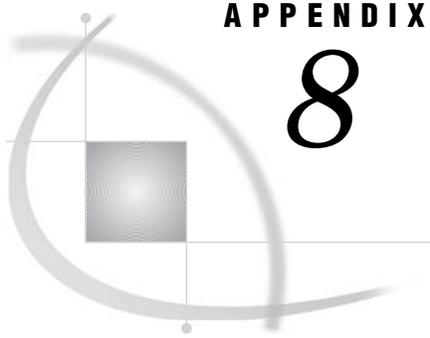
- 3 Review the remaining contents of the **C:\SAS\configuration directory name** directory and decide which files that you want to keep.
- 4 Copy the files that you want to keep to another location.
- 5 Delete the entire directory structure. For example, you can use the following commands:

- Windows:

```
rmdir /S/Q configuration directory location
```

- UNIX:

```
rm -r -f configuration directory location
```



APPENDIX

8

Recommended Reading

Recommended Reading 281

Recommended Reading

Here is the recommended reading list for this title:

- *SAS Intelligence Platform: Overview*
- *SAS Intelligence Platform: Installation Guide*
- *SAS Intelligence Platform: Troubleshooting Your Initial Installation*
- *SAS Intelligence Platform: Security Administration Guide*
- *SAS Intelligence Platform: Data Administration Guide*
- *SAS Intelligence Platform: Application Server Administration Guide*
- *SAS Intelligence Platform: Desktop Application Administration Guide*
- *SAS Intelligence Platform: Web Application Administration Guide*
- *SAS Open Metadata Interface: Reference*
- *SAS Language Reference: Concepts*
- *SAS Language Reference: Dictionary*
- *SAS Management Console: User's Guide*
- *SAS Solutions Services: System Administration Guide*

For a complete list of administration documentation for the SAS Intelligence Platform, see <http://support.sas.com/913administration>.

For a list of SAS documentation, see <http://support.sas.com/documentation/onlinedoc/sas9doc.html>.

For a complete list of SAS publications, see the current *SAS Publishing Catalog*. To order the most current publications or to receive a free copy of the catalog, contact a SAS representative at

SAS Publishing Sales
 SAS Campus Drive
 Cary, NC 27513
 Telephone: (800) 727-3228*
 Fax: (919) 677-8166
 E-mail: sasbook@sas.com
 Web address: support.sas.com/pubs

* For other SAS Institute business, call (919) 677-8000.

Customers outside the United States should contact their local SAS office.

Glossary

access control 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.

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.

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.

archive path

in the Publishing Framework, the complete path of the location to which a user or program can publish an archive package.

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.

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.

autocall macro

a macro whose uncompiled source code and text are stored in an autocall macro library. Unlike a stored compiled macro, an autocall macro is compiled before execution the first time it is called.

base path

the location, relative to a WebDAV server's URL, in which packages are published and files are stored.

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.

catalog

See SAS catalog.

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*.

client application

an application that runs on a client machine.

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.

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*.

DATA step

in a SAS program, a group of statements that begins with a DATA statement and that ends with either a RUN statement, another DATA statement, a PROC statement, the end of the job, or the semicolon that immediately follows lines of data. The DATA step enables you to read raw data or other SAS data sets and to use programming logic to create a SAS data set, to write a report, or to write to an external file.

database library

a collection of one or more database management system files that are recognized by SAS and that are referenced and stored as a unit. Each file is a member of the library.

database management system

See DBMS (database management system).

database server

a server that provides relational database services to a client. Oracle, DB/2 and Teradata are examples of relational databases.

DBMS (database management system)

a software application that enables you to create and manipulate data that is stored in the form of databases. See also relational database management system.

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.

development environment

a computing environment in which application developers use software tools to write, compile, and debug programs. See also testing environment, production environment.

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 of transforming intelligible data (plaintext) into an unintelligible form (ciphertext) by means of a mathematical process.

external file

a file that is created and maintained by a host operating system or by another vendor's software application. SAS can read data from and route output to external files. External files can contain raw data, SAS programming statements, procedure output, or output that was created by the PUT statement. A SAS data set is not an external file. See also fileref (file reference).

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.

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.

foundation services

See SAS Foundation Services.

generated transformation

in SAS Data Integration Studio, a transformation that is created with the Transformation Generator wizard, which helps you specify SAS code for the transformation. See also transformation.

grid computing

a type of computing in which large computing tasks are distributed among multiple computers on a network.

home directory

under OpenVMS and UNIX operating systems, the directory in which a user is placed after logging in. The home directory is also called the login directory.

identity

See metadata identity.

information map

a collection of data items and filters that describes and presents data in a form that is relevant and meaningful to a business user. A user of a query and reporting application such as SAS Web Report Studio can easily build a business report by using the parts of an information map as the building blocks for queries.

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 application

a stand-alone program that is written in the Java programming language.

JavaServer page

See JSP (JavaServer page).

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.

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.

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.

middle tier

in a SAS business intelligence system, the tier in which J2EE Web applications and J2EE enterprise applications execute.

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 multidimensional database (MDDB) tables.

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.

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.

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.

planned installation

a method of installing a SAS business intelligence system. This type of installation requires a planning file that contains information about the different hosts that are included in the system, the software to be installed on each host, and the SAS

servers that should be configured on each server-tier host. The planning file then serves as input to an installation tool called the SAS Software Navigator and to a configuration tool called the SAS Configuration Wizard. See also basic installation.

planning file

an XML file that contains a list of the products to be installed and the components to be configured at a site. This file serves as input to both the SAS Software Navigator and the SAS Configuration Wizard.

pool

a group of server connections that can be shared and reused by multiple client applications. A pool consists of one or more puddles. See also puddle.

pooling

the act or process of creating a pool. See pool.

portlet

a Web component that is managed by a Web application and which is aggregated with other portlets to form a page within the application. A portlet processes requests from the user and generates dynamic content.

production environment

a computing environment in which previously tested and validated software is used (typically on a daily basis) by its intended consumers. See also development environment, testing environment.

project repository

a repository that must be dependent on a foundation repository or custom repository that will be managed by the Change Management Facility. A project repository is used to isolate changes from a foundation repository or from a custom repository. The project repository enables metadata programmers to check out metadata from a foundation repository or custom repository so that the metadata can be modified and tested in a separate area. Project repositories provide a development/testing environment for customers who want to implement a formal change management scheme. See also custom repository, foundation repository.

promotion

See metadata promotion.

Public Kiosk

a public page that is displayed when a user starts the SAS Information Delivery Portal but has not yet logged on.

publication channel

an information repository that has been established using the SAS Publishing Framework and which can be used to publish information to users and applications. See also publish.

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.

replication

See metadata replication.

resource template

an XML file that specifies the information that is needed for creating a metadata definition for a SAS resource.

SAS Application Server

in the SAS Intelligence Platform, a logical entity that represents the SAS server tier. This logical entity contains specific servers (for example, a SAS Workspace Server and a SAS Stored Process Server) that execute SAS code. A SAS Application Server has relationships with other metadata objects. For example, a SAS library can be assigned to an application server. When a client application needs to access that library, the client submits code to the application server to which the library is assigned.

SAS catalog

a SAS file that stores many different kinds of information in smaller units called catalog entries. A single SAS catalog can contain several different types of catalog entries. See also SAS catalog entry.

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 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 Open Metadata Interface

a set of methods that enable users to read metadata from or write metadata to the SAS Metadata Server.

SAS Stored Process

a SAS program that is stored in a central location and which can be executed from the SAS Information Delivery portal at the user's request. When a stored process is executed, it creates a report that includes the most current data that is available. Stored processes can display input forms that enable users to customize the contents of reports.

SAS Stored Process Server

a SAS IOM server that is launched in order to fulfill client requests for SAS Stored Processes. See also IOM server.

SAS system option

an option that affects the processing of an entire SAS program or interactive SAS session from the time the option is specified until it is changed. Examples of items that are controlled by SAS system options include the appearance of SAS output, the handling of some files that are used by SAS, the use of system variables, the processing of observations in SAS data sets, features of SAS initialization, and the way SAS interacts with your host operating environment.

SAS table

another term for SAS data set. See SAS data set.

SAS Workspace Server

a SAS IOM server that is launched in order to fulfill client requests for IOM workspaces. See also IOM server, workspace.

SAS/ACCESS software

a group of software interfaces, each of which makes data from a particular external database management system (DBMS) directly available to SAS, as well as making SAS data directly available to the DBMS.

SAS/CONNECT server

a server that provides SAS/CONNECT services to a client. When SAS Data Integration Studio generates code for a job, it uses SAS/CONNECT software to submit code to remote computers. SAS Data Integration Studio can also use SAS/CONNECT software for interactive access to remote libraries.

SAS/CONNECT spawner

a program that runs on a remote computer and that listens for SAS/CONNECT client requests for connection to the remote computer. When the spawner program receives a request, it invokes a SAS session on the remote computer.

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.

SASROOT

a term that represents the name of the directory or folder in which SAS is installed at your site or on your computer.

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.

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.

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 Server

a SAS Scalable Performance Data Server. An SPD Server restructures data in order to enable multiple threads, running in parallel, to read and write massive amounts of data efficiently.

stored process

See SAS Stored Process.

target metadata server

the metadata server to which the metadata is promoted or replicated. See also metadata promotion, metadata replication, source metadata server.

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.

transformation

in SAS Data Integration Studio, a metadata object that specifies how to extract data, transform data, or load data into data stores. Each transformation that you specify in a process flow diagram generates or retrieves SAS code. You can specify user-written code in the metadata for any transformation in a process flow diagram.

trusted user

a special user of a metadata server who can acquire credentials on behalf of other users in a multi-tier server environment.

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.

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 Distributed Authoring and Versioning

See WebDAV (Web Distributed Authoring and Versioning).

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 WebFile Server, and Microsoft Corporation's Internet Information Server (IIS).

workspace

in the IOM object hierarchy for a SAS workspace server, an object that represents a single session in SAS. See also IOM (Integrated Object Model).

Index

- A**
- access controls
 - promotion of 174
 - access states
 - metadata repository 37
 - metadata server 40
 - repository manager 39
 - accessibility features 4
 - active sessions, viewing 100
 - Add Repository wizard 119
 - administration environment 140
 - setup for promotion jobs 147
 - administrative tasks, ongoing 25
 - Administrative User account 239
 - administrative users
 - promotion jobs 143
 - rights granted to 17
 - ADMINUSERS= option 254
 - adminUsers.txt file 13
 - configuring new privileged users 253
 - AIX scheduling issues 203
 - ALERTEMAIL= option 254
 - ALERTEMAILATTACHMENT= option 254
 - ALERTEMAILTEST= option 255
 - Apache Tomcat
 - starting servlet container 35
 - application log locations 248
 - Application Response Measurement (ARM) 77
 - application server
 - See SAS Application Server
 - ARM (Application Response Measurement) 77, 98
 - ARM logging 79
 - ARM_OMA subsystem 77
 - enabling other ARM subsystems 78
 - log examples 82
 - post-processing the log 88
 - records written by 79
 - starting 78
 - audit trails 49, 74
 - auditing metadata repositories 74
 - authentication events
 - logging 70
 - Authorization Manager plug-in for SAS Management Console 21
 - in configuration directory 13
 - locations for SAS servers 251
 - configuration properties 275
 - configuration properties file 276
 - content mapping
 - modifying for reports 161
 - copying metadata repositories 120
 - See also promotion jobs
 - See also Promotion Wizard
 - Promotion and Replication Wizards 121
 - tools for 120
 - cubes
 - modifying after promotion jobs 162
- B**
- backups for metadata server
 - alternating to different destinations 54
 - benefits of full backups 45
 - best practices for using %OMABAKUP 46
 - error logging 53
 - process for 57
 - warning messages 58
 - batch server 194
 - BEA WebLogic Server
 - See WebLogic Server
 - BI content
 - moving after promotion jobs 165
 - moving portal content 167
 - moving Xyθος WFS content 165
 - BI Manager plug-in for SAS Management Console 21
 - BLOCK_ADDDS= option 258
- C**
- change management 265
 - channels, report-scheduling tasks 226
 - clients cannot connect to metadata server 235
 - COMPARE procedure
 - restoring metadata server and 50
 - COMPRESS attribute
 - metadata repositories 107
 - configuration
 - default logging configuration 70, 97
 - logging of XML 72
 - metadata objects used in 18
 - metadata server 6, 7
 - modifying to optimize performance 23
 - privileged users 17
 - stored process servers 8
 - users and groups used in 14
 - workspace servers 8
 - workunit journaling 105
 - configuration directories 9
 - configuration files in 13
 - directory structure 267
 - server-tier machines 9
 - configuration environment
 - See environments
 - configuration files 12
 - backing up 47
 - for workspace server logging 93
- D**
- Data Library Manager plug-in for SAS Management Console 21
 - data sets
 - repairing 107
 - database management system 127
 - DB2
 - creating repository in 127, 128
 - creating repository manager in 128
 - RPOSMGR attributes for connecting 131
 - DBMS repository 127
 - creating in DB2 127, 128
 - creating in Oracle 127, 128
 - creating repository manager in Oracle or DB2 128
 - prerequisites for setting up 127
 - RPOSMGR attributes for DB2 131
 - RPOSMGR attributes for Oracle 130
 - DEBUG option 258
 - debugging options
 - omaconfig.xml file 258
 - default ports 243
 - demotion 167
 - dependent repositories, promoting 139
 - deployment directory 206
 - DESTINATIONPATH= option
 - %OMABAKUP macro 60
 - %OMARUNAN macro 115
 - directory permission defaults 273
 - directory structure 267
 - default directory permissions 273
 - Lev1 directory 11, 268

E

ENGINE= option 257
 environments 265
 basic concepts 265
 customizing properties of 275
 default directory permissions 273
 directory structure 267
 Lev1 directory structure 11, 268
 managing 275
 modifying configuration properties file 276
 promotion jobs 140
 re-creating 277
 uninstalling 279
 error logging 69
 event logging 69
 Export and Import Wizards (BI Manager) 121
 Export Wizard 169
 accessing 174
 BI Manager 121
 Metadata Manager 121
 preparing to use 176
 SAS Data Integration Studio 122
 selective promotion 171
 troubleshooting 188
 exporting metadata repositories 120

F

flow metadata for scheduling 196
 foundation services
 modifying after promotion jobs 162

G

GENERIC= option 258
 groups used in system configuration 14, 16
 GROUPSFORMEMBERASSOCDELETE= option 255

H

host environment
 porting metadata repository to different host 123

I

IBM WebSphere Application Server
 See WebSphere Application Server
 Import Wizard 169
 accessing 174
 Metadata Manager 121
 preparing to use 176
 SAS Data Integration Studio 122
 selective promotion 171
 troubleshooting 188
 importing metadata repositories 120
 invocation parameters for metadata server 259

J

Java Batch Server 222
 for SAS Web Report Studio 2.1 and 3.1 reports 228

L

Lev1 directory 11, 268
 SASMain contents 269
 web contents 273
 level, defined 265
 Library Manager plug-in for SAS Management Console 21
 LIBREF= option 257
 LOCK_DEPENDENT_REPOS= option 258
 log file, SAS OLAP Server 97
 logging
 administering for stored process servers 93
 enabling for object spawner 89
 enabling for SAS/CONNECT spawner 94
 enabling for workspace servers 92
 for ARM 79, 82
 SAS OLAP Server 97
 logging events and errors 69
 authentication events 70
 capturing XML in log 71
 default logging configuration 70, 97
 logical servers 266
 logs
 default locations for application logs 248
 default locations for server logs 247
 post-processing ARM_OMA log 88
 LSF Administrator account 239
 configuration and 14
 setting up for Platform Product Suite 199
 LSF User account 239

M

MAXACTIVETHREADS= option 103, 255
 changing settings of 104
 memory
 estimating for metadata queries 111
 estimating metadata storage requirements 110
 managing for performance 109
 optimizing usage 113
 recovering quickly in metadata server 115
 metadata
 defining Scheduling Admins group in 221
 overwriting 136
 metadata access files
 creating for promotion jobs 143
 metadata identities 18
 Metadata Manager
 Import and Export Wizards 121
 Metadata Manager plug-in for SAS Management Console 22
 metadata objects
 used in system configuration 18
 XML representation of 71
 METADATA procedure
 for checking repository access state 38

 for checking repository manager access state 40
 metadata repositories 6, 127
 access states 37
 auditing 74
 copying 120
 creating 119
 deleting 125
 importing and exporting 120
 moving to new location on same server 122
 performance issues 107
 porting to different host 123
 promoting 120, 136
 promoting dependent repositories 139
 replicating 136
 metadata server 6
 access states 40
 ARM_OMA subsystem 77
 backing up 46
 client cannot connect to 235
 configuration files 251
 configuring privileged users of 253
 defining source metadata server 148
 defining target metadata server 150
 description of 6
 how it works 7
 initial configuration 6, 7
 invocation parameters 259
 logging events and errors 69
 metadata storage requirements 110
 optimizing performance 103
 optional parameters 262
 options for invocation command 259
 recommended parameters 259
 recovering memory quickly 115
 restoring manually 63
 restoring with %OMABAKUP 48
 specifying changes to standard features 254
 starting 33, 233
 stopping 33
 threads 103
 troubleshooting 233
 UNIX parameters 261
 will not start 233
 Windows parameters 261
 workunit journaling 104
 z/OS parameters 261
 metadata type conversions 122
 METAOPERATE procedure 40
 deleting metadata repositories 125
 for checking metadata server access state 40
 monitoring
 log file 97
 SAS OLAP Server 97
 using ARM for SAS OLAP Server 98
 using the SAS OLAP Server Monitor 99

O

object spawner
 checking status 41
 configuration files 251
 defining source object spawner 149
 enabling logging 89
 OBJECTSERVERPARMS options 103

- OMA options, omaconfig.xml file 254
 - reserved options 258
 - %OMABAKUP macro 45
 - DESTINATIONPATH= option 60
 - executing on Windows or UNIX 52
 - reference 59
 - REORG= option 60
 - REPOSITORYLIST= option 61
 - RESTORE= option 61
 - RPOSMGRPATH= option 60
 - RUNANALYSIS= option 61
 - server connection statement 52, 56, 59
 - SERVERSTARTPATH= option 60
 - syntax 60
 - warning messages 58
 - omaconfig.xml file 12
 - DB2 131
 - debugging options 258
 - OMA options 254
 - Oracle 130
 - reference 254
 - reserved options 257
 - RPOSMGR options 257
 - %OMAPORT macro
 - details 125
 - porting metadata repository to different
 - host 123
 - reference 124
 - syntax 124
 - %OMARUNAN macro
 - DESTINATIONPATH option 115
 - details 115
 - reference 115
 - syntax 115
 - using 114
 - operating system accounts 148
 - operating system scheduling servers 211
 - cancelling a scheduled flow (UNIX) 216
 - cancelling a scheduled flow (Windows) 215
 - cancelling a scheduled flow (z/OS) 217
 - defining 212
 - manually submitting a flow (UNIX) 216
 - manually submitting a flow (Windows) 214
 - manually submitting a flow (z/OS) 217
 - using on UNIX systems 215
 - using on Windows systems 213
 - using on z/OS systems 216
 - optimizing performance
 - changes to initial configuration 23
 - managing memory 109
 - of metadata server 103
 - using the %OMARUN macro 113
 - OPTIONS= option 257
 - Oracle
 - creating repository in 127, 128
 - creating repository manager in 128
 - RPOSMGR attributes for connecting 130
 - OS Services Scheduling Server 194
 - Output Generation Tool 224
 - overwriting metadata 136
 - recommended for all hosts 259
 - UNIX 261
 - Windows 261
 - z/OS 261
 - PATH= option 257
 - performance
 - metadata repositories 107
 - SAS OLAP Server 98
 - performance monitoring 98
 - permissions
 - default directory permissions 273
 - PHYSICAL_DELETE= option 258
 - Platform Calendar Editor 197
 - Platform Flow Manager 196
 - Platform Grid Management Services 197
 - Platform LSF (Load Sharing Facility) 197
 - permissions for files 202
 - Platform Product Suite for SAS 196
 - AIX issues 203
 - error messages 201
 - installing and configuring 199
 - SAS exit behavior 203
 - Windows issues 202
 - Pool Administrator account 239
 - portal content
 - moving after promotion jobs 167
 - ports
 - default ports 243
 - numbering in multiple-level environment 246
 - post-configuration
 - scheduling reports 220
 - PostgreSQL User account 239
 - privileged users
 - configured by Configuration Wizard 17
 - configuring for metadata server 253
 - PROC METADATA
 - See METADATA procedure
 - PROC METAOPERATE
 - See METAOPERATE procedure
 - Process Manager Client 197
 - Process Manager Server 196
 - defining 201
 - process owners 15
 - promoting individual BI objects 169
 - promoting metadata repositories 120
 - See also promotion jobs
 - See also Promotion Wizard
 - See also Replication Wizard
 - Promotion and Replication Wizards 121
 - tools for 120
 - promotion jobs 140
 - administration environment setup 147
 - building 152
 - creating and running 152
 - environments 140
 - infrastructure for 141
 - modifying after running 159
 - moving associated BI content 165
 - running 158, 163
 - site preparation for 140
 - source environment setup 142
 - target environment setup 144
 - promotion process 139
 - Promotion Wizard 121, 136
 - features and capabilities 137
 - overwriting metadata 136
 - promoting dependent repositories 139
 - special considerations 137
 - when to use 136
 - properties
 - configuration properties 275
 - Publishing Framework plug-in for SAS Management Console 22
 - Puddle Log-in account 239
- ## Q
- queries
 - viewing 100
- ## R
- recovering repository records 74
 - refreshing
 - server connections 100
 - server list 100
 - REORG= option
 - %OMABAKUP macro 60
 - repairing data sets 107
 - replication administrator 148, 151
 - replication process 139
 - Replication Wizard 121, 136
 - See also promotion jobs
 - See also Promotion Wizard
 - when to use 136
 - report scheduling 219
 - channel-specific tasks 226
 - Java Batch Server 222, 228
 - Output Generation Tool 224
 - SAS Web Report Studio configuration 223
 - Scheduling Admins group 221
 - scheduling server configuration 224
 - tasks for enabling 220
 - testing the configuration 227
 - troubleshooting the configuration 227
 - updating logins for users 200, 221
 - updating the LSF password file 224
 - user IDs for 220
 - reports
 - modifying content mapping for 161
 - repository auditing 74
 - repository manager 6
 - access states 39
 - creating in Oracle or DB2 128
 - restoring 49
 - specifying changes to standard features 254
 - REPOSITORYLIST= option
 - %OMABAKUP macro 61
 - reserved options
 - debugging options 258
 - OMA options 258
 - omaconfig.xml file 257
 - RESTORE= option
 - %OMABAKUP macro 61
 - restoring metadata server
 - audit trails and 49
 - COMPARE procedure and 50
 - manually restoring a running server 64
 - manually restoring an unresponsive server 63
 - planning a restore 48

- REUSE attribute
 - metadata repositories 107
 - RPOSMGR attributes
 - connecting to DB2 131
 - connecting to Oracle 130
 - RPOSMGR options, omaconfig.xml file 257
 - RPOSMGRPATH= option
 - %OMABAKUP macro 60
 - RUNANALYSIS= option
 - %OMABAKUP macro 61
- S**
- SAS Administrator
 - as unrestricted user 17
 - SAS Administrator account 239
 - configuration and 14
 - SAS Application Server 8, 266
 - components 8
 - initial configuration 8
 - SAS Configuration Wizard
 - privileged users configured by 17
 - workspace and stored process servers 8
 - SAS/CONNECT
 - for promotion jobs 145
 - SAS/CONNECT Server
 - configuration files 251
 - defining target SAS/CONNECT Server 150
 - modifying after promotion jobs 159
 - SAS/CONNECT spawner
 - enabling increased logging 94
 - for promotion jobs 145
 - SAS Data Integration Studio
 - deploying jobs for scheduling 208
 - Export Wizard 169
 - Import and Export Wizards 122
 - Import Wizard 169
 - passwords and credentials 209
 - redeploying jobs after a promotion 167
 - scheduling 205
 - scheduling requirements 206
 - troubleshooting scheduling 209
 - SAS Data Step Batch Server 195
 - configuration files 251
 - defining 207
 - SAS Demo User account 239
 - configuration and 14
 - SAS exit behavior 203
 - SAS General Server User account 239
 - configuration and 14
 - SAS Guest account 239
 - SAS Guest User account
 - configuration and 14
 - SAS Information Delivery Portal
 - moving content after promotion jobs 167
 - SAS initialization failure 234
 - SAS Installer account 239
 - configuration and 14
 - SAS Management Console 21
 - creating metadata repositories 119
 - Export Wizard 169
 - Import Wizard 169
 - plug-ins for 21
 - Promotion Wizard 152
 - SAS OLAP Server Monitor 99
 - Schedule Manager 193
 - starting, stopping and pausing the metadata server 33
 - stopping the SAS OLAP Server 34
 - SAS Metadata Repository
 - See metadata repositories
 - SAS Metadata Server 6
 - See metadata server
 - SAS Object Spawner
 - See object spawner
 - SAS OLAP Server
 - checking status 42
 - configuration files 251
 - logging 97
 - monitoring performance with ARM 98
 - requirements for stopping 30
 - stopping 34
 - SAS OLAP Server Monitor 99
 - refreshing server connections 100
 - refreshing server list 100
 - viewing sessions and queries 100
 - SAS Open Metadata Architecture 6
 - SAS Replication Administrator account 239
 - SAS servers 15, 266
 - configuration file locations 251
 - default log locations 247
 - start-up parameters 30
 - start-up requirements 29
 - starting and stopping on UNIX 32
 - starting and stopping on Windows 31
 - starting and stopping on z/OS 33
 - starting in correct order 30
 - users and 15
 - SAS/SHARE Server
 - configuration files 251
 - SAS Stored Process Server
 - See stored process servers
 - SAS Stored Process User account 239
 - SAS Trusted User account 239
 - as trusted user 17
 - configuration and 14
 - SAS Web Administrator account 239
 - configuration and 14
 - SAS Web Report Studio
 - configuring for report scheduling 223
 - defining Java Batch Server for 2.1 and 3.1 reports 228
 - Output Generation Tool 224
 - SAS Workspace Server
 - See workspace server
 - SASMain directory 269
 - subdirectories 270
 - Schedule Manager plug-in for SAS Management Console 22, 193
 - Console 22, 193
 - scheduling 193
 - See also report scheduling
 - AIX and 203
 - architecture for 193
 - batch server definitions 195
 - deploying SAS Data Integration Studio jobs 208
 - deployment directory 206
 - deployment directory definitions 195
 - job and flow metadata for 196
 - metadata for 194
 - requirements for SAS Data Integration Studio jobs 206
 - SAS Data Integration Studio and 205
 - SAS Data Step Batch Server 207
 - SAS exit behavior 203
 - scheduling server definitions 194
 - servers for 194, 206
 - troubleshooting for SAS Data Integration Studio jobs 209
 - using operating system scheduling servers 211
 - Windows and 202
 - Scheduling Admins group
 - defining in metadata 221
 - scheduling server 194
 - configuring for report scheduling 224
 - security
 - server-tier files 13
 - server connection statement
 - %OMABAKUP macro 52, 56, 59
 - server connections
 - refreshing 100
 - server definitions
 - modifying after promotion jobs 159
 - server list
 - refreshing 100
 - Server Manager plug-in for SAS Management Console 22
 - server-tier machines
 - configuration directory 9
 - security-related files 13
 - servers 15, 266
 - checking status 37
 - for scheduling 194, 206
 - logical servers 266
 - operating system scheduling servers 211
 - pausing 29
 - starting and stopping 29
 - starting in correct order 30
 - SERVERSTARTPATH= option
 - %OMABAKUP macro 60
 - Servlet containers, starting 35
 - Solutions Installer account 239
 - Solutions Role Administrator account 239
 - source environment 140
 - setup for promotion jobs 142
 - start order for servers 30
 - starting
 - ARM_OMA subsystem 78
 - metadata server 33, 233
 - SAS servers 29
 - Servlet containers 35
 - WebLogic Server 35
 - WebSphere Application Server 35
 - state of the system 5
 - stopping
 - metadata server 33
 - SAS servers 29
 - stored process servers 41
 - administering logging 93
 - checking status 41
 - configuration 8
 - configuration files 251
 - system 5
 - administration tasks 25
- T**
- target administrator 146

- target environment 140
 - setup for promotion jobs 144
- threads
 - changing default number of 104
 - controlling number of 103
 - options set by default 103
- THREADSMAX option 103
 - changing settings of 104
- THREADSMIN option 103
 - changing settings of 104
- troubleshooting
 - Export and Import Wizards 188
 - metadata server 233
 - report schedule configuration 227
 - scheduling for SAS Data Integration Studio jobs 209
- trusted users
 - configured by SAS Configuration Wizard 17
 - rights granted to 17
- trustedPeers.txt file 14
- TRUSTEDUSERS= option 256
- trustedUsers.txt file 13
 - configuring new trusted users 253

U

- uninstalling environment 279
- UNIX
 - executing %OMABAKUP macro 52
 - parameters for metadata server 261
 - starting and stopping SAS servers 32

- unrestricted users
 - configured by SAS Configuration Wizard 17
 - rights granted to 17
- user accounts 239
- user IDs
 - for report scheduling 220
- User Manager plug-in for SAS Management Console 22
- users
 - modifying after promotion jobs 161
 - SAS servers and 15
 - special platform users 239
 - used in system configuration 14

W

- warning messages
 - generated by %OMABAKUP macro 58
- web directory 273
- WebLogic Server
 - starting 35
- WebSphere Application Server
 - starting 35
- Windows
 - executing %OMABAKUP macro 52
 - parameters for metadata server 261
 - scheduling issues 202
 - starting and stopping SAS servers 31
- workspace server
 - checking status 41

- configuration 8
- configuration files 251
- defining source workspace server 149
- enabling logging 92
- workunit journaling 104
 - configuring 105
 - turning on 104
- WORKUNIT_INDEXES= option 258
- WORKUNITJOURNALPATH= option 256
- WORKUNITJOURNALSIZE= option 256
- WORKUNITJOURNALYIELDCOUNT= option 256

X

- XML
 - capturing in log 71
 - configuring logging of 72
 - representation of metadata objects 71
- XMLMap Manager plug-in for SAS Management Console 22
- Xythos WebFile Server
 - moving content after promotion jobs 165

Z

- z/OS
 - parameters for metadata server 261
 - starting and stopping SAS servers 33

Your Turn

We want your feedback.

- If you have comments about this book, please send them to **yourturn@sas.com**. Include the full title and page numbers (if applicable).
- If you have comments about the software, please send them to **suggest@sas.com**.

SAS® Publishing gives you the tools to flourish in any environment with SAS!

Whether you are new to the workforce or an experienced professional, you need to distinguish yourself in this rapidly changing and competitive job market. SAS® Publishing provides you with a wide range of resources to help you set yourself apart.

SAS® Press Series

Need to learn the basics? Struggling with a programming problem? You'll find the expert answers that you need in example-rich books from the SAS Press Series. Written by experienced SAS professionals from around the world, these books deliver real-world insights on a broad range of topics for all skill levels.

support.sas.com/saspress

SAS® Documentation

To successfully implement applications using SAS software, companies in every industry and on every continent all turn to the one source for accurate, timely, and reliable information—SAS documentation. We currently produce the following types of reference documentation: online help that is built into the software, tutorials that are integrated into the product, reference documentation delivered in HTML and PDF—free on the Web, and hard-copy books.

support.sas.com/publishing

SAS® Learning Edition 4.1

Get a workplace advantage, perform analytics in less time, and prepare for the SAS Base Programming exam and SAS Advanced Programming exam with SAS® Learning Edition 4.1. This inexpensive, intuitive personal learning version of SAS includes Base SAS® 9.1.3, SAS/STAT®, SAS/GRAPH®, SAS/QC®, SAS/ETS®, and SAS® Enterprise Guide® 4.1. Whether you are a professor, student, or business professional, this is a great way to learn SAS.

support.sas.com/LE



**THE
POWER
TO KNOW®**

