



THE  
POWER  
TO KNOW.

# **SAS<sup>®</sup> 9.2**

## **Intelligence Platform**

### **System Administration Guide**

#### **Second Edition**



The correct bibliographic citation for this manual is as follows: SAS Institute Inc. 2010. *SAS® 9.2 Intelligence Platform: System Administration Guide, Second Edition*. Cary, NC: SAS Institute Inc.

**SAS® 9.2 Intelligence Platform: System Administration Guide, Second Edition**

Copyright © 2010, SAS Institute Inc., Cary, NC, USA

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 electronic book, November 2010

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/publishing](http://support.sas.com/publishing) 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>	<b>ix</b>
Overview	<b>ix</b>
New Logging Facility for SAS Servers	<b>ix</b>
New Server Management and Monitoring Features	<b>x</b>
New SAS Server Components	<b>x</b>
Enhancements and Changes Pertaining to Metadata Server Administration	<b>xi</b>
Enhancements and Changes Pertaining to Metadata Administration	<b>xi</b>
New Configuration Management Tool	<b>xii</b>
New Documentation for Administrators	<b>xiii</b>

## **PART 1    Getting Started    1**

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

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

### **Chapter 2   △   Understanding the State of Your System    5**

Overview of the State of Your System	<b>6</b>
Overview of SAS Management Console	<b>7</b>
Overview of SAS Deployment Manager and SAS Web Administration Console	<b>10</b>
Overview of the SAS Metadata Server and Its Initial Configuration	<b>11</b>
Overview of SAS Application Server Components and Their Initial Configuration	<b>15</b>
Overview of Other SAS Servers and Spawners	<b>17</b>
Overview of Ports	<b>19</b>
Overview of the Configuration Directory Structure	<b>19</b>
Overview of Configuration Files	<b>22</b>
Overview of Initial Roles, Groups, and Users	<b>24</b>

### **Chapter 3   △   Who Can Do What: Credential Requirements for SAS Management Console Tasks    33**

About Security for System Administration Tasks	<b>33</b>
Who Can Do What in the Metadata Manager Plug-in	<b>34</b>
Who Can Do What in the Server Manager Plug-in	<b>35</b>
About Adding Users to the SAS Administrators Group	<b>36</b>
Requirements for Accessing a Standard Workspace Server	<b>37</b>
Clearing the Credentials Cache	<b>37</b>

### **Chapter 4   △   What to Do Next: Administration Tasks    39**

Overview of Administration Tasks	<b>39</b>
First-Priority Setup Tasks	<b>40</b>

Standard Setup Tasks	43
Optional Setup Tasks	44
Ongoing System Administration Tasks	46
Best Practices for Ensuring the Integrity of Your System	47

## **PART 2    Starting, Stopping, and Checking the Status of Servers    49**

### **Chapter 5   △   Operating Your Servers    51**

Overview of Server Operation	52
Running Servers as Windows Services	55
Starting and Stopping SAS Servers on z/OS	57
Using the sas.servers Script on UNIX or z/OS to Start or Stop All Servers	58
Running Servers as UNIX Daemons	61
Using Scripts to Operate SAS Servers Individually	61
Using SAS Management Console to Operate SAS Servers	65
Starting and Stopping Web Application Servers	69

### **Chapter 6   △   Checking the Status of Servers    75**

About Checking the Status of Servers	75
Use a Server Script to Display the Status of a Server or Spawner	76
Checking the Access State of the SAS Metadata Server	76
Checking the Access State of Metadata Repositories and the Repository Manager	77
Check the Status of the Object Spawner	82
Validate the SAS Metadata Server, SAS Workspace Servers, SAS Pooled Workspace Servers, SAS Stored Process Servers, and SAS OLAP Servers	82
Validate the SAS Content Server	83

## **PART 3    System Monitoring and Logging    85**

### **Chapter 7   △   Monitoring the Activity of SAS Servers    87**

About SAS Server Monitoring	87
Using SAS Management Console to Monitor SAS Servers	88
Using the Server Performance Counters and Information Fields	96
Using the SAS OLAP Server Monitor	100
Using Third-Party Products to Monitor SAS Servers	101

### **Chapter 8   △   Administering Logging for SAS Servers    103**

About Server Logging	103
About Metadata Server Loggers	104
Enabling Server Logging	106
Initial Logging Configuration for SAS Servers	107
Modifying Your Server Logging Configurations	110
Enable More Detailed Logging for Server Troubleshooting	111
Create a Log File for Workspace Server Troubleshooting	112
Enable More Detailed Logging for SAS Object Spawner Troubleshooting	113
Collecting ARM Log Information for SAS Data Integration Studio Jobs	113

Collecting ARM Log Information in SAS OLAP Servers	115
Audit Server Logging Level Changes That Originate in Client Applications	116
Capture Metadata Server Management Events in the System Facility Appender	117
Capture XML Information in the SAS Metadata Server Log	117

## **PART 4    Backing Up and Restoring Your System    121**

### **Chapter 9   △   Best Practices for Backing Up and Restoring Your System    123**

About Backups and Restores	123
About the Metadata Server Backup and Restore Utilities	124
Using Operating System Commands to Back Up the Metadata Server	126
About Using the Export SAS Package Wizard to Back Up Specific SAS Folders	126
Best Practices for Backing Up Your SAS System	127
Best Practices for Restoring Your System	134

### **Chapter 10   △   Using the Backup Wizard in SAS Management Console    137**

About the Backup Wizard	137
Summary of Procedures for Using the Backup Wizard	138
Ensure Appropriate User Credentials	139
Use the Backup Wizard to Define a Backup Job and a Corresponding Restore Job	140
Manage Backup and Restore Jobs That Were Created by the Wizard	142
Run a Backup or a Restore Job That Was Created by the Wizard	143
Troubleshooting the Backup Wizard	144
Use Schedule Manager to Schedule a Backup Job That Was Created by the Backup Wizard	145

### **Chapter 11   △   Using the %OMABAKUP Macro to Perform Backups and Restores    147**

About %OMABAKUP	147
What %OMABAKUP Does	148
Ensuring Appropriate User Credentials	149
Running the backupServer.sas and restoreServer.sas Programs	150
Error Logging	151
Writing Custom Programs Using %OMABAKUP	151
Warning Messages	155
%OMABAKUP Macro: Reference	156

### **Chapter 12   △   Manually Restoring an Unresponsive SAS Metadata Server    161**

Manually Restoring an Unresponsive SAS Metadata Server	161
--	-----

## **PART 5    Optimizing the Performance of the SAS Metadata Server    165**

### **Chapter 13   △   Using Configuration Settings to Optimize Metadata Server Performance    167**

Overview of Performance-Related Configuration Settings	167
Configuring the Number of Threads Used by the Metadata Server	167
Configuring Metadata Server Journaling	170

## **Chapter 14 △ Managing Metadata Server Performance 173**

- Setting the Server's MEMSIZE Parameter 173
- Input/Output Balancing 173
- Configuring Concurrent Queries/Clients 174
- Note to Users of 32-bit Windows Systems 174
- Using %OMARUNAN (or the RUNANALYSIS Option of %OMABAKUP) to Optimize Memory Usage 175
- %OMARUNAN Autocall Macro: Reference 177
- Recovering Memory on the SAS Metadata Server 177

## **PART 6 Managing SAS Metadata Repositories 179**

### **Chapter 15 △ Creating, Registering, Moving, Copying, Renaming, and Deleting SAS Metadata Repositories 181**

- About SAS Metadata Repositories 181
- About SAS Metadata Repository Access Modes 182
- Create a Custom Repository 183
- Change the Properties of a Metadata Repository 184
- Copying, Promoting, Importing, and Exporting Your Metadata 185
- Moving a Metadata Repository to a New Location on the Same Metadata Server 188
- Porting a Metadata Repository to a Different Host Environment 189
- Registering a Metadata Repository 191
- Unregistering a Metadata Repository 192
- Deleting a Metadata Repository 193

## **PART 7 Managing Metadata Content 195**

### **Chapter 16 △ Working with SAS Folders 197**

- About the SAS Folder Structure 197
- Best Practices for Managing SAS Folders 201
- Best Practices for Maintaining Associations Among Objects in SAS Folders 202
- Using the BI Lineage Plug-in 203
- Customizing Your Folder Structure 208

### **Chapter 17 △ Using the Promotion Tools 213**

- Overview of Promotion Tools 214
- Special Considerations for Promoting Metadata 221
- Special Considerations for Promoting Metadata From SAS 9.1.3 to SAS 9.2 223
- Preparing to Use the Promotion Tools 226
- How to Access the Promotion Tools 229
- Using the Export SAS Package and Import SAS Package Wizards: Example Usage Scenario 229
- Using the Batch Export and Import Tools 238
- Troubleshooting the Promotion Tools 253

## **PART 8 Appendixes 257**

**Appendix 1 △ Default SAS Ports 259**

Overview 259

Default Port Numbers for SAS Servers and Spawners 259

Default Port Numbers for Third-Party Software 260

Port Numbering in a Multiple-Level Environment 261

**Appendix 2 △ Reference: Configuration Directories 263**

Introduction to the Configuration Directory Reference 263

Contents of the Levn Subdirectory 263

Contents of the Metadata Server Subdirectory (SASMeta) 265

Contents of the Application Server Subdirectory (SASApp or SASMain) 266

Contents of the Web Subdirectory 267

**Appendix 3 △ Default Locations for Server Logs 269**

Default Locations for Server Logs 269

**Appendix 4 △ Configuration Files 271**

Reference: Configuration Files for SAS Servers 271

Reference Information for omaconfig.xml 280

**Appendix 5 △ Options for the Metadata Server Invocation Command 289**

Modifying the Metadata Server Invocation Options 289

Recommended Parameters 289

Optional Parameters 292

**Appendix 6 △ Using the SAS Deployment Manager to Update Host Name References 295**

Overview of Updating Host Name References 296

Process for Updating Host Name References 304

Troubleshooting the Update Host Name References Tool 312

Example Scenarios for Updating Host Name References 312

**Appendix 7 △ Recommended Reading 319**

Recommended Reading 319

**Glossary 321****Index 333**





# What's New

---

---

## Overview

The SAS Intelligence Platform includes new and enhanced system administration tools, new features for server logging and management, new server components, changes to metadata repositories, and new documentation for administrators. The enhancements and changes include the following:

- a new logging facility for SAS servers
- new server management and monitoring features
- new SAS server components
- enhancements and changes pertaining to metadata server administration
- enhancements and changes pertaining to metadata administration
- a new configuration management tool
- new documentation for administrators

---

## New Logging Facility for SAS Servers

A new standard logging facility is provided for SAS servers. The facility supports problem diagnosis and resolution, performance and capacity management, and auditing and regulatory compliance. The logging facility provides the following capabilities for servers:

- Log events can be directed to multiple destinations, including files, operating system facilities, and client applications.
- For each log destination, you can configure the message layout, including the contents, the format, the order of information, and literal text.
- For each log destination, you can configure a filter to include or exclude events based on levels and message contents.
- For the metadata server, security-related events are captured, including authentication events, client connections, changes to user and group information, and permission changes.
- You can generate performance-related log events in a format that can be processed by an Application Response Measurement (ARM) 4.0 server.

- You can adjust logging levels dynamically by using the Server Management features of SAS Management Console.
- You can use SAS Management Console and third—party monitoring tools to view the contents of logs.

---

## New Server Management and Monitoring Features

The following new features are provided to help you manage and monitor your SAS server components, including the SAS Metadata Server, SAS OLAP Server, SAS Stored Process Server, SAS Workspace Server, SAS Pooled Workspace Server, SAS object spawner, and SAS Table Server:

- You can use the new Server Manager features of SAS Management Console to perform the following tasks:
  - quiesce, stop, pause, resume, and validate servers
  - test connections to servers
  - view information about current connections and processes
  - view performance counters that provide statistics about activity that has occurred since a server was last started
  - view server logs
  - change logging levels dynamically
  - refresh servers, so that metadata changes take effect without the need to restart

Server monitoring can also be performed using third-party enterprise systems management products.

- On UNIX and z/OS systems, you can use the new `sas.servers` script to start, stop, or restart all of your SAS servers in the correct order with a single command. You can also use the script to display the status of servers and to create a UNIX boot script that is tailored to your platform.

---

## New SAS Server Components

The initial configuration of your system includes these new server components:

- a server context for the metadata server called SASMeta. In multiple-machine installations, this server context includes a workspace server and other servers that are needed to run metadata utilities such as the Backup and Restore Wizard.
- a SAS Pooled Workspace Server, which is configured to use *server-side pooling*. In this new configuration, the SAS object spawner maintains a collection of workspace server processes that are available for clients. This server configuration is intended for use by query and reporting tools such as SAS Information Map Studio, SAS Web Report Studio, and the SAS Information Delivery Portal.
- a SAS Deployment Tester Server, which processes validation tasks that are requested either by the SAS Deployment Tester plug-in or by a SAS Deployment Tester batch client.

Depending on which SAS offerings you have installed, your configuration might also include the following new server components:

- a SAS Content Server, which is included with some offerings. This server stores digital content (such as documents, reports, and images) that is created and used by SAS Web applications. For example, the SAS Content Server stores report

definitions that are created by users of SAS Web Report Studio, as well as images and other elements that are used in reports. In addition, the SAS Content Server stores documents and other files that are to be displayed in the SAS Information Delivery Portal or in SAS solutions.

To interact with the SAS Content Server, client applications use WebDAV-based protocols for access, versioning, collaboration, security, and searching. Administrative users can use the browser-based SAS Web Management Console to create, delete, and manage permissions for folders on the SAS Content Server.

- a SAS Table Server, which is used by the SAS Shared Services component of the middle tier.
- a DataFlux Integration Server, which executes DataFlux jobs and real-time services that cleanse data or analyze data quality. The jobs and services are created using DataFlux applications (dfPower Architect and dfPower Profile) and are executed on the server through job flows that are created in SAS Data Integration Studio.
- a SAS In-Process Server, which provides the ability to schedule jobs from certain Web-based SAS applications without using a separate scheduling server. The scheduling functions run as a process within the application.

---

## Enhancements and Changes Pertaining to Metadata Server Administration

The following enhancements and changes pertain to administration of the SAS Metadata Server:

- Users can be assigned to administrator roles that enable them to have unrestricted access to metadata, act as metadata server operators, or perform user administration tasks.
- The metadata server can be paused to an Administration state, enabling unrestricted users (but not other users) to continue to perform tasks that require access to the server.
- The Backup and Restore Wizard in SAS Management Console enables you to create backup and restore jobs and execute them or deploy them for scheduling. In addition, SAS now provides customized backup and restore programs that you can launch from the command line. Metadata server configuration files are included in backups.
- Metadata server journaling is turned on by default.
- New options for internal user authentication have been added to the metadata server configuration file omaconfig.xml.

---

## Enhancements and Changes Pertaining to Metadata Administration

The following enhancements pertain to the administration of metadata and metadata repositories:

- Content is stored in a hierarchy of SAS folders that are visible in the **Folders** tab of SAS Management Console and in other client applications, as applicable. The folders segregate system information from business information, provide personal folders for individual users, and provide an area for shared data.
- BI Manager import and export wizards have been replaced with the Export SAS Package and Import SAS Package wizards. The wizards have been expanded to handle a larger number of object types. In addition, new batch tools enable you to

export and import folders, objects, or groups of folders and objects from an operating system command line or from a batch script.

- In the second maintenance release for SAS 9.2, the ExportCubes and ImportCubes batch tools are new. These tools provide a simplified method of promoting OLAP cubes from SAS 9.1.3 to SAS 9.2.
- Custom repositories are supported by all applications.
- It is not necessary to define dependencies between repositories. If a repository dependency was defined in SAS 9.1.3, then the dependency relationship is eliminated during the SAS 9.1.3 to SAS 9.2 migration process. However, the existing associations between metadata objects still remain in effect.
- The November 2010 release includes these features:
  - The new BI Lineage plug-in for SAS Management Console identifies connections between BI objects in the SAS Folders tree. The lineage information is stored in a repository called the BILineage repository. For details, see “Using the BI Lineage Plug-in” on page 203.
  - The Export SAS Package and Import SAS Package wizards can be used to promote dashboard objects (including dashboards, indicators, indicator data, ranges, indicator configurations, and dashboard components). For important details, see “Promoting SAS BI Dashboard 4.3 Content” in the *SAS Intelligence Platform: Web Application Administration Guide*.
  - The new content promotion tool for the SAS Information Delivery Portal 4.3 can be used to promote portal pages, page templates, portlet instances, and the portal’s application and user permissions trees. See “Promoting Portal Content” in the *SAS Intelligence Platform: Web Application Administration Guide*.
- The following features have been removed:
  - The Replication Wizard in SAS Management Console has been deprecated. This wizard includes the functions of the Replication Wizard and the Promotion Wizard that were available in SAS 9.1.3. The documentation for this feature was removed effective with the November 2010 edition of the *SAS Intelligence Platform: System Administration Guide*. For more information, see SAS Note 40834 or contact SAS Technical Support.
  - The use of third-party database management systems to store the repository manager and metadata repositories is no longer supported.
  - The repository auditing feature provided in previous releases has been removed.

---

## New Configuration Management Tool

The new SAS Software Deployment Manager enables you to remove configuration information (including metadata definitions, configuration files, and Windows services) for one or more components of your installation. You can use this feature to remove a component that you will no longer be using, or to move a component from one machine to another. You can also use the SAS Software Deployment Manager to rebuild SAS Web applications and to update a service account password that is included only in metadata.

In the October 2009 release, the SAS Deployment Manager includes a new option to renew software licenses for some SAS solutions that depend on a SAS middle tier.

In the third maintenance release for SAS 9.2, the SAS Deployment Manager includes a new option to change the host names (including the network domains to which they belong) of server machines in your deployment.

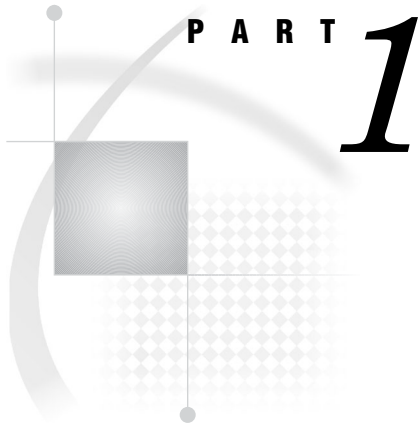
---

## New Documentation for Administrators

SAS 9.2 includes these changes to system administration documentation:

- All information about scheduling has been moved to *Scheduling in SAS*.
- Detailed information about configuring logging for SAS servers is now available in the *SAS Logging: Configuration and Programming Reference*.



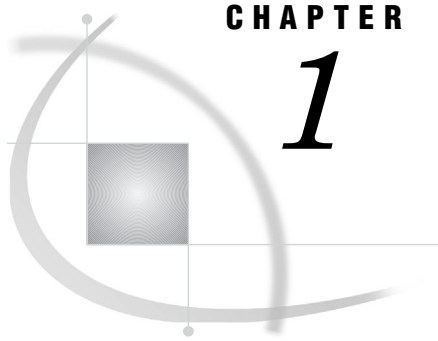


## Getting Started

<i>Chapter 1</i> . . . . .	<b>Introduction to the SAS Intelligence Platform: System Administration Guide</b>	<b>3</b>
<i>Chapter 2</i> . . . . .	<b>Understanding the State of Your System</b>	<b>5</b>
<i>Chapter 3</i> . . . . .	<b>Who Can Do What: Credential Requirements for SAS Management Console Tasks</b>	<b>33</b>
<i>Chapter 4</i> . . . . .	<b>What to Do Next: Administration Tasks</b>	<b>39</b>







## 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

The SAS Intelligence Platform is a comprehensive, end-to-end infrastructure for creating, managing, and distributing enterprise intelligence. The SAS Intelligence Platform comprises four software offerings:

- SAS BI Server
- SAS Enterprise BI Server
- SAS Data Integration Server
- SAS Enterprise Data Integration Server

Solutions for enterprise intelligence, customer intelligence, and financial intelligence are built on the SAS Intelligence Platform, as well as turnkey solutions for specialized markets in life sciences, health care, retail, manufacturing, and others. Therefore, if your SAS deployment plan includes a SAS solution, then the SAS Intelligence Platform documentation also pertains to you.

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
- monitoring server activity
- administering server logging
- administering the SAS Metadata Server, including the following tasks:
  - performing backups and restores
  - monitoring the server
  - optimizing the server's performance
- administering SAS Metadata Repositories and metadata content, including the following tasks:
  - creating repositories
  - working with SAS Folders
  - importing, exporting, and copying metadata

- promoting metadata and related content
- modifying your configuration

The appendixes of this guide provide reference information about log files, configuration files, 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
- scheduling
- logging

To access these documents, go to <http://support.sas.com/92administration>.

Additional information for specific operating systems is available in the SAS companion guides.

---

## 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 tasks that are described in the *SAS Intelligence Platform: Installation and Configuration Guide*.

To access these documents, go to <http://support.sas.com/92administration>.

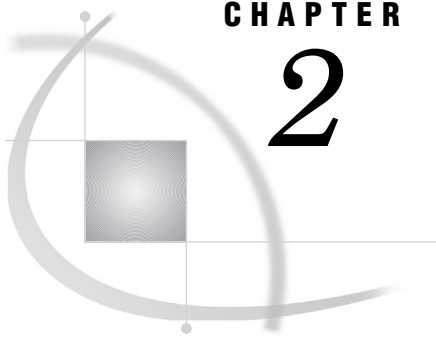
---

## 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 e-mail to [accessibility@sas.com](mailto:accessibility@sas.com).



## CHAPTER

## 2

# Understanding the State of Your System

<i>Overview of the State of Your System</i>	6
<i>Overview of SAS Management Console</i>	7
<i>About SAS Management Console</i>	7
<i>About SAS Management Console Plug-ins</i>	7
<i>Using the SAS Management Console Plug-ins</i>	9
<i>About the SAS Management Console Folders Tab</i>	9
<i>Overview of SAS Deployment Manager and SAS Web Administration Console</i>	10
<i>Overview of the SAS Metadata Server and Its Initial Configuration</i>	11
<i>What Is the SAS Metadata Server?</i>	11
<i>About the Metadata Manager Plug-in</i>	11
<i>About the SASMeta Server Context</i>	12
<i>About the SAS Metadata Repositories and the Repository Manager</i>	12
<i>How the SAS Metadata Server Works</i>	13
<i>The Initial Configuration of the SAS Metadata Server</i>	14
<i>Overview of SAS Application Server Components and Their Initial Configuration</i>	15
<i>About the SAS Application Server Hierarchy</i>	15
<i>About the SAS Application Server Components</i>	16
<i>Overview of Other SAS Servers and Spawners</i>	17
<i>Overview of Ports</i>	19
<i>Overview of the Configuration Directory Structure</i>	19
<i>About the Configuration Directory</i>	19
<i>About the Levn Subdirectory</i>	20
<i>About the Application Server Subdirectory</i>	21
<i>About the SASMeta Subdirectory</i>	21
<i>About the Web Subdirectory</i>	22
<i>Overview of Configuration Files</i>	22
<i>About the SAS Metadata Server Configuration File (omaconfig.xml)</i>	22
<i>About Other SAS Server Configuration Files</i>	23
<i>About the Security-Related Configuration Files</i>	23
<i>About Configuration Files for Client Applications</i>	24
<i>Overview of Initial Roles, Groups, and Users</i>	24
<i>About User Roles for System Administration</i>	24
<i>Overview of System Administration User Roles</i>	24
<i>Metadata Server: Unrestricted Role</i>	24
<i>Metadata Server: User Administration Role</i>	25
<i>Metadata Server: Operation Role</i>	25
<i>Management Console: Advanced Role</i>	26
<i>Management Console: Content Management Role</i>	26
<i>About the Initial User Groups</i>	26
<i>User Groups Initially Defined in the Operating System</i>	26
<i>User Groups Initially Defined in Metadata</i>	26

<i>About the Initial User Accounts</i>	27
<i>Overview of the Initial User Accounts</i>	27
<i>SAS Administrator</i>	28
<i>SAS Trusted User</i>	29
<i>SAS Installer</i>	29
<i>SAS Spawned Servers Account</i>	30
<i>SAS First User</i>	30
<i>SAS Anonymous Web User</i>	31
<i>LSF Administrator</i>	31
<i>lsfuser</i>	32

---

## 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 and deployment process:

- SAS Management Console, which is a Java application that provides a single point of control for administering the SAS Intelligence Platform.
- SAS Deployment Manager and SAS Web Administration Console, which are also used to administer the SAS Intelligence Platform.
- 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 that have been installed and configured in your environment to run SAS processes. These components can include SAS Workspace Servers, SAS Pooled Workspace Servers, SAS Stored Process Servers, SAS OLAP Servers, SAS/CONNECT servers, SAS Grid Servers, and SAS batch servers.

*Note:* In the SAS Intelligence Platform, the term *server* refers to a program or programs that wait for and fulfill requests from client programs for data or services. The term server does not necessarily refer to a specific computer, since a single computer can host one or more servers of various types.  $\triangle$

- other SAS servers that might have been installed in your environment, including a SAS/SHARE server, a SAS Content Server, a SAS Table Server, and a DataFlux Integration Server.
- the ports that are used by each server to listen for incoming requests.
- the configuration directory that stores configuration files, logs, scripts, and special-purpose SAS data sets on each SAS server machine and each middle-tier machine.
- the main configuration files that control the operation of the system.
- the initial SAS users, groups, and roles that have been defined, both on your host operating systems and in the SAS Metadata Repository.

Read the following topics to learn about these components and their initial configuration.

## Overview of SAS Management Console

### About SAS Management Console

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

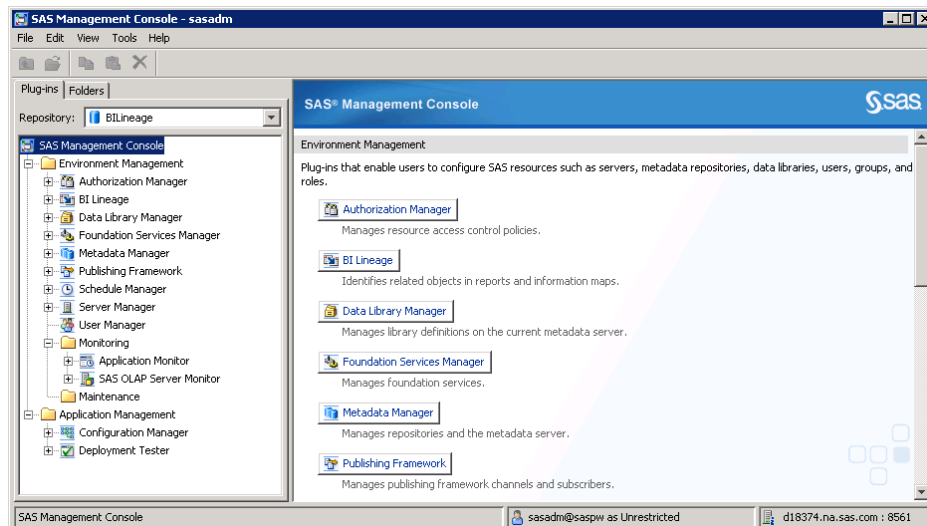
Whenever the SAS Metadata Server is running, you can use SAS Management Console to connect to the SAS Metadata Server and view and manage the metadata objects that are stored in the server's metadata repositories. The SAS Management Console user interface includes a **Plug-ins** tab and a **Folders** tab that you can use to access metadata, as described in the following topics.

### 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 and update metadata to define SAS servers. Some plug-ins include additional functionality. For example, the Server Manager plug-in enables you to stop, pause, and resume servers and to monitor server activity.

Only certain users can view and use plug-ins. A user's access to plug-ins depends on which roles the user is assigned to and which capabilities are assigned to those roles (see "Overview of Initial Roles, Groups, and Users" on page 24).

If you are authorized to use plug-ins, then the **Plug-ins** tab appears in the left pane when you open SAS Management Console and connect to a metadata server, as shown in this example:



The following table provides the name and purpose of each of the most frequently used SAS Management Console plug-ins. References to related documentation are also provided.

**Table 2.1** SAS Management Console Plug-ins

Plug-in Name	Plug-in Description
Authorization Manager	Define and maintain access rules to control how users and groups can access metadata definitions. For details, see the <i>SAS Intelligence Platform: Security Administration Guide</i> .
BI Lineage	Identify connections between BI objects in the SAS Folders tree. For details, see “Using the BI Lineage Plug-in” on page 203 and the SAS Management Console Help. This plug-in is new with the November 2010 release of SAS Management Console.
Configuration Manager	View and modify configuration attributes for SAS applications including SAS Information Delivery Portal, SAS Web Report Studio, SAS Financial Management, and SAS Strategic Performance Management. For details, see the <i>SAS Intelligence Platform: Web Application Administration Guide</i> .
Data Library Manager	Create and maintain definitions for SAS libraries and database schemas. For details, see the <i>SAS Intelligence Platform: Data Administration Guide</i> .
Foundation Services Manager	View and modify deployment configurations for infrastructure and extension services that are used by applications such as SAS Information Delivery Portal, SAS Web Report Studio, and SAS Web OLAP Viewer for Java. For details, see the <i>SAS Intelligence Platform: Web Application Administration Guide</i> .
Map Service Manager	Create and maintain map service definitions that link cubes to ESRI ArcGIS map services so that ESRI Geographic Information System (GIS) spatial map information can be accessed and returned by an OLAP server when the cubes are queried. For details, see the <i>SAS Intelligence Platform: Web Application Administration Guide</i> .
Metadata Manager	Perform administration tasks related to the SAS Metadata Server, including the following: <ul style="list-style-type: none"> <li>□ stop, pause, resume, or reset the metadata server</li> <li>□ check the status of the metadata server and the metadata repositories</li> <li>□ create and manage metadata repositories</li> <li>□ create and execute jobs to back up or restore your metadata repositories</li> </ul>
Publishing Framework	Create and maintain definitions for channels, package subscribers, group subscribers, and event subscribers. For details, see the <i>SAS Intelligence Platform: Web Application Administration Guide</i> .
Schedule Manager	Create and maintain schedules for running jobs that are created in SAS Data Integration Studio and SAS Web Report Studio. For details, see <i>Scheduling in SAS</i> .

Plug-in Name	Plug-in Description
Server Manager	<p>Perform administration tasks related to SAS servers, including the following:</p> <ul style="list-style-type: none"> <li>□ check the status of servers and validate them to ensure that they are configured correctly</li> <li>□ stop, pause, quiesce, resume, or refresh the SAS object spawner or the SAS Table Server; and stop, pause, resume, or quiesce a process on a SAS OLAP Server, SAS Workspace Server, or SAS Stored Process Server</li> <li>□ connect to the metadata server, to components of SAS Application Servers, to the SAS object spawner, or to a SAS Table Server to perform monitoring activities</li> </ul> <p>Details about the preceding tasks are provided in this guide.</p> <p>You can also use the Server Manager plug-in to create and maintain definitions for servers. For details, see the <i>SAS Intelligence Platform: Application Server Administration Guide</i>.</p>
User Manager	Create and maintain definitions for users, groups, and roles. For details, see the <i>SAS Intelligence Platform: Security Administration Guide</i> .

## Using the SAS Management Console Plug-ins

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), right-click the item and select an action from the menu that appears.
- 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 SAS 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 and the applicable administration guides.

## About the SAS Management Console Folders Tab

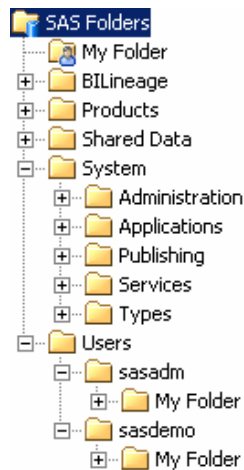
The **Folders** tab in SAS Management Console displays the hierarchy of SAS folders. These folders are used to store metadata for business intelligence content such as libraries, tables, jobs, cubes, information maps, and reports. The folders also store SAS system content that is not directly accessed by business users.

As a system administrator, you will use this view for tasks such as the following:

- setting up a folder structure for your users

- setting permissions on folders and objects
- importing and exporting metadata and associated files (for example, when promoting content to production)

For a newly installed system, the folder organization is similar to the following:



For details, see “About the SAS Folder Structure” on page 197.

---

## Overview of SAS Deployment Manager and SAS Web Administration Console

In addition to SAS Management Console, the following interfaces are provided to help you administer the SAS Intelligence Platform:

- SAS Deployment Manager, which is a graphical user interface that enables you to do the following:
  - remove one or more components of a SAS Intelligence Platform configuration from your environment. For details, see “Removing a SAS Configuration” in the *SAS Intelligence Platform: Installation and Configuration Guide*.
  - update passwords for the service accounts that were configured when you ran the SAS Deployment Wizard. For details, see “Manage Passwords” in the *SAS Intelligence Platform: Security Administration Guide*.
  - rebuild Web applications. For details, see “Rebuilding the SAS Web Applications” in the *SAS Intelligence Platform: Web Application Administration Guide*.
  - renew your software license for some SAS solutions that depend on a SAS middle tier. For details, see “Updating the SID File in Metadata for SAS Solutions” in the *SAS Intelligence Platform: Installation and Configuration Guide*.
  - change the host names (including the network domains to which they belong) of server machines in your deployment. For details, see Appendix 6, “Using the SAS Deployment Manager to Update Host Name References,” on page 295.
  - configure a product after you have applied maintenance or after you have upgraded the product to a new release. For details, see “Configuring SAS



Products” in the *SAS Intelligence Platform: Installation and Configuration Guide*.

- SAS Web Administration Console, which is a Web-based interface that enables you to monitor which users are logged on to SAS Web applications; manage folders and permissions for content in the SAS Content Server; and view the current configuration of Web applications. For details, see “Using the SAS Web Administration Console” in the *SAS Intelligence Platform: Web Application Administration Guide*.

---

## Overview of the SAS Metadata Server and Its Initial Configuration

---

### What Is the SAS Metadata Server?

The SAS Metadata Server is the most critical software component in the SAS Intelligence Platform. All of the client applications and the other SAS servers that are part of the platform depend on the SAS Metadata Server and cannot operate without it.

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 SAS Metadata Server enables centralized control so that all users access consistent and accurate data.

The functionality of the SAS Metadata Server is provided through the SAS Open Metadata Architecture, which is a 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.

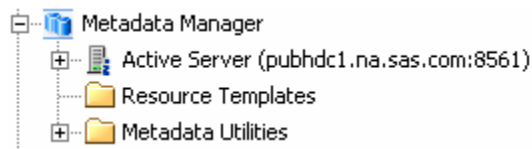
The SAS Metadata Server stores information about the following:

- the enterprise data sources and data structures that are accessed by SAS applications
- resources that are created and used by SAS applications, including information maps, OLAP cubes, report definitions, stored process definitions, and scheduled jobs
- the servers that run SAS processes
- the users and groups of users that use the system, and the levels of access that users and groups have to resources

---

### About the Metadata Manager Plug-in

In SAS Management Console, the SAS Metadata Server appears in the Metadata Manager plug-in, as shown here:



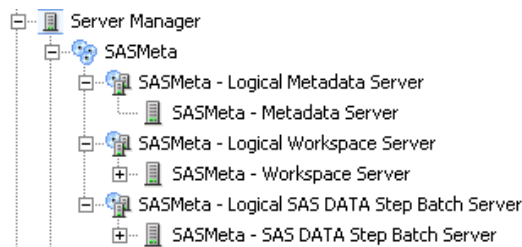
You can use this plug-in to stop, pause, resume, or reset the metadata server; to check the status of the metadata server and the metadata repositories; and to create and manage metadata repositories. You can also access metadata utilities such as the Backup and Restore Wizard.

*Note:* In the Plug-ins view, the **Properties** action generally displays an object's metadata definition. However, when you select **Properties** on the **Active Server** node of the Metadata Manager plug-in, the information that is displayed is supplied directly by the metadata server. △

---

## About the SASMeta Server Context

So that you can make use of SAS Management Console's server monitoring features, the metadata server also appears in the SASMeta application server tree (also referred to as the SASMeta server context) in the Server Manager plug-in, as shown here:



As shown in the preceding display, the SASMeta server context also includes a workspace server and a DATA step batch server. This workspace server is only visible to and accessible by users who are in the SAS Administrators group. This server is used to execute metadata utilities including the Backup and Restore Wizard and the Import SAS Package and Export SAS Package wizards.

### CAUTION:

**Use the SASMeta workspace server only for running the metadata utilities. Do not use it for other SAS processing.** △

The DATA step batch server in the SASMeta context is provided for scheduling backup jobs that are created by using the Backup Wizard.

---

## 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.

Metadata repositories 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.

The SAS Metadata Server supports three types of metadata repositories:

**foundation repository**

the required metadata store for a SAS Metadata Server. Each metadata server has one foundation repository. As a default, the name of the foundation repository is Foundation. (If you performed a custom installation, then you might have assigned a different name.)

**project repository**

a metadata repository that serves as an isolated work area or playpen. Project repositories are available for SAS Data Integration Studio only. In general, each user who participates in change management has his or her own project repository.

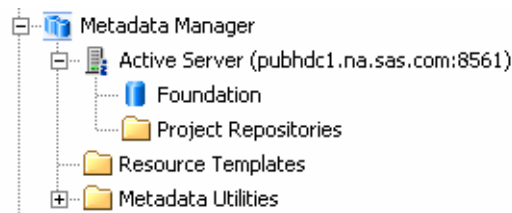
A project repository enables a user to check out and lock metadata from a foundation repository or a custom repository so that the metadata can be modified and tested in a separate area. When the metadata is ready, it is checked in and unlocked so that it is available to other users.

**custom repository**

an optional metadata store for a SAS Metadata Server, in addition to the foundation repository. Custom repositories are useful for physically segregating metadata for storage or security purposes.

Custom repositories appear as separate root folders in the **Folders** tab of SAS Management Console.

In SAS Management Console, the repositories are displayed in the Metadata Manager plug-in, as shown in this example:



In SAS 9.2, it is not necessary to define dependencies between repositories. If a repository dependency was defined in SAS 9.1.3, then the dependency relationship is eliminated during the SAS 9.1.3 to SAS 9.2 migration process. However, the existing associations between metadata objects still remain in effect.

---

## 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 data sets that contain the requested records are read from repository data sets on disk into the server’s memory, where they remain until the server is paused and resumed or until the server is stopped.

In the default configuration, metadata updates that are requested by client applications are recorded immediately in memory and in a journal file so that client access can be resumed. The more time-consuming updates to the repository data sets on disk are performed later in the background.

## The Initial Configuration of the SAS Metadata Server

The SAS Deployment Wizard provides the following initial configuration for your metadata server:

- Your metadata server contains a foundation metadata repository called Foundation. (If you performed a custom installation, then you might have assigned a different name.) This repository contains the following items:
  - metadata definitions for initial users, groups, and roles
  - default authorization settings that give the initial users, groups, and roles access to different levels of functionality
  - metadata definitions for your servers
  - 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 typically located in the following path:

*SAS-configuration-directory*\Lev1\SASMeta\**MetadataServer\MetadataRepositories\Foundation**

In this path, *SAS-configuration-directory* refers to the SAS configuration directory (see “About the Configuration Directory” on page 19).

- Journaling has been enabled.

When journaling is not enabled, clients that request metadata updates must wait for updates to be written to the repository data sets on the file system before client access is resumed. When journaling is enabled, access is returned to clients as soon as the metadata updates are written to the in-memory database and the journal file. The more time-consuming updates to the repository data sets are performed later in the background.

The default settings for journaling are as follows:

- The name of the journal file is **MetadataJournal.dat**.
- The journal file is written to the following path:

*SAS-configuration-directory*\Lev1\SASMeta\**MetadataServer\Journal**

- On 32-bit machines and z/OS systems, 200 megabytes of storage on the metadata server are reserved for the journal file and cannot be used for other processes. On 64-bit machines, the default setting is 500 megabytes. To modify these settings, see “Configuring Metadata Server Journaling” on page 170.
- Two special users of the metadata server have been set up: the SAS Administrator and the SAS Trusted User.
- The metadata server has been configured to write daily logs. The default location for logs is as follows:

*SAS-configuration-directory*\Lev1\SASMeta\**MetadataServer\Logs**

For details about the default logging configuration, see “Initial Logging Configuration for SAS Servers” on page 107.

# Overview of SAS Application Server Components and Their Initial Configuration

## About the SAS Application Server Hierarchy

The SAS Deployment Wizard creates metadata definitions and configuration files for a set of SAS servers that are referred to collectively as a SAS Application Server. The server hierarchy includes the following levels:

### SAS Application Server

a logical framework (or *server context*) under which SAS applications execute. This framework enables you to specify metadata that applies to all of the logical servers and servers that the SAS Application Server contains. You can also attach libraries, schemas, directories, and other resources that are available to SAS servers, regardless of the type of server. As a result, you can deploy applications in several modes while ensuring that applications will execute properly in each mode.

The default name of the initial SAS Application Server is **SASApp**. However, the SAS Deployment Wizard enables you to provide a different name during installation. (In single-machine installations that are migrated from SAS 9.1.3, the SAS Application Server is typically called **SASMain**).

Your installation also includes a SAS Application Server called SASMeta, which is used to execute metadata utilities (see “About the SASMeta Server Context” on page 12).

### logical servers

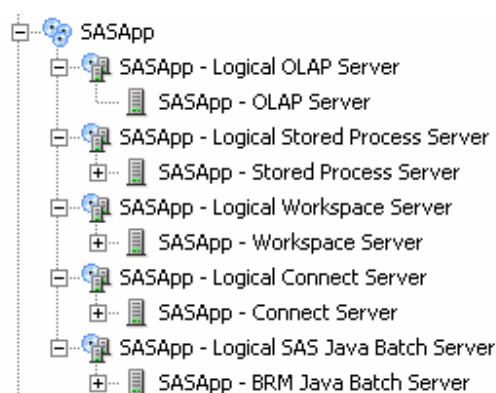
a logical framework within SAS Application Server metadata that contains definitions for one or more similar servers. A logical server is referenced when an application requests a specific launch mode, such as interactive or batch.

### servers

specific process instances that perform the requested work. A server definition contains the server metadata that is required to connect to a SAS server on a particular machine. The definition includes details about where the process is executing and how a client should contact the server, as well as options that describe how the server should behave.

*Note:* In the SAS Intelligence Platform, the term *server* refers to a program or programs that wait for and fulfill requests from client programs for data or services. The term server does not necessarily refer to a specific computer, since a single computer can host one or more servers of various types. △

The SAS Application Server hierarchy appears in SAS Management Console in the Server Manager plug-in, as in the following example:



## About the SAS Application Server Components

The most common components of a SAS Application Server are SAS Workspace Servers, SAS Pooled Workspace Servers, and SAS Stored Process Servers. An application server can also contain SAS/CONNECT servers, SAS OLAP Servers, SAS DATA step batch servers, and SAS Grid Servers. The purpose of each server component is described in the following table.

**Table 2.2** SAS Application Server Components

Server Name	Server Description
SAS Workspace Server	SAS Workspace Servers fulfill client requests for SAS sessions. When these servers are configured as standard workspace servers (without workspace pooling), each client establishes a connection to a single-user server process, uses the server, and then disconnects. For details, see the <i>SAS Intelligence Platform: Application Server Administration Guide</i> .
SAS Pooled Workspace Server	<p>SAS Pooled Workspace Servers are workspace servers that use <i>server-side pooling</i>, which is new with SAS 9.2. The SAS object spawner maintains a pool of reusable workspace server processes called a pool, thus avoiding the overhead that is associated with creating a new process for each connection. Workspace server load balancing is built in to server-side pooling.</p> <p>Pooled workspace servers are intended for use by query and reporting tools such as SAS Web Report Studio and SAS Information Delivery Portal when accessing relational information maps.</p> <p>For details, see the <i>SAS Intelligence Platform: Application Server Administration Guide</i>.</p>
SAS Stored Process Servers	<p>SAS Stored Process Servers fulfill client requests to execute SAS stored processes. A stored process is a SAS program that is stored on a server and can be executed as required by requesting applications.</p> <p>Stored process servers are configured to be load balanced. By default, the object spawner balances workloads across three stored process server processes.</p> <p>For details, see the <i>SAS Intelligence Platform: Application Server Administration Guide</i>.</p>
SAS/CONNECT server	SAS/CONNECT servers enable clients to execute code on a remote host, or to move data between client and server machines. For details, see <i>SAS/CONNECT User's Guide</i> .

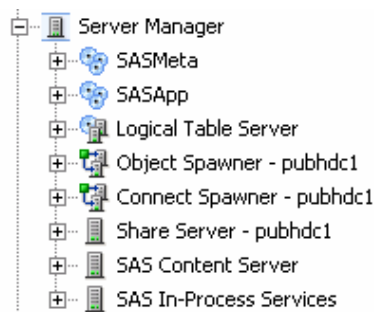
Server Name	Server Description
SAS OLAP Server	SAS OLAP Servers provide access to multidimensional data. For details, see the <i>SAS Intelligence Platform: Application Server Administration Guide</i> .
SAS batch server	SAS batch servers are part of the configuration for SAS scheduling. They are metadata objects that store information about an executable that the scheduling server can use to run scheduled tasks. For details, see <i>Scheduling in SAS</i> .
SAS Grid Server	A SAS Grid Server is configured if you have the SAS Grid Manager software package. SAS Grid Servers are metadata objects that store information about an executable that the scheduling server can use to start SAS/CONNECT on grid nodes. For details, see <i>Grid Computing in SAS</i> .

For additional details, see:

- “Initial Logging Configuration for SAS Servers” on page 107
- “Reference: Configuration Files for SAS Servers” on page 271

## Overview of Other SAS Servers and Spawners

The SAS Intelligence Platform includes additional servers and spawners that might appear in the Server Manager tree in SAS Management Console, as shown here:



The servers and spawners include the following:

**Table 2.3** Other SAS Servers and Spawners

Server or Spawner Name	Server or Spawner Description
SAS Content Server	<p>stores digital content (such as documents, reports, and images) that are created and used by SAS client applications. To interact with the server, clients use WebDAV-based protocols for access, versioning, collaboration, security, and searching.</p> <p>The SAS Content Server is used by SAS Information Delivery Portal and SAS Web Report Studio, as well as other SAS products and solutions.</p> <p>The server's repositories are stored in the <b>AppData</b> subdirectory of your configuration directory. For more information, see the <i>SAS Intelligence Platform: Web Application Administration Guide</i>.</p>
DataFlux Integration Server	<p>executes DataFlux jobs and real-time services that cleanse data or analyze data quality. The jobs and services are created using DataFlux applications (dfPower Architect and dfPower Profile) and are executed on the server through job flows that are created in SAS Data Integration Studio.</p> <p>For information about configuring this server, see "Administering SAS Data Integration Studio" in the <i>SAS Intelligence Platform: Desktop Application Administration Guide</i>.</p>
SAS In-Process Services	<p>provides the ability to schedule jobs from certain Web-based SAS applications without using a separate scheduling server. The scheduling functions run as a process within the application. For details, see <i>Scheduling in SAS</i>.</p>
SAS object spawner	<p>runs on the same machine as the SAS Application Server components. The spawner listens for incoming client requests for workspace servers, pooled workspace servers, and stored process servers, and it launches instances of these servers as needed.</p> <p>Although the object spawner is not part of the SAS Application Server hierarchy, it is essential to the operation of workspace servers, stored process servers, and pooled workspace servers.</p> <p>For details, see the <i>SAS Intelligence Platform: Application Server Administration Guide</i>.</p>
SAS Table Server	<p>is included in some SAS Intelligence Platform software offerings. It is used by the SAS Shared Services component of the middle tier.</p>
SAS/CONNECT spawner	<p>listens for SAS/CONNECT client requests and invokes SAS sessions on the remote computer. For details, see <i>Communications Access Methods for SAS/CONNECT and SAS/SHARE</i>.</p>
SAS/SHARE server	<p>SAS/SHARE enables multiple users to update a SAS file (or several files) at the same time. It also enables users to access SAS files on a server without requiring a separate SAS/CONNECT report login for each user. For details, see the <i>SAS/SHARE User's Guide</i>.</p>

In addition, the following server appears under the Deployment Tester plug-in under Application Management:

#### SAS Deployment Tester Server

processes validation tasks that are requested either by the SAS Deployment Tester plug-in or by a SAS Deployment Tester batch client. For usage instructions, see the SAS Management Console Help.



For additional details, see:

- “Initial Logging Configuration for SAS Servers” on page 107
- “Reference: Configuration Files for SAS Servers” on page 271

---

## 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. For information about the default ports, see Appendix 1, “Default SAS Ports,” on page 259. During installation, the SAS Deployment Wizard enables you to override the default port numbers with different values.

---

## 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. 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-configuration-directory* 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:

**usr/local/deployment-plan-name** or **usr/local/sas**

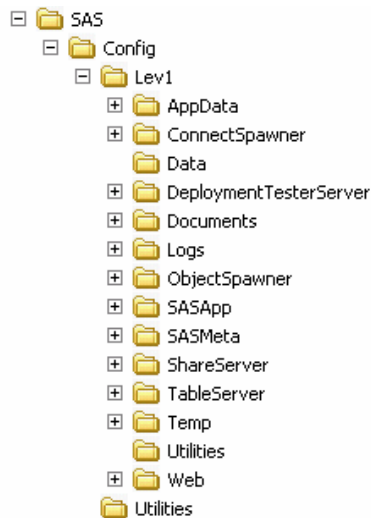
*Note:*

- In these paths, *deployment-plan-name* is the name that was specified when the deployment plan was created. If no name was specified, then the default name **Config** is used.
- You can specify a different path when you run the SAS Deployment Wizard. In UNIX environments, the installer generally overrides the path with the site's preferred location (for example, **opt/sas/Config**). The installer must have write permission to this path.

△

- on z/OS systems, the directory that is specified in the **CONFIG\_DIR** environment variable

In the configuration directory, the SAS Deployment 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:



*Note:* Depending on your operating system and which products you have installed, your site might have different subdirectories. △

---

## About the Levn Subdirectory

The **Levn** subdirectory contains configuration information and other files for a particular installation instance. **Lev1** is generally used for production environments. Additional levels, such as **Lev2** and **Lev3**, can be used for environments that you install for purposes such as development and testing. During installation, the SAS Deployment Wizard enables you to specify the level number.

The contents of the **Levn** directory depend on which components are installed on the machine. Its contents might include the following:

- on UNIX and z/OS machines, a script called `sas.servers` that you can use to start, stop, or restart all of the SAS servers and spawners on the machine in the correct order by using a single command.
- an **AppData** subdirectory, which contains data sets for the SAS Content Server repository. It also contains data that is installed for the use of specific applications (for example, SAS BI Dashboard).
- a subdirectory for the application server (see “About the Application Server Subdirectory” on page 21).
- a subdirectory for the metadata server (see “About the SASMeta Subdirectory” on page 21).
- subdirectories for other SAS servers and spawners, including the SAS object spawner, the SAS Deployment Tester Server, the SAS Table Server, the SAS/CONNECT spawner, and the SAS/SHARE server.
- a Web subdirectory for files related to Web applications (see “About the Web Subdirectory” on page 22) .
- logs and documents that were created during the installation and configuration process

For more details about the **Levn** directory, see “Contents of the Levn Subdirectory” on page 263.

---

## About the Application Server Subdirectory

On machines where components of a SAS Application Server are installed, the **Levn** directory contains an application server subdirectory. A SAS Application Server is a logical entity that contains specific servers (for example, a SAS Workspace Server and a SAS Stored Process Server) that execute SAS code. Application server subdirectories are named as follows:

- **SASApp** is the default name that the SAS Deployment Wizard assigns to directory for the first application server. If you assigned a different name when you ran the SAS Deployment Wizard, then the subdirectory is named accordingly.
- If you configure more than one application server, then additional directories are created that correspond to each of the application server names.
- If you have a single-machine installation that was migrated from SAS 9.1.3, then the application server subdirectory is typically called **SASMain**.

The application server subdirectory contains the following items:

- a subdirectory for each component of the application server, including the workspace server, the pooled workspace server, the OLAP server, the stored process server, the SAS/CONNECT server, batch servers, and grid servers (if applicable). Each of these subdirectories contains the server's start-up scripts, configuration files, logconfig.xml files to configure logging, and **Logs** directories to store log files.
- a subdirectory for data that is used by the application server.
- a **SASEnvironment** subdirectory to store deployed jobs, job steps, SAS format and informat catalogs, and user-created SAS autocall macros that are needed to perform processing on the application server
- configuration files that apply to all components of the application server

For more details about this subdirectory, see “Contents of the Application Server Subdirectory (SASApp or SASMain)” on page 266. For information about the server configuration files that are contained in this subdirectory, see “Configuration Files for Components of SAS Application Servers” on page 274.

---

## About the SASMeta Subdirectory

On the machines where the metadata server is installed, the **Levn** directory includes a subdirectory for the metadata server's files. **SASMeta** is the name that the SAS Deployment Wizard assigns to the metadata server context and, in turn, to this directory.

The **SASMeta** directory contains the following items:

- configuration files that apply to the SASMeta application server components, which include a workspace server and a DATA step batch server.
- a **WorkspaceServer** subdirectory, which contains files for the SASMeta workspace server. This server is used to execute metadata utilities. The subdirectory contains start-up scripts, configuration files, a logconfig.xml file, and a **Logs** directory for this server.
- a **BatchServer** subdirectory, which contains files for the SASMeta DATA step batch server. This server is used to schedule backup jobs that are created by the Backup Wizard. The subdirectory contains start-up scripts, configuration files, a logconfig.xml file, and a **Logs** directory for this server.
- **SASEnvironment** and **Data** subdirectories that can be used by the workspace server.

- a **MetadataServer** subdirectory, which contains the following:
  - the metadata server's start-up and management scripts, configuration files, logconfig.xml files to configure logging, and **Logs** directories to store log files.
  - programs to back up and restore the metadata server (for details, see "About the Metadata Server Backup and Restore Utilities" on page 124).
  - a subdirectory called **MetadataRepositories**, which contains a subdirectory for each metadata repository. These subdirectories contain the SAS data sets that store your metadata.
  - a subdirectory call **rposmgr**, which contains data sets for the repository manager.

**CAUTION:**

**The MetadataRepositories and rposmgr directories contain data sets that are critical to the operation of your system.** To safeguard the integrity of your system:

- Never move, delete, modify, or directly read the data sets in these directories. These data sets should be accessed only by metadata server processes.
- Do not place other files in these directories.
- Back up the contents of these directories regularly, as described in Chapter 9, "Best Practices for Backing Up and Restoring Your System," on page 123.

△

- a directory for the metadata server journal.
- a **SASBackup** directory that can be used as a destination for backing up the metadata server.

For more information about the metadata server configuration files, see "Configuration Files for the Metadata Server" on page 272. For a complete list of the contents of the MetadataServer subdirectory, see "Contents of the Metadata Server Subdirectory (SASMeta)" on page 265.

---

## About the Web Subdirectory

On middle-tier machines (that is, machines with installed Web components), the **Lev1** directory contains a subdirectory called **web**. For details, see "Contents of the Web Subdirectory" on page 267. 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 **SASMeta\MetadataServer** directory contains a file called omaconfig.xml. This file contains SAS Metadata Server settings such as the following:

- values for the libref and path of the metadata server's repository manager
- the names and locations of the adminUsers.txt and trustedUsers.txt files, which contain information about special users of the metadata server
- journaling options

For details about the omaconfig.xml file, see “Reference Information for omaconfig.xml” on page 280.

---

## About Other SAS Server Configuration Files

Server-tier machines contain configuration files for each SAS server. These files specify the SAS system options that are in effect when the server is launched. The following configuration files are present in each server's configuration directory:

sasv9.cfg

is the primary configuration file for a server. You should not edit this configuration file.

sasv9\_usermods.cfg

enables user modifications to a server configuration. If you need to modify an option or add a new option for a specific server, then edit this file.

logconfig.xml

specifies the logging configuration for the server. For details, see Chapter 8, “Administering Logging for SAS Servers,” on page 103.

The files sasv9.cfg and sasv9\_usermods.cfg are also present in the configuration directory for the SAS Application Server (for example, the **SASApp** directory) and in the *SAS-home* directory where SAS Foundation installed. Configuration files are set up to inherit options from one another.

For details about the server configuration files, see “Reference: Configuration Files for SAS Servers” on page 271.

---

## About the Security-Related Configuration Files

On the machine that hosts the SAS Metadata Server, three configuration files that affect security are located in the **SASMeta\MetaDataServer** directory:

*Note:* Be sure to limit access to these files to only those users who require access in order to perform their job functions. △

adminUsers.txt

defines your system's initial *unrestricted user*. The SAS Deployment Wizard creates this user, which is usually called the SAS Administrator. In the file, the *unrestricted user* name is preceded by an asterisk. For information about unrestricted users, see “About User Roles for System Administration” on page 24.

*Note:* If you migrated your system from SAS 9.1.3, then you might have additional users in this file. In SAS 9.2, role memberships are the preferred method for assigning *unrestricted user* status. △

trustedUsers.txt

conveys privileged status to a service account to enable that account to act on behalf of other users. For more information, see “About the Initial User Accounts” on page 27.

trustedPeers.xml

can be used to limit the scope of trust for connections from SAS sessions to the metadata server. Initially, the scope is not limited. For details, see the “Trusted Peer” topic in the “Authentication Mechanisms” chapter of the *SAS Intelligence Platform: Security Administration Guide*.

---

## About Configuration Files for Client Applications

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

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

## Overview of Initial Roles, Groups, and Users

---

### About User Roles for System Administration

#### Overview of System Administration User Roles

A user *role* is a set of capabilities. Some SAS applications make certain actions available only to users or groups that have a particular role. To enable a user or group to perform those actions, you add the user or group to the appropriate role.

During installation, the SAS Deployment Wizard creates metadata definitions for several user roles. The following initial roles are created for performing system administration tasks:

- Metadata Server: Unrestricted
- Metadata Server: User Administration
- Metadata Server: Operation
- Management Console: Advanced
- Management Console: Content Management

The SAS Deployment Wizard also creates some additional roles for users of specific client applications, including SAS Enterprise Guide, SAS Web Report Studio, and SAS Add-In for Microsoft Office.

*Note:*

- To understand how role assignments affect a user’s ability to perform the system administration tasks that are documented in this guide, see Chapter 3, “Who Can Do What: Credential Requirements for SAS Management Console Tasks,” on page 33.
- For information about how to add users to roles or to define additional roles, see “Managing Users, Groups, and Roles” in the *SAS Intelligence Platform: Security Administration Guide*.

△

#### Metadata Server: Unrestricted Role

*Note:* The initial name of this role is **META: Unrestricted Users Role**, and the initial display name for this role is **Metadata Server: Unrestricted**. △

The **Metadata Server: Unrestricted** role has access to all metadata regardless of SAS permissions settings. Users in this role, which are referred to as *unrestricted users*, can do the following:

- perform all of the functions that users in the **Metadata Server: User Administration** and **Metadata Server: Operation** roles can perform
- access all metadata except user passwords
- continue to access metadata repositories and use features of SAS Management Console when the metadata server is paused to the Administration state

Follow these important guidelines when using an account that is in the **Metadata Server: Unrestricted** role:

- This role is intended only for tasks that require unrestricted access to metadata (for example, adding other users to the **Metadata Server: Unrestricted** role, performing tasks when the metadata server is paused to the Administration state, and creating, deleting, formatting, and unregistering foundation repositories).
- Use the accounts in this role only to log on to SAS Management Console. You should not use these accounts to log on to other client applications.

The SAS Deployment Wizard places one user in this role. This user, which is generally called the SAS Administrator, is specified in the file `adminUsers.txt`. For details, see “About the Initial User Accounts” on page 27.

*Note:* The **Metadata Server: Unrestricted** role provides access to the metadata server, not to other SAS servers. Some administration tasks require access to a SAS Application Server, which might require additional credentials. For details, see Chapter 3, “Who Can Do What: Credential Requirements for SAS Management Console Tasks,” on page 33. △

## Metadata Server: User Administration Role

*Note:* The initial name of this role is **META: User and Group Administrators Role**, and the initial display name for this role is **Metadata Server: User Administration**. The SAS Deployment Wizard assigns the SAS Administrators group to this role. △

Users who are assigned to the **Metadata Server: User Administration** role can create and modify users, groups, and roles. Users in this role are authorized to update user passwords. They cannot read existing passwords, except the passwords for their own logins.

For details about the user administration tasks, see the *SAS Intelligence Platform: Security Administration Guide*.

## Metadata Server: Operation Role

*Note:* The initial name of this role is **META: Operators Role**, and the initial display name for this role is **Metadata Server: Operation**. The SAS Deployment Wizard assigns the SAS Administrators group to this role. △

Users who are assigned to the **Metadata Server: Operation** role can perform the following tasks:

- stop, pause, resume, and reset (or refresh) the metadata server
- add, delete, format, and unregister metadata repositories (except the foundation repository)

Users who perform these tasks must also be assigned to the SAS Management Console Advanced Role, which provides access to the Metadata Manager plug-in.

## Management Console: Advanced Role

The **Management Console: Advanced** role is initially configured to allow access to all of the plug-ins in SAS Management Console. The SAS Deployment Wizard assigns the SAS Administrators group to this role.

You must make the following additional role assignments to enable certain functions:

- the **Metadata Server: User Administration** role or the **Metadata Server: Unrestricted** role, to access functionality within the User Manager plug-in
- the **Metadata Server: Operation** role or the **Metadata Server: Unrestricted** role, to access some functions within the Metadata Manager plug-in

## Management Console: Content Management Role

The **Management Console: Content Management** role is initially configured to enable access to the following features of SAS Management Console:

- the User Manager, Authorization Manager, and Library Manager plug-ins
- the **Folders** tab

To have access to certain functionality within User Manager plug-in, the user must also be assigned to the **Metadata Server: User Administration** role or the **Metadata Server: Unrestricted** role.

---

## About the Initial User Groups

### User Groups Initially Defined in the Operating System

On some of the machines in your configuration, the following operating system user groups might have been defined during installation:

sas (UNIX only)

This group is used to control access to the configuration directories on UNIX machines. The group includes the installer (the sas user). Typically, you will not add any other users to this group.

SAS Server Users (Windows only)

This group might have been created on Windows machines that have stored process servers, pooled workspace servers, or standard workspace servers installed. During the installation process, you should have assigned this group the right to **Log on as a batch job**, which is required in order to start processes for those servers.

If you are not using Integrated Windows authentication, then you can add users to this group to enable them to start workspace server processes.

SASGRP (z/OS only)

On z/OS systems, this RACF group is used to control access to the configuration directory. The group is defined with an OMVS segment and is set as the default group for the SAS Installer and SAS Spawnd Servers accounts.

### User Groups Initially Defined in Metadata

The SAS Deployment Wizard creates the following user groups in metadata. These groups are part of the SAS Intelligence Platform security infrastructure. For information about how they are used to implement security, see the *SAS Intelligence Platform: Security Administration Guide*.



**PUBLIC**

a standard group with implicit membership. This group includes everyone who can access the metadata server, either directly or through a trust relationship. A user who does not have an individual identity uses the PUBLIC group identity.

**SASUSERS**

a standard group with implicit membership. This group includes all users who have individual identities.

**SAS Administrators**

a standard group for metadata administrators. By default, this group is granted broad access to the metadata and has all roles other than the **Metadata Server: Unrestricted** role.

**SAS System Services**

a standard group for service identities that need to read server definitions or other system resources.

**SAS General Servers**

a standard group whose members can be used for launching stored process servers and pooled workspace servers.

**Table Server Administrators**

a standard group that has permission to administer SAS Table Servers.

**LSF Services**

a group whose members can schedule jobs in the LSF component of Platform Suite for SAS. Beginning with the second maintenance release for SAS 9.2, this group is part of the standard configuration for sites that use Platform Suite for SAS to schedule SAS Web Report Studio reports.

The LSF Services group is not needed if you use SAS In-Process Services to schedule reports.

See also: “lsfuser” on page 32

---

## About the Initial User Accounts

### Overview of the Initial User Accounts

During installation, the SAS Deployment Wizard creates several initial user accounts. Some of these user accounts are created for all installations, some of the accounts are optional, and some of the accounts are created only if certain software components are installed.

- ☐ SAS Administrator
- ☐ SAS Trusted User
- ☐ SAS Installer
- ☐ SAS Spawnd Servers
- ☐ SAS First User
- ☐ SAS Anonymous Web User
- ☐ LSF Administrator
- ☐ lsfuser

These user accounts might have been assigned different names at your site. For each account, the following topics provide the default name and user ID, information about

whether or when the account is required, the account's purpose and use, and the locations where the account is set up.

## SAS Administrator

Type of Installation	Default User Name	Default User ID	Required?	Location of Account
New with default settings	SAS Administrator	sasadm@saspw	Yes	Metadata
New with external authentication selected	SAS Administrator	sasadm	Yes	Metadata and OS
Migrated from 9.1.3	SAS Administrator	sasadm	Yes	Metadata and OS

The SAS Administrator user account has privileges that are associated with the **Metadata Server: Unrestricted** role (see “Metadata Server: Unrestricted Role” on page 24). In addition, the SAS Administrator account is initially a member of the SAS Administrators group (see “User Groups Initially Defined in Metadata” on page 26) .

This user is defined in the following locations:

- in the file `adminUsers.txt`, which is typically located in the following path:

*SAS-configuration-directory*\Lev1\SASMeta\MetadataServer

This file ensures that your site will always have at least one user with the privileges of an *unrestricted user*, regardless of what is specified in metadata. You cannot override this user's privileges by modifying the user definition in SAS Management Console.

- in metadata.
- in the operating system of the metadata server machine, only in the following situations:
  - You selected the External authentication option for this user during a custom installation.
  - You migrated your system from SAS 9.1.3 to SAS 9.2.

In new, default installations of SAS 9.2, the SAS Administrator is an *internal user account* that is known only to SAS and that is authenticated internally in metadata. When internal authentication is used, it is not necessary for this user to have a local or network account.

*Note:* We recommend that you establish individual metadata administrators rather than sharing the SAS Administrator account. See “Security Tasks” in the *SAS Intelligence Platform: Security Administration Guide*. △

## SAS Trusted User

Type of Installation	Default User Name	Default User ID	Required?	Location of Account
New with default settings	SAS Trusted User	sastrust@saspw	Yes	Metadata
New with external authentication selected	SAS Trusted User	sastrust	Yes	Metadata and OS
Migrated from 9.1.3	SAS Trusted User	sastrust	Yes	Metadata and OS

The SAS Trusted User is a privileged service account that can act on behalf of other users on a connection to the metadata server. No user should log on directly as a *trusted user*, except to perform certain administrative tasks associated with the SAS Information Delivery Portal. For details about those tasks, see the *SAS Intelligence Platform: Web Application Administration Guide*.

The SAS Trusted User is defined in the following locations:

- ☐ in metadata.
- ☐ in the file `trustedUsers.txt`, which is typically located in the following path:

*SAS-configuration-directory*\Lev1\SASMeta\MetadataServer

A user is granted privileges as a *trusted user* only if the user is specified in this file.

*Note:* Typically, there is no reason to add more IDs to this file. In particular, do not add regular users to this file.  $\Delta$

- ☐ in the operating system of the metadata server machine, only in the following situations:
  - ☐ You selected the External authentication option for this user during a custom installation.
  - ☐ You migrated your system from SAS 9.1.3 to SAS 9.2.

In new, default installations of SAS 9.2, the SAS Trusted User is an *internal user account* that is known only to SAS and that is authenticated internally in metadata. When internal authentication is used, it is not necessary for this user to have a local or network account.

For detailed information about this user, see the *SAS Intelligence Platform: Security Administration Guide*.

## SAS Installer

Default User Name	Default User ID	Required?	Location of Account
SAS Installer	sas	Yes	OS

The SAS Installer is a user account that is used to install and configure SAS software. On UNIX and z/OS systems, this account is the owner of configuration directories and their contents and is the process owner for items such as the metadata server, the OLAP server, and the object spawner. The account should continue to be available after installation so that it can be used to apply maintenance.

The SAS Installer account must be defined in the operating systems of the following machines:

- the metadata server machine
- machines that host an OLAP server
- machines where the object spawner is installed

This user is not defined in metadata.

## SAS Spawnd Servers Account

Default User Name	Default User ID	Required?	Location of Account
SAS Spawnd Servers	sassrv	Yes	OS and metadata (as a login for the SAS General Servers group)

The SAS Spawnd Servers account is the initially configured process owner for pooled workspace servers and stored process servers. An account for this user must be defined in the operating system of the following machines:

- machines that host a stored process server
- machines that host a pooled workspace server

During the installation process on Windows machines, this user should have been assigned the right to **Log on as a batch job**. This right can also be assigned by adding the user to the SAS Server Users group (see “User Groups Initially Defined in the Operating System” on page 26) .

This user does not have an individual metadata identity. However, a login for this user is defined for the SAS General Servers group.

## SAS First User

Default User Name	Default User ID	Required?	Location of Account
SAS Demo User	sasdemo	No	Metadata and OS

The SAS First User is an optional account that can serve as a generic end user when you are testing any of the SAS client applications. During installation, the Software Deployment Wizard enables you to specify whether to create this user.

If you selected the option to create this user, then the user’s account is defined in the following locations:

- in metadata

- in the operating system of the metadata server machine and workspace server machine

During the installation process on Windows machines, this user should have been assigned the right to **Log on as a batch job**. This right can also be assigned by adding the user to the SAS Server Users group (see “User Groups Initially Defined in the Operating System” on page 26) .

## SAS Anonymous Web User

Type of Installation	Default User Name	Default User ID	Required?	Location of Account
New with default settings	SAS Anonymous Web User	webanon@saspw	No	Metadata
New with external authentication selected	SAS Anonymous Web User	webanon	No	Metadata and OS

The SAS Anonymous Web User is an optional account that is used to grant clients access to applicable SAS Web Infrastructure Platform components. When Web clients request access to Web services, they are not prompted for credentials but instead are granted access under this user account.

This user is defined in the following locations:

- in metadata. In default installations of SAS 9.2, the SAS Anonymous Web User is an internal user account that is known only to SAS and that is authenticated internally in metadata. When internal authentication is used, it is not necessary for this user to have a local or network account.
- in the operating system of the metadata server machine, only if you selected the External authentication option for this user during a custom installation.

## LSF Administrator

Default User ID	Required?	Location of Account
none	Yes, if Platform Suite for SAS is installed	OS

The LSF administrator is the primary administrator for the Platform scheduling server and the owner of the Process Manager server. This user is required only if you have installed Platform Suite for SAS in support of either scheduling or grid computing.

The LSF administrator account must be defined in the operating system of the machine where Platform Suite for SAS is installed. This user must have full control of the LSF and Process Manager directories. On Windows systems, this user must belong to the Administrators Group and must have rights to **Act as part of the operating system** and **Log on as a batch job**.

This user is not defined in metadata.

## lsfuser

Default User ID	Required?	Location of Account
lsfuser	Yes, if Platform Suite for SAS is installed and is used to schedule WRS reports	OS, metadata (as a login for the LSF Services group), and password file in LSF

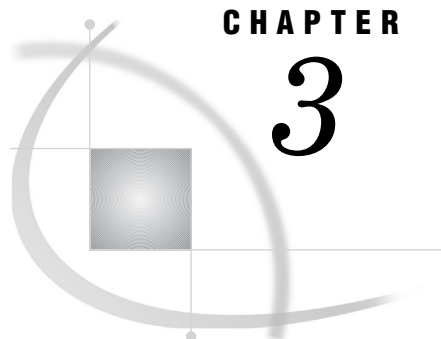
The lsfuser account is used by default when you schedule SAS Web Report Studio reports using the LSF component of Platform Suite for SAS. The lsfuser account must be defined in the operating system of the machine where Platform Suite for SAS is installed. On Windows machines, the account must also be added to the password file in the LSF software. For details, see “Enabling Report Scheduling With Platform Suite for SAS” in *Scheduling in SAS*.

This user does not have an individual metadata identity. However, a login for this user is defined for the LSF Services group, effective with the second maintenance release for SAS 9.2.

### *Note:*

- If you have not yet installed the second maintenance release for SAS 9.2, follow the instructions in SAS Installation Note 35283 to set up credentials for scheduling reports in LSF.
- The lsfuser account is not needed if you use SAS In-process Services to schedule reports.

△



## CHAPTER

## 3

## Who Can Do What: Credential Requirements for SAS Management Console Tasks

---

<i>About Security for System Administration Tasks</i>	33
<i>Who Can Do What in the Metadata Manager Plug-in</i>	34
<i>Who Can Do What in the Server Manager Plug-in</i>	35
<i>About Adding Users to the SAS Administrators Group</i>	36
<i>Requirements for Accessing a Standard Workspace Server</i>	37
<i>Clearing the Credentials Cache</i>	37

---

### About Security for System Administration Tasks

The system administration tasks that can be performed using SAS Management Console affect the stability of your servers, the integrity of your data and metadata, and the performance of client applications. Therefore, it is important to limit access to system administration tasks to only those users who:

- have the necessary skills, background, and knowledge
- need to perform system administration tasks in order to fulfill their job functions

The security model for the SAS Intelligence Platform model enables you to control access to system administration tasks by adding users to administrative roles, adding users to groups, and granting permissions on metadata. If necessary, you can change the capabilities that are assigned to a role. You can also create new roles.

This chapter describes the group memberships, role memberships, and other criteria that are required for users to perform the system administration tasks that are documented in this guide. Requirements are provided for two different approaches to system administration:

- *centralized administration*, which means that a single administrative user (or a small group of users) performs all administrative functions for your SAS deployment, with no division of responsibility
- *delegated administration*, which means that various types of administrative functions for your SAS deployment are divided among different administrative users or groups of users

For some functions, the requirements are the same regardless of whether you are using centralized or delegated administration.

The requirements in this chapter reflect initial settings in a default installation. Role definitions and group membership might be configured differently at your site. For an in-depth description of the security model, or for information about modifying the security structure, see the *SAS Intelligence Platform: Security Administration Guide*.

## Who Can Do What in the Metadata Manager Plug-in

Use the following table to determine the group memberships, role memberships, and other criteria that enable users to perform tasks in the Metadata Manager plug-in in SAS Management Console:

**Table 3.1** Who Can Do What in the Metadata Manager Plug-in

Actions	Requirements <sup>1</sup>	
	Centralized Administration Approach	Delegated Administration Approach
Check the access state of the metadata server, metadata repositories, and repository manager	Membership in the SAS Administrators group	Membership in the <b>Management Console: Advanced</b> role.
Create, register, and delete custom repositories	Membership in the SAS Administrators group	Membership in the <b>Management Console: Advanced</b> and <b>Metadata Server: Operation</b> roles, and WriteMetadata permission on the Root folder of the SAS Folders tree.
Change a repository's properties, including the access mode, the location of users' folders, and mapping for notes and documents		
Stop, pause, or resume the metadata server	Membership in the SAS Administrators group	Membership in the <b>Management Console: Advanced</b> and <b>Metadata Server: Operation</b> roles.
Continue to use all functions of SAS Management Console when the metadata server is paused to the Administration state	Membership in the <b>Metadata Server: Unrestricted</b> role	
Analyze, repair, or upgrade metadata		
Initialize or unregister a custom repository		
Add, initialize, register, unregister, and delete a foundation repository. <sup>2</sup>	Membership in the SAS Administrators group, specification as an unrestricted user in adminUsers.txt, and an external user ID that can be authenticated on the metadata server host machine	
Use the Backup Wizard	Membership in the SAS Administrators group and credentials to start a workspace server (see "Requirements for Accessing a Standard Workspace Server" on page 37). For additional requirements, see Chapter 10, "Using the Backup Wizard in SAS Management Console," on page 137.	

<sup>1</sup> These columns list minimum requirements. The **Metadata Server: Unrestricted** role can be substituted for any of the roles and groups that are specified here.

<sup>2</sup> Deleting or unregistering the foundation repository makes the metadata server unusable. These actions should be taken only in special situations.



As a best practice, users who are members of the SAS Administrators group should be defined as internal users (rather than external users) in metadata. For more information, and to understand what role memberships and permissions are derived from membership in this group, see “About Adding Users to the SAS Administrators Group” on page 36.

## Who Can Do What in the Server Manager Plug-in

Use the following table to determine the group memberships, role memberships, and other criteria that enable users to perform tasks in the Server Manager plug-in in SAS Management Console:

**Table 3.2** Who Can Do What in the Server Manager Plug-in

Actions	Requirements <sup>1</sup>	
	Centralized Administration Approach	Delegated Administration Approach
Create or update server metadata (except the SASMeta workspace server and the SASMeta DATA step batch server)	Membership in the SAS Administrators group	Membership in the <b>Management Console: Advanced</b> role
Validate servers (except the SASMeta workspace server)	Validation of a standard (non-pooled) workspace server also requires credentials to start a workspace server (see “Requirements for Accessing a Standard Workspace Server” on page 37).	Validation of a standard (nonpooled) workspace server also requires credentials to start a workspace server (see “Requirements for Accessing a Standard Workspace Server” on page 37).
Connect to servers and spawners and view the following tabs <sup>2</sup> : <ul style="list-style-type: none"> <li>Metadata server <b>Connections</b> and <b>Options</b> tabs</li> <li>Object spawner <b>Options</b> tab</li> <li>OLAP server <b>Connections</b>, <b>Sessions</b>, and <b>Options</b> tabs</li> </ul>	Membership in the SAS Administrators group	Membership in the <b>Management Console: Advanced</b> role, and Administer permission on server metadata
Connect to servers and spawners (except the metadata server and the SASMeta workspace server) and view all tabs <sup>2</sup>	Membership in the SAS Administrators group	Membership in the <b>Management Console: Advanced</b> role, and Administer permission on server metadata
Adjust the logging level for a server, spawner, or server process (except the metadata server)		
Stop, pause, resume, or quiesce a server, spawner, or server process (except the metadata server and the SASMeta workspace server)		
Refresh the object spawner		

Actions	Requirements <sup>1</sup>	
	Centralized Administration Approach	Delegated Administration Approach
Connect to the metadata server and view the <b>Clients</b> , <b>Loggers</b> , and <b>Logs</b> tabs <sup>2</sup>	Membership in the SAS Administrators group	Membership in the <b>Management Console: Advanced</b> and <b>Metadata Server: Operation</b> roles
Adjust the metadata server's logging levels		
Stop, pause, or resume the metadata server		
Validate the SASMeta workspace server	Membership in the SAS Administrators group	Membership in the <b>Management Console: Advanced</b> role; and ReadMetadata and Administer permission on the SASMeta workspace server
Connect to the SASMeta workspace server and view tabs <sup>2</sup>		
Stop, pause, resume, or quiesce a process on the SASMeta workspace server		

1 These columns list minimum requirements. The **Metadata Server: Unrestricted** role can be substituted for any of the roles and groups that are specified here.

2 To view tabs, you must right-click on the server or spawner and click **Connect**.

As a best practice, users who are members of the SAS Administrators group should be defined as internal users (rather than external users) in metadata.

For more information, and to understand what role memberships and permissions are derived from membership in this group, see the next topic.

## About Adding Users to the SAS Administrators Group

When you add a user to the SAS Administrators group, the user derives the following memberships and permissions (if the group's initial settings have not been changed):

- membership in the **Metadata Server: Advanced** role
- membership in the **Metadata Server: Operation** role
- membership in the **Metadata Server: User Administration** role. This role is required in order to perform most actions in the User Manager plug-in.
- WriteMetadata permission on the Root folder of the SAS Folders tree
- Administer permission on server metadata
- ReadMetadata on server metadata in the SASMeta context

If you use a centralized (rather than delegated) approach to system administration, then the SAS Administrators group provides a convenient mechanism for enabling users to perform administrative tasks. Centralized administration means that a single administrative user (or a small group of users) performs all administrative functions for your SAS deployment, with no division of responsibility.

As a best practice, users who are members of the SAS Administrators group should be defined as internal users (rather than external users) in metadata whenever possible. For details about creating administrative users, see “Add Administrators” in the *SAS Intelligence Platform: Security Administration Guide*.

## Requirements for Accessing a Standard Workspace Server

Special rights are required for tasks that involve running processes on a standard (non-pooled) workspace server. In SAS Management Console, these tasks include the following:

- validating a workspace server
- using the Backup Wizard
- using the Import SAS Package and Export SAS Package wizards or the batch promotion tools to import or export physical content that is associated with stored processes, tables, jobs, libraries, or external files

The requirements for performing these tasks depend on which authentication method is used and which server is being accessed:

- If you use host (credential-based) authentication, the following requirements must be met:
  - The user's account must be known to the workspace server host. On Windows, the account must have rights to **Log on as a batch job** on the host machine. Typically, you would add the user to an operating system group that has this right (for example, the SAS Server Users group).
  - The user's account must correspond to a metadata identity that has the ReadMetadata permission for the server definition.
- If you use Integrated Windows authentication (IWA), the following requirements must be met:
  - The user's account must be known to the workspace server host. It is not necessary to have rights to **Log on as a batch job**.
  - The user's account must correspond to a metadata identity that has the ReadMetadata permission for the server definition.
  - Integrated Windows authentication must be selected in the connection profile that the user uses to log on to SAS Management Console. In addition, Integrated Windows authentication must be fully configured on the workspace server.
- If the workspace server is configured to use SAS token authentication, then no credentials on the workspace server host are necessary. The user can log on to SAS Management Console with either an internal account or an external account. The account must correspond to a metadata identity that has the ReadMetadata permission for the server definition.
- If you are using a workspace server that is part of the metadata server context (for example, SASMeta), then the user must also be a member of the SAS Administrators group. The use of this server is limited to metadata server administration tasks such as using the Backup and Restore Wizard and using the Import SAS Package and Export SAS Package wizards.

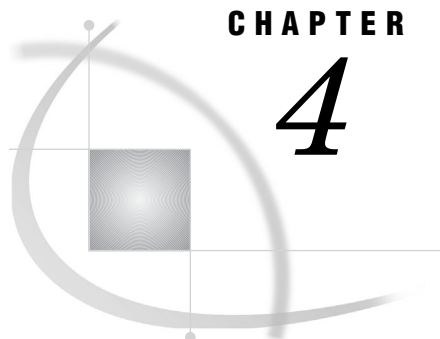
For detailed information about configuring and using the authentication methods for SAS servers, see the *SAS Intelligence Platform: Security Administration Guide*.

## Clearing the Credentials Cache

If a user is denied access when trying to connect to or validate a server or spawner, and the user's credentials are then updated to allow access, the credentials cache must

be cleared before the user tries to connect again. To clear the credentials cache, click **File ► Clear Credentials Cache** in SAS Management Console.

If the object spawner denies access to a server because of the lack of ReadMetadata permissions, and the user is then granted the needed permission, you must reset the object spawner's authorization cache before the user tries to connect again. To do so, expand the Server Manager tree in SAS Management Console. Next, expand the object spawner, right-click the host node, and click **Connect**. After you have connected, right-click the host node again, and click **Refresh Spawner**.



## CHAPTER

## 4

# What to Do Next: Administration Tasks

<i>Overview of Administration Tasks</i>	39
<i>First-Priority Setup Tasks</i>	40
<i>Summary of First-Priority Setup Tasks</i>	40
<i>Recommended Operating System Protections for Windows Machines</i>	40
<i>Default Operating System Protections for UNIX and z/OS Machines</i>	42
<i>Standard Setup Tasks</i>	43
<i>Optional Setup Tasks</i>	44
<i>Ongoing System Administration Tasks</i>	46
<i>Best Practices for Ensuring the Integrity of Your System</i>	47
<i>Allow Only the Metadata Server to Access Your Metadata Repositories</i>	47
<i>Use Best Practices for Working with SAS Folders</i>	47

## Overview of Administration Tasks

Before you perform the administration tasks that are outlined in this chapter, you should have successfully completed the following tasks:

- Use the SAS Deployment Wizard to install and configure your software on each machine in your environment.
- Complete the post-installation tasks that are described in your **Instructions.html** file, which is located in the **Documents** subdirectory of your configuration directory. These tasks include performing manual configuration steps (if applicable), validating your SAS servers, validating your Web applications, and backing up your system.

After the SAS Intelligence Platform has been installed, configured, and validated, you will have a functional system. You can now begin performing administration tasks that are necessary to fully implement the SAS Intelligence Platform in your environment. These include the following categories of tasks:

- first-priority setup tasks that are necessary to protect the integrity of your system. You should perform these tasks first, as soon as possible after you have completed your installation.
- standard setup tasks that enable the users in your organization to begin using SAS Intelligence Platform client applications to access and analyze your data.
- optional setup tasks that you might want to perform to address specific requirements in your environment.
- ongoing tasks to keep the SAS Intelligence Platform operational.

## First-Priority Setup Tasks

### Summary of First-Priority Setup Tasks

The following tasks are necessary to protect the integrity of your system. Complete these steps as soon as possible after installation, before you complete any of the other tasks that are outlined in this chapter.

**Table 4.1** First-Priority Setup Tasks

Task	Description
Secure the SAS configuration on each server machine.	<p>For a secure deployment, the configuration directory on each server machine must be protected by operating system controls. These controls will prevent inappropriate access to repository data sets, server scripts, server logs, and configuration files.</p> <p>On Windows systems, all configuration directories, files, and scripts are owned by the user who performs the installation. You must update the permissions as shown in “Recommended Operating System Protections for Windows Machines” on page 40. These recommendations assume that your SAS servers and spawners run as services under the Local System account.</p> <p>On UNIX and z/OS systems, the SAS Deployment Wizard automatically applies the appropriate permissions. The default permissions are shown in “Default Operating System Protections for UNIX and z/OS Machines” on page 42.</p>
Establish a formal, regularly scheduled backup process.	<p>Establish a formal, regularly scheduled backup process that includes your metadata repositories as well as the associated physical files.</p> <p>SAS provides backup and restore utilities that enable you to perform correct backups and restores of your metadata repositories, the repository manager, the metadata journal file, and the metadata server configuration files while minimizing disruptions in service. It is important to also back up the physical data that is associated with the metadata so that related information will be synchronized if a restore becomes necessary.</p> <p>Before you back up your SAS Intelligence Platform, read Chapter 9, “Best Practices for Backing Up and Restoring Your System,” on page 123.</p>

### Recommended Operating System Protections for Windows Machines

On Windows server machines, we recommend that you apply the following operating system protections to your configuration directory. All of these directories are located in *SAS-configuration-directory\Lev1*.

**Table 4.2** Recommended Operating System Protections on Windows

Directories	Users	Recommended Permissions
<i>SAS-configuration-directory</i>	SYSTEM and Administrators	Full Control
<i>SAS-configuration-directory\Lev1</i>		
<b>Lev1</b> subdirectories: <b>Documents</b> , <b>ReportBatch</b> , <b>SASApp</b> , <b>SASMeta</b> , <b>Utilities</b> , <b>Web</b>	All other users	List Folder Contents, Read
<b>Lev1</b> subdirectories: <b>ConnectSpawner</b> , <b>Logs</b> , <b>ObjectSpawner</b> , <b>SASApp\OLAPServer</b> , <b>SASMeta\MetadataServer</b> , <b>SASTS</b> , <b>ShareServer</b>	SYSTEM and Administrators	Full Control Remove all other users and groups
<b>SASApp</b> subdirectories : <b>PooledWorkspaceServer</b> , <b>StoredProcessServer</b>	SYSTEM, Administrators, and SAS Spawned Servers (sassrv)	Full Control Remove all other users and groups
<b>SASApp</b> subdirectories: <b>ConnectServer\Logs</b> , <b>Data\wrsdist</b> , <b>Data\wrstemp</b> , <b>PooledWorkspaceServer\Logs</b> , <b>PooledWorkspaceServer\sasuser</b> , <b>StoredProcessServer\Logs</b> , <b>StoredProcessServer\sasuser</b> , and <b>WorkspaceServer\Logs</b>	SYSTEM, Administrators, and SAS Spawned Servers (sassrv)	Full Control
<b>SASMeta\WorkspaceServer\Logs</b>		
<i>sasv9_meta.cfg</i> file	SYSTEM and Administrators	Read and Write Remove all other users and groups
<b>SASMeta</b> subdirectories: <b>MetadataServer</b> , <b>MetadataServer\rposmgr</b> , <b>MetadataServer\MetadataRepositories\Foundation</b>	The user who backs up the metadata server. You can add this user to the Administrators	Full Control
Backup destination (for example, <b>SASMeta\MetadataServer\SASBackup</b> )	group to provide the required access.	

*Note:*

- These recommendations assume that your SAS servers and spawners run as services under the Local System account. If servers and spawners are run under a different account, then grant that account the permissions that are recommended for SYSTEM.
- You might have selected the custom installation option to place all of your log files in a single directory. If you selected this option, then you will need to grant the SAS Spawned Servers (sassrv) user Full Control of the central log destination (for example, *SAS-configuration-directory\Lev1\Logs*).
- If users will be using SAS Enterprise Guide to create stored processes, then the SAS Spawned Servers (sassrv) account must have Write access to the directory in which stored processes will be stored.
- If you enable logging for a workspace server, then you will need to grant all users of the workspace server Full Control of the log directory. (See “Create a Log File for Workspace Server Troubleshooting” on page 112).

△

For information about backups, see Chapter 10, “Using the Backup Wizard in SAS Management Console,” on page 137 and Chapter 11, “Using the %OMABAKUP Macro to Perform Backups and Restores,” on page 147.

For details about the configuration directory structure, see “Overview of the Configuration Directory Structure” on page 19.

## Default Operating System Protections for UNIX and z/OS Machines

The following table shows the default operating system protections that are provided automatically for configuration directories on UNIX and z/OS machines. All of these directories are located in *SAS-configuration-directory/Lev1*.

**Table 4.3** Default Operating System Protections on UNIX and z/OS

Directories	Users	Default Permissions
<i>SAS-configuration-directory</i>	SAS Installer	Read, Write, and Execute
<i>SAS-configuration-directory/Lev1</i>	All other users	Read and Execute
<b>Lev1 subdirectories: Documents, ReportBatch, SASApp, SASMeta, Utilities, Web</b>		
<b>Lev1 subdirectories: ConnectSpawner, Logs, ObjectSpawner, SASApp/OLAPServer, SASMeta/MetadataServer, SASTS, ShareServer</b>	SAS Installer	Read, Write, and Execute
	All other users	No access
<b>SASApp subdirectories : PooledWorkspaceServer, StoredProcessServer</b>	SAS Installer	Read, Write, and Execute
	sas group	Read and Execute
<b>SASApp subdirectories : ConnectServer/Logs, Data/wrsdist, Data/wrstemp, PooledWorkspaceServer/Logs, PooledWorkspaceServer/sasuser, StoredProcessServer/Logs , StoredProcessServer/sasuser, and WorkspaceServer/Logs</b>	SAS Installer	Read, Write, and Execute
	sas group	Read, Write, and Execute
<b>SASMeta/WorkspaceServer/Logs</b>		
<i>sasv9_meta.cfg</i> file	SAS Installer	Read and Write
	All other users	no access

*Note:*

- Make sure that the SAS Spawnd Servers account (sassrv) is a member of the sas group, which has the necessary permissions to server configuration files and log directories.
- You might have selected the custom installation option to place all of your log files in a single directory. If you selected this option, then you will need to grant either the sas group or the SAS Spawnd Servers (sassrv) user Read, Write, and Execute permission on the central log destination (for example, *SAS-configuration-directory/Lev1/Logs*).



- If users will be using SAS Enterprise Guide to create stored processes, then the SAS Spawmed Servers (sassrv) account must have Write access to the directory in which stored processes will be stored.
- If you enable logging for a workspace server, then you will need to grant all users of the workspace server Read, Write, and Execute permission on the log directory. (See “Create a Log File for Workspace Server Troubleshooting” on page 112).
- The user who backs up the metadata server must have full access to *SAS-configuration-directory/Lev1/SASMeta/MetadataServer*, to its subdirectories **rposmgr** and **MetadataRepositories/Foundation**, and to the backup destination (for example, *SAS-configuration-directory/Lev1/SASMeta/MetadataServer/SASBackup*). The SAS Installer user has the required access.

**CAUTION:**

**Do not run a backup or a restore as the Root user. Doing so will change ownership of the metadata server files.**   △

For more information, see Chapter 10, “Using the Backup Wizard in SAS Management Console,” on page 137 and Chapter 11, “Using the %OMABAKUP Macro to Perform Backups and Restores,” on page 147.

- For details about the configuration directory structure, see “Overview of the Configuration Directory Structure” on page 19.

△

---

## Standard Setup Tasks

The following administration tasks are essential to enable the users in your organization to begin using SAS Intelligence Platform client applications to access and analyze your data.

**Table 4.4** Standard Setup Tasks

Task	Description
Add users and manage access.	<p>Standard security setup activities include the following:</p> <ul style="list-style-type: none"> <li>□ creating SAS administrators and regular SAS users</li> <li>□ managing access to metadata, data, and application functionality</li> </ul> <p>For details, see “Security Tasks” in the <i>SAS Intelligence Platform: Security Administration Guide</i>.</p>
Establish connectivity to your data sources.	<p>To enable the client applications in the SAS Intelligence Platform to access your data sources (including SAS data sets, third-party relational databases, and data from Enterprise Resource Planning systems), you must create metadata objects that represent your data. For example, to enable users to access data in a library of SAS data sets, you must define metadata objects that represent the SAS library and each of the tables in the library. For details, see “Connecting to Common Data Sources” in the <i>SAS Intelligence Platform: Data Administration Guide</i>.</p>
Set up your metadata folder structure.	<p>SAS Intelligence Platform clients use a hierarchy of SAS folders to store metadata for content such as libraries, tables, OLAP schemas, jobs, information maps, and reports. The initial structure provides private folders for individual users and provides a separate area for shared data. Within these folders, you should create a customized folder structure that meets your specific needs. For details, see Chapter 16, “Working with SAS Folders,” on page 197.</p> <p><i>Note:</i> Be sure to secure access to the folders as described in the <i>SAS Intelligence Platform: Security Administration Guide</i>. △</p>

## Optional Setup Tasks

It might be necessary for you to modify your initial configuration to meet specific requirements in your environment. Optional administration and configuration tasks include the following.

**Table 4.5** Optional Setup Tasks

Task	Description
Optimize the performance and reliability of the metadata server.	<p>The following are some steps that you might need to take to optimize the performance and reliability of the metadata server:</p> <ul style="list-style-type: none"> <li>□ Move the metadata server journal file to another file system (see “Move the Journal File to Another File System” on page 171).</li> <li>□ Adjust the maximum number of threads used by the metadata server (see “Configuring the Number of Threads Used by the Metadata Server” on page 167).</li> </ul> <p>For other performance-related tasks, see Chapter 14, “Managing Metadata Server Performance,” on page 173.</p>
Modify the configuration of your processing servers.	<p>The following are some server configuration changes that you might want to make for performance optimization or other reasons:</p> <ul style="list-style-type: none"> <li>□ 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. For details, see “Understanding Server Load Balancing” in the <i>SAS Intelligence Platform: Application Server Administration Guide</i>.</li> <li>□ Modify load balancing or algorithms for stored process servers and pooled workspace servers. For details, see “Modify Logical Pooled Workspace or Logical Stored Process Server Load-Balancing Properties” in the <i>SAS Intelligence Platform: Application Server Administration Guide</i>.</li> <li>□ If your SAS server metadata contains characters other than those typically found in the English language, then you must start your SAS server with an ENCODING= or LOCALE= system option that accommodates those characters. For details, see “Encoding and Locale Information” in the <i>SAS Intelligence Platform: Application Server Administration Guide</i>.</li> <li>□ If you will be submitting large jobs (for example, JAVA GRAPH jobs) on z/OS servers, then you might need to specify a larger region size. For details, see “Managing Memory” in the <i>SAS Companion for z/OS</i>.</li> </ul>
Optimize Web application performance.	For information about configuring middle-tier components for better efficiency and performance, see “Best Practices for Configuring Your Middle Tier” in the <i>SAS Intelligence Platform: Web Application Administration Guide</i> .
Adjust server logging.	You can change your server configurations to obtain additional logging information for troubleshooting, performance management, or other purposes. For details, see Chapter 8, “Administering Logging for SAS Servers,” on page 103 and <i>SAS Logging: Configuration and Programming Reference</i> .
Enable job and report scheduling.	To enable SAS scheduling features, you can install and configure scheduling software, configure operating system scheduling, or configure in-process scheduling. For details, see <i>Scheduling in SAS</i> .

Task	Description
Increase Java heap memory allocation for desktop applications.	To process large amounts of data, you might need to modify the Java heap memory allocation that is configured for SAS Management Console, SAS Data Integration Studio, SAS OLAP Cube Studio, and SAS Information Map Studio. For details, see “Managing the Java Heap Used By Desktop Applications” in the <i>SAS Intelligence Platform: Desktop Application Administration Guide</i> .
Set up change management for SAS Data Integration Studio users.	If you want to implement change management for SAS Data Integration Studio, then you must set up project repositories and at least one change-managed folder. For details, see “Administering Data Integration Studio” in the <i>SAS Intelligence Platform: Desktop Application Administration Guide</i> .
Collect ARM log information for SAS Data Integration Studio batch jobs.	If you want to collect ARM log information for SAS Data Integration Studio jobs that are run on a batch basis, then you must enable logging for the batch server that executes the jobs. For details, see .

## Ongoing System Administration Tasks

The following table lists important system administration tasks that you need to perform on an ongoing basis to keep the SAS Intelligence Platform operational.

**Table 4.6** Ongoing System Administration Tasks

Task	Documentation Reference
Start, stop, pause, resume, and refresh the servers that are used in the system.	Chapter 5, “Operating Your Servers,” on page 51
Check the status of a server or of a metadata repository.	Chapter 6, “Checking the Status of Servers,” on page 75
Monitor the activity of servers.	Chapter 7, “Monitoring the Activity of SAS Servers,” on page 87
Perform regular full backups of the metadata server.	Chapter 9, “Best Practices for Backing Up and Restoring Your System,” on page 123
Use server logs and configure logging options.	Chapter 8, “Administering Logging for SAS Servers,” on page 103 and <i>SAS Logging: Configuration and Programming Reference</i>
Create a new metadata repository.	Chapter 15, “Creating, Registering, Moving, Copying, Renaming, and Deleting SAS Metadata Repositories,” on page 181
Promote individual metadata objects or groups of objects.	Chapter 17, “Using the Promotion Tools,” on page 213

*Note:* The preceding table includes only the tasks that are documented in this guide. Other important administration tasks are described in the following documents:

- *SAS Intelligence Platform: Security Administration Guide.* In this guide, you can use the “Checklists for a More Secure Deployment” to ensure that your environment remains secure on an ongoing basis.
- *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.*
- *Scheduling in SAS.*
- *SAS/SHARE User’s Guide.*
- *SAS/CONNECT User’s Guide.*

To access these documents, go to <http://support.sas.com/92administration>. △

---

## Best Practices for Ensuring the Integrity of Your System

---

### Allow Only the Metadata Server to Access Your Metadata Repositories

The **MetadataRepositories** and **rpsomgr** subdirectories, which are located in the SAS configuration directory, are critical to the operation of your system. These data sets contain metadata that defines your servers, users, access levels, enterprise data sources, and data structures. They also contain metadata for resources that are created and used by SAS applications, including information maps, OLAP cubes, report definitions, stored process definitions, and jobs.

To safeguard the integrity of your system:

- Never move, delete, modify, or directly read the data sets in the **MetadataRepositories** and **rpsomgr** directories. These data sets should be accessed only by metadata server processes.
- Do not place other files in these directories.
- Back up the contents of these directories regularly, as described in Chapter 9, “Best Practices for Backing Up and Restoring Your System,” on page 123.

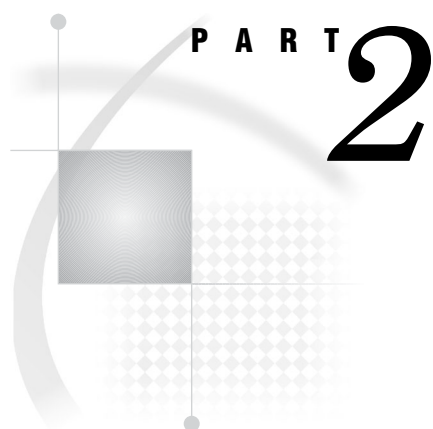
---

### Use Best Practices for Working with SAS Folders

The folders that appear in the **Folders** tab of SAS Management console are used by client applications to store business intelligence content and system information. Inappropriate renaming, deleting, or moving of these folders or their contents could cause client applications to malfunction.

When interacting with folders in SAS Management Console, be sure to follow the best practices that are provided in Chapter 16, “Working with SAS Folders,” on page 197. If you need to move or copy the contents of these folders, use the procedures that are provided in Chapter 17, “Using the Promotion Tools,” on page 213.





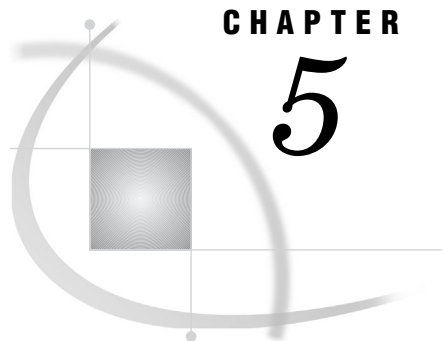
## **Starting, Stopping, and Checking the Status of Servers**

*Chapter 5*.....**Operating Your Servers** 51

*Chapter 6*.....**Checking the Status of Servers** 75







## CHAPTER

## 5

# Operating Your Servers

<i>Overview of Server Operation</i>	<b>52</b>
<i>About Required Servers</i>	<b>52</b>
<i>Starting Servers in the Correct Order</i>	<b>52</b>
<i>About Server Start-up Parameters</i>	<b>53</b>
<i>About SAS Metadata Server States</i>	<b>53</b>
<i>Summary of Procedures for Server Operation</i>	<b>54</b>
<i>Running Servers as Windows Services</i>	<b>55</b>
<i>Starting and Stopping SAS Servers on z/OS</i>	<b>57</b>
<i>Using the sas.servers Script on UNIX or z/OS to Start or Stop All Servers</i>	<b>58</b>
<i>About the sas.servers Script</i>	<b>58</b>
<i>Use the sas.servers Script to Start, Stop, or Restart All Servers or to Display the Status of Servers</i>	<b>58</b>
<i>Using generate_boot_scripts.sh to Regenerate the sas.servers Script</i>	<b>59</b>
<i>About generate_boot_scripts.sh</i>	<b>59</b>
<i>About the server-name.srv Files</i>	<b>60</b>
<i>Regenerate the sas.servers Script</i>	<b>60</b>
<i>Install sas.servers as a Boot Script</i>	<b>60</b>
<i>Running Servers as UNIX Daemons</i>	<b>61</b>
<i>Using Scripts to Operate SAS Servers Individually</i>	<b>61</b>
<i>About Using Scripts to Operate Servers</i>	<b>61</b>
<i>Use Individual Scripts to Operate the SAS Metadata Server, SAS Object Spawner, SAS OLAP Server, and SAS Table Server</i>	<b>61</b>
<i>Use Individual Scripts to Operate Servers and Spawners</i>	<b>63</b>
<i>Starting or Stopping the SAS Services Application</i>	<b>64</b>
<i>Using SAS Management Console to Operate SAS Servers</i>	<b>65</b>
<i>Obtain the Appropriate User Credentials, and Clear Cached Credentials (If Necessary)</i>	<b>65</b>
<i>Use SAS Management Console to Stop, Pause, Resume, or Reset the SAS Metadata Server</i>	<b>65</b>
<i>Use SAS Management Console to Stop, Pause, Resume, Quiesce, or Refresh the SAS Object Spawner</i>	<b>66</b>
<i>Use SAS Management Console to Stop, Pause, Resume, Quiesce, or Refresh the SAS OLAP Server</i>	<b>67</b>
<i>Use SAS Management Console to Control Processes on Servers</i>	<b>68</b>
<i>Starting and Stopping Web Application Servers</i>	<b>69</b>
<i>Starting or Stopping a JBoss Application Server</i>	<b>69</b>
<i>Starting or Stopping an IBM WebSphere Application Server or an IBM WebSphere Application Server Network Deployment (ND)</i>	<b>70</b>
<i>Starting or Stopping an IBM WebSphere Application Server or an IBM WebSphere Application Server Network Deployment (ND) on Windows or UNIX</i>	<b>70</b>
<i>Starting or Stopping an IBM WebSphere Application Server Network Deployment (ND) on z/OS</i>	<b>71</b>
<i>Starting or Stopping an Oracle WebLogic Server</i>	<b>72</b>

## Overview of Server Operation

### About Required Servers

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

- a SAS Metadata Server
- the SAS object spawner, which acts as a listener for SAS Workspace Servers, SAS Pooled Workspace Servers, and SAS Stored Process Servers

Depending on which SAS products you have installed, one or more instances of the following additional components might also be required to be running on network-accessible machines: SAS OLAP Server, SAS Table Server, SAS Services Application (also referred to as Remote Services), SAS/CONNECT spawner, SAS/SHARE server, SAS Content Server, and a Web application server.

If you want to use the SAS Deployment Tester utility, then the SAS Deployment Tester server must also be running.

### Starting Servers in the Correct Order

Because of dependencies, it is important to start the servers in the correct order. Server dependencies are shown in the following table:

Server	Server Dependencies
SAS Metadata Server	None
SAS OLAP Server	SAS Metadata Server
SAS object spawner	SAS Metadata Server
SAS Table Server	SAS Metadata Server
SAS Services Application (Remote Services)	SAS Metadata Server
Web application server	SAS Services Application (Remote Services)
SAS/CONNECT spawner	SAS Metadata server
SAS/SHARE server	SAS Metadata server
SAS Deployment Tester server	None

The following start-up order meets the dependency requirements:

- 1 Start the SAS Metadata Server.
- 2 Start the SAS OLAP Server.
- 3 Start the SAS object spawner.
- 4 Start the SAS/SHARE server
- 5 Start the SAS/CONNECT spawner.
- 6 Start the SAS Table Server.
- 7 Start the SAS Services Application (Remote Services).

- 8 Start the SAS Deployment Tester server.
- 9 Start the Web application server. (The SAS Content Server starts automatically when the application server is started.)

Servers should be stopped in the inverse order, with the metadata server stopped last.

---

## 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 “Reference: Configuration Files for SAS Servers” on page 271.

For detailed information about start-up parameters for each server, see the documentation references in the table in “Summary of Procedures for Server Operation” on page 54.

---

## About SAS Metadata Server States

A SAS Metadata Server can exist in any of the following states:

### Online

The metadata server is available to accept client requests from authenticated users.

External systems such as Windows Services Manager report the Online state as **running** or **started**.

### Offline

The metadata server is paused and is not available to accept client requests. When the metadata server is paused to Offline status, the server does the following:

- retains all active client connections.
- temporarily changes the access state of all repositories to Offline.
- 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.
- 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.

Utilities such as the Backup Wizard automatically pause the metadata server to the Offline state because they require repository containers to be closed. The Offline state is also useful if you need to disable users’ access while you perform maintenance on the metadata server machine.

External systems such as Windows Services Manager report the Offline state as **paused**.

### Administration

The metadata server is paused and will accept requests only from an unrestricted administrative user. When the metadata server is paused to Administration status, the server does the following:

- retains all active client connections.
- temporarily stops receiving client requests to read or write metadata, except for requests from users in the **Metadata Server: Unrestricted** role. (For information about this role, see “About User Roles for System

Administration” on page 24). If the user is not unrestricted, then the client displays a message indicating that the server is paused.

The Administration state is useful when you need to disable users’ access while you connect to the metadata server to perform administrative tasks such as the following:

- run diagnostic and repair tools
- upgrade metadata to a new version
- promote metadata and related content
- modify metadata for components of a SAS Application Server

External systems such as Windows Services Manager report the Administration state as **paused**.

#### Stopped

The Stopped state terminates the SAS Metadata Server process and breaks all client connections. The metadata server, and all SAS servers that depend on it, must be started before clients can connect again.

---

## Summary of Procedures for Server Operation

To operate servers in SAS Management Console, you must have the appropriate user credentials. To determine what credentials are required for each task, see Chapter 3, “Who Can Do What: Credential Requirements for SAS Management Console Tasks,” on page 33.

The following table summarizes the server operation methods for each server. For detailed information about startup parameters for each server, see the documentation references that are provided in the last column.

Server Type	Available Methods for Server Operation	Documentation Location for Startup Parameters
SAS Metadata Server, SAS OLAP Server, and SAS object spawner	<ul style="list-style-type: none"> <li>□ Windows service<sup>1</sup></li> <li>□ z/OS started task<sup>2</sup></li> <li>□ sas.servers script<sup>3</sup> (UNIX or z/OS)</li> <li>□ Individual server script (or Windows shortcut)</li> <li>□ SAS Management Console (except startup)</li> </ul>	<p>Appendix 5, “Options for the Metadata Server Invocation Command,” on page 289</p> <p>“System Options for Application Server Components” in the <i>SAS Intelligence Platform: Application Server Administration Guide</i></p> <p>“Spawner Invocation Options” in the <i>SAS Intelligence Platform: Application Server Administration Guide</i></p>
SAS Workspace Server, SAS Pooled Workspace Server, and SAS Stored Process Server	<ul style="list-style-type: none"> <li>□ Started by the object spawner</li> <li>□ SAS Management Console (except startup)</li> </ul>	<p>“System Options for Application Server Components” in the <i>SAS Intelligence Platform: Application Server Administration Guide</i></p>

---

Server Type	Available Methods for Server Operation	Documentation Location for Startup Parameters
SAS Table Server	<input type="checkbox"/> Windows service <sup>1</sup> <input type="checkbox"/> z/OS started task <sup>2</sup> <input type="checkbox"/> sas.servers script <sup>3</sup> (UNIX or z/OS) <input type="checkbox"/> Individual server script (or Windows shortcut) <input type="checkbox"/> SAS Management Console	
SAS Services Application (Remote Services)	<input type="checkbox"/> Windows service <sup>1</sup> <input type="checkbox"/> z/OS started task <sup>2</sup> <input type="checkbox"/> sas.servers script <sup>3</sup> (UNIX or z/OS) <input type="checkbox"/> Remote Services script (or Windows shortcut)	<i>SAS Intelligence Platform: Web Application Administration Guide</i>
Web application server	<input type="checkbox"/> Server console or script <input type="checkbox"/> Windows service (JBoss and Oracle WebLogic only) <sup>1</sup>	<i>SAS Intelligence Platform: Web Application Administration Guide</i>
SAS Content Server	Starts automatically when the Web application server is started	<i>SAS Intelligence Platform: Web Application Administration Guide</i>
SAS/CONNECT spawner, SAS/SHARE server, and SAS Deployment Tester server	<input type="checkbox"/> Windows service <sup>1</sup> <input type="checkbox"/> z/OS started task <sup>2</sup> <input type="checkbox"/> sas.servers script <sup>3</sup> (UNIX or z/OS) <input type="checkbox"/> Script (or Windows shortcut)	<i>Communications Access Methods for SAS/CONNECT and SAS/SHARE</i> <i>SAS/SHARE User's Guide</i>

1 On Windows systems, this is the recommended method.

2 On z/OS systems, this is the recommended method.

3 On UNIX systems, this is the recommended method.

For details about the server operation procedures that are summarized in this table, see the remaining topics in this chapter.

## Running Servers as Windows Services

On Windows systems, depending on the options that were selected during the installation process, the following might be installed as services: the SAS/CONNECT spawner, the SAS Metadata Server, the SAS object spawner, the SAS OLAP Server, the SAS Services Application (Remote Services), the SAS Table Server, the SAS/SHARE server, and the SAS Deployment Tester server. This installation option is highly recommended. If these servers have been installed as services, then they start automatically when you restart the machines. To ensure that services are always started in the correct order, the required dependencies are built into the services.

The JBoss Application Server also might be installed as a service. This service is set up to be started manually.

The third maintenance release for SAS 9.2 includes scripts for installing the Oracle WebLogic Managed Server and WebLogic Administration Server as Windows services. You can install these services by following instructions that are provided in the `Instructions.html` file, which is located in `SAS-configuration-directory/Levn/Documents`. After you install the services, they will start up automatically when you restart the machine.

For each service, the following table shows the built-in dependencies and the actions that you can take from within Windows Services Manager:

Server	Dependencies	Available Net Commands
SAS Metadata Server	None	Start, stop, pause, and continue
SAS OLAP Server	Metadata server	Start, stop, pause, and continue
SAS object spawner	Metadata server	Start, stop, pause, and continue
SAS Services Application (Remote Services)	Metadata server	Start and stop
SAS Table Server	Metadata server	Start, stop, pause, and continue
SAS/CONNECT spawner	Metadata server	Start and stop
SAS/SHARE server	Metadata server	Start and stop
SAS Deployment Tester server	None	Start and stop
JBoss Application Server	SAS Services Application (Remote Services)	Start and stop
Oracle WebLogic Administration Server	None	Start and stop
Oracle WebLogic Managed Server	Oracle WebLogic Administration Server	Start and stop

The service names have the following format:

**SAS** [*deployment-name-and-level*] <*server-context* -> *server-name*

For example, your services might be displayed in Windows Services Manager as follows:

```
SAS [Config-Levl] Deployment Tester Server
SAS [Config-Levl] Object Spawner
SAS [Config-Levl] Remote Services
SAS [Config-Levl] SASApp - OLAP Server
SAS [Config-Levl] SASMeta - Metadata Server
SAS [Config-Levl] SASSTS - Table Server
```

If you installed the JBoss Application Server as a service, the service is displayed as **JBoss - SASServer1**. If you installed the WebLogic Administration Server and WebLogic Managed Server as services, the service names consist of the domain name, the server name, and the configuration directory path, as in the following examples: **SASDomain AdminServer (sas.Config.Lev1)** and **SASDomain SASServer1 (sas.Config.Lev1)**.

SAS provides batch scripts that you can use to manage these services from the command line (see “Using Scripts to Operate SAS Servers Individually” on page 61). You can also use the Windows Net command, which has the following syntax:

```
net start|stop|pause|continue "service-name"
```

For example, if the service name for the object spawner is **SAS [Config-Lev1] Object Spawner**:

- ☐ You would use the following command to stop the service:  
**net stop "SAS [Config-Lev1] Object Spawner"**
- ☐ You would use the following command to start the service:  
**net start "SAS [Config-Lev1] Object Spawner"**

---

## Starting and Stopping SAS Servers on z/OS

On z/OS systems, the SAS Deployment Wizard initially starts the SAS servers under UNIX System Services (USS) by using scripts that are provided for each server. After the initial installation and configuration has been completed and validated, SAS recommends that you stop the servers and restart them as started tasks.

The SAS Deployment Wizard creates started task definitions for each server. The definitions are stored in *SAS-installation-location*.W0.SRVPROC. To deploy the started tasks, see the instructions.html document that is created by the SAS Deployment Wizard.

After the started tasks are deployed, you can operate your servers as follows:

- 1 Log on as the SAS Installer user (for example, sas).
- 2 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.
- ☐ Stopping the metadata server stops other components, including object spawners, OLAP servers, table servers, SAS/CONNECT spawners, and SAS/SHARE servers.

$\Delta$

- 3 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:

```
SAH011001I SAS (9.2) Metadata Server <(port-number)>, State, starting
SAH011999I SAS (9.2) Metadata Server <(port-number)>, State, running
SAH019001I SAS (9.2) Metadata Server <(port-number)>, State, stopping
SAH019999I SAS (9.2) Metadata Server <(port-number)>, State, stopped
```

When servers are running under USS, you can start and stop the servers as follows:

- ☐ by using individual scripts that are provided for each server. See “Using Scripts to Operate SAS Servers Individually” on page 61.
- ☐ by using the sas.servers script. See “Using the sas.servers Script on UNIX or z/OS to Start or Stop All Servers” on page 58.

These methods are not recommended for production environments.

For more information about using SAS on z/OS, see the *SAS Companion for z/OS*.

---

## Using the sas.servers Script on UNIX or z/OS to Start or Stop All Servers

---

### About the sas.servers Script

During installation, the SAS Deployment Wizard creates a sas.servers script on each UNIX and z/OS server tier machine. You can perform the following tasks with this script:

- ☐ start, stop, or restart all of the SAS servers and spawners on the machine in the correct order by using a single command
- ☐ display the status of all the SAS servers and spawners on the machine by using a single command

*Note:* To use the script on z/OS systems, you must run the servers under UNIX System Services (USS). On z/OS systems, SAS recommends operating servers as started tasks. See “Starting and Stopping SAS Servers on z/OS” on page 57.  $\triangle$

The script performs the preceding tasks for all instances of the following servers, spawners, and services that are installed on the machine:

- ☐ SAS Metadata Server
- ☐ SAS object spawner
- ☐ SAS OLAP Server
- ☐ SAS/CONNECT spawner
- ☐ SAS/SHARE server
- ☐ SAS Table Server
- ☐ SAS Services Application (Remote Services)
- ☐ SAS Deployment Tester server

The sas.servers script is located in the top level of your configuration directory. For example, in a Level 1 deployment, the script is located in the following path:

*SAS-configuration-directory/Lev1/.*

#### CAUTION:

**You should not directly update the sas.servers script.** If the script needs to be updated (for example, to add new servers or remove servers), then regenerate the script by using generate\_boot\_scripts.sh.  $\triangle$

---

### Use the sas.servers Script to Start, Stop, or Restart All Servers or to Display the Status of Servers

To use the sas.servers script to start, stop, or restart all servers on a UNIX or z/OS machine or to display the status of all servers on the machine, follow these steps:

- 1 Log on as the SAS Installer user (for example, sas).
- 2 Go to the directory *SAS-configuration-directory/Lev1/*.
- 3 Type **./sas.servers** followed by one of the following arguments:

#### **start**

starts the servers and spawners in the following order: SAS Metadata Server, SAS OLAP Server, SAS object spawner, SAS/SHARE server, SAS/CONNECT spawner, SAS Table Server, SAS Remote Services, SAS Deployment Tester server.



*Note:* The OLAP server, object spawner, SAS/SHARE server, SAS/CONNECT spawner, and table server require that the metadata server be running. If the metadata server is configured on a different host machine, make sure that the server is running before you use this script. △

#### **stop**

stops the servers and spawners in the following order: SAS Deployment Tester server, SAS Remote Services, SAS Table Server, SAS/CONNECT spawner, SAS/SHARE server, SAS object spawner, SAS OLAP Server, SAS Metadata Server.

*Note:*

- When you stop the metadata server, any dependent servers and spawners (whether they are on the same host machine or a different machine) will stop running.
- Before you stop an 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**.

△

#### **restart**

stops and then starts the servers and spawners in the following order: SAS Metadata Server, SAS OLAP Server, SAS object spawner, SAS/SHARE server, SAS/CONNECT spawner, SAS Table Server, SAS Remote Services.

#### **status**

displays the current status of the servers and spawners in the following format:

*server-name server-instance (process-ID) is running|is stopped.*

For example, the following messages mean that two instances of the SAS OLAP Server are configured on the machine. The first instance is running, and the second instance is stopped:

```
SAS OLAP Server 1 (pid 29529) is running
SAS OLAP Server 2 (pid 29530) is stopped
```

*Note:* To use the script on z/OS systems, you must run the servers under UNIX System Services (USS). On z/OS systems, SAS recommends operating servers as started tasks. See “Starting and Stopping SAS Servers on z/OS” on page 57. △

---

## Using generate\_boot\_scripts.sh to Regenerate the sas.servers Script

### About generate\_boot\_scripts.sh

The generate\_boot\_scripts.sh script enables you to automatically regenerate the sas.servers script for a particular UNIX or z/OS host machine. This feature is useful in the following types of situations:

- You have configured a new server or spawner on the machine. In this situation, the script generation process detects the new server or spawner and adds it to the sas.servers script.
- You have removed a server or spawner configuration from the machine. In this situation, the script generation process detects the absence of the server or spawner and removes it from the sas.servers script.
- You want to remove a server or spawner from the script so that you can operate it separately, or you want to add a server or spawner that you previously removed.
- You have changed the location of the server's log.

## About the server-name.srv Files

On UNIX and z/OS systems, each server's configuration directory contains a file called *server-name.srv*. For example, the metadata server's configuration directory contains a file called *MetadataServer.srv*, and the workspace server's configuration directory contains a file called *WorkspaceServer.srv*.

The *generate\_boot\_scripts.sh* script accesses each of the *server-name.srv* files to determine which servers and spawners are to be included in the sas.servers script. If a server's *server-name.srv* file contains the keyword **ENABLE**, then the server is included in the script. If the file contains the keyword **DISABLE**, then the server is excluded from the script.

In the initial configuration, all *server-name.srv* files contain the keyword **ENABLE**.

The *server-name.srv* files also contain the path to the server's startup script and the path to the server's log files.

## Regenerate the sas.servers Script

To regenerate the sas.servers script, follow these steps:

- 1 Log on as the SAS Installer user (for example, sas).
- 2 If you want to remove a server or spawner from the script, or if you want to add a server or spawner that you previously removed, follow these steps:
  - a Navigate to the server's configuration directory and open the file *server-name.srv* in an editor.  
 For example, if you want to remove the SAS/SHARE server from the script, you would go to *SAS-configuration-directory/Lev1/ShareServer* and open the file *ShareServer.srv*.
  - b Locate the line that contains the keyword **ENABLE** or **DISABLE**. If you want to exclude the server from the sas.servers script, change the keyword to **ENABLE**. If you want to include the server in the sas.servers script, change the keyword to **DISABLE**. Save the file.
- 3 Go to the directory *SAS-configuration-directory/Lev1/*.
- 4 Enter the following command: **./generate\_boot\_scripts.sh**

A new sas.servers script is created to reflect the servers that are currently configured on the machine. If any of the *server-name.srv* files contain the keyword **DISABLE**, the following message appears:

```
<number-of-disabled-servers> SAS servers were explicitly disabled in
their .srv files
```

---

## Install sas.servers as a Boot Script

On UNIX machines, you can choose to install the sas.servers script as a boot script so that the servers will start automatically when you start the machine. The comments in

the `sas.servers` script contain installation instructions that are tailored to your operating environment.

---

## Running Servers as UNIX Daemons

On UNIX systems, as an alternative to using the `sas.servers` script, you can 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 machine. 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.

---

## Using Scripts to Operate SAS Servers Individually

---

### About Using Scripts to Operate Servers

The configuration directory for each SAS server and spawner contains a script that enables you to operate the server or spawner individually from a command line.

On Windows machines, depending on the options that were selected during installation, the SAS Deployment Wizard might have also added shortcuts to your **Start** menu to enable you to start, stop, pause and resume servers and spawners.

On UNIX and z/OS systems, you can use the `sas.servers` script to start and stop servers with a single command (see “Using the `sas.servers` Script on UNIX or z/OS to Start or Stop All Servers” on page 58). On z/OS systems, SAS recommends operating servers as started tasks (see “Starting and Stopping SAS Servers on z/OS” on page 57).

For information about the startup parameters for each server, see the documentation references that are provided in “Summary of Procedures for Server Operation” on page 54.

---

### Use Individual Scripts to Operate the SAS Metadata Server, SAS Object Spawner, SAS OLAP Server, and SAS Table Server

You can use scripts to start, stop, pause, or resume the metadata server, an object spawner, an OLAP server, or a table server from the DOS command line, the UNIX console, or the z/OS console. The scripts also enable you to install or remove a server or spawner as a Windows service and to display the server or spawner's current status.

*Note:*

- The metadata server must be running before you can start an object spawner, an OLAP server, or a table server.
- Before using scripts to operate the metadata server, see “About SAS Metadata Server States” on page 53.

△

Follow these steps:

- 1 On Windows, log on as a user who is a member of the host operating system's Administrators group. On UNIX or z/OS, log on as the SAS Installer user (for example, sas).
- 2 Use one of the following scripts, which are located in the path *SAS-configuration-directory/Lev1*:

**Table 5.1** Scripts for Operating Servers and Spawners

Server or Spawner	Script Name and Location
SAS Metadata Server	Windows: <b>SASMeta\MetaDataServer\MetaDataServer.bat</b> <sup>1,2</sup> UNIX and z/OS: <b>SASMeta/MetaDataServer/MetaDataServer.sh</b>
SAS object spawner	Windows: <b>ObjectSpawner\ObjectSpawner.bat</b> UNIX and z/OS: <b>ObjectSpawner/ObjectSpawner.sh</b>
SAS OLAP Server	Windows: <b>SASApp\OLAPServer\OLAPServer.bat</b> <sup>1,2</sup> UNIX and z/OS: <b>SASApp/OLAPServer/OLAPServer.sh</b> <sup>1,2</sup>
SAS Table Server	Windows: <b>SASTS\TableServer\TableServer.bat</b> <sup>1,2</sup> UNIX and z/OS: <b>SASTS/TableServer/TableServer.sh</b> <sup>1,2</sup>

- 1 Your installation might have different names for the **SASApp** directory.
- 2 In single-machine installations that were migrated from SAS 9.1.3, the default location for the OLAP server script is **SASMain\OLAPServer** on Windows and **SASMain/OLAPServer** on UNIX or z/OS.

You can use any of the following arguments with these scripts:

*Note:* The hyphen in front of each argument is optional. △

**-install**

installs the server or spawner as a Windows service.

**-remove**

removes the server or spawner service from Windows.

**-start**

starts the server or spawner.

*Note:* Before you start the object spawner, the OLAP server, or the table server, make sure that the metadata server is running. △

**-stop**

completely stops the server or spawner.

*Note:*

- Stopping the metadata server stops other components, including object spawners, OLAP servers, table servers, SAS/CONNECT spawners, and SAS/SHARE servers.
- Before you stop an OLAP server, you should first stop the active sessions, as follows:
  - a In SAS Management Console, expand the **Monitoring** node and the **SAS OLAP Server Monitor** node.
  - b Right-click the name of the logical SAS OLAP Server, and select **Connect**.
  - c In the dialog box that appears, enter your user name and password and click **OK**.

- d Right-click the name of each session that you want to close, and select **Close Session**.

$\triangle$

**-restart**

stops and then starts a running server or spawner.

**-pause**

causes the server or spawner to finish processing all active client sessions and to stop accepting new requests.

When this option is used with the metadata server, the server is placed in the Offline state.

**-resume**

resumes the server or spawner process when it is in the paused state.

When this option is used with the metadata server, the server is returned to the Online state if the current state is Offline or Administration.

**-status**

displays the current status of the server or spawner.

The metadata server script has two additional options:

**-backup**

executes the program backupServer.sas, which is located in your metadata server startup directory. This program pauses the metadata server and copies your metadata repositories, the repository manager, the metadata journal file, and the metadata server's configuration files to a subdirectory called **SASBackup** in the metadata server's configuration directory. For details about this and other methods of performing backups, see "About the Metadata Server Backup and Restore Utilities" on page 124.

**CAUTION:**

**For production backups and restores, it is highly recommended that you follow the best practices that are provided in Chapter 9, "Best Practices for Backing Up and Restoring Your System," on page 123.    $\triangle$**

**-restore**

executes the program restoreServer.sas, which is located in your metadata server startup directory. This program pauses the metadata server and copies your metadata repositories, the repository manager, the metadata journal file, and the metadata server's configuration files from the backup directory to the metadata server configuration directory. For details about this and other methods of performing restores, see "About the Metadata Server Backup and Restore Utilities" on page 124.

---

## Use Individual Scripts to Operate Servers and Spawners

To use an individual script to operate the SAS/CONNECT spawner, SAS/SHARE server, and SAS Deployment Tester server, follow these steps:

- 1 On Windows, log on as a user who is a member of the host operating system's Administrators group. On UNIX and z/OS, log on as the SAS Installer user (for example, sas).
- 2 Use one of the following scripts, which are located in the path *SAS-configuration-directory/Lev1* (your installation might have a different name for **SASApp** subdirectory):

**Table 5.2** Scripts for Operating Servers and Spawners

Server or Spawner	Script Name and Location
SAS/CONNECT spawner	Windows: <b>SASApp\ConnectServer.bat</b> UNIX and z/OS: <b>SASApp/ConnectServer.sh</b>
SAS/SHARE server	Windows: <b>SASApp\ShareServer.bat</b> UNIX and z/OS: <b>SASApp/ShareServer.sh</b>
SAS Deployment Tester server	Windows: <b>DeploymentTesterServer\DeploymentTesterServer.bat</b> UNIX and z/OS: <b>DeploymentTesterServerDeploymentTesterServer.sh</b>

You can use any of the following arguments with these scripts:

*Note:* The hyphen in front of each argument is optional. △

**-install**

installs the server or spawner as a Windows service.

**-remove**

removes the server or spawner service from Windows.

**-start**

starts the server or spawner.

**-stop**

stops the server or spawner.

**-restart**

restarts the server or spawner.

**-status**

displays the current status of the server or spawner.

For more information, see *Communications Access Methods for SAS/CONNECT and SAS/SHARE* and *SAS/SHARE User's Guide*.

## Starting or Stopping the SAS Services Application

The SAS Services Application deploys a set of services called Remote Services that are used by the SAS Information Delivery Portal, the SAS Stored Process Web application, and other Web applications. This application must be started before you start your Web application server. To start the SAS Services Application:

- 1 Make sure that the metadata server is running.
- 2 On Windows, log on as a user who is a member of the host operating system's Administrators group. On UNIX and z/OS, Log on as the SAS Installer user (for example, sas).
- 3 Use one of the following scripts:
  - On Windows: **SAS-configuration-directory\Lev1\Web\Applications\RemoteServices\RemoteServices.bat**
  - On UNIX or z/OS: **SAS-configuration-directory/Lev1/Web/Applications/RemoteServices/RemoteServices.sh**

You can use any of the following arguments with these scripts:

*Note:* The hyphen in front of each argument is optional. △

**-install**

installs the SAS Services Application as a Windows service.

**-remove**

removes the SAS Services Application service from Windows.

**-start**

starts the SAS Services Application.

**-stop**

stops the SAS Services Application.

**-restart**

restarts the SAS Services Application.

**-status**

displays the current status of the SAS Services Application.

---

## Using SAS Management Console to Operate SAS Servers

---

### Obtain the Appropriate User Credentials, and Clear Cached Credentials (If Necessary)

To connect to a server or spawner in SAS Management Console, you must have the appropriate user credentials. Membership in the SAS Administrators group typically provides sufficient rights and permissions. For details, see “Who Can Do What in the Server Manager Plug-in” on page 35.

If you attempt to connect to a server or spawner and you are denied access, make sure you have the appropriate credentials. After your credentials are updated, be sure to clear the credentials cache before you try to connect again. To clear the credentials cache, click **File ► Clear Credentials Cache** in SAS Management Console.

---

### Use SAS Management Console to Stop, Pause, Resume, or Reset the SAS Metadata Server

You can use SAS Management Console to stop, pause, or resume the SAS Metadata Server. You can also use SAS Management Console to reset the metadata server in order to recover memory on the host machine.

*Note:* Before using these features to operate the metadata server, see “About SAS Metadata Server States” on page 53. △

Follow these steps:

- 1 Log on to SAS Management Console as a user with the appropriate credentials.
- 2 On the **Plug-ins** tab, expand the **Metadata Manager** node, and right-click the **Active Server** node.
- 3 Select one of the following actions:

**Stop**

completely stops the metadata server.

*Note:*

- Stopping the metadata server stops other components, including object spawners, OLAP servers, table servers, SAS/CONNECT spawners, and SAS/SHARE servers.
- To start the metadata server after stopping it, you must use a script. If you want to be able to resume processing without using a script, use **Pause ► Offline** instead.

△

#### **Pause ► Offline**

pauses the metadata server to the Offline state. In the Pause Comment dialog box, you can enter a comment that provides information about the pause action. When a client attempts to read or write metadata, the comment is appended to the appropriate message and displayed as follows:

**Error accessing metadata: The GetMetadataObjects request failed because the server is paused for OFFLINE use. The reason for the server pause is *comment*.**

#### **Pause ► Administration**

pauses the metadata server to the Administration state. In the Pause Comment dialog box, you can enter a comment that provides information about the pause action. When a user who is not in the **Metadata Server: Unrestricted** role attempts to read or write metadata, the comment is appended to the appropriate message and displayed as follows: **Error accessing metadata: The GetMetadataObjects request failed because the server is paused for ADMINISTRATION use. The reason for the server pause is *comment*.**

#### **Resume**

returns the metadata server to the Online state (if the current state is Offline or Administration). The metadata server resumes accepting client requests to read and write metadata and re-opens repository containers as clients request metadata from them.

#### **Reset**

unloads from memory and closes the containers of all registered repositories in order to recover memory on the metadata server host. The metadata server then re-opens repository containers as clients request metadata from them. This option is useful if the host computer experiences an out-of-memory condition.

To start the metadata server, see “Running Servers as Windows Services” on page 55 and “Using Scripts to Operate SAS Servers Individually” on page 61. For information about metadata server startup parameters, see Appendix 5, “Options for the Metadata Server Invocation Command,” on page 289.

---

## **Use SAS Management Console to Stop, Pause, Resume, Quiesce, or Refresh the SAS Object Spawner**

You can use SAS Management Console to stop, pause, resume, quiesce, or refresh the object spawner. Follow these steps:

- 1 Log on to SAS Management Console as a user with the appropriate credentials.
- 2 On the **Plug-ins** tab, expand the **Server Manager** node and the **Object Spawner** node.



- 3 Right-click the object spawner instance that you want to administer, and click **Connect**.

- 4 After you are connected, you can select any of the following actions:

**Refresh**

updates the SAS Management Console display with current information about the spawner.

**Disconnect**

discontinues your current connection to the spawner.

**Stop**

stops the spawner process, including any active client requests.

*Note:* If you want active requests to be completed first, use **Quiesce**. △

**Pause**

causes the spawner to finish processing all active client requests and to stop accepting new requests.

**Resume**

resumes the spawner process when it is in the paused state.

**Quiesce**

stops the spawner process after the spawner has finished processing all active client requests.

**Refresh Spawner**

re-initializes the spawner. During re-initialization, the spawner stops any spawned servers that are currently running, rereads the metadata in the spawner definition, and applies any new options that have been set.

*Note:* If the spawner is in a paused state when the **Refresh Spawner** action occurs, the spawner remains paused after it is re-initialized. △

To start the object spawner, see “Running Servers as Windows Services” on page 55 and “Using Scripts to Operate SAS Servers Individually” on page 61. For information about startup options for the object spawner, see “Spawner Invocation Options” in the *SAS Intelligence Platform: Application Server Administration Guide*.

---

## Use SAS Management Console to Stop, Pause, Resume, Quiesce, or Refresh the SAS OLAP Server

You can use SAS Management Console to stop, pause, resume, quiesce, or refresh the SAS OLAP Server. Follow these steps:

- 1 Log on to SAS Management Console as a user with the appropriate credentials.
- 2 On the **Plug-ins** tab, expand the **Server Manager** node, the application server node (for example, **SASApp**), and the node for the logical OLAP server.
- 3 Right-click the OLAP server instance that you want to administer, and click **Connect**.
- 4 After you are connected, you can select any of the following actions:

**Refresh**

updates the SAS Management Console display with current information about the OLAP server.

**Disconnect**

discontinues your current connection to the OLAP server.

**Stop**

stops the OLAP server process, including any active client requests.

*Note:* If you want active sessions to be completed first, use **Quiesce**. △

**Pause**

causes the OLAP server to finish processing all active sessions and to stop accepting new client requests.

**Resume**

resumes the OLAP server process when it is in the paused state.

**Quiesce**

stops the OLAP server process after the server has finished processing all active sessions.

*Note:* To start the OLAP server after quiescing it, you must use a script. If you want to be able resume processing without using a script, use **Pause** instead. △

To start the OLAP server, see “Running Servers as Windows Services” on page 55 and “Using Scripts to Operate SAS Servers Individually” on page 61. For information about OLAP server startup parameters, see “System Options for Application Server Components” in the *SAS Intelligence Platform: Application Server Administration Guide*.

---

## Use SAS Management Console to Control Processes on Servers

You can use SAS Management Console to stop, pause, resume, or quiesce a process on a workspace server, pooled workspace server, or stored process server. Follow these steps:

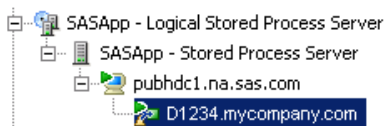
- 1 Log on to SAS Management Console as a user with the appropriate credentials.
- 2 On the **Plug-ins** tab, expand the **Server Manager** node and the application server node (for example, **SASApp**).
- 3 Expand the node for the logical server that is running the process, and then expand the node for the server instance that is running the process. The server’s available connections are displayed in the server tree and in the right pane.
- 4 Highlight the appropriate host machine name, as shown in this example.




Right-click the mouse, and click **Connect**. For each process that is active for that connection, the right pane displays the process ID, the process owner, and the port, as shown in this example:

Processes			
	ID	Owner	Port
	3772	sassrv@D1234	8611

- 5 In the Server Manager tree, expand the node for the connection that you are working with. The active processes for the connection are displayed in the tree.
- 6 Highlight the process that you want to stop, pause, resume, or quiesce, as shown in this example:



In the right pane, the following tabs become active: **Clients**, **Options**, and **Loggers**. For pooled workspace servers and stored process servers, the **Sessions** and **Log** tabs also become active.

*Note:* If some tabs are disabled, then you need additional user credentials. See “Who Can Do What in the Server Manager Plug-in” on page 35. For details about the tabs, see “Use the Server Manager Tabs to Obtain Monitoring Information and to Adjust Logging Levels (If Needed)” on page 91. 

- 7 Right-click the mouse and select one of the following actions:

**Stop**

stops the server process, including any active client requests.

*Note:*

- ☐ If you want active requests to be completed first, use **Quiesce**.



**Pause**

causes the server process to finish processing all active client requests and to stop accepting new requests.

**Resume**

resumes the server process when it is in the paused state.

**Quiesce**

stops the server process after the server has finished processing all active client requests.

---

## Starting and Stopping Web Application Servers

---

### Starting or Stopping a JBoss Application Server

On Windows systems, follow these steps to start a JBoss Application Server:

- 1 Make sure that the SAS Services Application (Remote Services) has been started (see “Starting or Stopping the SAS Services Application” on page 64).
- 2 Use one of the following scripts:
  - ☐ On Windows:

*JBoss-home-directory\bin\SASServer1.bat*

- $\square$  On UNIX:

*JBoss-home-directory/bin/SASServer1.sh*

You can use any of the following arguments with these scripts:

*Note:* The hyphen in front of each argument is optional.  $\Delta$

**-install**

installs the server as a Windows service.

**-remove**

removes the service from Windows.

**-start**

starts the server.

**-stop**

stops the server.

If you selected the option to install the JBoss Application Server as a service, see “Running Servers as Windows Services” on page 55.

---

## Starting or Stopping an IBM WebSphere Application Server or an IBM WebSphere Application Server Network Deployment (ND)

### Starting or Stopping an IBM WebSphere Application Server or an IBM WebSphere Application Server Network Deployment (ND) on Windows or UNIX

If you are using an IBM WebSphere Application Server or an IBM WebSphere Application Server Network Deployment on Windows or UNIX, follow these steps to start the server:

- 1 Make sure that the SAS Services Application (Remote Services) has been started.
- 2 If the WebSphere deployment manager is not already running, start it by using one of the following methods:

- $\square$  If the deployment manager is installed as a Windows service, start the appropriate service.
- $\square$  Use one of the following commands, where *WAS\_INSTALL\_ROOT* is the home directory of the deployment manager:

- $\square$  On Windows:

*WAS\_INSTALL\_ROOT\bin\startManager.bat -profileName  
"SASDmgr01"*

- $\square$  On UNIX:

*WAS\_INSTALL\_ROOT/bin/startManager.sh -profileName  
"SASDmgr01"*

SASDmgr01 is the default profile name for the deployment manager. The name in your installation might be different.

- 3 If the WebSphere node agent is not already running, start it by using one of the following methods:
  - $\square$  If the node agent is installed as a Windows service, start the appropriate service.

- Use one of the following commands, where *WAS\_INSTALL\_ROOT* is the home directory of the application node:

- On Windows:

```
WAS_INSTALL_ROOT\bin\startNode.bat -profileName
"SAShost-name01"
```

- On UNIX:

```
WAS_INSTALL_ROOT/bin/startNode.sh -profileName
"SAShost-name01"
```

*SAShost-name01* is the default profile name for the managed node, which contains the node agent and the application server that is used to run SAS. The profile name for the managed node in your installation might be different.

#### 4 Start the WebSphere Application Server by using one of the following methods:

- Use the WebSphere administrative console to start the application server. The default server name on Windows and UNIX is *SASServer1*. If additional servers are installed, the default server names are *SASServer2*, *SASServer3*, and so on.

- Use one of the following commands, where *WAS\_INSTALL\_ROOT* is the home directory of the application node:

- On Windows:

```
WAS_INSTALL_ROOT\bin\startServer.bat "server-name"
-profileName "SAShost-name01"
```

- On UNIX:

```
WAS_INSTALL_ROOT/bin/startServer.sh "server-name"
-profileName "SAShost-name01"
```

The default server name on Windows and UNIX is *SASServer1*. If additional servers are installed, the default server names are *SASServer2*, *SASServer3*, and so on.

*SAShost-name01* is the default profile name on Windows and UNIX. If multiple servers are installed, then the default names for the additional profiles are *SAShost-name02*, *SAShost-name03*, and so on.

To stop the server, use the WebSphere administrative console or the appropriate script (*stopServer.bat* or *stopServer.sh*). The scripts are located in the same directory as the startup script.

*Note:* When you use the SAS Deployment Wizard to deploy or migrate a SAS system that uses WebSphere, you must first stop all WebSphere application servers. For instructions, see the IBM documentation for WebSphere Application Server. △

## Starting or Stopping an IBM WebSphere Application Server Network Deployment (ND) on z/OS

If you are using an IBM WebSphere Application Server Network Deployment on z/OS, follow these steps to start the application server:

- 1 Make sure that the SAS Services Application (Remote Services) has been started.
- 2 If the WebSphere deployment manager is not already running, start it by using one of the following methods:
  - Run the following script, where *WAS\_INSTALL\_ROOT* is the home directory of the deployment manager:

**`WAS_INSTALL_ROOT/bin/startManager.sh`**

- Issue the MVS START command to start the deployment manager. Specify the appropriate deployment manager procedure and related parameters.
- 3 If the WebSphere node agent is not already started, start it by using one of the following methods:
  - Run the following script, where `WAS_INSTALL_ROOT` is the home directory of the application node:
 

**`WAS_INSTALL_ROOT/bin/startNode.sh`**
  - Issue the MVS START command to start the node agent. Specify the the appropriate node agent procedure and related parameters.
- 4 Start the WebSphere Application Server by using one of the following methods:
  - Use the WebSphere administrative console to start each application server.
  - Run the following script for each application server, where `WAS_INSTALL_ROOT` is the home directory of the application node:
 

**`WAS_INSTALL_ROOT/bin/startServer.sh "server-name"`**
  - Issue the MVS START command for each server. Specify the appropriate application server procedure and related parameters.

To stop the application server, use one of the following methods:

- Use the WebSphere administrative console to stop the application server.
- Run the following script for the application server, where `WAS_INSTALL_ROOT` is the home directory of the application node:
 

**`WAS_INSTALL_ROOT/bin/stopServer.sh "server-name"`**
- Issue the MVS STOP command against the application server control region.

*Note:* When you use the SAS Deployment Wizard to deploy or migrate a SAS system that uses WebSphere, you must first stop all WebSphere application servers. For instructions, see the IBM documentation for WebSphere Application Server. △

---

## Starting or Stopping an Oracle WebLogic Server

By default, the SAS Deployment Wizard configures the WebLogic domain in the following path: `SAS-configuration-directory/Lev1/Web/SASDomain`. Your domain might have been configured in a different location.

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

- 1 Make sure that the SAS Services Application (Remote Services) has been started.
- 2 If the WebLogic Administration Server is not running, start it by using the following command:

**`domain-home-directory\startWebLogic.cmd`**

- 3 Start the WebLogic Managed Server by using **`startManagedWebLogic.cmd`**, which is located in `domain-home-directory\bin\`. The syntax is as follows:

**`startManagedWebLogic.cmd server-name administration-server-URL`**

For example:

**`startManagedWebLogic.cmd SASServer1 http://mycompany.com:7501`**

*Note:* The third maintenance release for SAS 9.2 includes scripts that you can use to install the WebLogic Managed Server and WebLogic Administration Server as Windows services. If you used these scripts to install the services, see “Running Servers as Windows Services” on page 55. △

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

- 1 Make sure that the SAS Services Application (Remote Services) has been started.
- 2 If the WebLogic Administration Server is not running, start it by using the following command:

*domain-home-directory*/**startWebLogic.sh**

- 3 Start the WebLogic Managed Server by using **startManagedWebLogic.sh**, which is located in *domain-home-directory/bin/*. The syntax is as follows:

**startManagedWebLogic.sh** *server-name* *administration-server-URL*

For example:

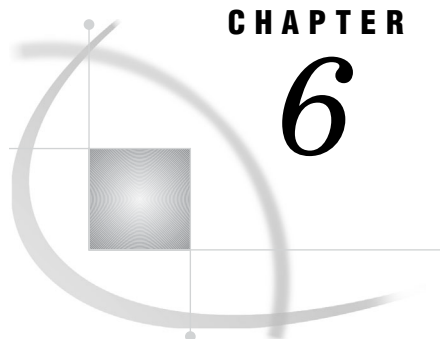
**startManagedWebLogic.sh** **SASServer1** **http://mycompany.com:7501**

You can also use other methods, including the administrative console or the WebLogic Scripting Tool (WLST), to start these servers. For details, see the WebLogic documentation at <http://edocs.beasys.com>.

To stop the server, use **stopManagedWebLogic.cmd** (on Windows); or **stopManagedWebLogic.sh** (on UNIX). These scripts are located in the same directories as the startup scripts.







## CHAPTER

## 6

## Checking the Status of Servers

<i>About Checking the Status of Servers</i>	<b>75</b>
<i>Use a Server Script to Display the Status of a Server or Spawner</i>	<b>76</b>
<i>Checking the Access State of the SAS Metadata Server</i>	<b>76</b>
<i>Use SAS Management Console to Check the Access State of the SAS Metadata Server</i>	<b>76</b>
<i>Use PROC METAOPERATE to Check the Access State of the SAS Metadata Server</i>	<b>76</b>
<i>Checking the Access State of Metadata Repositories and the Repository Manager</i>	<b>77</b>
<i>About Metadata Repository Access States and Access Modes</i>	<b>77</b>
<i>How the Access State of the Metadata Server Affects the Access States of Metadata Repositories</i>	<b>78</b>
<i>About Repository Manager Access States</i>	<b>78</b>
<i>Use SAS Management Console to Check the Access State of a SAS Metadata Repository</i>	<b>79</b>
<i>Use PROC METADATA to Check the Status of Metadata Repositories and the Repository Manager</i>	<b>80</b>
<i>Create and Run a PROC METADATA Program</i>	<b>80</b>
<i>Example: PROC METADATA Output</i>	<b>81</b>
<i>Check the Status of the Object Spawner</i>	<b>82</b>
<i>Validate the SAS Metadata Server, SAS Workspace Servers, SAS Pooled Workspace Servers, SAS Stored Process Servers, and SAS OLAP Servers</i>	<b>82</b>
<i>Validate the SAS Content Server</i>	<b>83</b>

### About Checking the Status of Servers

SAS provides a number of tools that you can use to determine the status of your SAS server components. SAS Management Console includes features that enable you to do the following:

- ❑ use a server script to display the status of a server from the command line
- ❑ check the access state of the SAS Metadata Server to determine whether it is online, paused for administration, or paused to an Offline state
- ❑ check a SAS Metadata Repository to determine whether it is in an Online, Administration, Administration (ReadOnly), ReadOnly, or Offline access state
- ❑ check an instance of the SAS object spawner to ensure that you can make a connection
- ❑ submit SAS code to validate that instances of a SAS Metadata Server, SAS Workspace Server, SAS Pooled Workspace Server, SAS Stored Process Server, or SAS OLAP Server are set up correctly
- ❑ determine whether the SAS Content Server is listening on the specified host and port, and that the first configured base path exists on the server

You can also use SAS programming features to check the status of the metadata server, metadata repositories, and the repository manager.

*Note:* To check the status of a SAS/SHARE server, see the *SAS/SHARE User's Guide*.  $\Delta$

---

## Use a Server Script to Display the Status of a Server or Spawner

To display the status of a server or spawner from a command line, you can use the **-status** option of the script for that server or spawner. For details, see “Using Scripts to Operate SAS Servers Individually” on page 61.

On UNIX and z/OS, you can use the **-status** option of the `sas.servers` script to display the status of all servers and spawners on a machine. For details, see “Use the `sas.servers` Script to Start, Stop, or Restart All Servers or to Display the Status of Servers” on page 58.

---

## Checking the Access State of the SAS Metadata Server

---

### Use SAS Management Console to Check the Access State of the SAS Metadata Server

To use SAS Management Console to determine the access state of the SAS Metadata Server:

- 1 In the SAS Management Console **Plug-ins** tab, expand the **Metadata Manager** node.
- 2 Right-click the **Active Server** node and select **Properties**.

The current state is displayed in the **State** field on the **General** tab of the Active Server Properties dialog box. If the state is Administration or Offline, then the **Comment** field displays any comments that were entered by the administrator who paused the server.

For a description of the Offline, Administration, and Online states, see “About SAS Metadata Server States” on page 53. To change the state of the metadata server, see “Use SAS Management Console to Stop, Pause, Resume, or Reset the SAS Metadata Server” on page 65.

---

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

To check the access state of the SAS Metadata Server, you can use PROC METAOPERATE to issue a STATUS action. The syntax is as follows:

```
PROC METAOPERATE
  SERVER='host-name'
  PORT=port-number
  USERID='user-ID'
  PASSWORD='password'
  PROTOCOL=BRIDGE
  ACTION=STATUS;
RUN;
```

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

If the SAS Metadata Server is running normally, information similar to the following is written to the SAS Output window:

```
NOTE: Server dXXXX.us.company.com SAS Version is 9.02.02B0P082107
NOTE: Server dXXXX.us.company.com SAS Long Version is 9.02.02B0P08212007 .
NOTE: Server dXXXX.us.company.com Operating System is XP_PRO.
NOTE: Server dXXXX.us.company.com Operating System Family is WIN.
NOTE: Server dXXXX.us.company.com Client is userid@DOMAIN.
NOTE: Server dXXXX.us.company.com Metadata Model is Version 11,01.
NOTE: Server dXXXX.us.company.com is RUNNING on 10Sep07:14:47:57.
```

If the SAS Metadata Server is paused, information similar to the following is written to the SAS Output window:

```
NOTE: Server dXXXX.us.company.com SAS Version is 9.02.02B0P082107
NOTE: Server dXXXX.us.company.com SAS Long Version is 9.02.02B0P08212007 .
NOTE: Server dXXXX.us.company.com Operating System is XP_PRO.
NOTE: Server dXXXX.us.company.com Operating System Family is WIN.
NOTE: Server dXXXX.us.company.com Client is userid@DOMAIN.
NOTE: Server dXXXX.us.company.com Metadata Model is Version 11,01.
NOTE: Server dXXXX.us.company.com is PAUSED on 10Sep07:14:47:57.
```

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

For more information about PROC METAOPERATE, see *SAS Language Interfaces to Metadata*.

---

## Checking the Access State of Metadata Repositories and the Repository Manager

---

### About Metadata Repository Access States and Access Modes

The *access state* of a metadata repository indicates its current state with regard to client access. A metadata repository can exist in any of the following access states:

#### Online

Clients can read, write, and update metadata.

#### ReadOnly

Clients can read metadata but cannot write, update, or delete metadata.

#### Administration

Only users who are in the **Metadata Server: Unrestricted** role can read or write metadata. (For information about this role, see “Overview of Initial Roles, Groups, and Users” on page 24.)

#### Administration (ReadOnly)

Only users who are in the **Metadata Server: Unrestricted** role can read metadata. Users cannot write, update, or delete metadata.

#### Offline

Clients can neither read nor write metadata to the repository.

The access state of a repository depends on both the current state of the metadata server and the repository's registered access mode. The *access mode* of a SAS metadata repository is the repository's intended access state (see "About SAS Metadata Repository Access Modes" on page 182). When the metadata server is Online, the access state of a repository is equal to its access mode. When the metadata server is paused to the Offline state or the Administration state, the access state of a repository can change temporarily. When the metadata server is resumed, the repository's access state returns to its registered access mode.

## How the Access State of the Metadata Server Affects the Access States of Metadata Repositories

Generally, the access state of a repository is the same as the access mode that is recorded in the repository's metadata. However, a repository's access state can change temporarily when the metadata server is paused to an Offline state (for example, while a backup process is executing) or when the metadata server is paused for administration. When the metadata server is resumed, the repository's access state returns to its registered access mode.

The following table shows how the access state of the metadata server affects the access state of a repository, based on the repository's access mode:

**Table 6.1** The Effect of Metadata Server States on Metadata Repository Access States

Metadata Server Access State	Metadata Repository Registered Access Mode	Metadata Repository Access State
Online	Online	Online
Online	Administration	Administration
Online	ReadOnly	ReadOnly
Online	Offline	Offline
Administration	Online	Administration
Administration	Administration	Administration
Administration	ReadOnly	Administration (ReadOnly)
Administration	Offline	Offline
Offline	Online	Offline
Offline	Administration	Offline
Offline	ReadOnly	Offline
Offline	Offline	Offline

## 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. An access state of ReadOnly or Offline for the repository manager is typically reflected in the following behavior:

ReadOnly

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

#### Offline

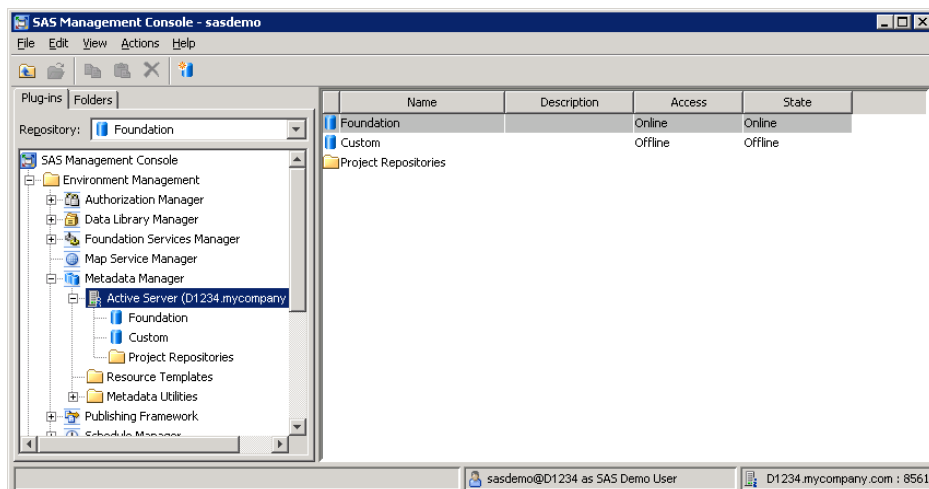
The behavior can vary depending on whether a repository is available in memory. Assume that repositories are unavailable for reading or writing to all users.

## Use SAS Management Console to Check the Access State of a SAS Metadata Repository

If the metadata server is online or is paused for administration, then you can use SAS Management Console to determine the current access state of a metadata repository, as follows:

- 1 In the SAS Management Console **Plug-ins** tab, expand the **Metadata Manager** node.
- 2 To view the access state of the foundation repository and any custom repositories that you have created, select the **Active Server** node. To view the access state of project repositories, expand the **Active Server** node and select the **Project Repositories** folder.

For each repository, the access mode and the current access state are displayed in the table view, as shown in the following example:



The table lists the following information for each registered repository:

#### Access

indicates the access mode of the repository as it is recorded in the Access attribute of the repository's metadata definition

#### State

indicates the current access state of the repository. Generally, the access state of a repository is equal to its access mode. However, if the metadata server is paused to an Offline state or is paused for administration, then the repository's access state might be changed temporarily (see "How the Access State of the Metadata Server Affects the Access States of Metadata Repositories" on page 78).

The **Access** and **State** fields are also displayed in the metadata repository properties. To see these properties, right-click the repository name in the Active Server tree, select **Properties**, and select the **Registration** tab.

*Note:*

- If the metadata server is paused to the Administration state, then you must be in the **Metadata Server: Unrestricted** role to view the server's status in SAS Management Console.
- If the metadata server is paused to the Offline state, then the access state of repositories cannot be displayed in SAS Management Console.

△

---

## Use PROC METADATA to Check the Status of Metadata Repositories and the Repository Manager

### Create and Run a PROC METADATA Program

If the metadata server is online, then you can use PROC METADATA to check the status of metadata repositories. To do so, use PROC METADATA to issue a GetRepositories method call that sets the OMI\_ALL flag to 1. The GetRepositories method is part of the SAS Open Metadata Interface.

The syntax is as follows:

```
PROC METADATA
    SERVER="host-name"
    PORT=port-number
    USERID="your-user-ID"
    PASSWORD="your-password"
    IN="<GetRepositories>
        <Repositories/>
        <Flags>1</Flags>
        <Options/>
        </GetRepositories>";
RUN;
```

*Note:*

- To run this program, you must have ReadMetadata permission to all repositories on the SAS Metadata Server.
- If the metadata server is paused to an Offline state, then you cannot use PROC METADATA to issue a GetRepositories method call.
- If the metadata server is paused to an Administration state, then special user credentials are required to use PROC METADATA to issue a GetRepositories method call. The user must either (1) be a member of the **Metadata Server: Unrestricted** role or (2) be listed in adminUsers.txt, either with or without an asterisk.

△

For each repository that is registered to the metadata server, 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 repository's intended access mode, which is specified in the Access attribute of the repository's metadata definition. This field corresponds to the repository's **Access** field in SAS Management Console. The possible values are **OMS\_FULL** (which is displayed as **Online** in SAS Management Console), **OMS\_READONLY**, **OMS\_ADMIN**, and **OMS\_OFFLINE**.

#### **CurrentAccess**

contains information that is used by metadata server processes. This field is displayed because it is sometimes used by SAS Technical Support in the resolution of metadata repository issues.

#### **PauseState**

indicates the current access state of the repository. This field corresponds to the repository's **State** field in SAS Management Console. The access state depends on both the current state of the metadata server and the repository's registered access mode (see "How the Access State of the Metadata Server Affects the Access States of Metadata Repositories" on page 78). An empty string means that the repository is online.

#### **Path**

indicates the path where the repository's data sets are located.

#### **RepositoryFormat**

contains the revision level of the metadata model that is used by the repository. A RepositoryFormat level that is less than the level of the server can cause the repository to have a **PauseState** (access state) of ReadOnly.

For more information about the METADATA procedure and the GetRepositories method, see *SAS Language Interfaces to Metadata*.

## **Example: PROC METADATA Output**

The following is an example of output from PROC METADATA when the metadata server is online:

```
<GetRepositories>
  <Repositories>
    <Repository Id="A00000001.A00000001"
      Name="REPOSMGR"
      Desc="The Repository Manager"
      DefaultNS="REPOS"
      RepositoryType=""
      RepositoryFormat="11"
      Access="OMS_FULL"
      CurrentAccess="OMS_FULL"
      PauseState=""
      Path='C:\SAS\EBIservice\Levl\SASMeta\MetadataServer\rposmgr' />
    <Repository Id="A00000001.A50Q21KN"
      Name="Foundation"
      Desc=""
      DefaultNS="SAS"
      RepositoryType="FOUNDATION"
      RepositoryFormat="11"
      Access="OMS_FULL"
      CurrentAccess="OMS_FULL"
```

```

    PauseState=""
    Path='C:\SAS\EBIServer\Lev1\SASMeta\MetadataServer\Foundation' />
<Repository Id="A00000001.A5IPLNM1"
    Name="Test Repository"
    Desc=""
    DefaultNS="SAS"
    RepositoryType="CUSTOM"
    RepositoryFormat="11"
    Access="OMS_READONLY"
    CurrentAccess="OMS_READONLY"
    PauseState="READONLY"
    Path='C:\SAS\EBIServer\Lev1\SASMeta\MetadataServer\TestRepository' />

```

This output in this example provides the following information:

- For the repository manager (**REPOSMGR**): **PauseState=""** indicates that it is available.
- For the Foundation repository: **Access="OMS\_FULL"** indicates a registered access mode of Online, and **PauseState=""** indicates that the current access state is Online.
- For the Test Repository: **Access="OMS\_READONLY"** indicates a registered access mode of ReadOnly, and **PauseState="READONLY"** indicates that the current access state is ReadOnly.

---

## Check the Status of the Object Spawner

The SAS object spawner is a program that listens for incoming client requests for SAS Workspace Servers, SAS Pooled 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, follow these steps:

- 1 Log on to SAS Management Console as a user with the appropriate credentials (see “Who Can Do What in the Server Manager Plug-in” on page 35).
- 2 On the **Plug-ins** tab, expand the **Server Manager** node and the **Object Spawner** node.
- 3 Right-click the object spawner instance that you want to check, and click **Connect**.

---

## Validate the SAS Metadata Server, SAS Workspace Servers, SAS Pooled Workspace Servers, SAS Stored Process Servers, and SAS OLAP Servers

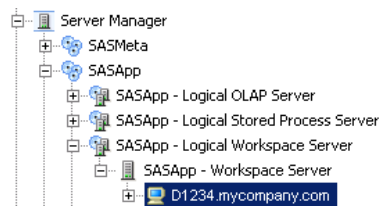
To determine whether your metadata server, workspace servers, pooled workspace servers, stored process servers, and OLAP servers are set up correctly, you can use the Validate feature. This feature submits a small SAS program to the server to determine whether it can process the code. You can validate either a logical server or a particular instance of a server.

*Note:* You cannot validate a workspace server that is configured to use client-side (rather than server-side) pooling. To determine whether a workspace server uses client-side pooling, right-click the server’s logical server node and click **Properties**. If the properties dialog box contains a **Pooling** tab that lists at least one puddle, then the server uses client-side pooling. △



To validate a server, follow these steps:

- 1 Log on to SAS Management Console as a user with the appropriate credentials (see “Who Can Do What in the Server Manager Plug-in” on page 35).
- 2 In SAS Management Console, expand the **Server Manager** node and the node for the application server (for example, **SASApp**).
- 3 To validate a logical server:
  - a Right-click the logical workspace server, logical pooled workspace server, logical stored process server, or logical OLAP server that you want to validate.
  - b Select **Validate**.
- 4 To validate a particular server instance:
  - a Expand the appropriate logical server node.
  - b In the tree that appears under the logical server definition, expand the server definition node.
  - c In the tree that appears under the server definition, select the server instance that you want to validate, as shown in this example:



- d Right-click the server instance and select **Validate**.
- 5 SAS Management Console sends a small SAS program to the server for execution. If the program executes correctly, the following message is displayed:

Validation Successful!

If the program does not execute correctly, an error message is displayed. For assistance in resolving the problem, see “Troubleshooting the SAS Server Tier” in the *SAS Intelligence Platform: Installation and Configuration Guide*.

If you are denied access, make sure you have the appropriate credentials. After your credentials are updated, click **File ► Clear Credentials Cache** in SAS Management Console before you try to validate again.

*Note:* To run more extensive validation tests, you can use the SAS Deployment Tester plug-in in SAS Management Console or the SAS Deployment Tester batch clients. These tools enable you to validate components on both the middle tier and the SAS server tier. For more information, see the SAS Management Console Help. The SAS Deployment Tester Server must be installed on the same machine as the components that you are validating. △

---

## Validate the SAS Content Server

If your installation includes a SAS Content Server, you can use the Validate feature to determine whether the server is set up correctly, as follows:

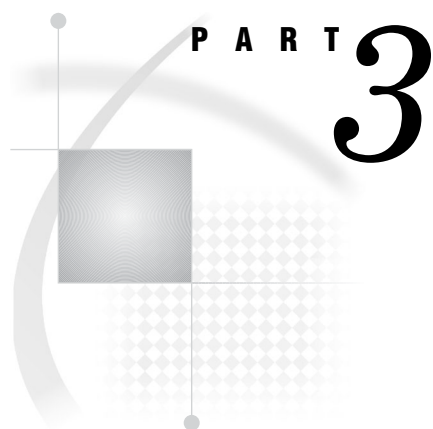
- 1 Log on to SAS Management Console, and expand the **Server Manager** node.
- 2 Right-click the **SAS Content Server** node.

- 3 SAS Management Console attempts to make a WebDAV connection to the server. If the connection is successful, the following message is displayed:

Validation Successful!

This message indicates that a WebDAV server is listening on the specified host and port, and that the first configured base path exists on the server.

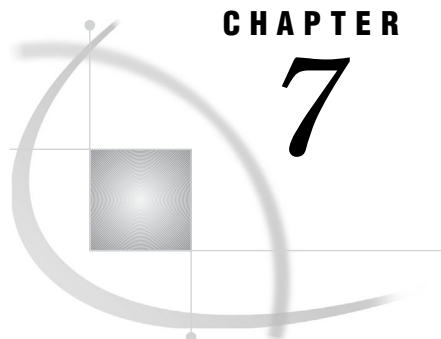
If the connection is not successful, an error message is displayed.



## **System Monitoring and Logging**

<i>Chapter 7</i> .....	<b>Monitoring the Activity of SAS Servers</b>	<i>87</i>
<i>Chapter 8</i> .....	<b>Administering Logging for SAS Servers</b>	<i>103</i>





## CHAPTER

## 7

# Monitoring the Activity of SAS Servers

<i>About SAS Server Monitoring</i>	<b>87</b>
<i>Using SAS Management Console to Monitor SAS Servers</i>	<b>88</b>
<i>About SAS Management Console's Server Management Features</i>	<b>88</b>
<i>Summary of Steps for Using SAS Management Console Server Monitoring</i>	<b>88</b>
<i>Obtain the Appropriate User Credentials, and Clear Cached Credentials (If Necessary)</i>	<b>89</b>
<i>Use SAS Management Console to Connect to Servers and Display Server Monitoring Information</i>	<b>89</b>
<i>Use the Server Manager Tabs to Obtain Monitoring Information and to Adjust Logging Levels (If Needed)</i>	<b>91</b>
<i>About the Server Manager Tabs</i>	<b>91</b>
<i>Use the Clients Tab in the Server Manager</i>	<b>91</b>
<i>Use the Sessions Tab in the Server Manager</i>	<b>92</b>
<i>Use the Spawned Server Activity Tab in the Server Manager</i>	<b>92</b>
<i>Use the Options Tab in the Server Manager</i>	<b>93</b>
<i>Use the Loggers Tab in the Server Manager, and (If Necessary) Change Logging Levels for Individual Loggers</i>	<b>93</b>
<i>Use the Log Tab in the Server Manager</i>	<b>94</b>
<i>Return Logging to Normal Levels, Disconnect from the Servers, and Stop Displaying Log Messages</i>	<b>96</b>
<i>Using the Server Performance Counters and Information Fields</i>	<b>96</b>
<i>Using the SAS OLAP Server Monitor</i>	<b>100</b>
<i>Access the SAS OLAP Server Monitor</i>	<b>100</b>
<i>View OLAP Server Sessions and Queries</i>	<b>101</b>
<i>Refresh the List of OLAP Servers or Server Connections</i>	<b>101</b>
<i>Using Third-Party Products to Monitor SAS Servers</i>	<b>101</b>

## About SAS Server Monitoring

To monitor the activity of SAS servers, including the metadata server, object spawners, OLAP servers, pooled workspace servers, stored process servers, table servers, and workspace servers, you can use the following capabilities:

- the server management features of SAS Management Console. For details, see “Using SAS Management Console to Monitor SAS Servers” on page 88.
- third-party monitoring tools. For details, see “Using Third-Party Products to Monitor SAS Servers” on page 101

To monitor and manage SAS/SHARE servers, see the *SAS/SHARE User's Guide*.

---

## Using SAS Management Console to Monitor SAS Servers

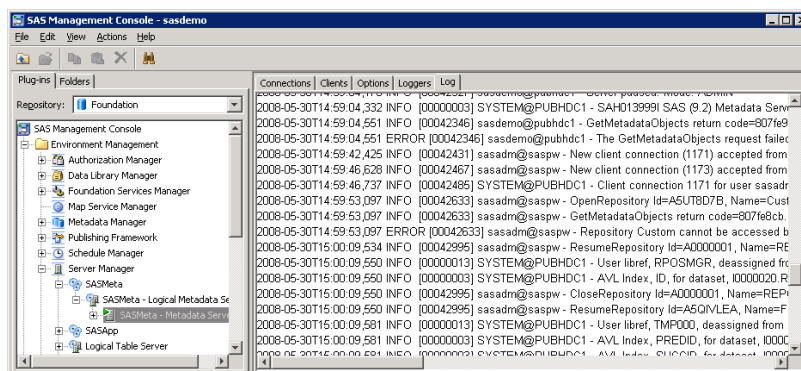
---

### About SAS Management Console's Server Management Features

You can use SAS Management Console to monitor the activity of SAS servers, including the metadata server, object spawners, OLAP servers, pooled workspace servers, stored process servers, table servers, and workspace servers. You can view the following kinds of information:

- ☐ currently connected clients
- ☐ graphs showing the number of spawned server sessions over time
- ☐ sessions that are active, inactive, or terminated
- ☐ server options and properties
- ☐ performance counters
- ☐ loggers and associated logging levels
- ☐ logging messages, at the level that you specify

This information is displayed on tabs in SAS Management Console, as shown in the following example:



You can also use the server management features to dynamically change logging levels and some server properties.

---

### Summary of Steps for Using SAS Management Console Server Monitoring

To use the server management features of SAS Management Console, follow these steps:

- 1 Obtain the appropriate user credentials.
- 2 Set the session display options, the logging level (if needed), and the logging start and end times
- 3 Use SAS Management Console to connect to servers and display server monitoring information.
- 4 Use the Server Manager tabs to obtain monitoring information and to adjust logging levels (if needed).

- 5 Return logging to normal levels, disconnect from the servers, and stop displaying log messages.

Each of these steps is described in the following topics.

---

## Obtain the Appropriate User Credentials, and Clear Cached Credentials (If Necessary)

To perform server monitoring in SAS Management Console, you must have the appropriate user credentials. To determine what credentials are required for each group of tasks, see “Who Can Do What in the Server Manager Plug-in” on page 35.

If you attempt to connect to a server or spawner and you are denied access, make sure you have the appropriate credentials. After your credentials are updated, be sure to clear the credentials cache before you try to connect again. To clear the credentials cache, click **File ► Clear Credentials Cache** in SAS Management Console.

---

## Use SAS Management Console to Connect to Servers and Display Server Monitoring Information


To monitor an instance of a server or spawner, you must establish a connection. Follow these steps:

- 1 Log on to SAS Management Console as a user with the appropriate credentials (see “Who Can Do What in the Server Manager Plug-in” on page 35).
- 2 On the **Plug-ins** tab, expand the **Server Manager** node and the application server node (for example, **SASApp** or **SASMeta**).
- 3 Expand the node for the logical server that you want to monitor.

The instances of that server that have been defined are displayed in the server tree. When you highlight a server instance in the tree, the right pane displays the available connections, as well as the port and the host machine for each connection.

- 4 To connect to the metadata server, an OLAP server, or a table server and display monitoring information:
  - a In the Server Manager tree, right-click the server instance that you want to monitor.
  - b Click **Connect**.

The following server monitoring tabs become active: **Clients**, **Options**, **Loggers**, and **Log**. For OLAP servers and table servers, a **Sessions** tab also appears.


*Note:* If some tabs are disabled, then you need additional user credentials. See “Who Can Do What in the Server Manager Plug-in” on page 35. 

To use the information on the tabs, see “Use the Server Manager Tabs to Obtain Monitoring Information and to Adjust Logging Levels (If Needed)” on page 91.

- 5 To connect to a workspace server, pooled workspace server, or stored process server and display monitoring information, follow these steps:
  - a In the Server Manager tree, expand the node for the server instance that you want to monitor.
 

The server’s available connections are displayed in the server tree and in the right pane.
  - b In the Server Manager tree, highlight a connection and click **Connect**.

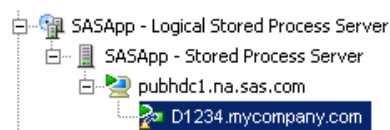
For each process that is active for that connection, the right pane displays the process ID, the process owner, and the port, as shown in this example:

Processes			
	ID	Owner	Port
	3772	sassrv@D1234	8611


- c In the Server Manager tree, expand the node for the connection that you are working with.

The active processes for the connection are displayed in the tree.

- d In the Server Manager tree, highlight the process that you want to monitor, as shown in this example:



The following server monitoring tabs become active: **Clients**, **Options**, and **Loggers**. For stored process servers, the **Sessions** and **Log** tabs also become active.


*Note:* If some tabs are disabled, then you need additional user credentials. See “Who Can Do What in the Server Manager Plug-in” on page 35. 

To use the information on the tabs, see “Use the Server Manager Tabs to Obtain Monitoring Information and to Adjust Logging Levels (If Needed)” on page 91.

- 6 To connect to an instance of the object spawner and display monitoring information, follow these steps:

- a On the **Plug-ins** tab, expand the **Server Manager** node and the **Object Spawner** node.
- b Right-click the object spawner instance that you want to administer, and click **Connect**.

The following server monitoring tabs become active: **Clients**, **Spawned Server Activity**, **Options**, **Loggers**, and **Log**.

*Note:* If some tabs are disabled, then you need additional user credentials. See “Who Can Do What in the Server Manager Plug-in” on page 35. 

To use the information on the tabs, see “Use the Server Manager Tabs to Obtain Monitoring Information and to Adjust Logging Levels (If Needed)” on page 91.



## Use the Server Manager Tabs to Obtain Monitoring Information and to Adjust Logging Levels (If Needed)


### About the Server Manager Tabs

After you have set the Server Manager options and connected to a server or a spawner, as described in the preceding topics, SAS Management Console displays server monitoring information in the right pane in a series of tabs.

Depending on which server component you are monitoring, SAS Management Console might display the following tabs:










- ☐ **Clients**
- ☐ **Sessions**
- ☐ **Spawned Server Activity**
- ☐ **Options**
- ☐ **Loggers**
- ☐ **Log**

The following topics describe the tabs and how to use them.

*Note:* If some tabs are disabled, then you might need to connect to the server (see “Use SAS Management Console to Connect to Servers and Display Server Monitoring Information” on page 89) or you might need additional user credential (see “Who Can Do What in the Server Manager Plug-in” on page 35). 

### Use the Clients Tab in the Server Manager

The **Clients** tab in the Server Manager lists the clients that are or have been connected to the currently selected server component. The tab displays each client’s user name, host machine, and entry time, as shown in the following example:

Connections   <b>Clients</b>   Options   Loggers   Log			
	User	Host	Entry Time
	sastrust@saspw	10.16.14.117:1747	May 30, 2008 10:00:07 AM
	sastrust@saspw	10.16.14.117:1698	May 30, 2008 9:56:07 AM
	sastrust@saspw	10.16.14.117:1971	May 30, 2008 10:15:23 AM
	sastrust@saspw	10.16.14.117:2004	May 30, 2008 10:16:18 AM
	sastrust@saspw	10.16.14.117:1949	May 30, 2008 10:14:46 AM
	sasadm@saspw	10.16.14.117:1696	May 30, 2008 9:56:04 AM
	sasdemo@pubhdc1	10.16.14.117:2934	May 30, 2008 2:01:17 PM
	sastrust@saspw	10.16.14.117:1934	May 30, 2008 10:13:00 AM
	sasdemo@PUBHDC1	10.16.14.117:2081	May 30, 2008 10:24:45 AM

The **Clients** tab is displayed for the following server components:

- ☐ metadata server
- ☐ object spawner connections
- ☐ OLAP servers
- ☐ pooled workspace server processes
- ☐ stored process server processes
- ☐ table servers
- ☐ workspace server processes

## Use the Sessions Tab in the Server Manager

The **Sessions** tab in the Server Manager lists the sessions that are active on the currently selected server or server process. Depending on your options settings, inactive and terminated sessions might also be displayed, as shown in this example:

Connections	Clients	Sessions	Options	Loggers	Log
ID	Owner	State	Idle Since /	Start Time	End Time
c3af3010-f7cd-4cf2-b03...	sasdemo@D1234	Ended		May 30, 2008 3:17:31 PM	May 30, 2008 3:17:31 PM

To specify which types of sessions to display:

- 1 Right-click the **Server Manager** node, and select **Options**. The Options dialog box appears.
- 2 On the **Sessions** tab of the Options dialog box, select one of the following options:

**Active and Inactive**

to display only active and inactive sessions in the **Sessions** tab. When sessions terminate, they will be removed from the tab.

**Active, Inactive, and Ended**

to display active, inactive, and terminated sessions in the **Sessions** tab.

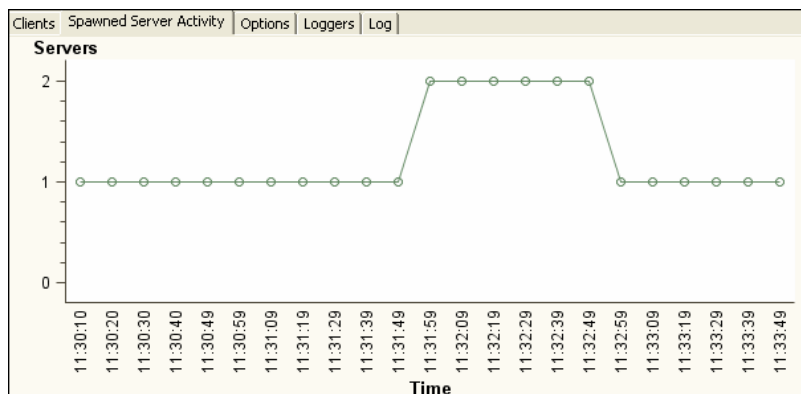
If terminated sessions are displayed, you can remove them by right-clicking the selected server node or process node in the server tree and selecting **Refresh**.

The **Sessions** tab is displayed for the following server components:

- ☐ OLAP servers
- ☐ stored process server processes
- ☐ table servers

## Use the Spawned Server Activity Tab in the Server Manager

The **Spawned Server Activity** tab in the Server Manager displays a graph that shows server activity associated with the currently selected object spawner connection. The graph shows how many spawned workspace servers, pooled workspace servers, and stored process servers are active at a given point in time, as shown in the following example:



The graph is updated every 10 seconds. This tab is displayed only for object spawner connections.

## Use the Options Tab in the Server Manager

The **Options** tab displays performance counters and other information fields for the currently selected server or spawner session, as shown in the following example:

Connections   Clients   Options   Loggers   Log				
Name	Description	Value	Category	
IOM.CounterResets	The number of times the ResetPerformanceCounters method has been ...	0	Counters	
IOM.CurrentClients	The number of currently connected clients.	10	Counters	
IOM.IdleTime	The amount of time the server has been idle.	0.0	Counters	
IOM.TimeInCalls	The amount of time spent in server method calls.	0.0	Counters	
IOM.TotalCalls	The number of method calls the server has processed.	0	Counters	
IOM.TotalClients	The number of clients the server has serviced.	0	Counters	
Metadata.CounterAddMetadata	A Metadata counter that tracks the number of AddMetadata calls.	0	Counters	
Metadata.CounterAddResponsibleP...	A Metadata counter that tracks the number of AddResponsibleParty calls.	0	Counters	
Metadata.CounterAddUserFolders	A Metadata counter that tracks the number of AddUserFolders calls.	0	Counters	
Metadata.CounterAnalyzeMetadata	A Metadata counter that tracks the number of AnalyzeMetadata calls.	0	Counters	
Metadata.CounterCheckinLogical	A Metadata counter that tracks the number of CheckinLogical calls.	0	Counters	
Metadata.CounterCheckoutLogical	A Metadata counter that tracks the number of CheckoutLogical calls.	0	Counters	
Metadata.CounterDeleteMetadata	A Metadata counter that tracks the number of DeleteMetadata calls.	0	Counters	

For more information about the performance counters and information fields, see “Using the Server Performance Counters and Information Fields” on page 96.

Some types of properties can be modified in this tab. To change the value of a property, right-click the property entry and click **Edit**.

The **Options** tab is displayed for the following server components:

- ☐ metadata server
- ☐ object spawner connections
- ☐ OLAP servers
- ☐ pooled workspace server processes
- ☐ stored process server processes
- ☐ table servers
- ☐ workspace server processes

## Use the Loggers Tab in the Server Manager, and (If Necessary) Change Logging Levels for Individual Loggers

The **Loggers** tab in the Server Manager displays the loggers that are in use for the selected server component, as shown in the following example:

Connections   Clients   Options   Loggers   Log				
	Name ▲	Description	Level	
	<Root>		Error	
	Admin		Information	
	Admin.Operations		<Inherited>	
	App		Information	
	App.FilteredList		<Inherited>	
	App.FilteredList.FilteredList		<Inherited>	
	App.FilteredList.FilteredList.Close		<Inherited>	
	App.FilteredList.FilteredList.Filter		<Inherited>	
	App.FilteredList.FilteredList.Filter.Get		<Inherited>	
	App.FilteredList.FilteredList.Filter.Set		<Inherited>	
	App.FilteredList.FilteredList.GetAttrib...		<Inherited>	

For each logger that is associated with the server component, this tab shows the logging level that is in effect. If **<Inherited>** appears in the third column, then the logging level is not set explicitly for the logger; instead, the level is inherited from a

parent logger. If you want to see which logging level is in effect through inheritance, then right-click the entry and select **Properties**.

The logging levels that are in effect at server start-up are based on the server's logging configuration file. In most cases, there is no need to change the logging levels. However, SAS Technical Support might request that you change a logging level temporarily to assist in problem resolution. For example, you might be requested to increase the level for the object spawner's **App** logger.

To change the logging level temporarily for a logger:

- 1 Right-click the entry for the logger whose level you want to change, and select **Properties**.
- 2 In the Properties dialog box, select the **Assigned** radio button. Then select the appropriate level from the drop-down box.

**CAUTION:**

**Excessive logging can degrade performance. You should not use the TRACE and DEBUG logging levels unless you are directed to do so by SAS Technical Support.** △

The logging level changes that you enter here have the following effects on logging:

- ☐ For the duration of the server process, the changes override the logging levels that are defined in the server's logging configuration file.
- ☐ The changes affect messages that are displayed in the **Log** tab, as well as messages that are sent to other appenders (as configured in the server's logging configuration file).
- ☐ The changes remain in effect until the server component or process stops. When the server component is restarted, the logging levels revert to those that are defined in the server's logging configuration file.

*Note:* The logging levels that are set in the default configuration files should be sufficient for most sites. If you need to permanently change logging levels for a component, then you must change the logging configuration file. To maintain system performance, use extreme caution in making these changes. △

The **Loggers** tab is displayed for the following server components:

- ☐ metadata server
- ☐ object spawner connections
- ☐ OLAP servers
- ☐ pooled workspace server processes
- ☐ stored process server processes
- ☐ table servers
- ☐ workspace server processes

For more information, see Chapter 8, "Administering Logging for SAS Servers," on page 103 in this guide, as well as the *SAS Logging: Configuration and Programming Reference*.


## Use the Log Tab in the Server Manager

The **Log** tab displays log events for the currently selected server or server process, as shown in the following example:

Connections   Clients   Options   Loggers   Log			
2008-01-30T14:01:25,175	INFO	[00001622]	Audit.Authentication (pebclient.c:1670) - Client connection 31 for user SASServerOp@
2008-01-30T14:01:25,175	INFO	[00001624]	Audit.Authentication (pebclient.c:1229) - New client connection (32) accepted from ser
2008-01-30T14:01:25,253	INFO	[00001636]	Audit.Authentication (pebclient.c:1670) - Client connection 30 for user SASServerOp@
2008-01-30T14:01:25,800	INFO	[00001664]	App.Meta (tkomiApiFolders.c:928) - The requested user folder 'My Folder' does not exi
2008-01-30T14:01:25,800	INFO	[00001664]	App.Meta (tkomiMthBase.c:5189) - GetUserFolders return code=807fe9b6....
2008-01-30T14:01:25,800	INFO	[00001664]	App.Meta (tkomiCmpBase.c:2119) - DoRequest return code=807fe9b6....
2008-01-30T14:01:25,800	INFO	[00001664]	App.Meta (tkomiCmpBase.c:2119) - The requested user folder 'My Folder' does not exi
2008-01-30T14:01:25,816	INFO	[00001665]	App.Meta (tkomiApiFolders.c:928) - The requested user folder 'My Folder' does not exi
2008-01-30T14:01:25,816	INFO	[00001665]	App.Meta (tkomiApiFolders.c:928) - The requested user folder 'Home Folder' does not
2008-01-30T14:01:25,972	INFO	[00001665]	Audit.Meta.Security.AccCtrlAdm (tkomsWorkunit.c:1509) - Access Control change on
2008-01-30T14:02:30,129	INFO	[00002017]	Audit.Authentication (pebclient.c:1211) - New client connection (33) accepted from ser
2008-01-30T14:03:13,785	INFO	[00002055]	Audit.Authentication (pebclient.c:1229) - New client connection (34) accepted from ser

To display log events on this tab:

- 1 Use the Server Manager options to specify a message level or threshold level filter. You can also specify a starting and ending time.

*Note:* These settings filter the events that are already being generated based on the server's logging configuration file and the **Loggers** tab; they do not generate additional events. If you want to generate additional events, use the **Loggers** tab to change the level for one or more loggers, or modify the server's logging configuration file. 

Set the filters as follows:


- a Right-click the **Server Manager** node, and select **Options**. The Options dialog box appears.
- b Click the **Logging** tab of the Options dialog box, and specify the following options:

#### Message Level

specifies a specific level of messages that are to be displayed in SAS Management Console.


#### Threshold Level

specifies the lowest level of messages that are to be displayed in SAS Management Console. Messages that have a level equal to or higher than the specified level are displayed in the **Log** tab.

*Note:* You can specify either a threshold level or a message level, but you cannot specify both. 

#### Start Time


specifies the date and time that log messages are to start appearing in

SAS Management Console. Click the date and time icon (), specify a date and time, and then click outside of the dialog box to save your entry.

If you do not specify a start time, then the message level or threshold level takes effect immediately.

#### End Time

specifies the date and time that log messages are to stop appearing in

SAS Management Console. Click the date and time icon (), specify a date and time, and then click outside of the dialog box to save your entry.

If you do not specify an end time, then the message level or threshold level remains in effect until you change it.

Click **OK** in the Options dialog box.

- 2 Click **View ► Refresh** on the menu bar to display the current contents of the message buffer.

- 3 Click **View ► Refresh** again each time you want to see the latest events.

The events that are displayed in this tab depend on the following:

- the loggers and logging levels that are specified in the server's logging configuration file.
- adjustments to logging levels that you might have made on the **Loggers** tab (see "Use the Loggers Tab in the Server Manager, and (If Necessary) Change Logging Levels for Individual Loggers" on page 93). These adjustments override the logging configuration file and remain in effect for the duration of the server session.
- additional filtering (based on message level, threshold level, and starting and ending times) that you specify in the Server Manager options.

For example, if you specify a threshold level of TRACE in the Server Manager Options, but none of the loggers are set to TRACE in the logging configuration file or on the **Loggers** tab, then no TRACE events will appear in SAS Management Console. You must first adjust the logging level for the appropriate loggers either in the logging configuration file or the **Loggers** tab.

**CAUTION:**

**Excessive logging can degrade performance. Therefore, you should not use the TRACE and DEBUG logging levels unless you are directed to do so by SAS Technical Support.** △

The **Log** tab is displayed for the following server components:

- metadata server
- object spawner connections
- OLAP servers
- stored process server processes
- table servers

For more information, see Chapter 8, "Administering Logging for SAS Servers," on page 103 in this guide, as well as the *SAS Logging: Configuration and Programming Reference*.

---

## Return Logging to Normal Levels, Disconnect from the Servers, and Stop Displaying Log Messages

When you are finished monitoring servers, then you should do the following:

- 1 If you changed the logging levels using the **Loggers** tab, then manually reset the loggers to their normal levels (see "Use the Loggers Tab in the Server Manager, and (If Necessary) Change Logging Levels for Individual Loggers" on page 93).

*Note:* You can also reset loggers to their normal levels by restarting the server component. △

- 2 Reset the message level, threshold level, or the start and end times in the Server Manager options (see "Use the Log Tab in the Server Manager" on page 94).
- 3 Disconnect from any server components that you are connected to.

---

## Using the Server Performance Counters and Information Fields

Various types of performance counters and information fields are provided for sessions of SAS servers and object spawners. The performance counters provide statistics about activity that has occurred since the server or spawner was last started.

The information fields provide static information about the server or spawner configuration or about the current server or spawner session.

The description and the current value of each performance counter and information field are displayed on the server's **Options** tab, which appears on the right pane of SAS Management Console. To access this tab, follow the instructions in "Using SAS Management Console to Monitor SAS Servers" on page 88.

You can also use third party monitoring tools to view the counters and information fields and to reset the performance counters (see "Using Third-Party Products to Monitor SAS Servers" on page 101).

The following types of performance counters and information fields are provided:

**IOM.***information-type*

provides status and static information about the selected metadata server, OLAP server, object spawner, pooled workspace server, or stored process server session, including the following information:

- ☐ the server or server start time
- ☐ the current execution state
- ☐ the port on which the server or spawner is listening
- ☐ the time that counters were last reset
- ☐ the name of the server or spawner component

**IOM.Counter***counter-name*

records activity for the selected metadata server, OLAP server, object spawner, pooled workspace server, or stored process server session, including the following information:

- ☐ the number of times that the counters were reset
- ☐ the number of currently connected clients
- ☐ the amount of time that the server or spawner has been idle
- ☐ the amount of time that has been spent processing server method calls
- ☐ the number of method calls that have been processed
- ☐ the number of clients that have been serviced

**LogicalServer.***logical-server-name.counter-name*

records object spawner activity for each logical server that is serviced by the object spawner. This type of counter appears in the object spawner's **Options** tab.

For logical pooled workspace servers and stored process servers, these counters provide the following information:

- ☐ the number of clients that are connected, pending, and waiting
- ☐ the number of clients that have been redirected, that have timed out, and that have been denied authorization
- ☐ the number of servers and the number of active peers that are running in the cluster
- ☐ the total waiting time for clients

For logical pooled workspace servers, the counters provide information on the total waiting time for clients and the number of clients that have been denied authorization. For logical non-pooled workspace servers, a counter provides the number of clients that have been denied authorization.

**LogicalServer.***logical-server-name.information-type*

provides static information about each logical server that is serviced by the object spawner. This type of field appears in the object spawner's **Options** tab. These fields provide information such as the logical server's class identifier, authorization

service, security settings, maximum wait time, and load balancing configuration (if applicable).

**LogicalServerName.n**

provides the names of each logical server that is serviced by the object spawner. This field appears in the object spawner's **Options** tab.

**Metadata.Countermethod-name**

records the number of times that specific metadata server methods are called. This type of counter provides information such as the following:

- the number of times that SAS Data Integration Studio calls the change management methods that are used to check out and check in metadata
- the number of times that the metadata server is paused and resumed

**Metadata.CounterIOtransaction-type**

records low-level input and output activity on metadata repository data sets. This type of counter is updated as follows:

- Each time a repository, metadata object, or association is created, deleted, updated, or read in any metadata repository on the server, the appropriate counter is incremented by one.
- If a metadata object is the first object of its type to have an operation performed, then counters are incremented by more than one. For example, if an object is added and the object's metadata type does not yet exist in the repository, then Metadata.CounterIOAdd is incremented by 3.
- When a repository is deleted, the Metadata.CounterIODelete counter is incremented by one and the Metadata.CounterIOAdd counter is incremented by one because a new container registry is created.

**Metadata.CounterJournalcounter-name**

records activity and status information for the metadata server's journal file. This type includes the following counters:

**Metadata.CounterJournalDataAvailable**

indicates the number of bytes of data that are in the journal file that have not yet been applied to the repository data sets on disk.

**Metadata.CounterJournalDataAvailableMax**

indicates the largest amount of metadata, in bytes, that has been committed to memory but not to repository data sets on disk. This value is equal to the highest value that has been placed in the Metadata.CounterJournalDataAvailable counter during the current server session. You can use this counter to determine the appropriate value for the JOURNALFILESIZE parameter of the omaconfig.xml file. If the omaconfig.xml JOURNALFILESIZE option specifies a journal file size that is too small, the metadata server slows down to allow the background task to catch up.

**Metadata.CounterJournalDataWritten**

indicates the total number of bytes of metadata that have been written to the journal file since the file was created. This number continues to grow throughout the life of the journal file and is not reset when the server is stopped and restarted.

**Metadata.CounterJournalSpaceAvailable**

indicates either the number of bytes of space left in the journal file (if the file is a fixed size), or 999999999 if the journal file does not have a fixed size.

**Metadata.CounterJournalTransProcessed**



indicates the number of transactions that have been fully processed to the data sets on disk since the metadata server was last started.

**Metadata.CounterJournalTransQueued**

indicates the number of transactions in the journal file that are currently waiting to be applied to metadata repositories on disk. This counter corresponds to the Metadata.CounterJournalDataAvailable counter, which measures the outstanding workload in bytes.

**Metadata.CounterJournalTransWritten**

indicates the number of transactions that have been processed since the server was started with journaling enabled. The number includes transactions that are still in the journal file and those that have been applied to repositories on disk. This number is reset when the server is paused to an Offline state and resumed, and also when the server is stopped and restarted.

**Metadata.CounterRetryUpdateLock***counter-name*

provides information to help diagnose performance issues with metadata updates that involve multiple repositories, particularly if messages indicate that repositories were not properly locked. This type includes the following counters:

**Metadata.CounterRetryUpdateLockMiss**

indicates the number of missing repository update locks that have been encountered.

**Metadata.CounterRetryUpdateLockMissCount**

indicates the number of times that requests were restarted because a repository update lock was missing.

**Metadata.CounterRetryUpdateLockRequest**

indicates the number of requests that were restarted because a repository update lock was missing.

**Metadata.CounterRetryUpdateLockHighWater**

indicates the highest number of update lock retries that occurred for a single request. This value cannot exceed the update retry limit, which is specified in the omaconfig.xml file's MAXIMUM\_RETRY\_COUNT option.

**Metadata.CounterRetryUpdateLockFailed**

indicates the number of requests that could not be completed within the update retry limit, which is specified in the omaconfig.xml file's MAXIMUM\_RETRY\_COUNT option.

**Metadata.CounterRetryUpdateLockExtra**

indicates the number of repository update locks that have been obtained but not used. This number includes update locks for requests that only needed to read information from the repository.

**Metadata.CounterRetryUpdateLockExtraCount**

indicates the number of requests that obtained but did not use an update lock. This includes requests that only needed to read information from the repository.

For more information about repository update locks, see MAXIMUM\_RETRY\_COUNT on page 284.

**Metadata.CounterRetryQueryLock***counter-name*

provides information to help diagnose performance issues with metadata queries that involve multiple repositories, particularly if messages indicate that repositories were not properly locked. This type includes the following counters:

**Metadata.CounterRetryQueryLock**

indicates the number of query locks that were needed after requests were started.

**Metadata.CounterRetryQueryLockMiss**

indicates the number of missing repository query locks that could not be obtained, and the number of times that query requests were restarted because of missing query locks.

**Metadata.CounterRetryQueryLockRequest**

indicates the number of query requests that were restarted because query locks could not be obtained.

**Metadata.CounterRetryQueryLockHighWater**

indicates the highest number of query lock retries that occurred for a single request. This value cannot exceed the query retry limit, which is specified in the omaconfig.xml file's MAXIMUM\_QUERY\_RETRY\_COUNT option.

**Metadata.CounterRetryQueryLockFailed**

indicates the number of query requests that could not be completed within the query retry limit, which is specified in the omaconfig.xml file's MAXIMUM\_QUERY\_RETRY\_COUNT option. The default for this option is 7 or the number of repositories that were registered when the server was last started, whichever is greater. If the Metadata.CounterRetryQueryLockFailed counter is greater than zero, and if there is a possibility that new repositories were created since the server was last started, then you might want to restart the metadata server in order to reset the default.

For more information about repository query locks, see OMA MAXIMUM\_QUERY\_RETRY\_COUNT on page 284.

**OBJSPAWN.counter-name**

records activity for the selected object spawner session, including the number of servers launched, the number of servers abandoned, and the number of servers that failed to launch.

**Server.information-type**

provides static information about the selected metadata server, OLAP server, object spawner, pooled workspace server, or stored process server session, including the following information:

- the number of CPUs that are available
- the executable that was used to launch the server or spawner
- the identity under which the server or spawner process is executing
- the process ID
- the name of the computer where the server or spawner process is executing

---

## Using the SAS OLAP Server Monitor

---

### Access 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 Administer permission and the Advanced role.
- 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 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. 

---

## View OLAP Server Sessions and Queries

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


- 1 Access the SAS OLAP Server Monitor, as described in the preceding topic.
- 2 Under **Monitoring** and **SAS OLAP Server Monitor**, expand the **Logical OLAP Server** node and double-click the **OLAP Server**. The right pane displays a list of active OLAP sessions. In the Session Owner column, if the server is not secured, the owner is displayed as **<unknown>**.
- 3 To list the active queries for a session, double-click the session in the tree. 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 accessed 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*. 

---

## Using Third-Party Products to Monitor SAS Servers

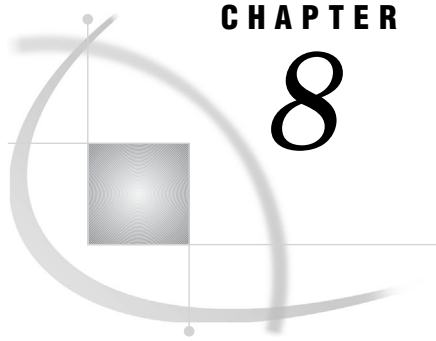
In addition to using SAS Management Console to monitor SAS servers, you can use the following enterprise systems management products:

- BMC Performance Manager (formerly BMC PATROL)

- HP Insight Control Environment (ICE)
- HP OpenView
- Hyperic HQ
- IBM Tivoli
- Xymon

For information about how to enable the use of these products with the SAS Intelligence Platform, see <http://support.sas.com/rnd/emi>.

In addition, SAS provides MBeans that can be used to manage object spawners, metadata servers, workspace servers, pooled workspace servers, stored process servers, and OLAP servers. For details, see “Using JMX Tools to Manage SAS Resources” in the *SAS Intelligence Platform: Web Application Administration Guide*.



## CHAPTER

## 8

# Administering Logging for SAS Servers

<i>About Server Logging</i>	103
<i>About Metadata Server Loggers</i>	104
<i>Enabling Server Logging</i>	106
<i>Initial Logging Configuration for SAS Servers</i>	107
<i>Modifying Your Server Logging Configurations</i>	110
<i>Enable More Detailed Logging for Server Troubleshooting</i>	111
<i>Create a Log File for Workspace Server Troubleshooting</i>	112
<i>Enable More Detailed Logging for SAS Object Spawner Troubleshooting</i>	113
<i>Collecting ARM Log Information for SAS Data Integration Studio Jobs</i>	113
<i>About Collecting ARM Log Information for SAS Data Integration Studio Jobs</i>	113
<i>Configure DATA Step Batch Servers on Windows or UNIX to Collect ARM Log Information</i>	114
<i>Configure DATA Step Batch Servers on z/OS to Collect ARM Log Information</i>	114
<i>Collecting ARM Log Information in SAS OLAP Servers</i>	115
<i>Overview</i>	115
<i>Collect ARM Log Information Without Restarting the SAS OLAP Server</i>	116
<i>Audit Server Logging Level Changes That Originate in Client Applications</i>	116
<i>Capture Metadata Server Management Events in the System Facility Appender</i>	117
<i>Capture XML Information in the SAS Metadata Server Log</i>	117
<i>About Capturing XML Information</i>	117
<i>Configure the Logging of XML Information in the Metadata Server Log</i>	118
<i>Examples of XML Log Entries</i>	119

## About Server Logging

The SAS 9.2 Intelligence Platform uses a standard logging facility to perform logging for SAS servers. The facility supports problem diagnosis and resolution, performance and capacity management, and auditing and regulatory compliance. The logging facility provides the following capabilities for servers:

- Log events are categorized using a hierarchical naming system that enables you to configure logging at a broad or a fine-grained level.
- Log events can be directed to multiple output destinations, including files, operating system facilities, and client applications. For each output destination, you can specify the following logging facility components:
  - the categories and levels of log events to report
  - the message layout, including the types of data to be included, the order of the data, and the format of the data
  - filters based on criteria such as diagnostic levels and message content
- Performance-related log events can be generated for processing by the Application Response Measurement (ARM) 4.0 server.

- You can use SAS Management Console and third party monitoring tools to view the contents of logs.
- You can adjust logging levels dynamically by using the Server Management features of SAS Management Console.

Each server has a logging configuration file that controls the destination, contents, and format of the log for that server. The configuration file specifies options for the following components of the logging facility:

#### *appenders*

are named entities that represent a specific output destination for messages. Destinations include fixed files, rolling files, operating system facilities, and client applications. You can configure appenders by specifying thresholds, filters, log directories and filenames, pattern layouts, and other parameters that control how messages are written to the destination.

#### *loggers*

are named entities that identify message categories. Loggers are named using a hierarchical system that enables you to configure logging at a broad or a fine-grained level.

The logging facility includes a set of high-level loggers for SAS servers, including Audit, Admin, App, IOM, and Perf. Some loggers are subdivided into lower-level (child) loggers. For example, the Audit logger has descendant loggers called Audit.Meta and Audit.Authentication, and Audit.Meta has descendant loggers called Audit.Meta.Security and Audit.Meta.Mgmt. The Root logger is the highest-level logger and does not represent a specific message category.

Loggers inherit settings from their higher-level (ancestor) loggers.

#### *pattern layouts*

are templates that you create to format messages. The pattern layout identifies the types of data, the order of the data, and the format of the data that is generated in a log event and is delivered as output.

For each logger and appender that you specify, you can specify a threshold level that limits the events that will be created or written. Logging levels, from lowest to highest, are ALL, TRACE, DEBUG, INFO, WARN, ERROR, FATAL, and OFF. For example:

- If you specify a threshold level of INFO for a logger, then the logger only creates events with levels of INFO, WARN, ERROR, and FATAL.
- If you specify a threshold level of ERROR for an appender, then events are written to the appender only if they have a level of ERROR or FATAL.

The SAS Deployment Wizard provides default logging configuration files for your SAS servers. You can modify these files as needed to adjust your logging configuration. You can also adjust logging levels dynamically by using the Server Management features of SAS Management Console (see “Using SAS Management Console to Monitor SAS Servers” on page 88).

The following topics describe the initial logging configuration for SAS servers and provide high-level instructions for administering server logging. For detailed information about the logging facility, see the *SAS Logging: Configuration and Programming Reference*.

---

## About Metadata Server Loggers

The logging facility includes loggers for the following categories of events that are specifically related to the SAS Metadata Server:

- security auditing events
- metadata server and metadata repository management events
- change management events
- input and output events
- performance-related events

The initial logging configuration is set up to log these event categories, except for performance-related events, at the INFO, WARN, ERROR and FATAL levels. The loggers are as follows:

#### App.Meta

is the parent logger for metadata server events. Logging levels that are defined for this logger are inherited by its child loggers (including App.Meta.CM, App.Meta.IO, and App.Meta.Mgmt) unless they are explicitly overridden.

#### App.Meta.CM

logs change management events including check-in and check-out.

#### App.Meta.IO

logs low-level input and output activity.

#### App.Meta.Mgmt

logs metadata server management activity such as the following:

- server operation actions such as pause, resume, and refresh
- creating and deleting repositories
- modifying repository access modes
- repository backup and migration

If you want to capture these events in the operating system facility, see “Capture Metadata Server Management Events in the System Facility Appender” on page 117.

#### Audit.Meta.Security

is the parent logger for metadata server security events. No events are written directly to this logger. Logging levels that are defined for this logger are inherited by its child loggers (for example, Audit.Meta.Security.AccCtrlAdm) unless the levels are explicitly overridden.

#### Audit.Meta.Security.AccCtrlAdm

logs changes to permissions, permission settings, and access control templates (ACTs); and changes to passwords for objects such as tables and connections (including adding, deleting, and modifying passwords, as well as failed attempts to perform these actions).

#### Audit.Meta.Security.GrpAdm

logs changes to group or role membership (including adding and removing members, and failed attempts to perform these actions).

#### Audit.Meta.Security.UserAdm

logs changes to users, groups, roles, logins, and authentication domains (including adding, deleting, and modifying these objects, as well as failed attempts to perform these actions).

#### Perf.Meta

logs performance-related events. This information previously was available through the OMA\_ARM subsystem.

The following are examples of security auditing messages.

```
2008-07-22T14:22:14,222 INFO [00222108] 2504:User123@saspw - Added
IdentityType=Person Name=Test User, ObjId=A5LL3EE0.AN00000D.
```

```
2008-07-22T14:22:14,222 INFO [00222108] 2504:User123@saspw - Added Member
IdentityType=Person Name=Test User, ObjId=A5LL3EE0.AN00000D to IdentityGroup
Name=BI Dashboard Users, ObjId=A5LL3EE0.A3000000.
```

```
2008-07-22T14:22:14,222 INFO [00222108] 2504:User123@saspw - Added Login with
UserId=Test User, ObjId=A5LL3EE0.AQ00000F, AuthDomain=DefaultAuth to
IdentityType=Person Name=Test User, ObjId=A5LL3EE0.AN00000D.
```

These messages record the addition of a new user identity (Test User), the addition of the user to the BI Dashboard Users group, and the creation of a login for the user. Each message includes:

- the event date and time
- the logging level
- the thread identifier (in brackets)
- the client ID
- the user identity that is associated with the current thread or task
- the generated message, preceded by a hyphen

*Note:* The metadata server also generates performance counters. For details, see “Using the Server Performance Counters and Information Fields” on page 96. △

---

## Enabling Server Logging

The SAS Deployment Wizard automatically enables logging for most SAS servers. The following steps are necessary to enable server logging:

- 1 A logging configuration file must be created for the server. The logging configuration file is an XML file that specifies and configures loggers and appenders.

The SAS Deployment Wizard automatically creates an initial logging configuration file for each server. Each server's configuration file is named `logconfig.xml` and is located in the directory where the servers' startup script is located. For example, the metadata server's logging configuration file is located in the following path:

*SAS-configuration-directory\Lev1\SASMeta\MetadataServer*

- 2 The `LOGCONFIGLOC=` system option must be specified in the server's `sasv9.cfg` file. This system option specifies the path and filename of the logging configuration file, as in the following example:

```
-logconfigloc "logconfig.xml"
```

If you want to change the option to specify a different path or filename for the logging configuration, then modify the `LOGCONFIGLOC=` option in the server's `sasv9_usermods.cfg` file.

For details about the initial configuration files, see the next topic.



## Initial Logging Configuration for SAS Servers

The following table shows the logging configuration that is created initially for each SAS server. For each server, the table displays the types of events that are written to each log destination.

Server	Log Destination		
	Rolling Log File and SAS Management Console Log Tab	Operating System Facility	ARM Logging Facility
SAS Metadata Server	<input type="checkbox"/> Admin, App, Audit, and IOM events (INFO, WARN, ERROR, and FATAL) <input type="checkbox"/> all other events with a level of ERROR or FATAL	None	None
SAS object spawner	<input type="checkbox"/> App, Audit, and IOM events (INFO, WARN, ERROR, and FATAL) <input type="checkbox"/> all other events with a level of ERROR or FATAL	Admin events (INFO, WARN, ERROR, or FATAL)	None
SAS OLAP Server	<input type="checkbox"/> Admin events (ERROR and FATAL) <input type="checkbox"/> App, Audit, and IOM events (INFO, WARN, ERROR, and FATAL)	Admin.Operation events (ERROR and FATAL)	Performance-related events to support cube testing and tuning (captured in a log file if the logging level is modified. See “Collecting ARM Log Information in SAS OLAP Servers” on page 115.
SAS Pooled Workspace Server	<input type="checkbox"/> Admin events (ERROR and FATAL) <input type="checkbox"/> App, Audit, and IOM events (INFO, WARN, ERROR, and FATAL) <input type="checkbox"/> App.Program events (ERROR and FATAL) <input type="checkbox"/> all other events with a level of ERROR or FATAL  Events are sent only to the rolling log file.	None	None

Server	Log Destination		
	Rolling Log File and SAS Management Console Log Tab	Operating System Facility	ARM Logging Facility
SAS Stored Process Server	<input type="checkbox"/> Admin events (ERROR and FATAL) <input type="checkbox"/> App, Audit, and IOM events (INFO, WARN, ERROR, and FATAL) <input type="checkbox"/> App.Program events (FATAL) <input type="checkbox"/> all other events with a level of ERROR or FATAL		
SAS Table Server	<input type="checkbox"/> Admin events (ERROR or FATAL) <input type="checkbox"/> App, Audit, and IOM events (INFO, WARN, ERROR, and FATAL) <input type="checkbox"/> all other events with a level of ERROR or FATAL	None	None
SAS Workspace Servers	None	Admin.Operation events (ERROR and FATAL)	Performance-related events (available for display in SAS Data Integration Studio)
SAS DATA step batch server	App events (INFO, WARN, ERROR, and FATAL)	Admin.Session events (ERROR and FATAL)	Performance-related events (captured in a log file if the server configuration is modified. See “Collecting ARM Log Information for SAS Data Integration Studio Jobs” on page 113.
SAS/CONNECT server	None	Admin.Session events (INFO, WARN, ERROR, and FATAL)	Performance-related events (available for display in SAS Data Integration Studio)

The log destinations that are shown in the preceding table are configured as follows:

Rolling log files

For metadata servers, object spawners, OLAP servers, pooled workspace servers, stored process servers, and table servers, the logging configuration file includes a definition of `RollingFileAppender`. This appender routes events to a rolling log file, which is configured as follows:

- A new log file is created when the date changes and when a new server or spawner process is started.
- Events are written using a layout that includes the following information:
  - the event date and time
  - the logging level
  - the thread identifier (in brackets)
  - the client ID
  - the user identity that is associated with the current thread or task
  - the generated message, preceded by a hyphen

Here is an example of a metadata server event that was created with the specified layout:

```
2008-10-15T11:42:49,294 INFO [00005671] User123@D1234 - Server
paused. Mode: OFFLINE
```

- The rolling log files are named *server-type\_current-date\_process-id.log*.  
For example, events for metadata server process 6188 on October 12, 2008, are written to a file called `MetadataServer_2008-10-12_6188.log`.
- As a default, the rolling log files are placed in the path *server-configuration-directory\Logs*. For example, log files for the metadata server are written to the following path:  
*SAS-configuration-directory\Lev1\SASMeta\MetadataServer\Logs*  
For a list of default log locations for servers and spawners, see Appendix 3, “Default Locations for Server Logs,” on page 269.  
*Note:* In a custom installation, you can choose to place all of your log files in a single directory. If this option was selected, then your log files are written to the path that was specified during installation (for example, *SAS-configuration-directory\Lev1\Logs*). △
- The log files include a header that identifies the server’s host machine, operating system, operating system release number, SAS release number, and server startup command.
- The workspace server is not initially configured to create a rolling log file. If you need to create a rolling log file for this server for troubleshooting purposes, see “Create a Log File for Workspace Server Troubleshooting” on page 112.

#### SAS Management Console (Log tab)

For metadata servers, object spawners, OLAP servers, stored process servers, and table servers, the logging configuration files include an `IOMServerAppender` definition. This appender routes events to the **Log** tab in SAS Management Console and displays them using the same layout that is used for the rolling files.

For information about viewing logs in SAS Management Console, see “Use the Server Manager Tabs to Obtain Monitoring Information and to Adjust Logging Levels (If Needed)” on page 91.

#### CAUTION:

**Do not modify the message layout in the `IOMServerAppender` definitions that are provided in default logging configuration files. Modifying the layout will result in unpredictable behavior.** However, you can adjust the number of messages that are captured by changing the value for the `MAXENTRIES` parameter. △

### Operating system facilities

For object spawners, OLAP servers, workspace servers, DATA step batch servers, and SAS/CONNECT servers, the logging configuration files include definitions of system facility appenders (WindowsEventAppender, UNIXFacilityAppender, or ZOSFacilityAppender). These appenders route events to the appropriate operating system facility.

The layout for these events includes the system date and time and the message text.

### ARM logging facility

The logging configuration files for the following servers include definitions of ARMAppender, which collects performance-related events that can be processed by an ARM 4.0 server.

- Workspace servers and SAS/CONNECT servers. The logging configuration files for these servers include definitions of sLogAppender. This appender is configured to make performance-related events that are collected by ARMAppender available for display in SAS Data Integration Studio. You do not need to make any configuration changes to enable capture of these events.

#### **CAUTION:**

**Do not modify sLogAppender or ARMAppender definitions that are provided in default logging configuration files. Modifying these definitions will result in unpredictable behavior. Because sLogAppender is a reserved class, you should not define new instances of this appender.**    △

- OLAP servers and DATA step batch servers. To enable the capture of ARM events for these servers, you must make configuration changes. See “Collecting ARM Log Information in SAS OLAP Servers” on page 115 and “Collecting ARM Log Information for SAS Data Integration Studio Jobs” on page 113.

For more information, see the *SAS Interface to Application Response Measurement (ARM): Reference*.

---

## Modifying Your Server Logging Configurations

As a best practice, you should use the initial logging configuration files that are installed by the SAS Deployment Wizard. However, in some situations it might be necessary to make changes to the logging configuration. You can use the following methods to modify a server’s logging configuration:

- You can adjust logging levels dynamically by using the Server Management features of SAS Management Console. This feature is useful when you want to change the configuration temporarily without stopping the server or spawner. See “Using SAS Management Console to Monitor SAS Servers” on page 88.
- You can permanently modify a server’s logging configuration by changing its logconfig.xml file. Before modifying a configuration file, be sure to make a backup copy. The following are some examples of changes that you might want to make to a server’s logging configuration:
  - configure RollingFileAppender to use a different log filename, to roll over log files more or less frequently, or to roll over log files based on file size rather than date
  - specify additional appenders (for example, you might want to define a separate rolling file to capture a particular category of messages)

- use filters to limit the events that are written to an appender
- configure a different message layout for an appender

For additional options and detailed configuration instructions, see the *SAS Logging: Configuration and Programming Reference*.

- You can use the alternative logging configuration files that are provided for troubleshooting. These files are useful when SAS Technical Support requests a higher level of logging. For details, see the following topics:
  - “Enable More Detailed Logging for Server Troubleshooting” on page 111
  - “Create a Log File for Workspace Server Troubleshooting” on page 112
  - “Enable More Detailed Logging for SAS Object Spawner Troubleshooting” on page 113

The following topics describe other logging changes that you might want to make:

- “Collecting ARM Log Information for SAS Data Integration Studio Jobs” on page 113
- “Collecting ARM Log Information in SAS OLAP Servers” on page 115
- “Audit Server Logging Level Changes That Originate in Client Applications” on page 116
- “Capture Metadata Server Management Events in the System Facility Appender” on page 117
- “Capture XML Information in the SAS Metadata Server Log” on page 117

**CAUTION:**

**Excessive logging can degrade performance. Therefore, you should not use the TRACE and DEBUG logging levels unless you are directed to do so by SAS Technical Support.**    △

---

## Enable More Detailed Logging for Server Troubleshooting

To assist in troubleshooting, alternative logging configuration files are provided for some servers, including metadata servers, OLAP servers, pooled workspace servers, stored process servers, table servers, and workspace servers. These configuration files, which are named `logconfig.trace.xml`, provide high-level logging messages (for example, DEBUG and TRACE messages) which can be used by SAS Technical Support to help resolve server issues. The messages are written to the server’s rolling log file.

*Note:* The workspace server is not initially configured to produce a log file. However, the alternative logging configuration file creates a rolling log file for this server for troubleshooting purposes. For details, see “Create a Log File for Workspace Server Troubleshooting” on page 112.    △

If SAS Technical Support requests that you enable detailed logging for server troubleshooting, follow these steps:

- 1 Stop the server, if it is running (for example, if you are enabling detailed logging for a metadata server or OLAP server).

**CAUTION:**

**Stopping the metadata server stops other components, including object spawners, OLAP servers, table servers, SAS/CONNECT spawners, and SAS/SHARE servers.**    △

- 2 Rename the server’s `logconfig.xml` file to `logconfig_orig.xml`.

- 3 Rename the server's logconfig.trace.xml file to logconfig.xml.
- 4 Restart the server if necessary (for example, if you are enabling detailed logging for a metadata server or OLAP server).
- 5 When troubleshooting is completed, stop the server if it is running, rename logconfig.xml to logconfig.trace.xml, rename logconfig\_orig.xml to logconfig.xml, and restart the server if necessary.

*Note:* You can also adjust logging dynamically by using the Server Management features of SAS Management Console. See “Using SAS Management Console to Monitor SAS Servers” on page 88. △

**CAUTION:**

**Do not use or modify the logconfig.trace.xml logging configuration files unless you are requested to do so by SAS Technical Support.** Performance issues can result from the use of these files. △

---

## Create a Log File for Workspace Server Troubleshooting

Standard (non-pooled) workspace servers are not initially configured to produce log files. However, you might need to capture workspace server messages in a log file for troubleshooting purposes. In these situations, you can use the alternative logging configuration file (logconfig.trace.xml) that is provided in each workspace server's configuration directory.

The alternative configuration provides high-level logging messages (for example, DEBUG and TRACE messages) which can be used by SAS Technical Support to help resolve server issues. The messages are written to a rolling log file. The file is written to a Logs directory within the server's configuration directory.

**CAUTION:**

**Do not use or modify the logconfig.trace.xml logging configuration files unless you are requested to do so by SAS Technical Support.** Performance issues can result from the use of these files. △

If SAS Technical Support requests that you enable detailed logging for workspace server troubleshooting, follow these steps:

- 1 If you are enabling workspace server logging to a log file for the first time, you must grant all users of the workspace server Read, Write, and Execute access to the server's **Logs** directory.
- 2 Rename the server's logconfig.xml file to logconfig\_orig.xml.
- 3 Rename the server's logconfig.trace.xml file to logconfig.xml.
- 4 When troubleshooting is completed, do the following:
  - a Rename logconfig.xml to logconfig.trace.xml.
  - b Rename logconfig\_orig.xml to logconfig.xml.

*Note:* You can also adjust logging dynamically by using the Server Management features of SAS Management Console. See “Using SAS Management Console to Monitor SAS Servers” on page 88. △

---

## Enable More Detailed Logging for SAS Object Spawner Troubleshooting

When resolving issues with the SAS object spawner, SAS Technical Support might request that you increase the logging level. To do so, you can use either of the following methods:

- To increase the logging level for multiple spawner sessions (if requested by SAS Technical Support):
  - 1 Open the spawner's logconfig.xml file, and locate the following lines:
 

```
<logger name="App">
  <level value="Info">
</logger>
```
  - 2 Change `<level value="Info">` to `<level value="Trace">`
  - 3 Restart the spawner.
  - 4 When troubleshooting is completed, open logconfig.xml again, change the logging level for the App logger back to **Info**, save the changes, and restart the spawner.
- To temporary increase the logging level (if requested by SAS Technical Support):
  - 1 Connect to the object spawner in SAS Management Console.
  - 2 On the **Loggers** tab, change the App logger to the logging level that was requested by SAS Technical Support. It is not necessary to restart the spawner.
  - 3 When troubleshooting is complete, return the logging level for the App logger to Information. It is not necessary to restart the spawner.

For details about using SAS Management Console to dynamically configure and view logs, see “Using SAS Management Console to Monitor SAS Servers” on page 88.

### CAUTION:

Excessive logging can degrade performance. Do not use the DEBUG or TRACE logging levels unless you are directed to do so by SAS Technical Support. △

---

## Collecting ARM Log Information for SAS Data Integration Studio Jobs

---

### About Collecting ARM Log Information for SAS Data Integration Studio Jobs

When jobs are run interactively in SAS Data Integration Studio, ARM logging is used to collect performance-related events, which are displayed in the SAS Data Integration Studio application. You do not need to change the initial server configuration to enable the capture and display of these events.

To collect performance-related events for SAS Data Integration Studio jobs that are run in batch mode, you must change the configuration of the SAS DATA step batch server that executes the jobs. The logging configuration files for SAS DATA step batch servers include specifications for collecting ARM log information and sending it to a log file. However, logging is not enabled by default for these servers. You must configure your server so that logging is enabled.

If you have selected the **Collect Runtime Statistics** option for a SAS Data Integration Studio job, but ARM logging is not configured correctly for the DATA step batch server, then messages similar to the following will appear in the log when the job is run:

**WARNING: ARMINIT: Negative appid returned - check macro parameters, appid=-1(0xFFFFFFFF)**

---

## Configure DATA Step Batch Servers on Windows or UNIX to Collect ARM Log Information

To configure your DATA step batch server on Windows or UNIX so that ARM logging is enabled for SAS Data Integration Studio Jobs, follow these steps:

- 1 Log on to SAS Management Console as a user who has the appropriate credentials to create or update server metadata (see “Who Can Do What in the Server Manager Plug-in” on page 35).
- 2 Expand the **Server Manager** node.
- 3 Expand the node for the application server that executes your SAS Data Integration Studio jobs (for example, **SASApp**). Then expand the node for the logical DATA step batch server (for example, **SASApp - Logical DATA Step Batch Server**).

*Note:* You cannot use the SASMeta application server to run SAS Data Integration Studio jobs. It is intended only for use in running metadata utilities. △

- 4 Right-click the node for the physical DATA step batch server (for example, **SASApp - SAS DATA Step Batch Server**, and select **Properties**.
- 5 Select the **Options** tab, and add the following expression to the contents of the **Command Line** field:

```
-logconfigloc "file-specification"
```

In *file-specification*, provide the fully qualified path to the logging configuration file. Here is an example command line for a Windows system with the logging option added:

```
C:\SAS\Config\Lev1\SASApp\BatchServer\sasbatch.bat
-logconfigloc "C:\SAS\Config\Lev1\SASApp\BatchServer\logconfig.xml"
```

Here is an example command line for a UNIX system with the logging option added:

```
/opt/SAS/Lev1/SASApp/BatchServer/sasbatch.sh
-logconfigloc "/opt/SAS/Lev1/SASApp/BatchServer/logconfig.xml"
```

- 6 On the **Options** tab, click **Advanced Properties**. In the Advanced Properties dialog box, delete the contents of the **Rolling Log Options** field and replace them with a single underscore character (\_).
- 7 Click **OK** to close the Advanced Properties dialog box, and click **OK** to close the **Options** tab.

---

## Configure DATA Step Batch Servers on z/OS to Collect ARM Log Information

You can use the following procedure to configure your DATA step batch server on z/OS so that ARM logging is enabled for a SAS Data Integration Studio job:



- 1 Create a configuration file with the following contents:

```
***** Top of Data *****
000001 logconfigloc=/<SAS-configuration-directory>/+
000002 Lev1/SASApp/BatchServer/logconfig.xml
000003 log=/<SAS-configuration-directory>/+
000004 Lev1/SASApp/BatchServer/Logs/<name-of-job>
***** Bottom of Data *****
```

*Note:*

- Replace **<SAS-configuration-directory>** with the location that is specified in the CONFIG\_DIR environment variable.
- If your application server has a name other than SASApp, then replace **SASApp** with the appropriate characters (for example, **SASMain**).
- Replace **<name-of-job>** with the name of the SAS Data Integration Studio job for which you want to collect ARM log information.

△

- 2 Save the configuration file in your SAS installation location.
- 3 In the JCL for each job that you want to collect ARM log information for, modify the //SASSTEP statement as follows so that it refers to the configuration file:

```
//SASSTEP EXEC PROCENW0,CONFIG=<SAS-installation-  
location>.<configuration-filename>(DSBS)
```

For example:

```
//SASSTEP EXEC PROCENW0,CONFIG=SVS02.V920.KJB.CONFIG(DSBS)
```

---

## Collecting ARM Log Information in SAS OLAP Servers

---

### Overview

By default, SAS OLAP Servers collect log information in a SAS system log and an ARM log. The system log is displayed in the **Log** tab of the Server Manager when you select a SAS OLAP Server. The system log is also stored in a file that is refreshed daily. The ARM log is stored in a file that is created at server start. To change the default logging behavior, see “Initial Logging Configuration for SAS Servers” on page 107.

The ARM log receives no messages under normal operation, because the logger (Perf.ARM.OLAP\_Server) is set to the WARN level.

During the testing and tuning of cubes, you change the logging level from WARN to INFO so that the ARM log begins to receive messages.

The Aggregation Tuning plug-in in SAS OLAP Cube Studio reads ARM logs to generate aggregations for common queries, as described in the *SAS OLAP Cube Studio: User's Guide*.

Be sure to truncate or delete large log files as necessary to maintain free space on disk.

### CAUTION:

To ensure that you capture accurate log information, configure load-balancing clusters of SAS OLAP Servers so that each server writes to a separate ARM log file. △

For information on configuring ARM logs in load-balancing clusters, see the “Understanding Load Balancing” chapter of the *SAS Intelligence Platform: Application Server Administration Guide*.

---

## Collect ARM Log Information Without Restarting the SAS OLAP Server

Follow these steps to enable ARM logging without restarting the SAS OLAP Server:

### Prerequisite

Required OLAP server system option: ARMAGENT=LOG4SAS. This value is set by default when you installed the server.

- 1 Start SAS Management Console and select a connection profile that has Administer permission. Authenticate yourself as directed to connect to a metadata server.
- 2 Expand the **Server Manager** node, the SAS Application Server node (for example, **SASApp**), the node for the logical OLAP server, and the OLAP server instance.
- 3 Right-click the OLAP server instance and select **Connect**. Tabs appear in the right pane.
- 4 In the **Loggers** tab, right-click the **Perf.ARM.OLAP\_Server** logger and select **Properties**.
- 5 In the Properties window, select the **Assigned** radio button.
- 6 In the **Assigned** field, click the down arrow and select **Information**.
- 7 Click **OK** to close the Properties window and begin logging.
- 8 Submit test queries to collect information in the ARM log file.
- 9 To disable logging, change the **Assigned** field back to the **Warning** level.

---

## Audit Server Logging Level Changes That Originate in Client Applications

For security purposes, you might want to audit changes to server logging levels that are initiated by remote client applications. SAS Management Console is the only SAS application that enables users to change logging levels. The `App.FilteredList.FilteredList.SetValue` logger captures any logging level changes that are initiated by SAS Management Console or by custom applications.

To capture these events, add the following lines to the server's logging configuration file:

```
<logger name="App.FilteredList.FilteredList.SetValue">
  <level value="Trace">
</logger>
```

If you want, you can direct these events to a separate log file. To do so, define a new appender, and reference the appender in the logger definition. For details, see the *SAS Logging: Configuration and Programming Reference*.

You can also use SAS Management Console to temporarily adjust the threshold level for this logger (see “Using SAS Management Console to Monitor SAS Servers” on page 88).

## Capture Metadata Server Management Events in the System Facility Appender

The App.Meta.Mgmt logger logs metadata server management activity such as the following:

- server operation actions such as pause, resume, and refresh
- creation and deletion of repositories
- modification of repository access modes
- repository backup and migration

You might want to capture these events in your operating system facility. To do so, add an instance of the appropriate system facility appender to the metadata server's logging configuration file, and direct App.Meta.Mgmt events to that appender.

For Windows systems, add the following lines to the logging configuration file:

```
<appender name="eventLog" class="WindowsEventAppender">
  <param name="Threshold" value="Warn"/>
  <layout>
    <param name="ConversionPattern" value="%d{yyyyMMdd:HH:mm:ss.SS}: %m"/>
  </layout>
</appender>
<logger name="App.Meta.Mgmt">
  <level value="Info"/>
  <appender-ref ref="eventLog"/>
</logger>
```

On UNIX systems, replace **WindowsEventAppender** with **UNXFacilityAppender**. On z/OS systems, replace **WindowsEventAppender** with **ZOSFacilityAppender**.

## Capture XML Information in the SAS Metadata Server Log

### About Capturing XML Information

To assist in troubleshooting, SAS Technical Support might request that you capture generated XML entries in the metadata server log.

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

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 a client requests 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.

These XML entries can be captured in the metadata server log and used for troubleshooting.

*Note:* For more information about the call interfaces, see the *SAS Language Interfaces to Metadata*. △

---

## Configure the Logging of XML Information in the Metadata Server Log

If SAS Technical Support requests that you capture XML entries in your metadata server log, you will need to do the following:

- Add the JNLSTRMAX and JNLLINEMAX object server parameters to the server's configuration file.
- Increase the logging level for the App.OMI logger to TRACE.
- Direct the XML entries to a separate log file, if you want to keep these entries separate.

Here are detailed instructions:

- 1 Add the following line to the metadata server's sasv9\_usermods.cfg file:

```
-objectserverparms "jnlStrMax=10000 jnlLineMax=10000"
```

*Note:* In configuration files on z/OS, do not precede **objectserverparms** with a hyphen. △

The file is located in the following path:

*SAS-configuration-directory*\Lev1\SASMain|SASMeta\MetadataServer

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 8.1** OBJECTSERVERPARMS Parameters Required to Capture XML in the Metadata Server's Log File

Parameter	Explanation
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: <b>*truncated to 500 characters (jnlStrMax)</b> .
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: <b>*truncated to 1000 characters (jnlLineMax)</b> .

- 2 Add the following lines to metadata server's logconfig.xml file, which is located in *SAS-configuration-directory*\Lev1\SASMain|SASMeta\MetadataServer::

```
<appender class="FileAppender" name="IOMCALLTRACE">
  <param name="File" value="omasrv_iomcalltrace.log"/>
  <param name="ImmediateFlush" value="true"/>
  <param name="Append" value="false"/>
</layout>
```

```

        <param name="ConversionPattern" value="%d %-5p [%t] %c (%F:%L) - %m"/>
    </layout>
</appender>
<logger name="App.OMI">
    <level value="trace"/>
    <appender-ref ref="IOMCALLTRACE"/>
</logger>

```

If you want, you can specify different values (for example, a different log filename) for the FileAppender parameters. For details, see “FileAppender” in the *SAS Logging: Configuration and Programming Reference*.

- 3 Restart the metadata server.

---

## 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 the receipt of a GetMetadataObjects request:

```

2007-11-06T15:37:55,815 TRACE [00000033] App.OMI.OMI.DoRequest
(CIOMomici.c:6239) - IOM CALL {compRef:5e274b0}->OMI::DoRequest():
  inString=<GetMetadataObjects><Reposid>A0000001.A0000001</Reposid>
<Type>RepositoryBase</Type><Objects/><ns>REPOS</ns>
<Flags>264</Flags><options/></GetMetadataObjects>

```

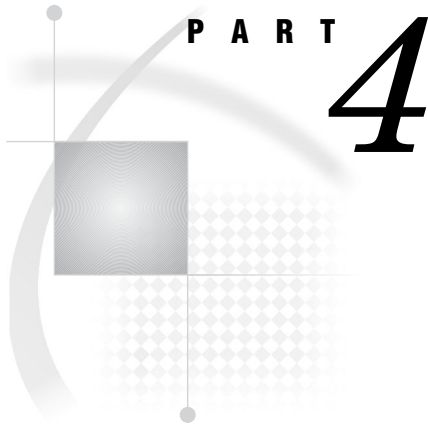
The following is an example of a log entry for a response to a client request:

```

2007-11-06T15:37:55,815 TRACE [00000033] App.OMI.OMI.DoRequest
(CIOMomici.c:6504) - IOM RETURN 0={compRef:5e274b0}->OMI::DoRequest():
  outString=<GetMetadataObjects><Reposid>A0000001.A0000001</Reposid>
<Type>RepositoryBase</Type><Objects><RepositoryBase Id="A0000001.A510HYM9"
Name="Foundation" Access="0" Desc="" Engine="BASE"
MetadataCreated="26Sep2007:16:12:48" MetadataUpdated="02Nov2007:18:10:37"
Options=""
Path="C:\SAS\DIServer\Lev1\SASMain\MetadataServer\MetadataRepositories\Foundation"
RepositoryFormat="11" RepositoryType="FOUNDATION"/><RepositoryBase
Id="A0000001.A5PJCVL" Name="Custom1" Access="0" Desc="" Engine="BASE"
MetadataCreated="26Sep2007:16:14:28" MetadataUpdated="02Nov2007:18:10:37"
Options=""
Path="C:\SAS\DIServer\Lev1\SASMain\MetadataServer\MetadataRepositories\Custom1"
RepositoryFormat="11" RepositoryType="CUSTOM"/><RepositoryBase
Id="A0000001.A5FIU304" Name="Project1" Access="0" Desc="" Engine="BASE"
MetadataCreated="26Sep2007:16:15:53" MetadataUpdated="02Nov2007:18:10:37"
Options=""
Path="C:\SAS\DIServer\Lev1\SASMain\MetadataServer\MetadataRepositories\Project1"
RepositoryFormat="11" RepositoryType="PROJECT"/></Objects><ns>REPOS</ns>
<Flags>264</Flags><options/></GetMetadataObjects> retval=0

```



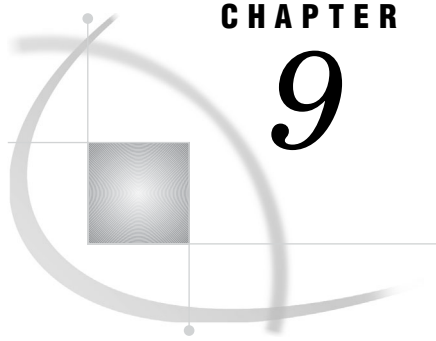


## Backing Up and Restoring Your System

- Chapter 9*.....**Best Practices for Backing Up and Restoring Your System** 123
- Chapter 10*.....**Using the Backup Wizard in SAS Management Console** 137
- Chapter 11*.....**Using the %OMABAKUP Macro to Perform Backups and Restores** 147
- Chapter 12*.....**Manually Restoring an Unresponsive SAS Metadata Server** 161







## CHAPTER

## 9

# Best Practices for Backing Up and Restoring Your System

<i>About Backups and Restores</i>	123
<i>About the Metadata Server Backup and Restore Utilities</i>	124
<i>Using Operating System Commands to Back Up the Metadata Server</i>	126
<i>About Using the Export SAS Package Wizard to Back Up Specific SAS Folders</i>	126
<i>Best Practices for Backing Up Your SAS System</i>	127
<i>Summary of Best Practices for Backing Up Your SAS System</i>	127
<i>Always Use the Backup Wizard or %OMABAKUP to Back Up the Repository Manager and the Metadata Repositories</i>	128
<i>Perform Regularly Scheduled Full Backups</i>	128
<i>Perform Backups Before and After Major Changes</i>	129
<i>Schedule Backups When Metadata Server Demand is Low</i>	129
<i>Specify a Backup Destination That Is Included in Daily System Backups</i>	129
<i>Use Alternating Backup Locations</i>	130
<i>Occasionally Use the REORG Option to Reclaim Unused Disk Space</i>	130
<i>Occasionally Use the RUNANALYSIS Option to Optimize Metadata Server Memory Usage</i>	130
<i>Back Up the SAS Content Server</i>	131
<i>Back Up the SAS Table Server and the SAS Shared Services Database</i>	131
<i>Back Up Data That Is Stored in File Systems</i>	132
<i>Synchronize Metadata Backups with SAS Content Server, SAS Table Server, and Physical File Backups</i>	132
<i>Recommended Approach for Synchronizing Metadata Backups with Physical Backups</i>	132
<i>Example of a PROC METAOPERATE Program that Pauses the Metadata Server to an Offline State</i>	133
<i>Example of a PROC METAOPERATE Program that Resumes the Metadata Server</i>	134
<i>Best Practices for Restoring Your System</i>	134
<i>Perform Full Restores</i>	134
<i>Restoring an Unresponsive Metadata Server</i>	135


## About Backups and Restores

To ensure the integrity of your SAS Intelligence Platform installation, you should establish a formal, regularly scheduled backup process. It is important to back up all of the following items at the same point in time so that related information will be synchronized if a restore becomes necessary:

- your metadata repositories, the repository manager, the metadata journal file, and the metadata server's configuration files. SAS provides a macro called %OMABAKUP that creates correct backups of these files while minimizing disruptions in service. Backup and restore programs that use this macro are provided with your installation. You can also use the Backup Wizard in SAS Management Console to create and run %OMABAKUP jobs without writing code,

or you can write your own backup and restore programs that use the %OMABAKUP macro.

**CAUTION:**

**If you choose to use operating system commands to back up your metadata repositories and the repository manager, instead of using the Backup Wizard or %OMABAKUP, then you must be sure to pause the metadata server to an Offline state before performing the backup. If the metadata server is Online or is paused to an Administration state, then the backup files will not be usable.** 

- the SAS Content Server, the SAS Table Server, and the physical content that is stored in files and databases. Because the Backup Wizard and %OMABAKUP do not back up these files, you must use operating system commands or third-party tools to back them up.

In addition to performing regular full backups, in certain situations it might be appropriate to back up specific objects or folders in the **SAS Folders** tree. In these situations, you can use the promotion tools, which include the Export SAS Package wizard, the Import SAS Package wizard, and the batch export and import tools. For details, see Chapter 17, “Using the Promotion Tools,” on page 213.

Before you back up your SAS Intelligence Platform installation, read the following information:

- “About the Metadata Server Backup and Restore Utilities” on page 124
- “Using Operating System Commands to Back Up the Metadata Server” on page 126
- “About Using the Export SAS Package Wizard to Back Up Specific SAS Folders” on page 126
- “Best Practices for Backing Up Your SAS System” on page 127
- “Best Practices for Restoring Your System” on page 134

---

## About the Metadata Server Backup and Restore Utilities

The SAS Intelligence Platform provides the following methods for performing a correct backup or restore of the metadata server while minimizing disruptions in service:

- You can execute the backup program (backupServer.sas) and restore program (restoreServer.sas) that are provided with your installation. The backupServer.sas program uses the %OMABAKUP macro to create a correct backup of your metadata repositories, the repository manager, the metadata journal file, and the metadata server’s configuration files, including omaconfig.xml, adminUsers.txt, trustedUsers.txt, and logconfig.xml. The restoreServer.sas program uses the %OMABAKUP macro to restore all of these items except the journal file.

The backupServer.sas and restoreServer.sas programs are customized for your environment and are located in the metadata server’s startup directory. If needed, you can modify these programs to specify a different backup location or to incorporate other %OMABAKUP options.

To execute these programs, you can use the **MetadataServer -backup** and **MetadataServer -restore** commands (on Windows), and the **MetadataServer.sh -backup** and **MetadataServer.sh -restore** commands (on UNIX and z/OS).

%OMABAKUP pauses the metadata server while retaining client connections, closes the repository data sets, and copies the metadata server files to the specified backup destination. It then automatically resumes the server and the handling of client requests. Your daily system backup process can then copy the files from the backup destination without needing to pause the server.

- You can use the Backup Wizard in SAS Management Console to create backup jobs that use %OMABAKUP without writing code. You can select most of the same options that are available in custom %OMABAKUP programs. However, you cannot back up repositories that have a registered access state of Offline; and you cannot create or run a backup job if the foundation repository has a registered access state of ReadOnly. For these types of backups, you must use a custom %OMABAKUP program.

From the wizard, you can choose to execute the backup job, save it as a .sas file, or deploy it for scheduling. For each backup job that you create, the wizard automatically creates a corresponding restore job. Jobs that are created by the wizard execute on a workspace server that is local to the metadata server machine.

- The backupServer.sas program and the Backup Wizard should be sufficient for most situations. However, if the need arises, you can write custom backup and restore programs that contain the %OMABAKUP macro.

The following table summarizes the differences and similarities among jobs that are created by the Backup Wizard and custom programs (including backupServer.sas) that are created with %OMABAKUP:

Type of Repository or File	Registered Access Mode	Job Created with Backup Wizard	Custom Program Created with %OMABAKUP*
Foundation	Online	Always backed up	Always backed up
	Administration	Always backed up	Always backed up
	ReadOnly	Cannot be backed up	Backed up unless DISABLEREADONLYBACKUP is set to YES
	Offline	Cannot be backed up	Backed up unless the DISABLEOFFLINEBACKUP option is set to YES
Project or custom	Online	Always backed up	Always backed up
	Administration	Always backed up	Always backed up
	ReadOnly	Backed up, unless the <b>Back Up READONLY Repositories</b> option is deselected	Backed up unless the DISABLEREADONLYBACKUP option is set to YES
	Offline	Cannot be backed up	Backed up unless the DISABLEOFFLINEBACKUP option is set to YES
Metadata journal file	Not applicable	Backed up, unless the <b>Back Up Metadata Journal File</b> option is deselected	Backed up, unless the BACKUPMETADATAJOURNALFILE option is set to NO

Type of Repository or File	Registered Access Mode	Job Created with Backup Wizard	Custom Program Created with %OMABAKUP*
Repository manager	Not applicable	Always backed up	Always backed up
Metadata server configuration files	Not applicable	Backed up, unless the <b>Back Up Configuration Files</b> option is deselected	Backed up, unless the DISABLECONFIGURATIONFILEBACKUP option is set to YES

\* An example of a custom program is **backupServer.sas**, which is located in your metadata server startup directory.

For usage documentation for these backup methods, see the following:

- “Running the backupServer.sas and restoreServer.sas Programs” on page 150
- Chapter 10, “Using the Backup Wizard in SAS Management Console,” on page 137
- “Writing Custom Programs Using %OMABAKUP” on page 151

Before using any of these utilities, be sure to read “Best Practices for Backing Up Your SAS System” on page 127.

## Using Operating System Commands to Back Up the Metadata Server

Instead of using the Backup Wizard or %OMABAKUP, you can choose to use operating system commands to back up your metadata repositories and the repository manager. If you use this method, then you must be sure to pause the metadata server to an Offline state before performing the backup. If the metadata server is Online or is paused to an Administration state, then the backup files will not be usable.

*Note:* External systems such as Windows Services Manager report the Offline state as **paused**. △

You can use PROC METAOPERATE to pause the server to an Offline state before the backup is taken and to resume the server to an Online state when the backup is complete. For details, see the following:

- “Example of a PROC METAOPERATE Program that Pauses the Metadata Server to an Offline State” on page 133
- “Example of a PROC METAOPERATE Program that Resumes the Metadata Server” on page 134

You can also pause and resume the server by using SAS Management Console or a server management script (see Chapter 5, “Operating Your Servers,” on page 51). Be sure to pause the server to an Offline state (rather than the Administration state).

## About Using the Export SAS Package Wizard to Back Up Specific SAS Folders

In certain situations it might be appropriate to make backups of specific SAS folders that are used to organize user-created SAS content. To perform selective backups and

restores of objects in these folders (as well as physical files associated with these objects), you can use the following tools:

- the Export SAS Package and Import SAS Package wizards, which are available in the **Folders** tab of SAS Management Console. The same wizards are also available in SAS Data Integration Studio and SAS OLAP Cube Studio.
- the batch export tool and the batch import tool, which enable you to perform exports and imports on a scheduled or repeatable basis. These tools provide most of the same capabilities as the Export SAS Package and Import SAS Package wizards.

The following are examples of situations in which you might want to back up individual folders:

- If you are creating new content (for example, new information maps and new reports) in a development environment, you might want to periodically back up the folders and associated physical files that contain this content.
- If you are about to import new content (for example, new information maps, stored processes, and reports) into production folders, you might want to first back up the current contents of the production folders and their associated physical files.

For details about using the Export SAS Package and Import SAS Package wizards and the batch import and export tools, see Chapter 17, “Using the Promotion Tools,” on page 213.

---

## Best Practices for Backing Up Your SAS System

---

### Summary of Best Practices for Backing Up Your SAS System

The following are the recommended best practices for backing up your SAS system:

- Always use the Backup Wizard or %OMABAKUP.
- Perform regularly scheduled full backups of the metadata server, including the repository manager, all active repositories, the metadata server’s configuration files, and the metadata journal file.
- Perform full backups of the metadata server before and after major changes.
- Schedule backups when metadata server demand is low.
- Specify a backup destination that is included in daily system backups.
- Use alternating backup locations if the backup destination is not included in daily system backups.
- Occasionally use the REORG option to reclaim unused disk space.
- Occasionally use the RUNANALYSIS option to optimize metadata server memory usage.
- Back up the SAS Content Server.
- Back up the SAS Table Server and the SAS Shared Services database.
- Back up data that is stored in file systems.
- Synchronize metadata backups with backups of the SAS Content Server, SAS Table Server, and physical files.

The following topics describe the best practices in detail.

---

## Always Use the Backup Wizard or %OMABAKUP to Back Up the Repository Manager and the Metadata Repositories

The processes that you use to make regular system backups of your file system cannot directly access the metadata repositories, because the repository data sets are normally locked. Therefore, the recommended method for backing up your metadata repositories, the repository manager, the metadata journal file, and the metadata server's configuration files is to use the %OMABAKUP macro. (To use this macro, you can execute the backupServer.sas program that is provided with your installation, run the Backup Wizard, or write a custom %OMABAKUP program.) Your daily system backup process can then copy the files from the backup destination without needing to pause the server.

### CAUTION:

**If you choose to use operating system commands to back up your metadata repositories and the repository manager, instead of using the Backup Wizard or %OMABAKUP, then you must be sure to pause the metadata server before performing the backup. Otherwise, the backup files will not be usable.** △

The %OMABAKUP macro and the Backup Wizard provide the following advantages:

- They automatically pause the repository manager and the repositories that are being backed up or restored, and they quickly resume 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.
- As a default, they back up the repository manager, the metadata journal file, your registered metadata repositories (depending on their access modes), and the metadata server's configuration files. For details about which repository access modes are supported by each tool, see “About the Metadata Server Backup and Restore Utilities” on page 124.
- 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.

---

## Perform Regularly Scheduled Full Backups

To safeguard the integrity of your metadata, make regular full backups (rather than incremental backups) of your metadata repositories, the repository manager, the metadata journal file, and the metadata server's configuration files. If your metadata is being updated continually (for example, if information maps, reports, or user identities are created or modified frequently), then you should perform full backups on a nightly basis.

By performing full backups, you will ensure that associations between metadata are correctly maintained and that your metadata repositories can be restored to a consistent state. The best way to ensure that your backups meet this requirement is to use the Backup Wizard to create your backup job, or use %OMABAKUP to create a custom backup program.

To maintain system integrity, you must also back up the SAS Content Server, the SAS Table Server, and the physical files that are associated with your metadata. See the following topics:

- “Back Up the SAS Content Server” on page 131

- “Back Up the SAS Table Server and the SAS Shared Services Database” on page 131
- “Back Up Data That Is Stored in File Systems” on page 132
- “Synchronize Metadata Backups with SAS Content Server, SAS Table Server, and Physical File Backups” on page 132

---

## Perform Backups Before and After Major Changes

In addition to your regular full backup schedule, you should also perform a full backup before and after making major changes to your system. For example, you should perform a backup on the following occasions:

- after you complete your initial installation and configuration
- before and after you upgrade your system to a new release or service pack
- after you add a new repository
- before you delete a repository

The Backup Wizard is a useful tool for performing ad hoc (unscheduled) backups. You can also use %OMABAKUP to create a custom backup job.

In unscheduled backup situations that involve associated content and files (for example, the addition of new reports or stored processes), be sure to include the SAS Content Server, the SAS Table Server, and the physical files in your backup. If the backup does not involve a large amount of data, then the Import and Export wizards can be useful for this type of backup (see “About Using the Export SAS Package Wizard to Back Up Specific SAS Folders” on page 126).

In unscheduled backup situations where the metadata has changed but no physical files are involved, it is sufficient to back up just the repository manager, the metadata repositories, and the metadata server’s configuration files. For example, no physical files are affected when you make changes to server definitions, user and group definitions, and access control templates. If you choose not to back up the physical files, you should make sure that no user activity that affects physical files has taken place since the last backup was taken. For example, make sure that no stored processes, jobs, or reports have been created or modified since the last backup.

---

## Schedule Backups When Metadata Server Demand is Low

When you run an %OMABAKUP job or a job created by the Backup Wizard, the metadata server is paused and the repositories are changed to Offline status. To minimize the impact of repository downtime on clients, you should execute these jobs during periods when demand for the metadata server is low.

---

## Specify a Backup Destination That Is Included in Daily System Backups

If the backup destination already contains backup files, the Backup Wizard or %OMABAKUP overwrites them with the new backup files. To ensure that you can restore metadata repository files from a previous backup, you should specify a backup destination that is included in your regular system backups. Then if a restore is necessary, you can obtain the files from the appropriate system backup.

As a default, the backupServer.sas program and the Backup Wizard write backup files to the **SASBackup** subdirectory of the metadata server startup directory. If you use the default destination, make sure that the **SASBackup** directory is included in your regular system backups.

---

## Use Alternating Backup Locations

If the backup destination is not included in daily system backups, then you should 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 metadata repositories and the repository manager from another location. For a sample backup program that uses this method, see “Example: Backup Program on Windows, Using Alternating Destinations” on page 154.

---

## Occasionally Use the REORG Option to Reclaim Unused Disk Space

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

- 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.

---

## Occasionally Use the RUNANALYSIS Option to Optimize Metadata Server Memory Usage

When you use the Backup Wizard or %OMABAKUP to perform a backup, you should occasionally use the RUNANALYSIS option. This option checks 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 the Backup Wizard or %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 use the RUNANALYSIS option 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 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 metadata server. To minimize the impact of the character variable analyses on the 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 information about using these features, see “Using %OMARUNAN (or the RUNANALYSIS Option of %OMABAKUP) to Optimize Memory Usage” on page 175.



## Back Up the SAS Content Server

If your deployment includes SAS Web applications, then it is important to back up the SAS Content Server when you back up the metadata server. The SAS Content Server contains content that is associated with metadata objects that are contained in the repositories, including the following:

- content for the SAS Information Delivery Portal
- report definition files
- other supporting files for reports, including PDF files and image files
- content for SAS solutions

Metadata cannot be used without the associated content that is stored on this server. If you need to recover from a failure, the metadata that you restore must be synchronized correctly with the content server.

To back up the SAS Content Server:

- 1 As a best practice, stop either the Web application server or the SAS Content Server application before making the backup.

If you are using an IBM WebSphere Application Server or an Oracle WebLogic Server, then you can stop the SAS Content Server application.

If you are using a JBoss Application Server, then you must stop the server.

- 2 Use operating system commands or third-party tools to copy all of the files and subdirectories from the following path:

*SAS-configuration-directory*\Lev1\AppData\SASContentServer\Repository

If you need to back up just a subset of the SAS Content Server, or if you want to create a backup for input to a system other than the SAS Content Server, you can use the WebDAVDump and WebDAVRestore utilities. For instructions, see SAS Usage Note 38667.

## Back Up the SAS Table Server and the SAS Shared Services Database

If your deployment includes SAS Web applications, then it is important to back up the SAS Table Server. Follow these steps:

- 1 Stop the SAS Table Server.
- 2 Use operating system commands or third-party tools to copy all of the files and subdirectories from the following path:

*SAS-configuration-directory*\Lev1\SASTS

You must also back up the Shared Services database that exists on the table server. Use the following command:

```
sastback -b SharedServices.fdb -mo read_only [-verbose] SharedServices.fbk
```

If you need to restore the Shared Services database, use the following command:

```
sastback -r SharedServices.fbk [-verbose] SharedServices.fdb
```

The SASTBACK executable is located in the following path:

- On Windows: *SAS-installation-directory*\SASFoundation\9.2\vulcan\bin
- On UNIX and z/OS: *SAS-installation-directory*/SASFoundation/9.2/vulcan/bin

For more information about the Shared Services database, see the *SAS Intelligence Platform: Web Application Administration Guide*.

---

## Back Up Data That Is Stored in File Systems

When you back up the metadata server, it is important to also back up the physical 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.

Since the Backup Wizard and %OMABAKUP do not back up the physical data that is associated with the metadata, you must use operating system commands or third-party tools to back up the physical data.

Your backup strategy should include the following data:

- portal content that is not stored in the SAS Content Server, including custom portlets, custom themes, and applications
- report definition files and other supporting files for reports (for example, PDF files and image files) that are stored in your file system, if you are not storing these items on the SAS Content Server
- 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

---

## Synchronize Metadata Backups with SAS Content Server, SAS Table Server, and Physical File Backups

### Recommended Approach for Synchronizing Metadata Backups with Physical Backups

To maintain system integrity in the event that a restore is needed, your backups of the SAS Content Server, SAS Table Server, and physical data must be synchronized correctly with your metadata backups. To ensure that the backups are synchronized, you can use one of the following approaches:

- Back up the metadata server, the SAS Content Server, the SAS Table Server, and the physical files concurrently (that is, in the same backup window).
- Back up the SAS Content Server, the SAS Table Server, and the physical files immediately after the metadata server is backed up, and do not allow clients to update metadata while you are performing these backups. You can accomplish this in either of the following ways:

- If you are running the backup on a batch basis (for example, as part of a daily schedule), then you can do the following:
  - 1 Write a program that uses PROC METAOPERATE to pause the metadata server to an Offline state. See “Example of a PROC METAOPERATE Program that Pauses the Metadata Server to an Offline State” on page 133.
 

If you use the Backup Wizard or %OMABAKUP to back up the metadata server, then the server is paused automatically. However, you can use this program to pause the metadata server while you back up the SAS Content Server, SAS Table Server, and associated physical data.

If you use operating system commands to back up the metadata server, then you can use this program to pause the server before running the backup.
  - 2 Write another program that resumes the metadata server to an Online state. See “Example of a PROC METAOPERATE Program that Resumes the Metadata Server” on page 134. You can use this program after using operating system commands to back up the metadata server, or you can use it after backing up the SAS Content Server, SAS Table Server, and associated physical data.
- If you are running an ad hoc (unscheduled) backup and you need to also back up associated data, then you can do the following:
  - 1 Use the Backup Wizard or %OMABAKUP to back up the metadata server. Then immediately use SAS Management Console to pause the metadata server (see “Using SAS Management Console to Operate SAS Servers” on page 65). As an alternative, you can use SAS Management Console to temporarily change the registered access mode of the repositories to ReadOnly (see “Change the Properties of a Metadata Repository” on page 184).
  - 2 Back up the SAS Content Server, the SAS Table Server, and the physical data.
  - 3 When you are finished backing up the SAS Content Server, the SAS Table Server, and the physical data, use SAS Management Console to resume the metadata server (or to return the registered access mode of the repositories to Online).

### Example of a PROC METAOPERATE Program that Pauses the Metadata Server to an Offline State

Here is an example program that uses PROC METAOPERATE to pause the metadata server to an Offline state.

If you use operating system commands to back up the metadata server, you can use this program to pause the server before running the backup.

If you use the Backup Wizard or %OMABAKUP to back up the metadata server, then the server is paused automatically. However, you can use this program to pause the metadata server while you back up the SAS Content Server, SAS Table Server, and associated physical data.

```
proc metaoperate
  server="host-name"
  port=8561
  userid="user-ID"
  password="encoded-password"
```

```

        protocol=bridge
        action=pause
        noautopause;

run;

```

The user that is specified in *user-ID* must be assigned to the **Metadata Server: Operation** role.

### Example of a PROC METAOPERATE Program that Resumes the Metadata Server

Here is an example program that uses PROC METAOPERATE to resume the metadata server to an Online state. You can use this program after using operating system commands to back up the metadata server, or you can use it after backing up the SAS Content Server, SAS Table Server, and associated physical data.

```

proc metaoperate
    server="host-name"
    port=8561
    userid="user-ID"
    password="encoded-password"
    protocol=bridge
    action=resume
    noautopause;

run;

```

The user that is specified in *user-ID* must be assigned to the **Metadata Server: Operation** role.

For detailed information about PROC METAOPERATE, see *SAS Language Interfaces to Metadata*.

---

## Best Practices for Restoring Your System

---

### Perform Full Restores

Like full backups, full restores are the best practice to follow to ensure that the metadata server returns to a consistent state and that metadata associations are correctly maintained. Full restores include the following:

- the repository manager, all of your metadata repositories, and the metadata server's configuration files (omaconfig.xml, adminUsers.txt, trustedUsers.txt, and logconfig.xml). If you use the Backup Wizard to create your backup job, then a restore job is automatically created that restores the repository manager and all of the repositories that are included in the backup. You can also use the RESTORE option of the %OMABAKUP macro to create your own restore job. As a default, this option restores the repository manager and all repositories with a registered access mode of Online or Administration.
- the SAS Content Server, the SAS Table Server, and the physical files that are associated with the metadata. Restore jobs that are created using the Backup Wizard and %OMABAKUP do not restore these files. Therefore, you must use operating system commands or third-party tools to restore them. For details, see

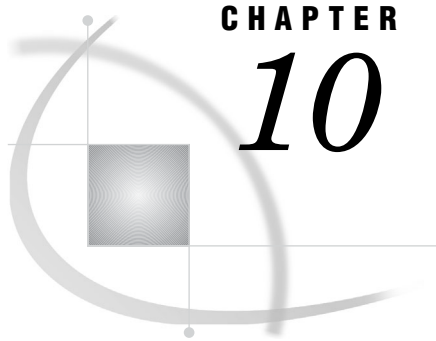
“Back Up the SAS Content Server” on page 131 and “Back Up Data That Is Stored in File Systems” on page 132.

---

## 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 metadata repositories and the repository manager. For instructions, see “Manually Restoring an Unresponsive SAS Metadata Server” on page 161.





## CHAPTER

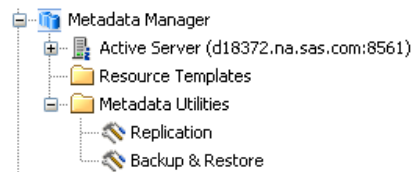
## 10

## Using the Backup Wizard in SAS Management Console

<i>About the Backup Wizard</i>	137
<i>Summary of Procedures for Using the Backup Wizard</i>	138
<i>Ensure Appropriate User Credentials</i>	139
<i>Use the Backup Wizard to Define a Backup Job and a Corresponding Restore Job</i>	140
<i>Manage Backup and Restore Jobs That Were Created by the Wizard</i>	142
<i>Run a Backup or a Restore Job That Was Created by the Wizard</i>	143
<i>Troubleshooting the Backup Wizard</i>	144
<i>Use Schedule Manager to Schedule a Backup Job That Was Created by the Backup Wizard</i>	145

### About the Backup Wizard

The Backup Wizard enables you to create backup and restore jobs that use the %OMABAKUP macro without the need to write code. The wizard is located under the **Metadata Manager** node on the **Plug-ins** tab of SAS Management Console, as shown here:



The Backup Wizard enables you to define and execute a job that backs up your metadata repositories, the repository manager, and the metadata server's configuration files, including omaconfig.xml, adminUsers.txt, trustedUsers.txt, and logconfig.xml. Jobs that are created by the wizard use the %OMABAKUP macro, which pauses the metadata server while retaining client connections, closes the repository data sets, and copies the metadata server files to the specified backup destination. It then automatically resumes the server and the handling of client requests.

The wizard allows you to specify backup options and a backup destination. You can then execute the job, save it for later execution, or deploy it for scheduling. Jobs that are created by the wizard execute on a workspace server that is local to the metadata server machine.

For each backup job that you create, the wizard automatically creates a corresponding restore job. Restore jobs that are created by the wizard use the %OMABAKUP macro with the RESTORE option. These jobs pause the metadata server; copy the metadata repository data sets, the repository manager data sets, and

the metadata server's configuration files from the backup destination; and then resume the metadata server.

You cannot use the Backup Wizard to back up repositories that have a registered access state of Offline; and you cannot use the wizard to create or run a backup job if the foundation repository has a registered access state of ReadOnly. For these types of backups, you must use a custom %OMABAKUP program.

Custom and project repositories that are registered as ReadOnly are included unless you choose to exclude them.

These options and restrictions are summarized in the following table:

Type of Repository or File	Registered Access Mode	Action Taken by the Backup Wizard
Foundation	Online	Always backed up
	Administration	Always backed up
	ReadOnly	Cannot be backed up
	Offline	Cannot be backed up
Project or custom	Online	Always backed up
	Administration	Always backed up
	ReadOnly	Backed up, unless the <b>Back Up READONLY Custom and Project Repositories</b> option is deselected
	Offline	Cannot be backed up
Metadata journal file	Not applicable	Backed up, unless the <b>Back Up Metadata Journal File</b> option is deselected
Repository manager	Not applicable	Always backed up
Metadata server configuration files	Not applicable	Backed up, unless the <b>Back Up Configuration Files</b> is deselected

Before you use the Backup Wizard, read the following information:

- “About the Metadata Server Backup and Restore Utilities” on page 124
- “Best Practices for Backing Up Your SAS System” on page 127
- “Best Practices for Restoring Your System” on page 134
- “About %OMABAKUP” on page 147
- “Summary of Procedures for Using the Backup Wizard” on page 138

## Summary of Procedures for Using the Backup Wizard

Follow these procedures to use the Backup wizard to perform a backup or a restore:

- 1 Read the overview and best practice information that is provided in Chapter 9, “Best Practices for Backing Up and Restoring Your System,” on page 123.
- 2 Ensure that you have the appropriate user credentials.
- 3 Use the Backup Wizard to define a backup job and a corresponding restore job.
- 4 Manage backup and restore jobs that were created by the wizard.



- 5 Run a backup or a restore job that was created by the wizard.
- 6 Perform troubleshooting if errors occur.
- 7 Use Schedule Manager to schedule a backup job that was created by the Backup Wizard.

The following topics provide detailed instructions for each of these procedures.

---


## Ensure Appropriate User Credentials

The user who launches the Backup Wizard (or who runs a job that was created by the wizard) must have the appropriate credentials. Follow these steps to set up a user who has the appropriate credentials:

- 1 Set up an operating system account that is known to the metadata server host machine.


On Windows systems, set up the user as follows:

- ☐ Make the user a member of the Administrators group on the metadata server host machine.
- ☐ Give the user the necessary credentials to access the workspace server on the metadata server host machine. In many cases, this means that the user must have rights to **Log on as a batch job**. Typically, you would add the user to an operating system group that has this right (for example, the SAS Server Users group). For details and exceptions, see “Requirements for Accessing a Standard Workspace Server” on page 37.

*Note:* If the user who launches the wizard or runs the job does not have credentials to access the workspace server, then the wizard displays a dialog box that you can use to enter the appropriate user ID and password. 

On UNIX or z/OS, set up an operating system account that has full access to the following directories:

- ☐ the metadata server directory (for example, *SAS-configuration-directory\Lev1\SASMeta\MetadataServer*) and the following subdirectories: **rposmgr**, **Foundation**, and the subdirectories for any custom or project repositories
- ☐ the backup destination (for example, *SAS-configuration-directory\Lev1\SASMeta\MetadataServer\SASBackup*)

*Note:* On UNIX and z/OS systems, the SAS Installer user (for example, sas) has these permissions by default. Therefore, you might want to use the SAS Installer account to run the Backup Wizard instead of setting up a different account. 

- 2 Use the User Manager plug-in in SAS Management Console to create a metadata identity for the user. The identity must include the following:

- a membership in the SAS Administrators group (or assignment to the unrestricted role)
- b a login for the operating system account that you created in step 1.

On Windows, make sure that the user ID includes the appropriate qualifier (for example, *Windows-domain-name\userID*, *userID@company.com*, or *machine-name\userID*).

On UNIX and z/OS systems, you might want to use the login for the SAS Installer (sas) user.

**CAUTION:**

On UNIX and z/OS systems, do not run a backup or a restore as the Root user. Doing so will change ownership of the metadata server files. △

---

## Use the Backup Wizard to Define a Backup Job and a Corresponding Restore Job

To use the Backup Wizard to define a backup job and a corresponding restore job, follow these steps:

- 1 Log on to SAS Management Console as a user with the appropriate credentials (see “Who Can Do What in the Metadata Manager Plug-in” on page 34).
- 2 In the Plug-ins view, expand the **Metadata Manager** node and the **Metadata Utilities** node.
- 3 Right-click the **Backup & Restore** node, and select **New Definition**.
- 4 On the first wizard page, you can select the following options:

### **Reorganize Repositories**

reclaims unused disk space that is left from previously deleted metadata objects. If this option is selected, the backup process 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 select this option for every backup. Consider using this option once a month and after large amounts of metadata have been deleted.

**Default:** Unselected

### **Run Analysis Procedure**

checks 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. Then the next time you run a backup, 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 use the RUNANALYSIS option 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.

**Default:** Unselected

### **Back Up Metadata Journal File**

backs up the metadata journal file. You can select this option only if the JOURNALPATH option is specified in the omaconfig.xml file. (For new installations, this option is specified by default.) For more information, see “Configuring Metadata Server Journaling” on page 170.

**Default:** Selected

### **Back Up READONLY Custom and Project Repositories**

includes project and custom repositories in the backup if they have a registered access mode of ReadOnly. If you do not select this option, the

backup includes only repositories that have a registered access mode of Online or Administration.

*Note:*

- ☐ You cannot create or run a backup job if the foundation repository has a registered access mode of ReadOnly. If your foundation repository is registered as ReadOnly, then you must change its registered access mode to Online. See “Change the Properties of a Metadata Repository” on page 184.
- ☐ You cannot use the Backup Wizard to back up repositories that have a registered access mode of Offline.
- ☐ If you need to back up an Offline repository or a ReadOnly foundation repository, you can use the %OMABAKUP macro directly by submitting the macro in the program editor or from a .sas file (see Chapter 11, “Using the %OMABAKUP Macro to Perform Backups and Restores,” on page 147).

$\Delta$

**Default:** Selected

#### **Back Up Configuration Files**

includes the metadata server’s configuration files, including omaconfig.xml, adminUsers.txt, trustedUsers.txt, and logconfig.xml, in the backup

**Default:** Selected

#### **Generate Debug Output**

creates detailed log messages that you can use to troubleshoot the backup program. The messages are placed in the SAS log, which is written to the backup destination directory.

**Default:** Selected

- 5 Click **Next**. The next wizard page specifies the SASMeta application server, which contains a workspace server that is local to the metadata server machine and that is configured to run operating environment commands. This workspace server is used by default to run metadata utilities such as the Backup Wizard. (For more information, see “About the SASMeta Server Context” on page 12).

You do not need to make any entries or selections on this page. When you click **Next**, the wizard validates the workspace server. You might be prompted to provide credentials.

- 6 The next wizard page lists the repositories that will be backed up. Verify that these are the correct repositories.

*Note:* You cannot make any changes on this page. However, you can go back to the first wizard page and check or uncheck the option to back up ReadOnly repositories.  $\Delta$

- 7 On the next wizard page, specify the backup directory, which is the directory location to which the backup is to be written. The default location is as follows:

*SAS-configuration-directory\Lev1\SASMeta\MetadataServer\SASBackup*

You can use this location or specify another location, keeping the following information in mind:

- ☐ As a best practice, you should direct the backup to a location that is included in your regular system backups.
- ☐ If the backup directory already contains backup files from a previous backup, those files will be deleted.

- ☐ You can specify any network location that is accessible from the workspace server. If you want to write the backup files to a path that is remote to the workspace server, then specify a UNC path (for example, `\\D2345\C$\SASBackup`).
 

*Note:* If your metadata server and workspace server are installed on a Windows host, do not specify a mapped drive.    △
  - ☐ If the backup directory does not exist, then you can create it by using the **Browse** button.
- 8 On the next wizard page, enter a unique name for the job definition. To identify the definition as a backup job, the wizard appends **\_B** to the name that you enter. The wizard also creates a corresponding definition for a restore job and appends **\_R** to its name.
- After you enter the name, select one of the following options:
- ☐ **Do not run the job now, but save the definition.**
  - ☐ **Run the backup now and save the job definition.**
- 9 On the next wizard screen, confirm the information that you entered. Then click **Finish**. The wizard does the following:
- ☐ saves the job definition with the suffix **\_B**
  - ☐ creates and saves a corresponding definition for a restore job. The wizard gives the restore job the same name that you entered for the backup job and adds the suffix **\_R** to the end of the name.
- 10 If you selected **Run the backup now and save the job definition.**, the wizard displays the following prompt:
- Job definition was successful created. Run job now?**
- Select **Yes** if you want to begin executing the backup job now.
- For additional information about running the job, including information about entering SAS Application Server credentials and verifying the backup results, see “Run a Backup or a Restore Job That Was Created by the Wizard” on page 143.

---

## Manage Backup and Restore Jobs That Were Created by the Wizard

After you have used the Backup Wizard to create backup and restore jobs, you can view the job properties; rename, run, or delete the jobs; save the jobs to files; or deploy the jobs for scheduling. To perform these tasks, follow these steps:

- 1 Log on to SAS Management Console as a user who is in the SAS Administrators group. If you are going to run the job, you must also have full operating system access to the **MetadataServer** directory, the **rposmgr** directory, and all repository and backup directories.
- 2 In the Plug-ins view in SAS Management Console, open **Metadata Manager**, open **Metadata Utilities**, and open the **Backup & Restore** node.
 

A list of previously created backup and restore jobs appears in the right pane.
- 3 Right-click the job that you want to run or manage. You can select one of the following:

### **Run Job**

runs the job. For details, see “Run a Backup or a Restore Job That Was Created by the Wizard” on page 143.

### **Save to File**

enables you to browse to a directory location of your choice and save the job as a SAS program. You can then execute the saved program on a batch basis or manage it with a third-party scheduling utility.

#### **Deploy for Scheduling**

enables you to use the scheduling features of SAS Management Console to schedule the job. For details, see “Use Schedule Manager to Schedule a Backup Job That Was Created by the Backup Wizard” on page 145.

#### **Delete**

deletes a backup job or a restore job. When you delete a backup job, a dialog box appears asking if you also want to delete the corresponding restore job. When you delete a restore job, a dialog box appears asking if you also want to delete the corresponding backup job.

**Note:** You cannot delete a backup job or a restore job if the foundation repository has a registered access state of ReadOnly.

#### **Properties**

enables you to view information about the job definition:

- ☐ The **General** tab displays the name of the job definition and enables you to enter a new name.
- ☐ The **Job** tab displays the options, servers, and paths that were specified when the job definition was created. It also displays the date the job was created and the date that the job was most recently run in Metadata Manager. (The Last Run Date is not updated when the job is executed by a scheduling system.)

---

## **Run a Backup or a Restore Job That Was Created by the Wizard**

To use SAS Management Console to run a backup or restore job that was created by the wizard:

- ☐ If you want to run a backup job immediately after it is created, then click **Yes** when the Backup wizard displays the following prompt: **Job definition was successfully created. Run job now?**
- ☐ To run a backup or restore job later, select the **Backup & Restore** node under **Metadata Utilities**. In the right pane, right-click an existing job definition, and select **Run Job**.

*Note:*

- ☐ You must have the appropriate credentials to run the job (see “Ensure Appropriate User Credentials” on page 139).
- ☐ You can also deploy the job for scheduling, if you have SAS scheduling configured (see “Use Schedule Manager to Schedule a Backup Job That Was Created by the Backup Wizard” on page 145). Or you can save the job as a SAS program and then execute it on a batch basis or manage it with a third-party scheduling utility (see “Manage Backup and Restore Jobs That Were Created by the Wizard” on page 142).
- ☐ You cannot run a restore job unless you have previously run the corresponding backup job.



After you start the job, you might be prompted for credentials to log on to the SAS Application Server. For information about the required credentials, see “Ensure Appropriate User Credentials” on page 139.

If the job completes successfully, the following message is displayed:

**Job completed without errors**

If the job that you executed is a backup job, then the following items should appear in the backup destination directory:

- a folder called **RPOSMGR**, which contains backups of the repository manager data sets
- a folder called **Foundation** (or the name of your foundation repository, if you have assigned a different name), which contains backups of the repository data sets for your foundation repository
- folders containing backups of any custom repositories or project repositories that were included in the backup
- the file `MetadataJournal.dat`, which is a backup of your metadata journal file (this file is not present if you unchecked the option to back up the journal file)
- the files `omaconfig.xml`, `adminUsers.txt`, `trustedUsers.txt`, and `logconfig.xml`, which are backups of your metadata server's configuration files
- the file `oma_backup.sas7bcart`, which stores information about the source and target destinations
- the file `oma_varan.sas7bcart`, which contains a job that SAS processes can use to analyze repository data sets in the backup subdirectories
- a SAS log file which contains messages generated by the backup job. If the **Generate Debug Output** option was selected, then the log contains detailed log messages that you can use to troubleshoot the backup program.
- a file with the suffix `.lst`, which provides information about which repositories were backed up.

When the job is completed, the current date is recorded in the Last Run Date field on the **Job** tab of the job definition's properties. In addition, metadata objects for the backup job and the corresponding restore job are placed in the **Administration** subfolder of the **System** folder in the SAS Folders tree.

*Note:* The Last Run Date is not updated when the job is executed by a scheduling system. △

If the job does not complete successfully, a message is displayed indicating that errors occurred. For details about the errors, open the file *name-of-backup-job.log* in the backup destination directory.

---

## Troubleshooting the Backup Wizard

In addition, the following error messages might be displayed when you run the Backup Wizard:

**Since all of your repositories are READONLY, no repositories will be backed up. To create a backup job, click Cancel and change the foundation repository to ONLINE. If you also want to back up READONLY custom and project repositories, select the Back Up READONLY Custom and Project Repositories option.**

This message appears if all of your repositories have a registered access mode of ReadOnly and if you did not select the option to **Back Up READONLY Custom and Project Repositories**. Do the following if this message appears:

- 1 Click **Cancel** to exit the wizard.
- 2 Change the foundation repository's registered access mode to Online (see "Change the Properties of a Metadata Repository" on page 184).

- 3 Start the wizard again.
- 4 If you have ReadOnly custom or project repositories that you want to back up, select the **Back Up READONLY Custom and Project Repositories** option on the first wizard page.

**Repository Foundation cannot be updated because the current or registered access is READONLY.**

**Due to the previous error, the job definition is incomplete for job *job-name*.**

These messages appear in succession if you are creating a new backup job and the foundation repository has a registered access mode of ReadOnly. Change the foundation repository's registered access mode to Online, and run the wizard again.

If a backup job fails due to a lack of disk space, and if the backup destination is local to the metadata server machine, follow these steps:

- 1 Check to see whether the metadata server is paused.
- 2 If the metadata server is paused, you must free up disk space (or configure additional disk space) before you can resume the server. To determine the amount of disk space that is needed to resume the server and run the backup, add the size of the metadata journal file to the total size of your repositories.
- 3 When the necessary amount of disk space has been made available, resume the server by using SAS Management Console, a script, or the Windows service manager (see Chapter 5, "Operating Your Servers," on page 51).
- 4 Rerun the backup.

For a detailed log, open *name-of-backup-job.log* in the backup destination directory.

---

## Use Schedule Manager to Schedule a Backup Job That Was Created by the Backup Wizard

If your deployment plan included either Platform Suite for SAS or an Operating System Services Scheduling Server, then you can use the Schedule Manager in SAS Management Console to schedule a backup job.

For example, if you have configured an Operating System Services Scheduling Server on Windows, then you would follow these steps to schedule a backup job:

- 1 Log on to SAS Management Console as a user who is in the SAS Administrators group. On Windows systems, the user must also be a member of the Windows Administrators group.
- 2 Make sure that the Server Manager tree includes an OS Scheduling Server. (This component must be installed and configured by using the SAS Deployment Wizard.)
- 3 Expand the **Metadata Manager** node, the **Metadata Utilities** node, and the **Backup & Restore** node.

A list of previously created backup and restore jobs appears in the right pane.

- 4 To deploy a job for scheduling, right-click the job and select **Deploy for Scheduling**. The job is placed in the deployment directory that has been defined for metadata server utility jobs.
- 5 To place the job in a flow, right-click the **Schedule Manager** node and select **New Flow**. In the New flow dialog box, do the following:
  - a Enter a name for the flow.
  - b In the **Scheduling Server** field, select **Operating System Services**.

- c Move the backup job from the **Available Items** list to the **Selected Items** list.
  - d Click **OK**.
- 6 To schedule the flow, right-click the flow in the Schedule Manager tree and select **Schedule Flow**. In the Schedule Flow dialog box, do the following:
- a Click **Options**.
  - b In the Trigger Options dialog box, click **New Time Event**.
  - c In the New Time Event dialog box, select the desired values for Recurrence, Interval, Time, and Date Range, as shown in the following example:

The screenshot shows the 'New Time Event' dialog box with the following settings:

- Recurrence:** Daily
- Interval:**
  - ☒ Daily interval: 1 day
  - ☐ Every weekday (Monday - Friday)
- Time:**
  - Hour: 2 p.m.
  - Minute: 15
- Date Range:**
  - Start Date: Day: 20, Month: November, Year: 2008
  - End Date: ☐ None
  - ☒ Day: 20, Month: November, Year: 2012

Buttons at the bottom: OK, Cancel, Help.

Click **OK** in the New Time Event, Trigger Options, and Schedule Flow dialog boxes.

- 7 On Windows systems, you must edit the scheduled job so that it will be run with the appropriate user credentials. By default, the Windows Task Scheduler uses the LOCAL SYSTEM user account to execute scheduled jobs. However, backup jobs must be run by a user who has the credentials that are described in “Ensure Appropriate User Credentials” on page 139.

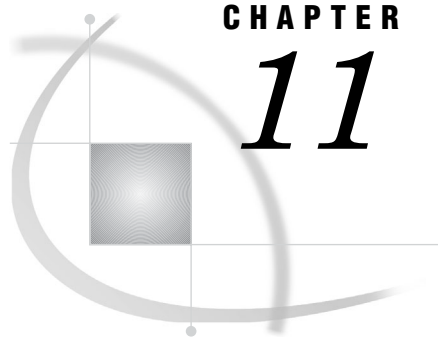
To provide the appropriate user ID and password, open the Windows Control Panel, select **Scheduled Tasks**, right-click the job that you just scheduled, and select **Properties**. In the job properties dialog box, enter the appropriate user ID in the **Run as** field, click **Set password** to enter the password, and click **OK** to save your changes.

When the job is executed, a log is written to the following location:

*SAS-configuration-directory\Lev1\SASMeta\BatchServer\Logs*

For information about additional scheduling options, such as setting up dependencies among jobs or using Platform Suite for SAS, see *Scheduling in SAS*.





## CHAPTER

## 11

# Using the %OMABAKUP Macro to Perform Backups and Restores

<i>About %OMABAKUP</i>	147
<i>What %OMABAKUP Does</i>	148
<i>Ensuring Appropriate User Credentials</i>	149
<i>Running the backupServer.sas and restoreServer.sas Programs</i>	150
<i>Error Logging</i>	151
<i>Writing Custom Programs Using %OMABAKUP</i>	151
<i>Writing a Custom Program to Perform a Full Metadata Server Backup</i>	151
<i>Example: Program for Full Backup</i>	153
<i>Example: Backup Program on Windows, Using Alternating Destinations</i>	154
<i>Example: Program to Restore All Repositories and the Repository Manager</i>	155
<i>Warning Messages</i>	155
<i>%OMABAKUP Macro: Reference</i>	156
<i>Server Connection Statement: Reference</i>	156
<i>%OMABAKUP Statement: Reference</i>	157

## About %OMABAKUP

The %OMABAKUP macro backs up or restores your metadata repositories, the repository manager, and the metadata server's configuration files, including omaconfig.xml, adminUsers.txt, trustedUsers.txt, and logconfig.xml. It also backs up (but does not restore) the metadata journal file. The %OMABAKUP macro is part of the SAS autocall libraries and runs in a dedicated SAS session. The macro must be executed on the computer that hosts the metadata server.

You can use %OMABAKUP in the following ways:

- Use the %OMABAKUP programs (backupServer.sas and restoreServer.sas) that are provided with your installation. These programs are customized for your operating environment and provide a full backup and restore of the metadata server. For details, see “Running the backupServer.sas and restoreServer.sas Programs” on page 150.
- Use the Backup Wizard to create a backup job that uses the %OMABAKUP macro. The wizard enables you to specify %OMABAKUP options without writing code. For details, see Chapter 10, “Using the Backup Wizard in SAS Management Console,” on page 137.
- The backupServer.sas program and the Backup Wizard should be sufficient for most situations. However, if the need arises, you can write a custom %OMABAKUP program and submit it in the program editor, from a .sas file, or from a scheduling utility. For details, see “Writing Custom Programs Using %OMABAKUP” on page 151.

Before you use %OMABAKUP, read the following information:

- “Best Practices for Backing Up Your SAS System” on page 127
- “Best Practices for Restoring Your System” on page 134
- “What %OMABAKUP Does” on page 148
- “Ensuring Appropriate User Credentials” on page 149

**CAUTION:**

If you choose to use operating system commands to back up your metadata repositories and the repository manager, instead of using the Backup Wizard or %OMABAKUP, then you must be sure to pause the metadata server before performing the backup. Otherwise, the backup files will not be usable. △

---

## What %OMABAKUP Does

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

- 1 pauses the metadata server to an Offline state. The server does the following when it is paused to Offline:
  - retains all active client connections.
  - temporarily changes the access state of each repository to Offline.
  - 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.
  - 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.
- 2 creates the directory that is specified in the DESTINATIONPATH option if the directory does not already exist.
- 3 creates a subdirectory called **REPOSMGR** in the destination path, and uses PROC COPY to copy the repository manager to this subdirectory.
- 4 determines which repositories to back up based on their registered access mode, as follows:
  - All Online repositories are automatically included.
  - All Administration repositories are automatically included.
  - All ReadOnly repositories are included, unless you set the DISABLEREADONLYBACKUP option to YES.

*Note:*

- When you use the Backup Wizard to create a backup job, you cannot back up a foundation repository that has a registered access mode of READONLY. If your foundation repository is registered as READONLY, then you must change its registered access mode to Online before running the backup. See “Change the Properties of a Metadata Repository” on page 184.
- When you use the Backup Wizard to create a backup job, READONLY project repositories and custom repositories are included only if you select the option **Back Up READONLY Custom and Project Repositories**.

△

- All Offline repositories are included, unless you set the `DISABLEOFFLINEBACKUP` option to YES.

*Note:* When you use the Backup Wizard, you cannot back up Offline repositories. △

5 performs the following steps for each repository that is being backed up:

- a 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 Windows or UNIX does not allow in directory names, %OMABAKUP replaces the unsupported characters with underscores when creating the directory name. These characters include the following: \, /, :, \*, ?, ", ', <, >, [, ], and |.

*Note:* %OMABAKUP supports repository **Name** values of up to 32 characters. △

- b uses PROC COPY to copy the repository's data sets to the subdirectory.
- c creates a file called `metimdb.sas7bcnt` for each repository in its registered location. This file is a control data set that specifies how character variables in repository data sets are to be treated in memory. The data set is used to load repositories back into memory when client activity is resumed on the metadata server.
- d creates a `DS_VCHAR` data set in the subdirectory if the `RUNANALYSIS` option is specified. For information about `DS_VCHAR` data sets, see "Using %OMARUNAN (or the `RUNANALYSIS` Option of %OMABAKUP) to Optimize Memory Usage" on page 175.

6 creates or updates the following catalogs in the destination directory:

- `oma_backup.sas7bcnt`, which stores information about repository source and target destinations
- `oma_varan.sas7bcnt`, which contains a job that SAS processes can use to analyze repository data sets in the backup subdirectories.

7 uses operating system commands to copy the metadata journal file (`MetadataJournal.dat`) to the backup destination, unless the `BACKUPMETADATAJOURNALFILE` option is set to NO .

8 uses operating system commands to copy the metadata server's configuration files to the backup destination, unless `DISABLECONFIGURATIONFILEBACKUP` is set to YES. These files include `omaconfig.xml`, `adminUsers.txt`, `trustedUsers.txt`, and `logconfig.xml`.

9 returns the metadata server to Online status. The metadata server resumes the handling of client requests.

To perform a restore, you can use %OMABAKUP with the `RESTORE` option. In SAS 9.2, the `RESTORE` option of %OMABAKUP restores the repository manager, the metadata repositories, and the metadata server's configuration files. The metadata journal file is not restored.

---

## Ensuring Appropriate User Credentials

The following requirements apply to the use of %OMABAKUP:

- The user who executes the %OMABAKUP macro must be authorized to pause the metadata server. The following categories of users have this authorization:
  - The user whose user ID was used to start the metadata server.
  - Users who are assigned to the **Management Console: Operation** role.
  - Users who are assigned to the **Management Console: Unrestricted** role.
  - Users who are specified in the adminUsers.txt file.
- On Windows systems, the user that executes the %OMABAKUP macro must be a member of the Windows Administrators group.
 

On Windows Server 2008 64-bit systems, you must also use the Run as Administrator option. To use this option, open a DOS command window. In the window, click the right mouse button and select **Run as Administrator**. In the Administrator: Command Prompt window, submit the appropriate batch job (for example, **MetadataServer -backup**).
- In all host environments, the user who executes the %OMABAKUP macro must have full operating system access to the metadata server directory (*SAS-configuration-directory\Lev1\SASMeta\MetadataServer*). Within that directory, the user must have full access to the following subdirectories: **rposmgr**, **Foundation**, and the subdirectories for any customer or project repositories.
 

On UNIX and z/OS systems, the SAS Installer (sas) user has the required access to these directories. On Windows systems, the user who runs the backup should be added to the Windows Administrators group to provide the required access.

**CAUTION:**

On UNIX and z/OS systems, do not run a backup or a restore as the Root user. Doing so will change ownership of the metadata server files. △

---

## Running the backupServer.sas and restoreServer.sas Programs

Your installation includes a backup program (backupServer.sas) and a restore program (restoreServer.sas) that use %OMABAKUP. These programs are customized for your environment and provide a full backup and restore of the metadata server. The programs are located in the metadata startup directory, which is in the following path:

*SAS-configuration-directory\Lev1\SASMeta\MetadataServer*

To run the programs, make sure that you have the appropriate user credentials, as described in the preceding topic. Then go to the metadata server startup directory and issue one of the following commands:

- On Windows: **MetadataServer -backup** or **MetadataServer -restore**
- On UNIX or z/OS: **MetadataServer.sh -backup** or **MetadataServer.sh -restore**

The backupServer.sas program does the following:

- connects to the metadata server by using credentials that are stored in the file metaparms.sas, which is located in the metadata startup directory.

**CAUTION:**

The metaparms.sas file contains the user ID and the encoded password for the SAS Administrator user. Be sure to protect this file from unauthorized access. △

- writes the backup files to a subdirectory called **SASBackup** in the metadata startup directory. (Be sure to include this directory in your site's daily backups.)

- writes a log to a subdirectory called **Logs** in the metadata startup directory. A file with the suffix .lst is also written to the metadata startup directory. This file provides information about which repositories were backed up. (When you use the Backup Wizard to create the job, these files are written to the backup destination instead.)

The restoreServer.sas program does the following:

- connects to the metadata server by using credentials that are stored in the file metaparms.sas, which is located in the metadata startup directory.
- restores the backup files from the **SASBackup** directory to the appropriate subdirectories in the metadata startup directory.
- writes a log to a subdirectory called **Logs** in the metadata startup directory. A file with the suffix .lst is also written to the metadata startup directory. This file provides information about which repositories were restored. (When you use the Backup Wizard to create the job, these files are written to the backup destination instead.)

You can further customize backupServer.sas and restoreServer.sas to specify a different backup location or to incorporate other %OMABAKUP options. For an explanation of each %OMABAKUP parameter, see “Server Connection Statement: Reference” on page 156.

---

## Error Logging

Errors are written to a file named *name-of-backup-job.log* in the directory where the %OMABAKUP program was executed. (When you use the Backup Wizard to create the job, the log is written to the backup destination instead.)

---

## Writing Custom Programs Using %OMABAKUP

---

### Writing a Custom Program to Perform a Full Metadata Server Backup

Use the following steps to create a custom program that performs a full metadata server backup:

- 1 Create and secure a directory in the path where the backup files will be stored. You can use the following directory, which is provided in your deployment:
  - On Windows:  
`SAS-configuration-directory\Lev1\SASMeta\MetadataServer\SASBackup`
  - On UNIX and z/OS: `SAS-configuration-directory/Lev1/SASMeta/MetadataServer/SASBackup`

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 server connection parameters and the %OMABAKUP macro, as follows:

- a Use the following syntax to include the file that contains server connection parameters:

```
FILENAME METAPARM "path-to-metaparms.sas-file";
%INCLUDE METAPARM;
```

The path to the metaparms.sas file is as follows:

- ☐ Windows: *SAS-configuration-directory\Lev1\SASMeta\MetadataServer\metaparms.sas*
- ☐ UNIX and z/OS: *SAS-configuration-directory/Lev1/SASMeta/MetadataServer/metaparms.sas*

The metaparms.sas file contains server connection statements that use the following syntax:

```
OPTIONS METASERVER="host-name"
        METAPORT=port-number
        METAPROTOCOL="bridge"
        METAUSER="user-ID"
        METAPASS="encoded-password"
        METAREPOSITORY="Foundation";
```

For an explanation of each parameter in the server connection statement, see “Server Connection Statement: Reference” on page 156.

**CAUTION:**

**The metaparms.sas file contains the user ID and the encoded password for the SAS Administrator user. Be sure to protect this file from unauthorized access.**  $\triangle$

- 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 157.

- 3 Name the program BackupRepositories.sas or RestoreRepositories.sas, and save it to the metadata server startup directory, which is in the following path:

*SAS-configuration-directory/Lev1/SASMeta/MetadataServer*

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 Use the appropriate command to submit the program.

*Note:*

- ☐ %OMABAKUP must be executed on the computer that hosts the metadata server.
- ☐ In the commands that are shown in this step, *SAS-installation-directory* is the directory where SAS Foundation is installed (for example, the directory might be **C:\Program Files\SAS\SASFoundation\9.2** on Windows or **/usr/local/SAS/SASFoundation/9.2** on UNIX).
- ☐ Line breaks have been inserted in the commands for presentation purposes. When you enter the commands, you must omit the line breaks.

$\triangle$

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

```
"SAS-installation-directory\sas.exe"
-config "SAS-configuration-directory\Levl\SASMeta\sasv9.cfg"
-sysin BackupRepositories.sas

"SAS-installation-directory\sas.exe"
-config "SAS-configuration-directory\Levl\SASMeta\sasv9.cfg"
-sysin RestoreRepositories.sas
```

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

```
"SAS-installation-directory/sas"
-config "SAS-installation-directory/sasv9.cfg"
-sysin BackupRepositories.sas

"SAS-installation-directory/sas"
-config "SAS-installation-directory/sasv9.cfg"
-sysin RestoreRepositories.sas
```

To execute the program on z/OS under the UNIX shell, submit one of the following commands:

```
SAS-installation-directory/sas.sh
-sysin SAS-configuration-directory/Levl/SASMeta/
MetadataServer/BackupRepositories.sas
-log SAS-configuration-directory/Levl/SASMeta/
MetadataServer/BackupRepositories.log
-print SAS-configuration-directory/Levl/SASMeta/
MetadataServer/BackupRepositories.lst

SAS-installation-directory/sas.sh
-sysin SAS-configuration-directory/Levl/SASMeta/
MetadataServer/RestoreRepositories.sas
-log SAS-configuration-directory/Levl/SASMeta/
MetadataServer/RestoreRepositories.log
-print SAS-configuration-directory/Levl/SASMeta/
MetadataServer/RestoreRepositories.lst
```

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

- 5 Be sure to include the backup destination in your site's daily backups.

---

## Example: Program for Full Backup

The following example program backs up the repository manager, each repository that is registered in the repository manager, and the metadata server's configuration files (including omaconfig.xml, adminUsers.txt, trustedUsers.txt, and logconfig.xml).

```
/* The metaparms.sas file contains the server connection options, including
the user ID and the encoded password for the SAS Administrator user.
Be sure to protect this file from unauthorized access. */
filename metaparm
      "SAS-configuration-directory\Levl\SASMeta\MetadataServer\metaparms.sas";
%include metaparm;
```

```

/* This %OMABAKUP command creates a backup on the local machine in
SAS-configuration-directory\Lev1\SASMeta\MetadataServer\SASBackup */

%omabakup(DestinationPath=
          "SAS-configuration-directory\Lev1\SASMeta\MetadataServer\SASBackup",
          ServerStartPath="SAS-configuration-directory\Lev1\SASMeta\MetadataServer",
          RposmgrPath="rposmgr")

```

*Note:* The preceding example uses paths that are appropriate for Windows systems. Use paths that are appropriate for your operating environment. △

During installation, the SAS Deployment Wizard created a similar backup program for your deployment (see “Running the backupServer.sas and restoreServer.sas Programs” on page 150). The program is called backupServer.sas and is located in *SAS-configuration-directory\Lev1\SASMeta\MetadataServer*.

---

## Example: Backup Program on Windows, Using Alternating Destinations

This example program creates one backup image on the local machine 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. 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**.

```

/* The metaparms.sas file contains the server connection options */
filename metaparm
          "SAS-configuration-directory\Lev1\SASMeta\MetadataServer\metaparms.sas";
%include metaparm;
/* This %OMABAKUP command creates a backup on the local machine in
SAS-configuration-directory\Lev1\SASMeta\MetadataServer\SASBackup */

%omabakup(DestinationPath=
          "SAS-configuration-directory\Lev1\SASMeta\MetadataServer\SASBackup",
          ServerStartPath="SAS-configuration-directory\Lev1\SASMeta\MetadataServer",
          RposmgrPath="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="SAS-configuration-directory\Levl\SASMeta\MetadataServer",
          RposmgrPath="rposmgr")

```

*Note:* The preceding example uses paths that are appropriate for Windows systems. Use paths that are appropriate for your operating environment. △

---

## Example: Program to Restore All Repositories and the Repository Manager

The following is an example of a program that restores the repository manager, all registered metadata repositories, and the metadata server's configuration files.

*Note:* In SAS 9.1.3, the RESTORE option of %OMABAKUP did not restore the repository manager when it was run in its default mode. In SAS 9.2, the RESTORE option restores the repository manager, all registered repositories, and the metadata server's configuration files by default. △

```

/* The metaparms.sas file contains the server connection options */
filename metaparm
    "SAS-configuration-directory\Levl\SASMeta\MetadataServer\metaparms.sas";
%include metaparm;

/* This %OMABAKUP command restores the repositories from the backup
files on the local machine in
SAS-configuration-directory\Levl\SASMeta\MetadataServer\SASBackup */

%omabakup(DestinationPath=
    "SAS-configuration-directory\Levl\SASMeta\MetadataServer\SASBackup",
    ServerStartPath="SAS-configuration-directory\Levl\SASMeta\MetadataServer",
    RposmgrPath="rposmgr",
    Restore=Y)

```

*Note:* The preceding example uses paths that are appropriate for Windows systems. Use paths that are appropriate for your operating environment. △

During installation, the SAS Deployment Wizard created a similar restore program for your deployment. For details, see “Running the backupServer.sas and restoreServer.sas Programs” on page 150.

---

## 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 before 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.

You might also see messages similar to the following: **NOTE: Data file filename is in a format that is native to another host, or the file encoding does not match the session encoding. Cross Environment Data Access will be used, which might require additional CPU resources and might reduce performance.** These messages are caused by a new encoding format that is used in SAS 9.2 for the repository container databases. The messages are not cause for concern.

---

## %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-user-ID"
      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.

**METAPROTOCOL**

Specify the network protocol for connecting to the metadata server. The default value, **BRIDGE**, is the only value that is valid in SAS 9.2.

**METAUSER**

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

- is assigned to the **Metadata Server: Operation** role
- 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=sas002;
run;
```

#### METAREPOSITORY

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

*Note:* Instead of directly coding the connection options, you can include the metaparms.sas file, as illustrated in "Example: Program for Full Backup" on page 153. △

---

## %OMABAKUP Statement: Reference

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

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

```
%OMABAKUP(DESTINATIONPATH="pathname",
  SERVERSTARTPATH="pathname",
  RPOSMGRPATH="pathname",
  <REORG="YES | NO">,
  <DISABLECONFIGURATIONFILEBACKUP="YES | NO">,
  <DISABLEREADONLYBACKUP="YES | NO">,
  <DISABLOFFLINEBACKUP="YES | NO">,
  <RESTORE="YES | NO">,
  <RUNANALYSIS="YES | NO">,
  <BACKUPMETADATAJOURNALFILE="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. If the directory does not already exist, %OMABAKUP will create the directory for you. If you are using the Backup Wizard, use the **Browse** button to create the directory.

For a restore, specify the pathname of the directory where backups are stored. The following values are recommended for DESTINATIONPATH:

- Windows:  
SAS-configuration-directory\Lev1\SASMeta\MetadataServer\SASBackup
- UNIX and z/OS: SAS-configuration-directory/Lev1/SASMeta/  
MetadataServer/SASBackup

Be sure to include this directory in your site's daily backups.

**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. In default installations, the path is as follows:

- Windows: *SAS-configuration-directory\Lev1\SASMeta\MetadataServer*
- UNIX and z/OS: *SAS-configuration-directory/Lev1/SASMeta/MetadataServer*

**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:

**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. △

- NO (or N) if you want to copy the repositories as they are.

**Default:** NO

**DISABLECONFIGURATIONFILEBACKUP**="YES|NO"

is an optional parameter. If you do not want to back up or restore the metadata server's configuration files, (omaconfig.xml, adminUsers.txt, trustedUsers.txt, and logconfig.xml), then set this parameter to YES. Otherwise, the files will be backed up.

**Default:** NO

**DISBLEREADONLYBACKUP**="YES|NO"

is an optional parameter. If you do not want to back up or restore custom and project repositories that have a registered access mode of ReadOnly, then set this parameter to YES. Otherwise, all custom and project repositories that have a registered access mode of ReadOnly will be included in the backup or the restore.

*Note:*

- You cannot use the Backup Wizard to create or run a backup job if the foundation repository has a registered access mode of ReadOnly. You must

first change its registered access mode to Online (see “Change the Properties of a Metadata Repository” on page 184).

- When you use the Backup Wizard to create a backup job, ReadOnly project repositories and custom repositories are included only if you select the option **Back Up READONLY Custom and Project Repositories**.

△

**Default:** NO

**DISABLEOFFLINEBACKUP="YES|NO"**

is an optional parameter. If you do not want to back up or restore repositories that have a registered access mode of Offline, then set this parameter to YES. Otherwise, all Offline repositories will be included in the backup or the restore.

*Note:* When you use the Backup Wizard, you cannot back up Offline repositories. △

**Default:** NO

**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 the repository manager and 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.

*Note:* In SAS 9.1.3, the RESTORE did not restore the repository manager when it was run in its default mode. In SAS 9.2, the RESTORE option restores the repository manager, all registered repositories, and the metadata server’s configuration files by default. △

**Default:** NO

**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.

For more information about this option, see “Using %OMARUNAN (or the RUNANALYSIS Option of %OMABAKUP) to Optimize Memory Usage” on page 175.

**Default:** NO

**BACKUPMETADATAJOURNALFILE="YES|NO"**

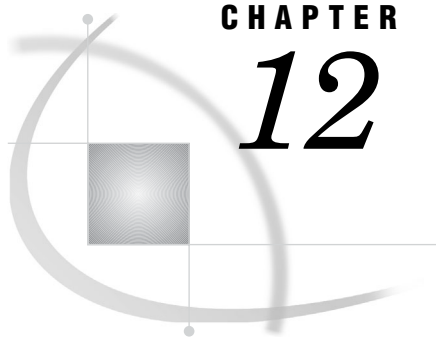
is an optional parameter that you can use to specify whether to back up the metadata journal file. You can set this option to YES only if the JOURNALPATH option is specified in the omaconfig.xml file. (In default installations, the JOURNALPATH option is specified.) For more information, see “Configuring Metadata Server Journaling” on page 170.

**Default:** YES

*Note:* The pathnames that are used in the DESTINATIONPATH, SERVERSTARTPATH, and RPOSMGRPATH parameters cannot contain the following characters:

- On Windows: asterisks (\*), backslashes (\), brackets ([]), colons (:), double quotation marks ("), forward slashes (/), greater-than signs (>), less-than signs (<), question marks (?), single quotation marks ('), or vertical bars (|).
- On UNIX: asterisks (\*), backslashes (\), brackets ([]), double quotation marks ("), forward slashes (/), question marks (?), or single quotation marks (').

(Backslashes and forward slashes are supported as directory delimiters on hosts that support them as delimiters.)  $\Delta$



## CHAPTER

## 12

# Manually Restoring an Unresponsive SAS Metadata Server

*Manually Restoring an Unresponsive SAS Metadata Server* 161

## Manually Restoring an Unresponsive SAS Metadata Server

If the SAS Metadata Server is not responding to client requests, then you cannot use the %OMABAKUP macro or the Backup Wizard to restore the repository manager and the foundation repository. Instead, you must stop the server and use operating system commands to restore these items. You must then use %OMABAKUP to restore any custom and project repositories.

**CAUTION:**

**This procedure assumes that you have a usable backup copy of the repository manager and your metadata repositories. △**

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

- 1 Follow the steps in “Checking the Access State of the SAS Metadata Server” on page 76 to ensure that the server is stopped.
- 2 Make sure that you have the appropriate user credentials, as described in “Ensuring Appropriate User Credentials” on page 149.
- 3 Open the directory where the SAS Metadata Server is started. In a default installation on Windows, this directory is *SAS-configuration-directory\Lev1\SASMeta\MetadataServer*. In a default installation on z/OS or UNIX, this directory is *SAS-configuration-directory/Lev1/SASMeta/MetadataServer*.
- 4 Make sure that the omaconfig.xml file, adminUsers.txt file, trustedUsers.txt file, and logconfig.xml files are intact. If they are not intact, then use operating system commands to restore them from the backup location to the metadata server startup directory.

If you do not have backup copies of these files, then you must use the SAS Deployment Wizard to re-install the metadata server before performing the rest of this procedure.

- 5 Use the appropriate operating system command to delete the current **MetadataRepositories** directory and its contents:

Windows command:

```
rmdir /S MetadataRepositories
```

UNIX and z/OS commands:

```
rm -rf MetadataRepositories
```

- 6 Use the appropriate operating system command to create a new, empty **MetadataRepositories** directory:

Windows command:

```
mkdir MetadataRepositories
```

UNIX and z/OS commands:

```
mkdir MetadataRepositories
```

- 7 Use the appropriate operating system command to delete the current **rposmgr** directory and its contents:

Windows command:

```
rmdir /S rposmgr
```

UNIX and z/OS commands:

```
rm -rf rposmgr
```

- 8 Use the following command to change directories to the backup directory.

```
cd backup-location
```

- 9 Use the appropriate operating system command to copy the backup of the repository manager to a subdirectory called **rposmgr** in the metadata server startup directory.

Windows command:

```
xcopy source-directory destination-directory /C /I /H /K /O /X /Y
```

UNIX and z/OS command:

```
cp -rp source-directory destination-directory
```

For *source-directory*, specify the subdirectory that contains the backup of the repository manager. If you used %OMABAKUP or the Backup Wizard to create the backup, then the name of this subdirectory is **REPOSMGR**.

For *destination-directory*, specify the full path of the metadata server startup directory, followed by the subdirectory name **rposmgr**.

**CAUTION:**

**Be sure to specify rposmgr as the subdirectory name in the destination path. On UNIX and z/OS, be sure to use lowercase letters.     △**

Here is an example Windows command to restore the repository manager from a backup that was created by %OMABAKUP or the Backup Wizard:

```
xcopy REPOSMGR "C:\SAS\Config\Lev1\SASMeta\MetadataServer\rposmgr" /C /I /H /K /O /X /Y
```

Here is an example UNIX or z/OS command to restore the repository manager from a backup that was created by %OMABAKUP or the Backup Wizard:

```
cp -rp REPOSMGR "/opt/sas/Config/Lev1/SASMeta/MetadataServer/rposmgr"
```

- 10 Use the appropriate operating system command to copy the backup data sets for the foundation repository to the metadata server startup directory.

Windows command:

```
xcopy source-directory destination-directory /C /I /H /K /O /X /Y
```



UNIX and z/OS command:

```
cp -rp source-directory destination-directory
```

For *source-directory*, specify the **Foundation** subdirectory, which contains the backup data sets for the foundation repository.

For *destination-directory*, specify the full path of the metadata server startup directory, followed by the subdirectory names **MetadataRepositories** and **Foundation**.

Here is an example Windows command to restore the foundation repository:

```
xcopy Foundation "C:\SAS\Config\Levl\SASMeta\
MetadataServer\MetadataRepositories\Foundation" /C /I /H /K /O /X /Y
```

Here is an example UNIX or z/OS command to restore the foundation repository:

```
cp -rp Foundation "/opt/sas/Config/Levl/SASMeta/MetadataServer/
MetadataRepositories/Foundation"
```

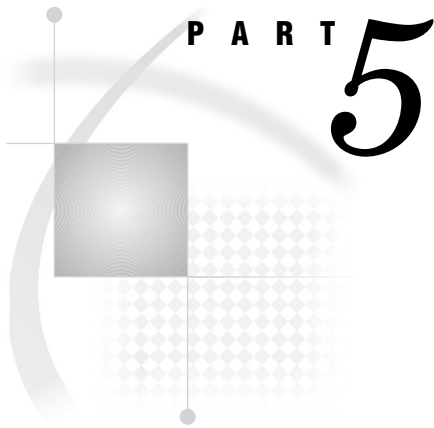
- 11 Use the instructions in Chapter 5, “Operating Your Servers,” on page 51 to start the metadata server.
- 12 If the metadata server includes any additional repositories (including custom or project repositories), then use the RESTORE option of %OMABAKUP to perform a full metadata server restore. For custom and project repositories, it is important to use the %OMABAKUP macro instead of operating system commands. The macro will ensure that each repository is restored to the correct path, as specified in the repository manager.

You can use either of the following methods to perform this step:

- Run the restoreServer.sas program that was provided with your installation. Before running the program, make sure that it accesses the same backup directory that you accessed in steps 9 and 10. For details, see “Running the backupServer.sas and restoreServer.sas Programs” on page 150.
- Write and execute a custom %OMABAKUP program that includes the RESTORE option. For an example, see “Example: Program to Restore All Repositories and the Repository Manager” on page 155. Be sure to access the same backup directory that you accessed in steps 9 and 10.

These methods pause the server and perform a full restore of the metadata server, including all of your metadata repositories, the repository manager, and the metadata server’s configuration files (including omaconfig.xml, adminUsers.txt, trustedUsers.txt, and logconfig.xml).



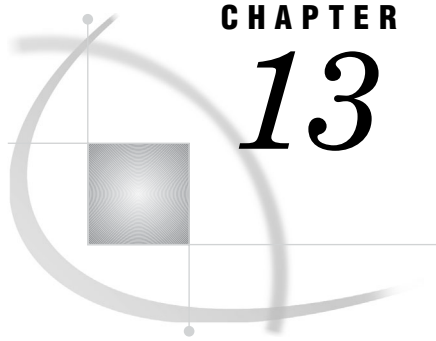


## **Optimizing the Performance of the SAS Metadata Server**

*Chapter 13* . . . . . **Using Configuration Settings to Optimize Metadata Server  
Performance** 167

*Chapter 14* . . . . . **Managing Metadata Server Performance** 173





## CHAPTER

## 13

# Using Configuration Settings to Optimize Metadata Server Performance

<i>Overview of Performance-Related Configuration Settings</i>	<b>167</b>
<i>Configuring the Number of Threads Used by the Metadata Server</i>	<b>167</b>
<i>About the MAXACTIVETHREADS, THREADSMIN, and THREADSMAX Options</i>	<b>167</b>
<i>Specify Values for the MAXACTIVETHREADS, THREADSMIN, and THREADSMAX Options</i>	<b>169</b>
<i>Configuring Metadata Server Journaling</i>	<b>170</b>
<i>About Metadata Server Journaling</i>	<b>170</b>
<i>Change the Journaling Options</i>	<b>170</b>
<i>Move the Journal File to Another File System</i>	<b>171</b>

## Overview of Performance-Related Configuration Settings

The following configuration settings can be used to optimize the performance of your metadata server:

- the MAXACTIVETHREADS, THREADSMIN, and THREADSMAX options, which control the way that the metadata server uses threads. See “Configuring the Number of Threads Used by the Metadata Server” on page 167.
- the metadata server journaling options. See “Configuring Metadata Server Journaling” on page 170.

## Configuring the Number of Threads Used by the Metadata Server

### About the MAXACTIVETHREADS, THREADSMIN, and THREADSMAX Options

When the metadata server is initialized, the server creates a pool of available threads to run client requests. The metadata server’s configuration settings control the minimum and maximum number of threads in the pool.

When a client request is received, one of the available threads is used to process the request. If no threads are available, the server creates a new thread. If the thread pool has reached the maximum size, the request waits until a thread is returned to the pool.

The client request is then executed in the thread as resources become available. The server’s configuration settings control the maximum number of threads that can be executed concurrently. When processing is complete, the thread is returned to the pool.

If the number of available threads exceeds the minimum thread pool size, unused threads are destroyed after a period of time to conserve system resources.

The management and use of threads is controlled by the following configuration settings:

**minimum number of threads**

determines the minimum number of threads that are maintained in the server's thread pool, regardless of the number of client requests that are received or accepted. If the number of available threads exceeds this minimum, then the excess unused threads are destroyed after a period of time to conserve system resources.

By default, the server uses this formula to compute the minimum number of threads:

$$\text{MAX}(5, ((\text{number-of-processors} * 2) + 1))$$

This means that the value is set to either 5 or to  $((\text{number-of-processors} * 2) + 1)$ , whichever is greater.

For example:

- ☐ If the host machine has eight processors, then the minimum number of threads is set to 17.
- ☐ If the host machine has two processors, then the minimum number of threads is set to 5.
- ☐ If the host machine has one processor, then the minimum number of threads is set to 5.

If the metadata server's invocation options include a value for THREADSMIN, then that value is used instead of the default.

**maximum number of threads**

determines the maximum number of threads that are maintained in the server's thread pool, regardless of the number of client requests that are received or accepted. If no threads are available when a client request is received, then a new thread is created only if the number of the threads in the thread pool is less than the maximum. If the maximum has already been reached, the client request waits for a thread to be returned to the pool.

As a best practice for the metadata server, THREADSMAX and THREADSMIN should be set to the same value to avoid thread resource thrashing.

By default, if no value is specified for maximum number of threads, it is set to the computed value of THREADSMIN.

**maximum number of active threads**

determines the number of threads that are allowed to run concurrently on the metadata server. This number controls the number of queries that can be active on the server at any one time.

By default, the server sets the maximum number of active threads as follows:

- ☐ If the host machine has one processor, then the maximum active number of threads is set to 2.
- ☐ If the host machine has two or more processors, then the maximum active number of threads is set to the number of processors.

If the metadata server's configuration file includes a value for MAXACTIVETHREADS, then that value is used instead of the default.

The default settings that are described in the preceding list provide a good starting point for most sites. To fine-tune the configuration to meet your specific workload and

performance requirements, you can set different values in your configuration files. For instructions and guidance, see the next topic.

---

## Specify Values for the MAXACTIVETHREADS, THREADSMIN, and THREADSMAX Options

To fine-tune the performance of the metadata server, you can specify values for the following:

- the MAXACTIVETHREADS option in the metadata server configuration file, omaconfig.xml
- the THREADSMIN and THREADSMAX parameters of the OBJECTSERVERPARMS metadata server invocation option

The values that you specify override the default configuration settings that are described in “About the MAXACTIVETHREADS, THREADSMIN, and THREADSMAX Options” on page 167.

The following are some examples of situations in which you might want to adjust these settings. You might encounter other situations in which changes are beneficial.

- If you want to dedicate only a portion of the host machine’s processors to metadata server processing, then specify values for MAXACTIVETHREADS, THREADSMIN, and THREADSMAX that are lower than the default settings. The following settings are recommended as a starting point:
  - If you want to dedicate more than one processor to metadata server processing, then set MAXACTIVETHREADS to the number of processors that you want to use. For example, if your host machine has eight processors, but you want to dedicate only five of those processors to metadata server processing, then set MAXACTIVETHREADS to 5.
 

*Note:* As a best practice, you should not set MAXACTIVETHREADS to a value less than 2. △
  - After you determine the correct setting for MAXACTIVETHREADS, specify a value for THREADSMIN and THREADSMAX that is equal to  $(\text{MAXACTIVETHREADS} * 2) + 1$ . For example, if you set MAXACTIVETHREADS to 5, then set THREADSMIN and THREADSMAX to 11.
- If clients of the metadata server are experiencing processing delays, and if you have observed that some of the processors that are dedicated to the server are underutilized, then you might want to specify values for MAXACTIVETHREADS, THREADSMIN, and THREADSMAX that are higher than the default settings. Follow these guidelines:
  - As a general rule, THREADSMIN and THREADSMAX should be set to a value that equals  $(\text{MAXACTIVETHREADS} * 2) + 1$ .
  - Increase the settings in small increments. For example, you could first set MAXACTIVETHREADS to *number-of-processors* + 2 and set THREADSMIN and THREADSMAX to a value that equals  $(\text{MAXACTIVETHREADS} * 2) + 1$ . After each change, check to see whether processor utilization increases and delay times decrease. If so, gradually increase the settings to determine which level results in the best performance. (When the settings are too high, the performance benefits will diminish.)

To specify a value for MAXACTIVETHREADS, update the metadata configuration file omaconfig.xml (see “Reference Information for omaconfig.xml” on page 280). To specify values for THREADSMAX and THREADSMIN, update the metadata server’s

sasv9\_usermods.cfg file (see “Modifying the Metadata Server Invocation Options” on page 289). Both of these files are located in the following path:

*SAS-configuration-directory\Lev1\SASMeta\MetadataServer*

After modifying the configuration files, you must restart the metadata server to enable the changes. The server reads the configuration settings only during initialization.

---

## Configuring Metadata Server Journaling

---

### About Metadata Server Journaling

In SAS 9.2, journaling is enabled by default for the SAS Metadata Server.

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. When journaling is not enabled, clients that request metadata updates must wait for updates to be written to both the in-memory records and the repository data sets on the file system before client access is resumed.

When journaling is enabled, access is returned to clients as soon as the metadata updates are written to the in-memory database and the journal file. The more time-consuming updates to the repository data sets are performed later in the background. If the metadata server fails before the update process has had a chance to apply all updates from the journal file, the metadata server automatically recovers them from the journal file when it is restarted.

For best performance, it is recommended that journaling be enabled at all times. The initial settings for journaling are as follows:

- ☐ The name of the journal file is MetadataJournal.dat.
- ☐ The journal file is written to the following path:  
*SAS-configuration-directory\Lev1\SASMeta\MetadataServer\Journal*
- ☐ 200 megabytes of storage on the metadata server are reserved for the journal file and cannot be used for other processes.
- ☐ If problems occur that prevent the repository data sets from being updated from the journal, an alert e-mail message is generated. A copy of the metadata server configuration file (omaconfig.xml) is attached to the message.

*Note:* For convenience and ease of management, the journal file is written by default to a subdirectory of the metadata server startup directory. To provide maximum server reliability and performance, you should consider changing the journal location to another file system. For details, see “Move the Journal File to Another File System” on page 171.  $\Delta$

You can also change other options such as the amount of storage that is reserved for the journal file and the alert e-mail options, as described in “Change the Journaling Options” on page 170.

---

### Change the Journaling Options

In SAS 9.2, journaling is enabled by default. If you need to change the initial settings, you can do so by modifying the metadata server configuration files. You can make the following types of changes:



Specify a different name or path for the journal file.

If you want to specify a different name or path for the journal file, you can change the value of the JOURNALPATH option in omaconfig.xml. For details about moving the journal file, see “Move the Journal File to Another File System” on page 171.

Change the alert notification settings.

The initial alert notification settings are based on values that were provided to the SAS Deployment Wizard during installation. You can change your alert notification configuration by providing different values for the following options:

- ALERTEMAIL and ALERTEMAILATTACHMENT in omaconfig.xml. You can also specify ALERTEMAILTEST, which enables you to determine whether the e-mail alert options that you have specified are valid.
- the EMAILSYS, EMAILHOST, and EMAILID metadata server invocation options.

Change the amount of storage on the metadata server that is allocated for the journal file.


The initial journal size of 200 MB (on 32-bit machines and z/OS systems) and 500MB (on 64-bit machines) should be sufficient for most sites. However, if clients regularly experience delays in getting their add, update, and delete requests processed, check the SAS Metadata Server log for messages similar to the following:

WARNING: insufficient file space available in journal file.

If this situation occurs, consider setting a higher value for the JOURNALSIZE option in omaconfig.xml.

Enable or disable journaling.

In SAS 9.2, journaling is enabled by default. If you want to disable journaling, then specify a blank string ("") for the JOURNALPATH option.

*Note:* For optimum performance, it is strongly recommended that journaling be enabled at all times. 

For details about updating omaconfig.xml, see “Reference Information for omaconfig.xml” on page 280. For details about updating the metadata server invocation options, see “Modifying the Metadata Server Invocation Options” on page 289. After modifying the configuration files, you must restart the metadata server to enable the changes.

---

## Move the Journal File to Another File System

For convenience and ease of management, the journal file (MetadataJournal.dat) is written by default to the following path on the metadata server machine:

*SAS-configuration-directory/Lev1/SASMeta/MetadataServer/Journal*

To provide maximum server reliability and performance, you should consider changing the journal location to another file system that is fast and highly reliable.

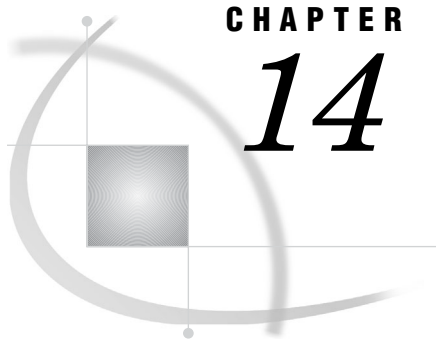
This recommendation is especially important if it is at all likely that the metadata server’s file system will fill up. If the server’s file system fills up, the journal file is not threatened because it is a fixed size. However, the metadata server log will need space to be able to record any disk-full messages in order to send them to the metadata server administrator.

To change the location of the journal file, follow these steps:

- 1 Create a directory for the journal. Make sure that the directory is in a file system that meets the following criteria:

- ☐ The file system should be fast and highly reliable.
  - ☐ The file system should be local to the metadata server machine. Do not use a network-mounted file system or network appliance.
- 2** Make sure that the new directory has the same permissions as the previous journal location.
- 3** Open the omaconfig.xml file, and edit the JOURNALPATH parameter to point to the new location and the filename MetadataJournal.dat. For details, see “Reference Information for omaconfig.xml” on page 280.
- 4** Stop the metadata server.
- 5** Physically copy the journal file (MetadataJournal.dat) from its current location to the new location.
- 6** Start the metadata server.
- 7** Check the server log to make sure that no errors occurred and that the new journal location is being used.

If the log indicates that no journal file was found and that a new empty journal file was created, then you might have specified an incorrect path in the JOURNALPATH option, or you might have copied the file to an incorrect location.



## CHAPTER

## 14

## Managing Metadata Server Performance

<i>Setting the Server's MEMSIZE Parameter</i>	173
<i>Input/Output Balancing</i>	173
<i>Configuring Concurrent Queries/Clients</i>	174
<i>Note to Users of 32-bit Windows Systems</i>	174
<i>Using %OMARUNAN (or the RUNANALYSIS Option of %OMABAKUP) to Optimize Memory Usage</i>	175
<i>About %OMARUNAN and the RUNANALYSIS Option of %OMABAKUP</i>	175
<i>When to Optimize Memory Usage</i>	175
<i>How Memory Optimization Works</i>	175
<i>Choosing Whether to Use %OMARUNAN or the RUNANALYSIS Option of %OMABAKUP</i>	176
<i>Using the %OMARUNAN Macro</i>	176
<i>%OMARUNAN Autocall Macro: Reference</i>	177
<i>Syntax</i>	177
<i>Syntax Description</i>	177
<i>Details</i>	177
<i>Recovering Memory on the SAS Metadata Server</i>	177

### Setting the Server's MEMSIZE Parameter

The SAS Metadata Server holds all queried and updated metadata in memory. Therefore, to avoid excessive paging, the amount of RAM on the server should exceed the amount of metadata.

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. For details, see Appendix 5, "Options for the Metadata Server Invocation Command," on page 289.

### 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 loaded into memory based on type. For example, when a user requests metadata for a specific column, metadata for all columns is loaded into memory. No additional I/O for column metadata queries will occur until the server is paused and resumed or stopped and restarted.

---

## Configuring Concurrent Queries/Clients

The number of processors that are available to the metadata 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.

By default, the number of queries (threads) that can be processed concurrently is set as follows:

- If the host machine has one processor, then the maximum number of active threads is set to 2.
- If the host machine has two or more processors, then the maximum number of active threads is set to the number of processors.

You can allow a different maximum number of active threads by specifying a value for the server's `MAXACTIVETHREADS` option. You can also configure the number of threads that are available in the server's thread pool, or you can change the configuration so that only a portion of the host machine's processors are dedicated to metadata server processing. For details, see "Configuring the Number of Threads Used by the Metadata Server" on page 167.

---

## Note to Users of 32-bit Windows Systems

If your metadata server is installed in a 32-bit Windows host, then you should be aware of the 2GB per-process address limit on these machines. The following 32-bit Windows hosts support a 4GT RAM Tuning feature that enables 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

To optimize the usage of memory on the metadata server, you should occasionally run the %OMARUNAN macro. This macro analyzes the handling of character variables, checking for items as unnecessary indexes, long character variables that can be stored as variable-length strings, and duplicate strings.

As an alternative to running the %OMARUNAN macro, you can set the RUNANALYSIS parameter to YES when you use %OMABAKUP to perform a backup; or select **Run Analysis Procedure** when you use the Backup Wizard. This causes the %OMARUNAN macro to be executed after the backup is complete and after client activity has resumed on the metadata server.

The next time %OMABAKUP or the Backup Wizard 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, anytime you use the **Run Analysis Procedure** option in the Backup Wizard, 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. △

---

### 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, when you use %OMABAKUP with the RUNANALYSIS option, or when you use the **Run Analysis Procedure** option in the Backup Wizard:

- 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. △

- 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 METAIMDB 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 METAIMDB 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 either %OMABAKUP or the Backup Wizard before you run %OMARUNAN.

%OMABAKUP also creates the OMA\_Varansas7bcat catalog 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\_Varansas7bcat reflects the last backup that was made. That is, if a full backup was made, then OMA\_Varansas7bcat 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")
```

---

### Syntax Description

#### 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. △

---

## Recovering Memory on the SAS Metadata Server

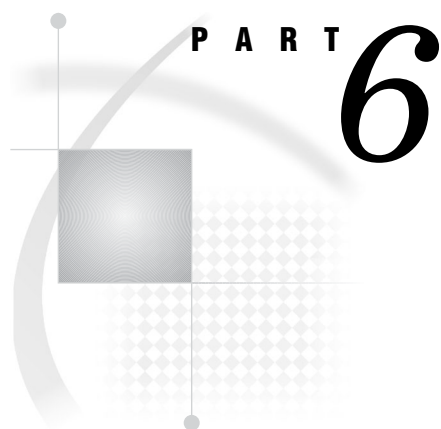
The SAS Metadata Server supports two approaches for quickly recovering memory on a running server:

- Issue a Pause action followed by a Resume action. The Pause action instructs the SAS Metadata Server to stop accepting new client requests and closes the repository containers. The Resume action resumes client activity. Repository containers are re-opened as clients request metadata from them.
- Issue a Reset action. The Reset action pauses and resumes the metadata server in one step.

Pause, Resume, and Reset actions can be issued in SAS Management Console. See “Use SAS Management Console to Stop, Pause, Resume, or Reset the SAS Metadata Server” on page 65. Pause and Resume actions can also be issued by using PROC METAOPERATE. See *SAS Language Interfaces to Metadata*.



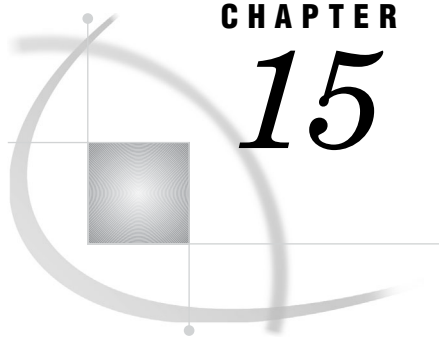




## Managing SAS Metadata Repositories

*Chapter 15* . . . . . **Creating, Registering, Moving, Copying, Renaming, and  
Deleting SAS Metadata Repositories 181**





## CHAPTER

## 15

# Creating, Registering, Moving, Copying, Renaming, and Deleting SAS Metadata Repositories

<i>About SAS Metadata Repositories</i>	181
<i>About SAS Metadata Repository Access Modes</i>	182
<i>Create a Custom Repository</i>	183
<i>Change the Properties of a Metadata Repository</i>	184
<i>Copying, Promoting, Importing, and Exporting Your Metadata</i>	185
<i>Tools for Copying, Promoting, Importing, and Exporting Metadata</i>	185
<i>About the Metadata Manager Import and Export Wizards</i>	187
<i>Moving a Metadata Repository to a New Location on the Same Metadata Server</i>	188
<i>Porting a Metadata Repository to a Different Host Environment</i>	189
<i>About the %OMAPORT Macro</i>	189
<i>%OMAPORT Autocall Macro: Reference</i>	189
<i>Syntax</i>	189
<i>Syntax Description</i>	189
<i>Details</i>	190
<i>Registering a Metadata Repository</i>	191
<i>Unregistering a Metadata Repository</i>	192
<i>Deleting a Metadata Repository</i>	193

## About SAS Metadata Repositories

A *metadata repository* is a physical location in which a collection of related metadata objects is stored. Metadata repositories 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.

The SAS Metadata Server supports three types of metadata repositories:

### Foundation repositories

are the required metadata stores for SAS Metadata Servers. Each metadata server has one foundation repository that is created by default when the metadata server is configured. The default name of the foundation repository is **Foundation**. (If you performed a custom installation, then you might have assigned a different name.) You cannot create more than one foundation repository.

### Custom repositories

are optional metadata stores for SAS Metadata Servers that can be configured in addition to the foundation repository. Custom repositories are useful for physically segregating metadata for storage or security purposes.

For example:

- If your metadata storage requirements exceed the capacity of a single metadata storage location, you could create custom repositories on separate storage devices that are accessible to the metadata server.
- If you have metadata that is highly sensitive, you could create a custom repository for that metadata and store it in a directory that has an exceptionally high level of protection from unauthorized access.

Custom repositories appear as separate folders in the Folders view of SAS Management Console and in other client applications.

To create a custom repository, see “Create a Custom Repository” on page 183.

#### Project repositories

are metadata repositories that serve as isolated work areas or playpens. Project repositories are available for SAS Data Integration Studio only. In general, each user who participates in change management has his or her own project repository.

A project repository enables a user to check out and lock metadata from a foundation repository or a custom repository so that the metadata can be modified and tested in a separate area. When the metadata is ready, it is checked in and unlocked so that it is available to other users.

For information about creating project repositories, see “Setting Up Change Management” in the *SAS Intelligence Platform: Desktop Application Administration Guide*.

The SAS Deployment Wizard creates just one repository, which is the foundation repository. If you create additional repositories, then the metadata in a given repository can be associated with metadata in other repositories as follows:

- Metadata in a custom repository can be associated with metadata in the foundation repository and with metadata in all other custom repositories.
- Metadata in the foundation repository or in any custom repository can be checked out to any project repository, as long as the user has the appropriate permissions.
- Metadata that is checked out to a project repository can be associated with metadata in the foundation repository or with metadata in any custom repository (but not with metadata that is checked out to a different project repository).

#### Note:

- In SAS 9.2, it is not necessary to define dependencies between repositories. If a repository dependency was defined in SAS 9.1.3, then the dependency relationship is eliminated during the SAS 9.1.3 to SAS 9.2 migration process. However, the existing associations between metadata objects still remain in effect
- Effective with the November 2010 release, an additional repository called BILineage is created automatically the first time an unrestricted administrative user logs on to SAS Management Console. This repository stores metadata lineage and impact analysis data that is generated by SAS Data Integration Studio, SAS OLAP Cube Studio, and the BI Lineage plug-in for SAS Management Console. The BILineage repository should not be used for any other purpose. For more information, see “Using the BI Lineage Plug-in” on page 203.

△

---

## About SAS Metadata Repository Access Modes

The *access mode* of a metadata repository is the repository’s intended access state. When the metadata server is online, the access state of a repository is equal to its

access mode. When the metadata server is paused to an Offline state or an Administration state, the access state of a repository can change temporarily (see “How the Access State of the Metadata Server Affects the Access States of Metadata Repositories” on page 78). When the metadata server is resumed, the repository’s access state returns to its registered access mode.

A repository’s access mode is recorded in the Access attribute of the repository’s metadata definition. When you create a new repository, you can place it in any of the following access modes:

#### *Online*

Clients can read, write, and update metadata. This access level is the default for new repositories.

#### *ReadOnly*

Clients can read metadata but cannot write, update, or delete metadata.

**Example Use Case** If work on a project is complete, you can assign an access mode of ReadOnly to the project’s repository so that users can continue to access the metadata without updating it.

#### *Administration*

Only users who are in the **Metadata Server: Unrestricted** role can read or write metadata. (For information about this role, see “Overview of Initial Roles, Groups, and Users” on page 24.)

**Example Use Case** If metadata problems occur (for example, if the metadata in a repository becomes corrupted), you can assign an access mode of Administration while the issue is being analyzed and repaired.

#### *Offline*

Clients can neither read nor write metadata to the repository.

**Example Use Case** If work on a project is complete and users no longer need to access the metadata, you can assign an access mode of Offline to free server memory. If the metadata is needed in the future, the access mode can be changed back to ReadOnly or Online.

For information about changing a repository’s access mode, see “Change the Properties of a Metadata Repository” on page 184.

---

## Create a Custom Repository

To organize your metadata, the recommended practice is to create folders instead of creating separate repositories (see Chapter 16, “Working with SAS Folders,” on page 197). However, you might want to create a custom repository in order to physically segregate metadata for storage or security purposes.

When you create a custom repository, you should be aware that the metadata server must add, update, and delete repositories when no other activity is taking place on the server. Therefore, the server delays other client requests until the repository changes are complete. This delay might have a small effect on server performance.

To create a new custom repository, follow these steps:

- 1 Log on to SAS Management Console as a user with the appropriate credentials (see “Who Can Do What in the Metadata Manager Plug-in” on page 34).
- 2 On the **Plug-ins** tab, expand the **Metadata Manager** node.
- 3 Right-click the **Active Server** node, and select **New Repository**.
- 4 Follow the instructions on the wizard pages. As you answer the wizard’s prompts, follow these guidelines:

- Be sure to specify a unique name for the repository.
- Generally, the path for a new repository should be `SAS-configuration-directory/Lev1/SASMeta/MetadataServer/MetadataRepositories/repository-name`. This is the default location for the initial foundation repository. By using this location, you will ensure that the directory inherits access permissions from the **MetadataServer** directory.  
If you choose to specify a different path, follow these guidelines:
  - Metadata repositories should be kept either on a local file system or a high-performance, high-availability network file system.
  - Do not place the directory under the **rposmgr** directory.
  - The directory should not contain any files other than the repository files.
  - Before completing the wizard, make sure that the process owner for the metadata server has credentials to create the directory (or to write to the directory, if it already exists).
- If you do not want the repository to be Online, you can select a different value in the Access attribute (see “About SAS Metadata Repository Access Modes” on page 182).

When you create a custom repository, SAS Management Console automatically does the following:

- creates a corresponding folder that appears in the **Folders** tab of SAS Management Console and in other client applications.
- creates a default access control template (ACT) that you can use to restrict user access to the repository. For details about repository access controls, see “Overview of the Metadata Authorization Model” in the *SAS Intelligence Platform: Security Administration Guide*.

You can also use SAS language elements to create a custom repository. If you use this method, then you must use SAS Management Console to initialize the repository. The initialization process creates the Default ACT and the resource templates.

---

## Change the Properties of a Metadata Repository

You can change the name or the registered access mode of a metadata repository. For foundation repositories, you can also change the location of users’ folders. Follow these steps:

- 1 Log on to SAS Management Console as a user with the appropriate credentials (see “Who Can Do What in the Metadata Manager Plug-in” on page 34).
- 2 On the **Plug-ins** tab, expand the **Metadata Manager** node and the **Active Server** node.
- 3 In the Active Server tree, right-click the repository whose properties you want to change, and select **Properties**.
- 4 If you want to change the repository’s name, enter the new name on the **General** tab and click **OK**. When you rename a custom repository, the corresponding folder in the **Folders** tab is also renamed.


### CAUTION:

**Do not rename a custom repository if the repository (or one of its subfolders) contains an object that has an association with another object. Doing so could cause the associated objects to malfunction.** △

When you rename a custom repository, the corresponding folder in the SAS Folders tree is automatically renamed to the new name.

- 5 If you want to change the repository's access mode, click the **Registration** tab and make a selection in the **Access** field (see “About SAS Metadata Repository Access Modes” on page 182). Then click **OK**.
- 6 As a default, users' home folders are located in the path **/Users** (see “About the Initial Folder Structure for New Installations” on page 198). If you want to store users' home folders in a different path (or if you chose not to create a **/Users** folder during installation but you would like to create one now):
  - a On the **Folders** tab, create a new folder where you want home folders to be stored, or rename the existing folder.

**CAUTION:**

**If any existing home folders contain objects that have associations with another objects, then renaming the Users folder could cause the associated objects to malfunction.** In this situation, you should create a new folder instead of renaming the existing folder. Then use the Export SAS Package and Import SAS Package functions to move users' home folders to the new location so that metadata associations are preserved. 

- b On the **Plug-ins** tab, open the **Properties** dialog box for the foundation repository. Click the **Registration** tab. In the **Metadata Location for Users' Folders** field, enter the new path for users' home folders.

---

## 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. The following table lists each tool, the purpose of each tool, the tool's location in SAS Management Console (if applicable), and a reference for documentation about the tool:

Tool Name	Purpose	Location in SAS Management Console	Documentation Reference
Backup and Restore Wizard	Make a backup copy of your repositories, or restore your repositories from a backup. These wizards enable you to easily create and run backup and restore jobs that use the %OMABAKUP macro.	Metadata Manager ► Metadata Utilities ► Backup & Restore	Chapter 10, “Using the Backup Wizard in SAS Management Console,” on page 137
%OMABAKUP macro	Make a backup copy of your repositories or restore your repositories from a backup. You can submit the macro either in the program editor or from a .sas file.	Not applicable	Chapter 11, “Using the %OMABAKUP Macro to Perform Backups and Restores,” on page 147
Import Metadata and Export Metadata <sup>1</sup>	Copy metadata for relational data and DBMSs to and from external sources.	Metadata Manager ► Active Server ► Import Metadata and Metadata Manager ► Active Server ► Export Metadata	“About the Metadata Manager Import and Export Wizards” on page 187
Export SAS Package and Import SAS Package <sup>2</sup>	Promote individual objects or groups of objects from one location to another, either on the same metadata server or a different metadata server. You can also include associated physical content in the promotion.	In the <b>Folders</b> tab, right-click a folder and select <b>Export SAS Package</b> or <b>Import SAS Package</b> .	Chapter 17, “Using the Promotion Tools,” on page 213 and SAS Management Console Help
Copy, Paste, and Paste Special	Copy and paste a group of objects or folders into a target folder on the same metadata server.	In the <b>Folders</b> tab, right-click a folder and select <b>Copy</b> . Then right-click another folder and select <b>Paste</b> or <b>Paste Special</b> .	SAS Management Console Help



Tool Name	Purpose	Location in SAS Management Console	Documentation Reference
%OMAPORT macro	Port a metadata repository from one operating system to another.	Not applicable	“Porting a Metadata Repository to a Different Host Environment” on page 189
Replication Wizard	The Replication Wizard has been deprecated. For more information, see SAS Note 40834 or contact SAS Technical Support.	Metadata Manager ► Metadata Utilities ► Replication	SAS Note 40834

- 1 SAS Data Integration Studio enables you to perform this function with change analysis. The change analysis feature enables you to compare imported metadata to existing metadata in the repository and selectively apply changes. For details, see the product Help for SAS Data Integration Studio.
- 2 This feature is also available in SAS Data Integration Studio.

## 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. To use this feature, you must license and install the appropriate SAS Metadata Bridge. For information about SAS Metadata Bridges, contact your SAS representative.

You can import and export relational metadata in any format that is accessible with a SAS Metadata Bridge. Relational metadata includes the metadata for the following objects: tables, columns, indexes, and keys (including primary keys and foreign keys). 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 you want to export it for use with other data warehousing software.

To export metadata, expand the **Metadata Manager** node in SAS Management Console, right-click the **Active Server** node, and select **Export Metadata**. In the Metadata Export Wizard, you specify the export format, navigate to the folders that contain the table objects to be exported, and select the tables. You then specify the path and filename to which the metadata is to be exported.

To import metadata, expand the **Metadata Manager** node in SAS Management Console, right-click the **Active Server** node, and select **Import Metadata**. In the Metadata Export Wizard, you specify the import format, specify the path and filename that contains the metadata to be imported, and specify the library to which the metadata is to be imported.

For more information about using the wizards, see the Help for SAS Management Console. Error messages generated by the wizards are written to SASMCErrLog.txt, which is located in one of the following paths:

- On Windows Vista systems:  
**C:\Users\user\AppData\Roaming\SAS\SASManagementConsole\9.2**
- On other Windows systems: **C:\Documents and Settings\user\Application Data\SAS\SASManagementConsole\9.2**
- On UNIX systems: *user's-home-directory*/**SAS/SASManagementConsole/9.2**

*Note:* SAS Data Integration Studio provides 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 details, see the product Help for SAS Data Integration Studio.  $\Delta$

---

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

You can move a repository to a new location while keeping its registration on the same metadata server. When choosing the new location, remember that metadata repositories should be kept either on a local file system or a high-performance, high-availability network file system.

You can use one of two approaches to move the repository, depending on whether repository availability or disk space is more important.

The first approach enables clients to continue to read 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. Follow these steps:

- 1 First, make a backup of the repository (see Chapter 9, “Best Practices for Backing Up and Restoring Your System,” on page 123).
- 2 Use SAS Management Console to change the repository’s access mode to ReadOnly (see “Change the Properties of a Metadata Repository” on page 184).
- 3 Use operating system commands to copy the contents of the repository to the new location.
- 4 When the files are in the new location, use SAS Management Console to delete the repository (see “Deleting a Metadata Repository” on page 193). The Delete action deletes the contents of the repository from the previous location and removes the repository’s registration from the repository manager.
- 5 Use SAS Management Console to re-register the repository (see “Registering a Metadata Repository” on page 191). When you are prompted for the repository path, specify the repository’s new location. The metadata server reads the files in the specified location to create the new repository’s registration.

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

- 1 First, make a backup of the repository (see Chapter 9, “Best Practices for Backing Up and Restoring Your System,” on page 123).
- 2 In the Metadata Manager, expand the **Active Server** node.
- 3 Unregister the repository (see “Unregistering a Metadata Repository” on page 192).
- 4 Use operating system commands to move the contents of the repository to the new location.
- 5 When the files are in the new location, re-register the repository (see “Registering a Metadata Repository” on page 191). When you are prompted for the repository path, specify the repository’s new location. The metadata server reads the files in the specified location to create the new repository’s registration.

---

## Porting a Metadata Repository to a Different Host Environment

---

### About 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* machine. The target machine is the metadata server machine to which you are porting the repository. The target machine must have a SAS Metadata Server defined on it. The *source* machine is the machine that contains the repository that you are porting to the target machine.

You can port a repository between any two hosts that are supported by the SAS Metadata Server. Ask your on-site SAS support personnel for a list of the platforms that are supported for SAS Intelligence Platform servers.

#### Note:

- The %OMAPORT macro can convert the data sets of only one repository at a time. If you are porting multiple repositories, you must copy and convert them separately.
- %OMAPORT cannot process repository data sets that are in the SAS 9.1 or SAS 9.1.3 format. While SAS 9.1 and SAS 9.1.3 used UTF-16 or UTF-32 encoding for binary data, SAS 9.2 uses UTF-8 encoding and compressed data sets. Repository data sets with outdated encoding will cause the macro to terminate abnormally with an error message.

The SAS 9.1.3-to-SAS 9.2 migration process automatically converts repositories to the new format.

△

---

### %OMAPORT Autocall Macro: Reference

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

#### Syntax

```
LIBNAME OLDRPOS "source-repository-path";
```

```
LIBNAME NEWRPOS "target-repository-path";
```

```
%OMAPORT(OLDRPOS,NEWRPOS,NULL)
```

#### Syntax Description

OLDRPOS	specifies the libref for the source repository. Specify the path to the backup repository data sets that you create as instructed in “Details” on page 190. If the backup location and the target machine are connected in a network such as a Network File System network, and you specify an absolute pathname for the <i>source-repository-path</i> ,
---------	--

then %OMAPORT references the repository on the machine that contains the backup. Otherwise, you must use one of the copy methods listed in “Details” on page 190 to move the source repository from the backup location to a working directory on the target machine and specify this working directory as the *source-repository-path*.

## NEWRPOS

specifies the libref for the target directory on the target machine in which to create the converted repository data sets. Create this directory before you run the macro; the macro will not create it for you.

For example, if you are porting a foundation repository to a Windows host, you should specify a *target-repository-path* similar to this:

*SAS-configuration-directory\Lev1\SASMeta\MetadataServer\MetadataRepositories\Foundation.*

### Note:

- We recommend that metadata repositories be kept either on a local file system or a high-performance, high-availability network file system.
- The repository directory name in the backup location might be different from the repository directory name on the source metadata server. This can occur because %OMABAKUP names the repository backup directory based on the Name attribute of the repository.

For example, suppose the Name attribute of a repository is Custom, but the repository data sets are stored in a directory called **Custom1**. If you use %OMABAKUP or the Backup Wizard to back up the repository, the repository data sets are stored in the backup location a directory called **Custom**.

If you want to ensure that *target-repository-path* specifies the same repository directory name that is used on the source metadata server, then check the source metadata server to determine the correct directory name. The directory name is also provided in the *name-of-backup-job.lst* file that is produced by the backup job.

△

## NULL

is ignored in SAS 9.2. In SAS 9.1.3, this parameter specifies the libref for the target metadata server’s repository manager.

## Details

Before using %OMAPORT, follow these steps:

- 1 Create an offline copy of the repository data sets on the source machine. We recommend using %OMABAKUP, which automatically pauses the metadata server to an Offline state, copies the data sets to a backup location, and then resumes the server. For instructions, see Chapter 11, “Using the %OMABAKUP Macro to Perform Backups and Restores,” on page 147. You can then access the offline (backup) copies of the repository data sets and avoid additional disruption to the source metadata server.

*Note:* If you choose to use a method other than %OMABAKUP to copy the repository data sets, first be sure to either pause the server to an Offline state or

change the repository's access mode to Offline or ReadOnly. Otherwise, the copies will be unusable. △

**CAUTION:**

**Do not use PROC CPORT or PROC CIMPORT to create the offline copy.** The CPORT and CIMPORT procedures do not preserve the required attributes and encoding of SAS 9.2 data sets. △

- 2 Ensure that the target machine has a SAS Metadata Server installed on it, and ensure that the target machine has a directory for the converted repository.
- 3 Ensure that the target machine can access a binary copy of the backup repository data sets that you created in step 1. Use one of the following methods to obtain this access:
  - Use a network protocol system such as Network File System to enable the target machine to access the backup copies of the repository data sets.
  - Before running %OMAPORT, use binary File Transfer Protocol to copy the backup repository data sets from the backup location to a working directory on the target machine.
  - Before running %OMAPORT, use SAS/CONNECT to establish a connection from the backup location to the target machine. Then use PROC UPLOAD to create a binary copy of the backup repository data sets in a working directory on the target machine. The code syntax is as follows:

```
filename source source "backup-repository-directory";
rsubmit;
    filename target "target-repository-directory";
    proc upload infile=source("*")
                outfile=target
                binary;

    run;
endrsubmit;
```

After you have established the necessary access, run %OMAPORT on the target machine to convert the repository data sets to the appropriate internal format.

After the port operation, take the following steps:

- 1 The %OMAPORT macro does not register the ported repository on the target metadata server. After the data set conversion is complete, you must use SAS Management Console to connect to the target metadata server and register the ported repository (see “Registering a Metadata Repository” on page 191).
- 2 %OMAPORT does not affect metadata values in the source repository. You must manually change any host-dependent values that are stored in the metadata (such as pathnames and machine names) so that they are correct for the target machine.
- 3 You can either delete or continue using the repository on the source metadata machine.

---

## Registering a Metadata Repository

When you register a metadata repository, information about the repository is added to the repository manager so that the repository can be accessed by clients.

You might need to register a repository in the following situations:

- You have moved the repository to a new location on the same metadata server and you want to register its new location (see “Moving a Metadata Repository to a New Location on the Same Metadata Server” on page 188).

- You have ported the repository to a different host environment and you want to register its new location (see “Porting a Metadata Repository to a Different Host Environment” on page 189).

To register a repository:

- 1 Log on to SAS Management Console as a user with the appropriate credentials (see “Who Can Do What in the Metadata Manager Plug-in” on page 34).
- 2 In the Metadata Manager, right-click the **Active Server** node and select **Register Repository**. Then follow the instructions that are provided by the wizard.

---

## Unregistering a Metadata Repository

When you unregister a metadata repository, information about the repository is removed from the repository manager, and the repository can no longer be accessed by clients.

You might need to temporarily unregister a repository in the following situations:

- You are moving the repository to a new location on the same metadata server (see “Moving a Metadata Repository to a New Location on the Same Metadata Server” on page 188).
- You are porting the repository to a different host environment (see “Porting a Metadata Repository to a Different Host Environment” on page 189).

### CAUTION:

**Unregistering the foundation repository makes the metadata server unusable.** Therefore, you should unregister a foundation repository only in special situations, such as those that are described in the preceding list. △

The following rules apply to unregistering a repository:

- You cannot unregister a foundation repository if other repositories (including custom repositories or project repositories) are registered. You must first unregister the other repositories.
- Unregistering a foundation repository requires special credentials (see “Who Can Do What in the Metadata Manager Plug-in” on page 34).
- You cannot unregister a repository that is in an Offline or ReadOnly state. You must first change the access mode of the repository to Online or Administration (see “Change the Properties of a Metadata Repository” on page 184).

To unregister a repository:

- 1 Log on to SAS Management Console as a user with the appropriate credentials (see “Who Can Do What in the Metadata Manager Plug-in” on page 34).
- 2 In the Metadata Manager, expand the **Active Server** node.
- 3 Right-click the repository that you want to unregister, and select **Unregister Repository**.

To re-register a repository:

- 1 Log on to SAS Management Console as a user with the appropriate credentials (see “Who Can Do What in the Metadata Manager Plug-in” on page 34).
- 2 In the Metadata Manager, expand the **Active Server** node.
- 3 Right-click the **Active Server** node, and select **Register Repository**. Then follow the instructions on the wizard pages.

## Deleting a Metadata Repository

When you delete a metadata repository, all of the repository's metadata and metadata containers are deleted. In addition, the repository's registration is removed from the repository manager.

You might want to delete a repository in the following situations:

- You have moved the repository to a new location on the same metadata server and you want to delete it from its previous location (see “Moving a Metadata Repository to a New Location on the Same Metadata Server” on page 188) .
- You have ported the repository to a different host environment and the repository is no longer needed in the original environment (see “Porting a Metadata Repository to a Different Host Environment” on page 189).

### CAUTION:

**Deleting the foundation repository makes the metadata server unusable.** Therefore, you should delete a foundation repository only in special situations, such as those that are described in the preceding list. △

The following rules apply to deleting a repository:

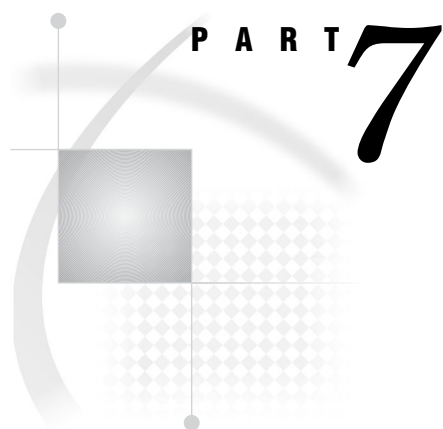
- You cannot delete a foundation repository if other repositories (including custom repositories or project repositories) are registered. You must first unregister the other repositories.
- Deleting a foundation repository requires special credentials (see “Who Can Do What in the Metadata Manager Plug-in” on page 34).
- You cannot delete a repository that is in an Offline or ReadOnly state. You must first change the access mode of the repository to Online or Administration (see “Change the Properties of a Metadata Repository” on page 184).
- You must have the appropriate user credentials (see “Who Can Do What in the Metadata Manager Plug-in” on page 34).

To delete a repository:

- 1 Log on to SAS Management Console as a user with the appropriate credentials (see “Who Can Do What in the Metadata Manager Plug-in” on page 34).
- 2 In the Metadata Manager, expand the **Active Server** node.
- 3 Right-click the repository that you want to delete, and select **Delete**.





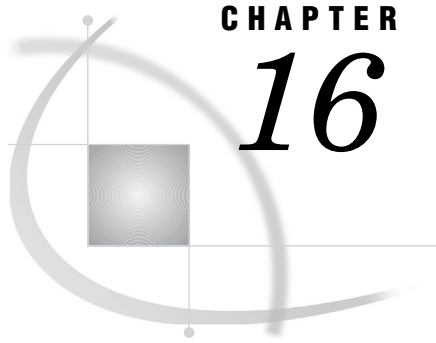


## Managing Metadata Content

*Chapter 16* . . . . . **Working with SAS Folders** 197

*Chapter 17* . . . . . **Using the Promotion Tools** 213





## CHAPTER

## 16

## Working with SAS Folders

<i>About the SAS Folder Structure</i>	197
<i>Purpose of the SAS Folder Structure</i>	197
<i>About the Initial Folder Structure for New Installations</i>	198
<i>About the Initial Folder Structure for Migrated Systems</i>	200
<i>Best Practices for Managing SAS Folders</i>	201
<i>Best Practices for Maintaining Associations Among Objects in SAS Folders</i>	202
<i>Using the BI Lineage Plug-in</i>	203
<i>About the BI Lineage Plug-in</i>	203
<i>Administering the BI Lineage Plug-in</i>	205
<i>Running and Scheduling Scans</i>	205
<i>Granting Users Permission to View Scan Results</i>	205
<i>Viewing the Results of BI Lineage Scans</i>	206
<i>Suggested Uses for the BI Lineage Plug-in</i>	208
<i>Customizing Your Folder Structure</i>	208
<i>Guidelines for Creating Custom Folders</i>	208
<i>Examples of Custom Folder Structures</i>	210
<i>Example 1: Multiple Lines of Business Across Multiple Regions</i>	210
<i>Example 2: Multiple Lines of Business with Separate Data Integration and Business Intelligence Users</i>	211
<i>Example 3: Separate Folder Structures for Development, Test, and Production</i>	211

## About the SAS Folder Structure

### Purpose of the SAS Folder Structure

SAS Intelligence Platform clients use a hierarchy of *SAS folders* to store metadata for the following:

- content such as channels, cubes, data explorations, generated transformations, information maps, jobs, libraries, mining results, OLAP schemas, prompts, reports, stored processes, subscribers, and tables
- SAS system content that is not directly accessed by business users

When you install SAS, a set of default SAS folders is created. The folders are arranged in a structure that segregates system information from business information, provides personal folders for individual users, and provides an area for shared data. Within this overall structure, you can create a customized folder structure that meets the information management, data sharing, and security requirements of your organization.

If you have installed Web applications such as SAS Web Report Studio, then some SAS folders (for examples, folders that contain metadata for reports) have associated physical content that resides either on the SAS Content Server or in a file system. In these instances, a procedure called *content mapping* maps the metadata folder structure to corresponding physical folders that have the same organization. Content mapping is automatically configured when you install your system.

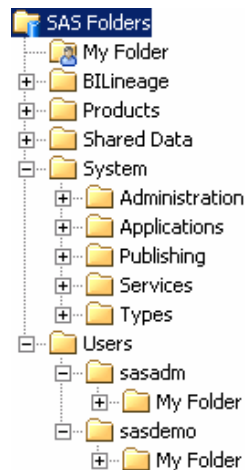
The **Folders** tab of SAS Management Console displays all SAS folders that the user has permission to view. Most other client applications display SAS folders only if they contain content that is relevant to the application, subject to the user's permissions.

*Note:* In SAS Data Integration Studio, metadata objects that are created in project repositories are not displayed in the **Folders** tab until they are checked in. △

---

## About the Initial Folder Structure for New Installations

For a newly installed system, the initial folder structure is similar to the following:



### CAUTION:

**Renaming, moving, or deleting SAS folders and the objects that they contain can cause unpredictable results.** Before renaming, moving, or deleting an object or a folder, see the guidelines in “Best Practices for Managing SAS Folders” on page 201 and “Best Practices for Maintaining Associations Among Objects in SAS Folders” on page 202. △

The initial folder structure includes the following main components:

#### SAS Folders

is the root folder for the folder structure. This folder cannot be renamed, moved, or deleted. It can contain other folders, but it cannot contain individual objects.

#### My Folder ( )

is a shortcut to the personal folder of the user who is currently logged on (for details, see the other instance of My Folder in this list)

#### BILineage

is the root folder for the BILineage metadata repository. This repository, which is new with the November 2010 release, stores results from scans that have been run using the BI Lineage plug-in. This folder should not be renamed, moved, or deleted. The repository and folder should not be used for any purpose other than

storing scan results. For more information about the BI Lineage plug-in, see “Using the BI Lineage Plug-in” on page 203.

### Products

contains folders for individual SAS products. These folders contain content that is installed along with the product. For example, some products have a set of initial jobs, transformations, stored processes, or reports which users can modify for their own purposes. Other products include sample content (for example, sample stored processes) to demonstrate product capabilities.

Where applicable, the content is stored under the product’s folder in subfolders that indicate the release number for the product.

*Note:* During installation, the SAS Deployment Wizard allows the installer to assign a different name to this folder. Therefore, your Products folder might have a different name. △

### Shared Data

is provided for you to store user-created content that is shared among multiple users. Under this folder, you can create any number of subfolders, each with the appropriate permissions, to further organize this content.

*Note:* You can also create additional folders under **SAS Folders** in which to store shared content. △

### Users

contains folders that belong to individual users. These folders are referred to as the users’ *home folders*. The name of each home folder is based on the value of the user’s Name field in User Manager.

The first time a user logs on to an application that requires a home folder, the user’s home folder is automatically created. That same folder is then used by other applications that the user logs on to.

As a default, users do not have write access to their home folders. However, each home folder contains a folder called My Folder (referred to as the user’s *personal folder*) to which the user has full access.

*Note:*

- When home folder names are created, any slash (/) or back-slash (\) characters in the user’s name are converted to hyphens. For example, if the user’s Name field contains **Jane/Doe**, then **Jane-Doe** is used as the folder name. Leading and trailing spaces, as well as unprintable characters, are also removed.
- In a custom installation, the SAS Deployment Wizard allows the installer to choose whether to create a **Users** folder. Therefore, your site might not have a **Users** folder. If you do not have a **Users** folder, but you would like to create one, see “Change the Properties of a Metadata Repository” on page 184.

△

### My Folder

is the user’s personal folder. This folder exists under each user’s home folder. It is the standard location for storing content (for example, information maps and reports) that the user creates, similar to the **My Documents** folder in Microsoft Windows.

The default security settings for **My Folder** are as follows:

- The folder is visible only to the owning user and to unrestricted users.
- The owning user has WriteMemberMetadata (WMM) permission on this folder. This means that only the owning user and unrestricted users have permission to add content to or remove content from the folder.

- The owning user cannot rename, move, delete, or change permissions on this folder. However, the owning user can create additional subfolders under this folder to organize content.

When the owning user is logged on, a pointer to the user's personal folder (**My Folder**) appears at the top of the folder hierarchy.

#### **Application Data**

is a standard location in which applications can store user preferences and other user-specific configuration information. The folder is created under the user's home folder the first time that an application needs to use it. The folder is to be accessed only by applications and is not accessed directly by the user.

#### **System**

contains SAS system objects that are not directly accessed by business users. This folder contains the following folders:

##### **Administration**

contains objects for backup jobs and restore jobs that are created by using the Backup Wizard.

##### **Applications**

contains folders for individual SAS applications that have system objects. Under these folders, the objects are stored in subfolders that correspond to individual release numbers.

##### **Publishing**

contains channel and subscriber objects that are used by the Publishing Framework.

##### **Types**

contains type definitions for public objects that exist on this metadata server.

If you have defined one or more custom repositories, then the folder structure also includes a top-level folder that corresponds to each custom repository. For more information about custom repositories, see "Create a Custom Repository" on page 183.

Project repositories do not appear in the **Folders** tab of SAS Management Console. They appear only in SAS Data Integration Studio.

---

## About the Initial Folder Structure for Migrated Systems

If you migrate your system from SAS 9.1.3, then the **Products**, **Users**, **Shared Data**, and **System** folders are created just as they are for newly installed system, as described in the previous topic. SAS 9.1.3 objects are handled as follows:

- The SAS 9.1.3 Shared Data folder structure is moved in its entirety to SAS 9.2. The folder structure and folder contents remain intact.
- SAS 9.1.3 objects that were not contained in folders (for example, OLAP schemas and cubes) are moved to the SAS 9.2 Shared Data folder.
- For each user who had objects (for example, report objects) in SAS 9.1.3 personal folders, a home folder and a personal folder are created in the SAS 9.2 Users folder. Users' objects are moved from the SAS 9.1.3 personal folders to the SAS 9.2 personal folders.
- The SAS 9.1.3 BIP tree folder structure is moved in its entirety to SAS 9.2 under a folder called **BIP Tree**. (This folder appears only in migrated systems.)

The folder structure and folder contents under **BIP Tree** remain intact, except for the contents of users' personal folders (as described in the previous bullet).

- Any content mapping that was in effect in SAS 9.1.3 is carried forward to SAS 9.2.

## Best Practices for Managing SAS Folders

To ensure the integrity of your system, follow these best practices when interacting with SAS folders in SAS Management Console:

Use personal folders for personal content and use shared folders for content that multiple users need to view.

By default, users cannot view other users' personal folders. Therefore, personal folders should be used for content that needs to be viewed and used only by the owning user. If the content needs to be viewed or used by other users, then it should be placed under the **Shared Data** folder or in a new folder structure that you create under **SAS Folders**.

To ensure secure and efficient sharing of content, the system administrator should create a folder structure for shared data that meets the needs of the organization. The appropriate permissions can then be assigned to each folder. For guidance, see “Customizing Your Folder Structure” on page 208.

Use folders, instead of custom repositories, to organize content.

In most cases, folders are the preferred method for organizing content. Custom repositories should be created only when there is an overriding reason to physically segregate repository data sets. For more information, see Chapter 15, “Creating, Registering, Moving, Copying, Renaming, and Deleting SAS Metadata Repositories,” on page 181.

It is recommended that you not delete or rename the Users folder, even if you have permission to do so.

If you have a reason to delete or rename the **Users** folder, then you must change the metadata repository configuration to reflect the change (see “Change the Properties of a Metadata Repository” on page 184).

Do not delete or rename the home folder or personal folder (My Folder) of an active user, even if you have permission to do so.

As a best practice, do not rename an active user's home folder or personal folder. If you do so, a new (empty) personal folder will be created the next time the user refreshes or logs on to an application that requires the folder. In addition, the contents of the renamed folder will not be visible to the user.

If you delete an active user's home folder or personal folder, the user will lose any existing personal content, and a new (empty) personal folder will be created the next time the user refreshes or logs on to an application that requires the folder.

Do not delete or rename the Products or System folders or their subfolders, even if you have permission to do so.

Deleting or renaming the **Products** or **System** folders or their subfolders could cause erroneous or unexpected behavior in client applications or solutions.

Use caution when renaming the Shared Data folder.

Renaming the **Shared Data** folder can affect associations and references to objects that are stored in this folder or its subfolders. For details, see “Best Practices for Maintaining Associations Among Objects in SAS Folders” on page 202.

When you create new folders, the security administrator should set permissions to ensure appropriate access.

For details, see “Permissions on Folders” in the *SAS Intelligence Platform: Security Administration Guide*.

## Best Practices for Maintaining Associations Among Objects in SAS Folders

To ensure that associations among objects are maintained, follow these best practices when interacting with folders and with objects that are stored in SAS folders.

Use available tools to identify associations among objects.

Effective with the November 2010 release, you can use the BI Lineage plug-in for SAS Management Console to identify objects that are associated with information maps and reports. For details, see “Using the BI Lineage Plug-in” on page 203.

To determine associations and dependencies for other objects, you can right-click the object and select **Export SAS Package**. On the second wizard page, highlight the object, and then use the **Dependencies** and **Used by** tabs to see the object’s associations. Click **Cancel** to exit the wizard.

Use the appropriate SAS applications to interact with metadata objects.

To interact with objects that are stored in SAS folders, users should use the specific SAS application that creates and manages those objects. For example:

- Use SAS Data Integration Studio to interact with jobs.
- Use SAS Information Map Studio to interact with information maps.
- Use SAS Web Report Studio to interact with reports.

Certain administrative tasks are exceptions. For example, administrators can use the **Folders** tab of SAS Management Console to perform the following tasks:

- manage permissions on folders (and, if necessary, on individual objects)
- use the Export SAS Package and Import SAS Package wizards to promote objects or groups of objects (and their associated physical data) from one metadata server to another
- use the Copy and Paste Special functions to copy objects or groups of objects (and their associated physical data) from one folder to another

Use caution when renaming folders and when moving objects from one folder to another.

Renaming folders, or moving objects from one folder to another, can affect associations and references to objects that are contained in a folder or its subfolders, as follows:

- Some types of objects are referenced using folder pathnames. Associations to these types of objects can break if you rename a folder that is part of the object’s path or if you move the object to a different folder. For example:
  - Reports use folder paths to locate information maps. If the location of an information map changes, then you might need to edit associated reports to point to the new information map location.
  - Information maps and reports use folder paths to locate objects such as cubes, prompts, and stored processes. If the location of one of these objects changes, then you might need to edit associated reports and information maps to point to the new location. In SAS Information Map Studio, you can use the Resource Replacement feature to change the location for some types of objects.
  - Some prompts (including library prompts, data source prompts, and prompts that use dynamically generated data values) use folder paths to locate libraries, tables, and information maps. If the location of a



library, table, or information map changes, you might need to edit the affected prompts to point to the new location.

- If you move a table or a library, or if you rename a folder that is in the path for a library or table, most associations to the library or folder will continue to work. Client applications will still be able to locate the library or table. Jobs, information maps, and reports that use the library or table will continue to function. The only exceptions are some types of prompts, as described in the preceding paragraph.
- Renaming folders and moving objects can affect the following SAS Information Delivery Portal content:
  - information maps that are displayed in information map viewer portlets
  - links to detailed data (for example, reports or stored processes) that are coded in portlets
  - report references that have been published to WebDAV

Use caution when renaming objects in SAS Management Console.

If you rename an object that is associated with other objects, then associated objects can malfunction. For example, if you rename a library or table that is used by an information map, then you will need to use the Resource Replacement feature of SAS Information Map Studio to specify the new library or table name.

*Note:* For information about renaming objects and folders in client applications other than SAS Management Console, see the documentation for that application. △

If you need to move content from one folder location to another (for example, from a user's personal folder to a shared folder), then use either **Export SAS Package** and **Import SAS Package** or **Copy** and **Paste Special**. Content movement should be carefully planned and executed so that metadata associations are maintained.

Correct use of **Export SAS Package** and **Import SAS Package** or **Copy** and **Paste Special** will maintain associations among objects as well as relationships between metadata and physical data.

For example, if you use one of these methods to move an information map that is associated with a report, then the association will be maintained. If you move the information map using a different procedure, then the report will no longer work.

For instructions for using **Export SAS Package** and **Import SAS Package**, see Chapter 17, "Using the Promotion Tools," on page 213.

---

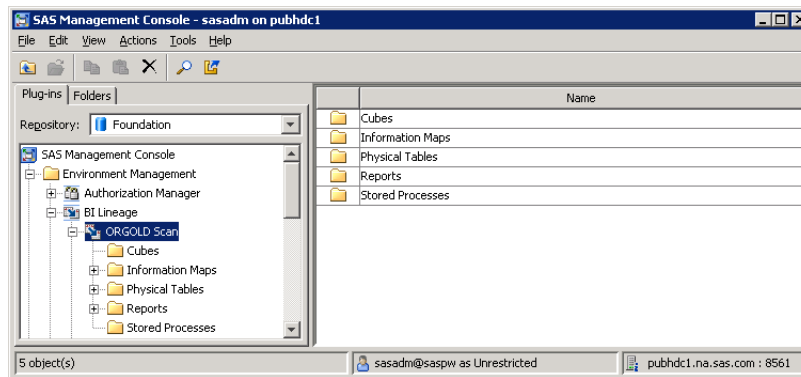
## Using the BI Lineage Plug-in

---

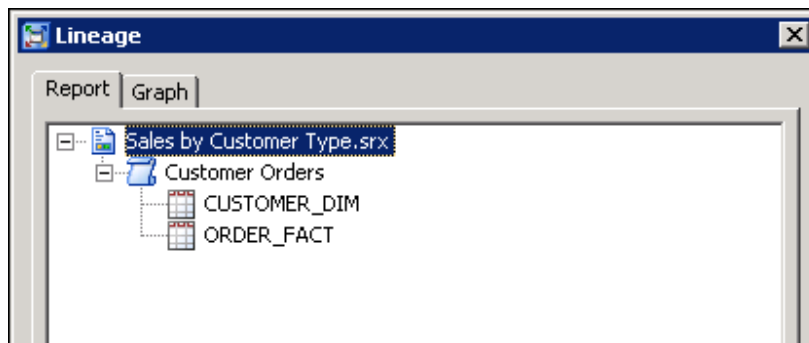
### About the BI Lineage Plug-in

The BI Lineage plug-in for SAS Management Console, which is new with the November 2010 release, identifies connections between BI objects. To generate lineage information, you use the BI Lineage plug-in to run a scan on a subset of folders in the SAS Folders tree. The scan examines reports and information maps that are stored in the selected folders, and identifies other objects (regardless of their location) that are connected to those reports and information maps. The starting point for the scan can be any folder (other than the root folder) in the SAS Folders tree.

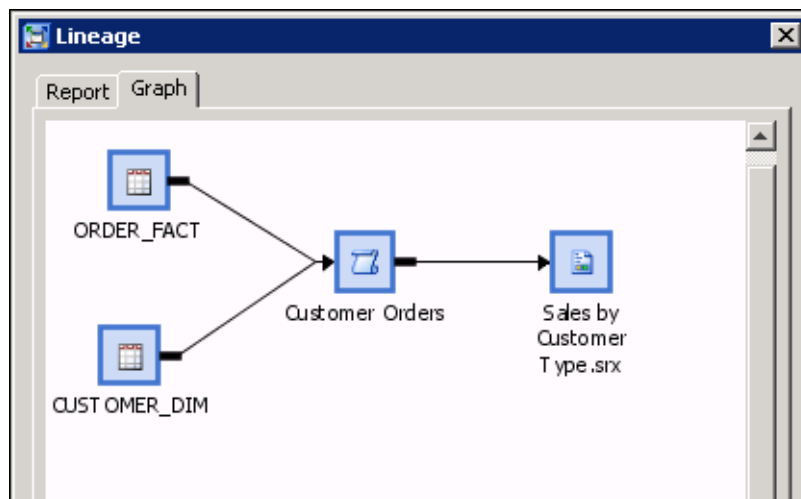
Scan results are stored in a tree structure that can be viewed on the Plug-ins tab, as shown in the following example:



Authorized users can navigate the tree for a particular scan, select a BI object that was included in the scan, and display the object's lineage or reverse lineage report. Here is an example of a lineage report for a report object, showing an information map and two tables that are connected to the report.



Information can also be displayed in graphical form, as shown here:



For details about information that is available in scan results, see "Viewing the Results of BI Lineage Scans" on page 206.

BI Lineage is similar to the impact analysis feature of SAS Data Integration Studio and SAS OLAP Cube Studio. The main difference is that BI Lineage analyzes objects starting with information maps and reports, and impact analysis starts with tables and cubes. If your deployment includes SAS Data Integration Studio or SAS OLAP Studio, then impact analysis data from that application is automatically incorporated into the BI Lineage scan results.

---

## Administering the BI Lineage Plug-in

### Running and Scheduling Scans

BI Lineage scans can be run only by an unrestricted administrative user (that is, a user who is in the **SAS Metadata Server: Unrestricted** role).

To run a scan, go the **Plug-ins** tab in SAS Management Console, right-click the **BI Lineage** node, and select **New Scan**. In the New Scan Wizard, specify a name for the scan. Then browse to a starting location, which can be any folder (other than the root folder) in the SAS Folders tree. The scan examines the reports and information maps that are stored in that folder and all of its subfolders, and identifies connected objects regardless of their location.

Because the lineage information is not generated in real time, it is important to keep the scan information updated. To make this task easier, you can create jobs and then schedule them to run at regular intervals. The plug-in can generate jobs for running, exporting, or deleting BI Lineage scans. After the jobs are generated, you can use the Schedule Manager plug-in to schedule the jobs. For details about these tasks, see the BI Lineage plug-in Help in SAS Management Console.

### Granting Users Permission to View Scan Results

Scan results are stored in a special metadata repository called the BILineage repository. After the November 2010 release or a later release has been installed, this repository is created automatically the first time an unrestricted administrative user logs on to SAS Management Console. The BILineage repository should not be used for any purpose other than storing scan results.

To give users permission to view scan results, you must update the BILineage repository's Default ACT to grant ReadMetadata permissions. Follow these steps to grant permission:

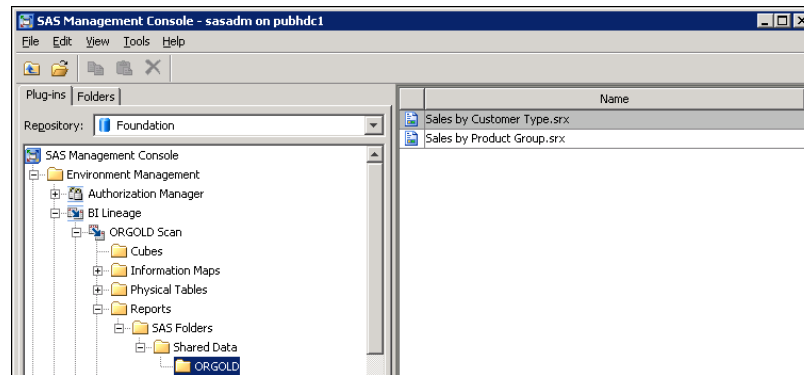
- 1 Log on to SAS Management Console as an unrestricted administrative user. On **Plug-ins** tab, select **BILineage** from the Repository drop-down list.
- 2 Navigate to **Environment Management ► Access Control Templates ► Default ACT**, and open the properties window for the Default ACT.
- 3 Select the **Permission Pattern** tab, and update the information as follows.
  - If you want to give all registered users access to scan results, then grant ReadMetadata permission to the SASUSERS group.
  - If you want to give access to a specific group of users, click **Add** and select the group. When you add the group, the Authorization Manager automatically grants it ReadMetadata permission.

*Note:* You cannot provide access by setting permissions on the BILineage folder that appears in the SAS Folders tree, because scan results are not stored in the folder. △

## Viewing the Results of BI Lineage Scans

To view scan results, expand the node for the BI Lineage plug-in, and then expand the node for the scan whose results you want to view.

Within the scan, expand the node for a particular object type (for example, Reports). Then select the folder that contains the object whose lineage information you want to view, as in the following example:



In the right pane, right-click the object whose lineage information you want to view, and select one of the following options:

### Lineage

identifies the objects that contribute to a particular report or information map (including objects that are located outside of the scanned folder and its subfolders).

For example:

- The lineage information for a report identifies all of the information maps, tables, cubes, columns, filters, and stored processes that contribute to the report.
- The lineage information for an information map identifies all of the tables, cubes, columns, filters, and stored processes that contribute to the information map.

### Reverse Lineage

identifies the objects within the scope of the scan that might be affected if a selected object were to change. (Affected objects that are stored outside of the scanned folder and its subfolders are not identified.)

For example, the reverse lineage information for a table shows information maps and reports that use the table (excluding those that are stored outside of the scanned folder and its subfolders). You can use this information to determine which information maps and reports might be affected if the table were to change.

### Properties

displays information about the object, as well as a list of the component objects that are members of the selected object (for example, the columns in a table, the data items in a cube, or the sections in a report).

Right-click any of the component objects to display a menu of options for that object. The options include the following:

#### High-level Lineage

identifies the high-level objects that contribute to a particular component object (such as a data item, column, or report section).

For example, high-level lineage for a report section shows all information maps and tables that contribute to the report section.

**High-level Reverse Lineage**

identifies the high-level objects within the scope of the scan that would be affected if the selected component object were to change. (Affected high-level objects that are stored outside of the scanned folder and its subfolders are not identified.)

For example, the high-level reverse lineage information for a data item shows all reports that use the data item (excluding reports that are stored outside of the scanned folder and its subfolders). You can use this information to determine which reports would be affected if the data item were to be changed.

**Low-level Lineage**

identifies other low-level member objects (such as data items, columns, and report sections) that contribute to a particular component object.

For example, low-level lineage information for a data item would show the column or columns that contribute to that data item.

**Low-level Reverse Lineage**

identifies other low-level objects within the scope of the scan that would be affected if a particular component object (such as a data item, column, or report section) were to change. (Affected objects that are stored outside of the scanned folder and its subfolders are not identified.)

For example, low-level reverse lineage for a data item shows report sections that would be affected if that data item were to change (excluding report sections that are stored outside of the scanned folder and its subfolders).

**Properties**

displays the general properties for the object.

You should be aware of the following limitations when using scans produced by the BI Lineage plug-in:

- Reverse lineage and low-level reverse lineage information only identifies objects that are stored within the scanned folder and its subfolders. Objects that are stored outside the scope of the scan are not identified. For example, reverse lineage for a table shows reports and information maps that might be affected if the table were to change, excluding reports and information maps that are stored outside of the scanned folder and its subfolders.

(By contrast, lineage and low level-lineage information includes all objects, regardless of their location in the SAS Folders tree. For example, the lineage information for a report includes all tables and information maps that contribute to the report, including those stored outside the scope of the scan.)

- If the data source for an information map is a table that was created by the SAS Information Maps LIBNAME engine, then only partial lineage information is provided for the map. The lineage information includes the table that the engine created. However, it does not include the original map that the engine accessed to create the table, nor does it include the table that was used to create the original map. Here is an example scenario:
  - Table A was used as a data source for Map B.
  - Map B was used as a data source by the Information Maps LIBNAME engine to create Table C.
  - Table C was used as a data source to create Map D.

For Map D, the BI Lineage scan would provide the following lineage information: **Map D ► Table C**. The lineage information would not include Map B or Table A.

*Note:* The SAS Information Maps LIBNAME engine provides read-only access to data in an information map. For more information, see *Base SAS Guide to Information Maps*. △

- Prompts that are used by information maps are not included in scans.
- Low-level lineages are not supported for sections of reports that directly use tables or cubes. Low-level lineages are supported only for report sections that are based on information maps.
- BI Dashboards and their component parts (such as ranges and indicators) are not included in scans.

For more details about using the BI Lineage plug-in, see the plug-in Help in SAS Management Console.

---

## Suggested Uses for the BI Lineage Plug-in

The BI Lineage plug-in is useful in the following types of situations:

- A report consumer who wants to know where the data on a report originated can use lineage information to identify the information maps, cubes, and tables that provided the source data, as well as any stored processes that processed the data. If your deployment includes SAS Data Integration Studio, the lineage information also includes any jobs and transformations that were used to process the data.
- If you modify a table, cube, or information map, you can use reverse lineage information to identify associated objects that might need to be retested or verified. For example, you can identify which information maps and reports (within the scope of the scan) use a particular table. In addition, you can use low-level reverse lineage information to identify which information maps, reports, or report sections (within the scope of the scan) use a particular column.
- For project documentation, you can use the BI Lineage plug-in to create a printable list of the objects in the project and their relationship to one other.
- BI Lineage can provide usage information to help set priorities and make support decisions. For example, you can find out how many cubes, information maps, or reports (within the scope of a scan) use a particular set of tables or a particular data column.

*Note:* The BI Lineage plug-in does not report the number of times that a particular object was accessed. This type of usage information can be obtained through tools such as the SAS Audit and Performance Measurement package (see <http://support.sas.com/rnd/emi>) and the Key User Action Log in SAS Web Report Studio (see “Configuring Logging in SAS Web Report Studio” in the *SAS Intelligence Platform: Web Application Administration Guide*). △

- When troubleshooting security issues, you can use lineage information to identify which objects’ permissions to examine. For example, if a user is having difficulty accessing a report, you can use lineage information to identify the information maps, tables, and cubes that the report depends on. To have full access to the report, the user must have the appropriate permission to the metadata and the physical data at each of these levels.

---

## Customizing Your Folder Structure

---

### Guidelines for Creating Custom Folders

When you install SAS, a default folder structure is provided. This structure segregates system information from business information, provides personal folders for

individual users, and provides an area for shared data. Within this overall structure, you can create a customized folder hierarchy that meets the information management, data sharing, and security requirements of your organization.

You can set up your folder structure under the **Shared Data** folder that is provided, or you can create additional folders under **SAS Folders** in which to store shared content.

The following are some guidelines for setting up a folder structure:

Keep the folder structure as simple as possible.

A simple folder structure makes your system easier to navigate and easier to administer. You should create new folders or levels of folders only if they provide a specific benefit for your organization.

Develop a folder structure that reflects the organization of your work.

Folders should be organized the way your work is organized. For example, your folder structure could reflect the following:

- your company's internal organization. For example, each division or department could have its own high-level folder.
- types of business activities. For example, you could have a separate folders for human resources, sales, research and development, and marketing.
- geography. For example, each country, sales region, or regional office could have its own folder.
- categories of products. For example, each product line or product group could have its own folder.
- time periods. For example, you could have a folder for each year, quarter, or month.
- categories of users. Generally, this type of folder structure is necessary only in large organizations that have a clear separation of responsibilities (for example, separate teams for data preparation, map creation, and report creation).
- change-control status. If you have just one deployment of the SAS Intelligence Platform (instead of separate deployments for development, test, and production), then you might want to use folders to separate production-status content from content that is in the development or testing stage. To do so, you can set up separate sets of folders for development, test, and production. Then, you can use the promotion tools to move content from development to test and from test to production. See "Example 3: Separate Folder Structures for Development, Test, and Production" on page 211.

Do not set up folders based on SAS client applications.

In SAS 9.2, it is not necessary or desirable to organize objects based on which SAS client applications were used to create them. Organizing folders on this basis can complicate administration tasks such as the assignment of permissions.

Do not set up folders based on object types unless it is necessary for access control.

Organizing folders based on object types can complicate administration tasks such as the assignment of permissions. As a general rule, you should avoid setting up folders on this basis.

Different types of objects such as libraries, information maps, reports, OLAP cubes, and stored processes can be stored together in the same folders without causing any problems. Because client applications such as SAS Information Map Studio and SAS Web Report Studio display only the object types that are used in that application, users will see only the object types that they need to see.

An additional advantage of mixing object types is that associations are easier to maintain. For example, if a folder contains objects that are associated with one another, you can rename the folder without breaking the associations.

However, in some large organizations you might need to use folders to restrict access to certain types of objects. For details, see the next guideline.

Develop a folder organization that reflects the access rules you want to enforce.

Folders enable you to easily restrict access to content. For example:

- If you want to prevent departments from accessing each other's content, then you can create a high-level folder for each department and apply different permissions to each of the folders.
- If you want to restrict access to sensitive content (for example, content related to a sensitive product line or a business activity such as human resources), then you can create a separate folder for that content and apply a restrictive access control template (ACT).
- If your organization requires a clear separation of content among different categories of SAS users, then you can create separate folders for each group. Generally, this type of folder structure is necessary only in large organizations that have separate teams of SAS users with different job responsibilities.

For example, suppose you have one group of users that works on data preparation tasks (such as creating libraries, tables, and cubes) and another group creates information maps, stored processes, and reports). To ensure that the groups do not interfere with one another's work, you can create a separate folder for each group and apply different permissions to each of the folders.

See “Permissions on Folders” in the *SAS Intelligence Platform: Security Administration Guide* for detailed information about creating Access Control Templates (ACTs) for folders and applying the appropriate permissions at each level.

If you have separate environments for development, test, and production, then use the same folder structure across environments.

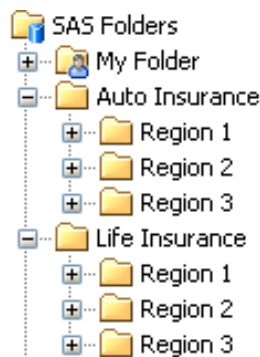
Using a uniform folder structure will make it easier to promote objects from one environment to another.

---

## Examples of Custom Folder Structures

### Example 1: Multiple Lines of Business Across Multiple Regions

Here is an example folder structure for an organization that has multiple lines of business across multiple regions, each with its own set of data definitions, information maps, stored processes, and reports. Each region and line of business has a single user or team of users that performs all data integration and business intelligence tasks.

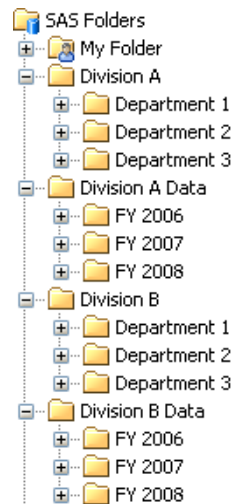




In this scenario, each low-level folder would have permissions that enable WriteMemberMetadata (WMM) access for the SAS users in that specific line of business and region.

## Example 2: Multiple Lines of Business with Separate Data Integration and Business Intelligence Users

Here is an example of a folder structure for an organization that has multiple divisions. Within each division, a specific group of SAS users performs data preparation tasks (including the creation of tables and cubes) and another group of users creates information maps, reports, and stored processes. The table and cube definitions are division-wide and are organized by time period. Separate information maps, reports, and stored processes are created for each department.

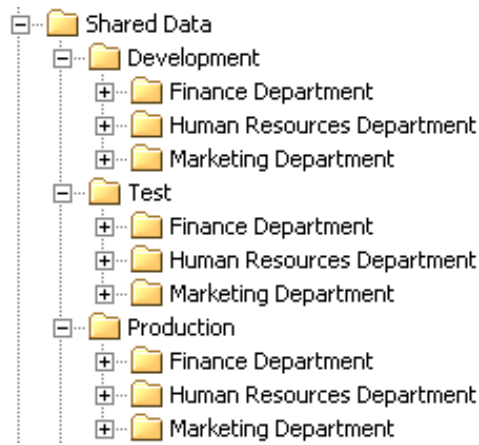


In this scenario, the following permissions would be granted for a division's data folder (for example, the **Division A Data** folder):

- only Read (R) and ReadMetadata (RM) access for users who create maps, reports, and stored processes. These would include users of SAS Enterprise Guide, SAS Information Map Studio, and SAS Web Report Studio.
- WriteMemberMetadata (WMM) access for the users who perform data integration tasks for the division (including users of SAS Data Integration Studio and SAS OLAP Cube Studio).

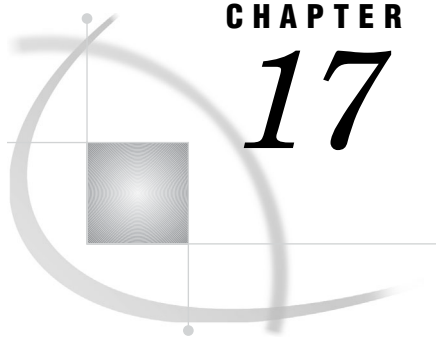
## Example 3: Separate Folder Structures for Development, Test, and Production

As an alternative to creating separate deployments of the SAS Intelligence Platform for development, test, and production, you can use SAS folders to create a change control environment within a single metadata server. The following example demonstrates how to set up folder structures for test and development that mirror your production folders:



In this scenario, you would give development and testing personnel the appropriate access to the **Development** and **Test** subfolders, and you would give business users the appropriate access to the **Production** subfolders.

You can then use the Export SAS Package and Import SAS Package wizards to move metadata and associated content from **Development** to **Test** and from **Test** to **Production**. For details, see Chapter 17, “Using the Promotion Tools,” on page 213. To facilitate promotion, use the same folder structure within each of the three folders.



## CHAPTER

## 17

# Using the Promotion Tools

<i>Overview of Promotion Tools</i>	<b>214</b>
<i>About the Promotion Tools</i>	<b>214</b>
<i>Overview of the Promotion Process</i>	<b>215</b>
<i>Selective Promotion</i>	<b>217</b>
<i>Promotion of Content Along With Metadata</i>	<b>218</b>
<i>Restoration of Metadata Associations</i>	<b>220</b>
<i>Optional Promotion of Access Controls</i>	<b>221</b>
<i>Special Considerations for Promoting Metadata</i>	<b>221</b>
<i>Special Considerations for Promoting Metadata From SAS 9.1.3 to SAS 9.2</i>	<b>223</b>
<i>About Promoting Metadata From SAS 9.1.3 to SAS 9.2</i>	<b>223</b>
<i>Types of Objects That Can Be Promoted from SAS 9.1.3 to SAS 9.2</i>	<b>224</b>
<i>Ensuring the Correct Software Version When Exporting from SAS 9.1.3</i>	<b>224</b>
<i>Moving SAS 9.1.3 Tables and Libraries to Folders</i>	<b>224</b>
<i>Special Considerations for Promoting Prompts from SAS 9.1.3</i>	<b>224</b>
<i>Promoting OLAP Cubes from SAS 9.1.3</i>	<b>224</b>
<i>Converting Data Explorations That Have Been Promoted from SAS 9.1.3</i>	<b>225</b>
<i>Promoting Portal Content from SAS 9.1.3</i>	<b>225</b>
<i>Using the Batch Import Tool to Import Metadata from SAS 9.1.3</i>	<b>225</b>
<i>Preparing to Use the Promotion Tools</i>	<b>226</b>
<i>Set Up Servers and Start Spawners</i>	<b>226</b>
<i>Ensure Appropriate User Credentials</i>	<b>226</b>
<i>Ensure Adequate Memory</i>	<b>226</b>
<i>Set Up Folders on the Target Metadata Server</i>	<b>227</b>
<i>For Reports, Set Up Content Mapping on the Target Metadata Server</i>	<b>227</b>
<i>Set Up Security for the Folders or Objects</i>	<b>227</b>
<i>Create Directories for Physical Files on the Target System</i>	<b>228</b>
<i>Create Directories and Move Physical Files for Cubes</i>	<b>228</b>
<i>Create Directories and Move Physical Files for Prompts</i>	<b>228</b>
<i>How to Access the Promotion Tools</i>	<b>229</b>
<i>Using the Export SAS Package and Import SAS Package Wizards: Example Usage Scenario</i>	<b>229</b>
<i>Description of Scenario</i>	<b>229</b>
<i>Prerequisite Actions</i>	<b>231</b>
<i>Step 1: Export the Folders and Objects</i>	<b>232</b>
<i>Step 2: Import the SPK File</i>	<b>235</b>
<i>Using the Batch Export and Import Tools</i>	<b>238</b>
<i>About the Batch Export and Import Tools</i>	<b>238</b>
<i>Preparing to Use the Batch Export Tool or the Batch Import Tool</i>	<b>239</b>
<i>Using the Batch Export Tool</i>	<b>239</b>
<i>Syntax for the Batch Export Tool</i>	<b>239</b>
<i>Syntax Description for the Batch Export Tool</i>	<b>240</b>
<i>Specifying Object Types</i>	<b>244</b>

<i>Specifying an Absolute Date (or an Absolute Date and Time)</i>	245
<i>Specifying a Relative Date</i>	245
<i>Specifying a Relative Time Period</i>	246
<i>Example: Batch Export Command</i>	246
<i>Using the Substitution Properties File</i>	247
<i>About the Substitution Properties File</i>	247
<i>Edit the Substitution Properties File</i>	247
<i>Apply the Substitution Properties File</i>	248
<i>Using the Batch Import Tool</i>	248
<i>Syntax for the Batch Import Tool</i>	248
<i>Batch Import Tool Syntax Description</i>	248
<i>Example: Batch Import Command</i>	251
<i>Viewing the Batch Export or Import Log</i>	252
<i>Using the Batch Import Tool to Import a Package That Was Created by the Export SAS Package Wizard</i>	252
<i>Using the Import SAS Package Wizard to Import a Package that Was Created by the Batch Export Tool</i>	253
<i>Using the Batch Import Tool to Import a Package that Was Created in SAS 9.1.3</i>	253
<i>Troubleshooting the Promotion Tools</i>	253

---

## Overview of Promotion Tools

---

### About the Promotion Tools

*Promotion* is the process of copying selected metadata and associated content within or between planned deployments of SAS software. This process is repeatable for a particular deployment.

The SAS Intelligence Platform provides tools that enable you to promote individual metadata objects or groups of objects from one metadata server to another, or from one location to another on the same metadata server. You can also promote the physical files that are associated with the metadata.

The promotion tools include the following:

- the Export SAS Package and Import SAS Package wizards, which are available in the **Folders** tab of SAS Management Console. The same wizards are also available in SAS Data Integration Studio and SAS OLAP Cube Studio.
- the batch import tool and the batch export tool, which enable you to perform promotions on a scheduled or repeatable basis. These tools provide most of the same capabilities as the Export SAS Package and Import SAS Package wizards.

The promotion tools support the following types of scenarios:

- promotion of objects from one location in the SAS Folders tree to another location in the same tree. For example, you might want to promote newly developed (or newly modified) objects from a user's home folder to a location under the **Shared Data** folder.
- promotion of objects from a test environment to a production environment. For example, you might want to promote newly developed (or newly modified) objects from a metadata server that is installed on a test machine to a metadata server that is installed on a production machine.
- importing objects that were exported from a SAS 9.1.3 environment. This feature is sometimes used to support migration from SAS 9.1.3 to SAS 9.2. It is also

useful for sites that are running two SAS releases in parallel. Before exporting objects from the SAS 9.1.3 environment, make that sure you have the correct software version (see “Special Considerations for Promoting Metadata From SAS 9.1.3 to SAS 9.2” on page 223).

*Note:* *Migration* is the process of moving data to a different server or storage device, converting the data for use in a different operating environment, or both. Migration can include installing newer software and converting applications for use with that software. △

- creating a backup of specific folders and objects. In this case, you would use the Export SAS Package wizard or the batch export tool to create a package file. If you later need to restore the folders or objects, you would use the Import SAS Package wizard or the batch import tool to restore the contents of the package file to the metadata server.

The following table shows the types of objects that you can promote:

*Note:*

- You might be able to export additional object types, depending on which applications you have installed.
- In SAS Data Integration Studio and SAS OLAP Cube Studio, you can import and export only the objects that pertain to the application.
- To promote portal pages, page templates, portlet instances, and the portal’s application and user permissions trees, see “Promoting Portal Content” in the *SAS Intelligence Platform: Web Application Administration Guide*.

△

**Table 17.1** Types of Objects That Can Be Promoted

Actions	Documents	Prompts
Applications	Event subscribers	Prompt groups
Burst definitions	Event subscriber groups	Queue managers
Channels	External files	Reports
Conditions	Folders	Report components
Condition action sets	Generated transformations	Report images
Content subscribers	Information maps	SAS Add-In for Microsoft Office documents
Content subscriber groups	Jobs	SAS Enterprise Guide projects
Cubes	Libraries	Stored processes
Dashboard objects*	Message queues	Tables
Data explorations	Mining results	Web services
Deployed flows	Notes	
Deployed jobs	OLAP schemas	

\* Dashboard objects (including dashboards, indicators, indicator data, ranges, indicator configurations, and dashboard components) can be promoted effective with the November 2010 release. However, these objects cannot be promoted within the same metadata server. For additional special considerations, see “Promoting SAS BI Dashboard 4.3 Content” in the *SAS Intelligence Platform: Web Application Administration Guide*.

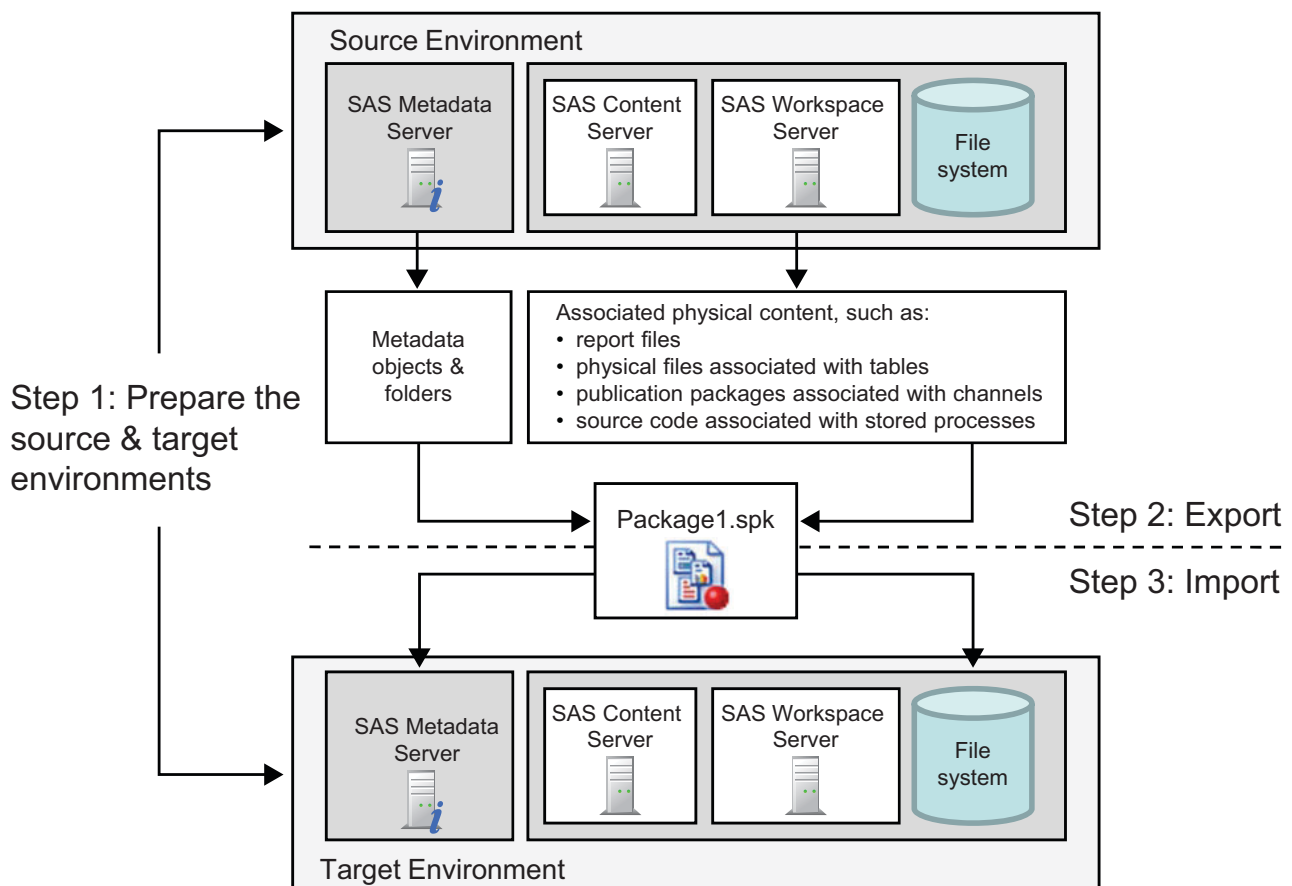
---

## Overview of the Promotion Process

Promotion is a three-step process:

- First, you must prepare the source and target environments. The preparation steps include ensuring that the required servers are configured and running. If you are promoting metadata from one metadata server to another, you must prepare the target environment by creating folders for the metadata, creating directories for the physical files, and setting up security. For details, see “Preparing to Use the Promotion Tools” on page 226.
- Second, you use the Export SAS Package wizard (or the batch export tool) to export the selected objects or folders from the source metadata server. The wizard or the batch tool places the exported objects in a SAS package (SPK) file. If you want the SPK file to be available remotely, then you can direct the package to a shared location within your file system. You can also e-mail the package after it has been created.
- Third, you use the Import SAS Package wizard (or the batch import tool) to import the package file. You can import the package to another metadata server, which can be located either on the same host machine or a different host machine; or you can import it to a different location in the SAS Folders tree on the source metadata server.

**Figure 17.1** Three-step Promotion Process



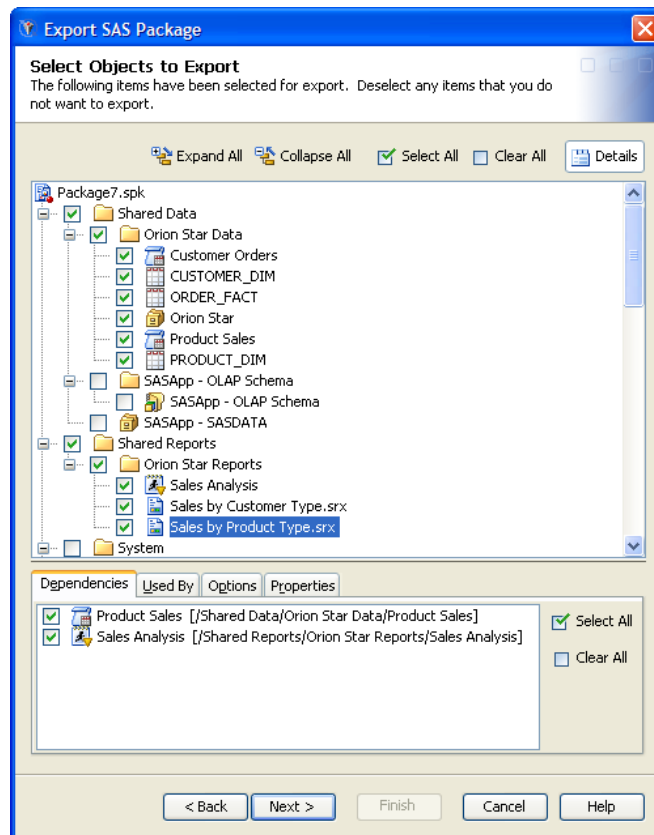
The package format is the same regardless of whether the package is created in SAS Management Console, in SAS Data Integration Studio, in SAS OLAP Cube Studio, or by the batch export tool. In addition, the package format is the same regardless of the host machine's operating system. Therefore, packages can be interchanged as follows:

- A package that is exported by using SAS Management Console can be imported by using SAS Data Integration Studio or SAS OLAP Cube Studio, and vice versa. However, SAS Data Integration Studio and SAS OLAP Cube Studio can import and export only the objects that pertain to the application, whereas SAS Management Console can import and export all object types that are listed in Table 17.1 on page 215.
- A package that is exported by using the Export SAS Package wizard can be imported by using the batch import tool.
- A package that is exported by using the batch export tool can be imported by using the Import SAS Package wizard.
- You can export a package from a Windows system and import it to a UNIX system, and vice versa.

## Selective Promotion

The Export SAS Package and the Import SAS Package wizards (and the batch export and import tools) provide the following options for specifying which objects are to be exported and imported:

- You can select multiple nested folders.
- You can include all of the objects that are contained in a folder or just selected objects, as shown in the following example:



- You can include or exclude dependent objects.

In the Export SAS Package wizard, as shown in the preceding example, you can highlight one or more objects and view their dependencies. You can then select which dependent objects you want to include in the export.

In the batch export tool, you can specify that all dependent objects are to be identified recursively and included in the export.

- You can use the Filter feature, as shown here, to select objects based on the object name, the object type, or the time period during which the object was created or last modified. In addition, you can select **Include empty folders** if you want to export folders that do not contain any subfolders or objects. (If you do not select this option, then empty folders are not exported.)

**Filter**

Select filter criteria: [Reset to defaults](#)

**Name**

All or part of a name:

☐ Include description

**Types**

☒ Action

☒ Application

☒ Burst Definition

☒ Channel

☒ Condition

☒ Condition action set

☒ Content subscriber

☐ Include empty folders

☒ Select All

☐ Clear All

**Dates**

☐ Created ☒ Last modified

Range type:

From:

To:

OK Cancel Help

- 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 target folder location).

---

## Promotion of Content Along With Metadata

For most object types, the Export SAS Package and Import SAS Package wizards (and the batch export and import tools) promote both the content and the metadata. When you use the batch tools, the promotion automatically includes all associated content except tables and external files. When you use the wizards, you can choose whether to include or exclude some of the content types.



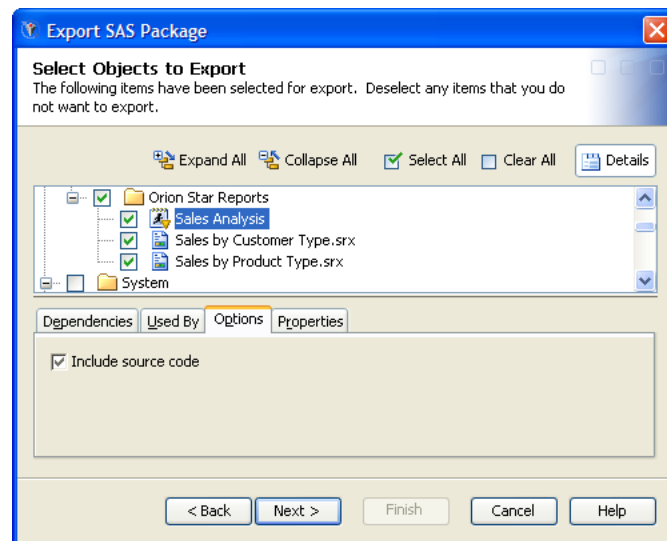
The following table lists the types of content that are exported and imported with metadata objects.

**Table 17.2** Promotion of Content with Metadata Objects

Object	Associated Content	Promotion Details
Channel	Packages published to the channel's archive or WebDAV location	Associated content is always promoted with batch tools and is optionally promoted with wizards.
Cube	Cube header files, MOLAP aggregation tables, drill-through tables, and ROLAP aggregation tables	Associated content is not promoted. For details about moving these files, see the <i>SAS OLAP Server: User's Guide</i> .
External file	Physical file	Associated content is not promoted with batch tools and is optionally promoted with wizards.
	External format file (EFF)	Associated content is not promoted with batch tools and is optionally promoted with wizards.
Job	Custom code, preprocesses, and postprocesses.	Associated content is always promoted.
Library	XML files and XML maps that are contained in the library	Associated content is always promoted with batch tools and is optionally promoted with wizards.
Prompt	Physical files that are associated with <b>File or directory</b> prompts	Associated content is not promoted. For details, see "Create Directories and Move Physical Files for Prompts" on page 228.
Report	XML (.srx) file that defines the report	Associated content is always promoted. The target is the path on the SAS Content Server or file system that is mapped to the target metadata folder via content mapping.

Object	Associated Content	Promotion Details
Stored process	Source code (.sas) file	<p>Associated content is optionally promoted with the wizards and is always promoted with the batch tools. The target is the source code repository that you designate in the wizard dialog box. If the repository definition does not exist on the target system, the Import SAS Package wizard allows you to add it.</p> <p>When you use the batch import tool, you can use the substitution file to designate the source code repository and to specify that the repository definition is to be created if it does not exist.</p>
Table	Physical table	<p>Associated content is not promoted with the batch tools and is optionally promoted with the wizards.</p> <p>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.</p>

If the promotion of an object's associated content is optional, information about the associated content is displayed in the wizard's **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 a table or an external file, the option to include the content is not selected by default.

## Restoration of Metadata Associations

Depending on the types of objects that you are importing, you might need to establish associations with the following objects and entities on the target system:

- a SAS Application Server
- a SAS Content Server
- base path on a SAS Content Server
- libraries

- tables
- external files
- physical locations for external files or for libraries
- physical locations for custom code that is associated with jobs, if the code is stored outside of the metadata
- OLAP schemas
- mining results
- a source code repository (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.

The Import SAS Package wizard displays prompts that assist you in associating imported objects with the appropriate objects and entities in the target environment. If you are using the batch tools, then you can specify the associations in the substitution properties file (see “Using the Substitution Properties File” on page 247).

---

## 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 using the Export SAS Package and Import SAS Package wizards and the batch export and import tools:

- The promotion tools 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 target metadata server.
- Access control entries and ACT associations that are inherited from a folder are not promoted. For inheritance to take effect in the target metadata server, 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 target metadata server.

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 target metadata server, 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.

---

## Special Considerations for Promoting Metadata

Before choosing to use the Export SAS Package and Import SAS Package wizards (or the batch import and export tools) to promote metadata, you should be aware of the following important considerations:

- The wizards and batch tools can promote only objects that appear in the SAS Folders tree. The wizards cannot promote metadata for users, roles, groups, servers, spawners, or Foundation Services configurations.

In addition, the wizards and batch tools cannot promote portal pages, page templates, portlet instances, or the portal’s application and user permissions trees. To promote these items, you can use the content promotion tool for the SAS

Information Delivery Portal 4.3. See “Promoting Portal Content” in the *SAS Intelligence Platform: Web Application Administration Guide*.

- 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 and batch tools do not export or import the servers or the physical directories (including source code repositories and deployment directories) that are associated with objects. Instead, the tools allow you to map objects to existing servers and directories in the target environment, as needed.
- The following special considerations apply to promoting libraries and tables:
  - Library names must be unique within a SAS Application Server. When you import a library, the promotion tools check to determine whether the target environment contains a library that has the same name and is located on the same application server. If such a library exists, then the promotion tool renames the imported library to **Copy of library-name**.
  - Table names must be unique within a library. When you import a table, the promotion tools check to determine whether the target environment contains a table that has the same name and is part of the same library. If such a table exists, then the promotion tool renames the imported table to **Copy of table-name**.
  - The promotion tools do not create the physical directory paths that are needed for libraries. Instead, the tools determine whether the physical directory exists, and if it does exist, they create the metadata for the directory.
  - 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).

*Note:* When you use the SAS Package Export wizard to export table objects, you can choose whether to also export the physical tables. When you use the export batch tool to export table objects, the physical tables are not exported. △

- If you import metadata for a table or an external file that is already registered in the target environment, then any matching column metadata is overwritten. Any non-matching metadata in the target table or file is preserved. The matching of column names is case-insensitive. For example, metadata for a column called **SALES** will overwrite metadata for a column called **Sales**.
- An environment can contain only one OLAP schema for each SAS Application Server. If an application server has an OLAP schema and you try to import another one for the same server, the Import SAS Package wizard prompts you to either deselect the schema or select a different application server.
- The following special considerations apply to promoting OLAP cubes:
  - OLAP cubes must be promoted along with their associated jobs. The Export SAS Package wizard will not allow you to continue if you do not select both objects.
  - After you promote the metadata and jobs for an OLAP cube, you must rebuild the cube.

For more information about promoting cubes, see the *SAS OLAP Server: User's Guide*.

- If you export a queue manager for WebSphere object or a queue manager for MSMQ object, then all of the message queues that are defined for that object are also exported, regardless of whether they were selected. When you import the package, the message queues might be added to your metadata. If you do not want to use the message queues, you should delete them when the import is complete.
- If you promote an object (for example, a stored process or a shared prompt) that includes a **File or directory** prompt, then you must do the following:
  - After the promotion is complete, you must edit the prompt to ensure that a valid workspace server is specified.
  - If the **File or directory** prompt has a default file or directory value specified, then you must move the associated physical file or directory to the target environment, if it is not already present. After the promotion, you must edit the prompt to associate it with the correct file or directory.
- After you promote a SAS Enterprise Guide project or a SAS Add-In for Microsoft Office document, you might need to update the application server names for tables that are associated with the project or document. This step is necessary if the target environment uses different application server names than the source environment. After you perform the promotion, use one of the following methods in the target environment to adjust the metadata:
  - Run the migration wizard that is provided with SAS Enterprise Guide and SAS Add-In for Microsoft Office. For details, see “Using the Migration Wizard” in *Administering SAS Enterprise Guide*; or see “Migrating SAS Content” in the SAS Add-In for Microsoft Office Help.
  - For SAS Enterprise Guide projects, use the Project Maintenance dialog box to edit the table metadata. For details, see the SAS Enterprise Guide Help.
- Dashboard objects (including dashboards, indicators, indicator data, ranges, indicator configurations, and dashboard components) can be promoted effective with the November 2010 release. However, these objects cannot be promoted within the same metadata server. For additional special considerations, see “Promoting SAS BI Dashboard 4.3 Content” in the *SAS Intelligence Platform: Web Application Administration Guide*.

Additional considerations apply if you are promoting metadata from a SAS 9.1.3 system to a SAS 9.2 system. For details, see the next topic.

---

## Special Considerations for Promoting Metadata From SAS 9.1.3 to SAS 9.2

---

### About Promoting Metadata From SAS 9.1.3 to SAS 9.2

You can export a package from a SAS 9.1.3 metadata server and import it to a SAS 9.2 metadata server. This feature is useful in the following situations:

- You are upgrading from SAS 9.1.3 to SAS 9.2, but you chose not to use the migration utility that is provided in the SAS Deployment Wizard. (See “Overview of Migrating Content to SAS 9.2” in the *SAS Intelligence Platform: 9.1.3 to 9.2 Migration Guide*.)
- You are running SAS 9.1.3 and SAS 9.2 in parallel, and you want to update your SAS 9.2 system with new content that you created in SAS 9.1.3.

The following topics outline the special considerations for promoting metadata from SAS 9.1.3 to SAS 9.2.

---

## Types of Objects That Can Be Promoted from SAS 9.1.3 to SAS 9.2

In SAS 9.1.3, only the following types of objects can be exported: data explorations, deployed jobs, documents, external files, folders, generated transformations, information maps, jobs, jobs flows, libraries, mining results, notes, reports, stored processes, and tables. (In SAS 9.2, a larger number of object types can be imported and exported.)

---

## Ensuring the Correct Software Version When Exporting from SAS 9.1.3

If you use BI Manager in SAS Management Console to perform the export, then the SAS 9.1.3 environment must have SAS Foundation Services 1.4. To check for this version:

- On Windows, go to **Settings ► Control Panel ► Add or Remove Programs** and find the entry for SAS Foundation Services 1.4.
- On UNIX machines, go the home directory of the user who maintains the SAS installation. Enter the following command:

```
more vpd.properties | grep Foundation
```

The following string should be present in the output that is displayed:

```
SAS Foundation Services 1.4
```

If you use SAS Data Integration Studio to perform the export, then you must have SAS Data Integration Studio 3.4. To check the version, click **Help ► About SAS Data Integration Studio**, and make sure that **Version=3.4** is displayed in the window.

---

## Moving SAS 9.1.3 Tables and Libraries to Folders

If you use SAS Management Console to export tables and libraries from a SAS 9.1.3 system, the tables and libraries must be in folders. To move them to folders, right-click the **BI Manager** node and select **Collect Ungrouped Library Metadata**. BI Manager places pointers to the libraries and tables in a folder called **Shared Data**, from which you can perform the export.

---

## Special Considerations for Promoting Prompts from SAS 9.1.3

When you import a prompt from a SAS 9.1.3 system, the following special considerations apply:

- The default values for multiple-value prompts are saved in reverse order.
- If the prompt's static (custom) list of values contains a row that is invalid in SAS 9.2 (for example, if a row contains blank or null unformatted values), then the row is deleted.
- If the prompt's static (custom) list of values contains duplicate formatted values, then each formatted value is appended with its corresponding unformatted value, which is enclosed in square brackets ([]).

---

## Promoting OLAP Cubes from SAS 9.1.3

To promote OLAP cubes from SAS 9.1.3 to SAS 9.2, use the ExportCubes and ImportCubes batch tools. These tools are included in the second maintenance release

for SAS 9.2. For detailed instructions, see “Importing and Exporting SAS OLAP Cubes” in the *SAS OLAP Server: User’s Guide*.

If you have not installed the second maintenance release for SAS 9.2, then follow the instructions in SAS Usage Note 36842 to promote your OLAP cubes.

---

## Converting Data Explorations That Have Been Promoted from SAS 9.1.3

In the November 2010 release of SAS 9.2, SAS Web OLAP Viewer is discontinued. After you have promoted your data explorations from SAS 9.1.3 to SAS 9.2, you can convert the bookmarks in your data explorations to SAS reports. The reports can then be viewed and edited using SAS Web Report Studio 4.3.

To ensure that the data explorations can be converted successfully, do the following when you promote them:

- Import the information maps on which the data explorations depend to the same path in which they were located in the SAS 9.1.3 environment.

Or, if you choose to import the information maps to a different location, import the information maps first and then import the data explorations in a separate package. When you import the data explorations, make sure to specify the correct location for the information maps.

- Rebuild any OLAP cubes that the data explorations depend on.

To perform the conversion, follow the manual steps that are described in “Manually Convert Data Explorations to Reports” in the *SAS Intelligence Platform: 9.1.3 to 9.2 Migration Guide*.

---

## Promoting Portal Content from SAS 9.1.3

To promote portal pages, page templates, portlet instances, and the portal’s application and user permissions trees from SAS 9.1.3, use the content promotion tool for the SAS Information Delivery Portal 4.3. See “Promoting Portal Content” in the *SAS Intelligence Platform: Web Application Administration Guide*.

---

## Using the Batch Import Tool to Import Metadata from SAS 9.1.3

You can use the batch import and export tools only with a SAS 9.2 metadata server. You can use the batch import tool to import a package that was created in SAS 9.1.3. For details, see “Using the Batch Import Tool to Import a Package that Was Created in SAS 9.1.3” on page 253.

### **CAUTION:**

**Use of the batch import tool to import SAS 9.1.3 packages can result in a significant number of broken metadata associations, such as tables with no associated library, jobs with missing tables, transformations within jobs with missing mappings, and expressions within mappings with missing columns.** As a best practice, you should use the batch import tool to import a SAS 9.1.3 package only if the resources associated with the imported objects have the same values in the two environments. Otherwise, it is strongly recommended that you use the Import SAS Package wizard in either SAS Management Console or SAS Data Integration Studio. △

---

## Preparing to Use the Promotion Tools

---

### Set Up Servers and Start Spawners

Before using the Export SAS Package and Import SAS Package wizards (or the batch import and export tools), make sure that the appropriate servers are set up and running, as follows:

- The object spawner must be running in both the source environment and the target environment.
- If you are promoting physical content that is associated with stored processes, tables, jobs, libraries, or external files, then a workspace server must exist in both the source environment and the target environment.
- If your stored processes create packages that are stored permanently on the SAS Content Server, then a SAS Content Server must exist and must be running in both the source environment and the target environment.
- If you are promoting reports and want to store report content on a SAS Content Server, then a SAS Content Server must exist and must be running in both the source environment and the target environment.

---

### Ensure Appropriate User Credentials

Make sure that the user who launches the wizard (or whose credentials are specified in the batch export or import command) meets the following criteria:

- The user must have ReadMetadata access to the objects that are being exported and WriteMetadata access to the objects that are being imported.
- The user must have WriteMemberMetadata access to the folders where objects are imported.
- The user must have the appropriate access to associated content (for example, tables and source code files) that is being exported or imported.
- To export or import physical content that is associated with stored processes, tables, jobs, libraries, or external files, the user must have the necessary credentials to access the workspace server. For information about these credentials, see “Requirements for Accessing a Standard Workspace Server” on page 37.

---

### Ensure Adequate Memory

If you are planning to export or import a large number of sizable objects, such as information maps or SAS Data Integration Studio jobs, then you might need to increase the amount of memory that is allocated to the import or export process. Follow these steps:

- 1 Open the appropriate INI file in a text editor, as follows:
  - If you are using SAS Management Console, open `sasmc.ini`, which is located in `SAS-installation-directory\SASManagementConsole\9.2`.
  - If you are using SAS Data Integration Studio, open `distudio.ini`, which is located in `SAS-installation-directory\SASDataIntegrationStudio\4.2`.



- If you are using SAS OLAP Cube Studio, open `olapa.ini`, which is located in `SAS-installation-directory\SASOLAPCubeStudio\4.2`
  - If you are using the batch import or export tool, then open `ImportPackage.ini` or `ExportPackage.ini`. These files are located in `SAS-installation-directory\SASPlatformObjectFramework\9.2`.
- 2 Locate the following line:  
`JavaArgs_1=-Xmx512m`
  - 3 Replace **512** with a higher value (for example, **1024**), and save your changes.  
*Note:* If your desktop applications run in the Citrix application delivery environment, then you can increase the size of your Java heap to the Citrix limit. While some versions of Citrix support higher limits, all versions support a maximum Java heap size of 512 megabytes. △

As an alternative to increasing memory, you can perform multiple imports or exports that include a smaller number of objects instead of importing or exporting all of your objects at the same time.

---

## Set Up Folders on the Target Metadata Server

If you will be using folders to control access to metadata, then you might want to set up folders in the SAS Folders tree on the target metadata server before you import your metadata. When you use the wizards, you should import objects at the same level in the folder hierarchy that you exported them from.

---

## For Reports, Set Up Content Mapping on the Target Metadata Server

The SAS Deployment Wizard automatically sets up content mapping on the metadata server. Content mapping establishes the location of the XML (.srx) files that define the reports. The location can be either a path on a SAS Content Server or a path on the file system.

If you are importing reports, then you should check the target metadata server to make sure that content mapping has been set up properly. Generally, content mapping is specified in the root folder (**SAS Folders**). To check the content mapping:

- 1 Right-click **SAS Folders** (or a specific folder, if applicable) and click **Properties**.
- 2 Open the **Content Mapping** tab, and make any necessary changes. If you specify a location on the SAS Content Server, you must enter the user ID and password for the SAS Trusted User (sastrust) 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 import the objects, select the **Include access controls** option in the Import SAS Package wizard (or specify **-includeACL** in the batch import command). 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 Import SAS Package wizard or specify **-includeACL** in the batch import command.

This option involves fewer steps, but it does not provide the opportunity to test your security settings before promotion.

---

## Create Directories for Physical Files on the Target System

On the target system, set up directories for the physical content that you plan to promote with the metadata. For example, you might need to create the following:

- directories for physical locations of libraries
- directories for physical files and external format files (EFFs) that are associated with external files
- directories for custom code that is associated with jobs, if the code is stored outside of the metadata
- directories for source code repositories for stored processes

The user who will execute the import should have Write access to these directories.

---

## Create Directories and Move Physical Files for Cubes

If you are promoting cubes, then you must create directories on the target system for the associated physical files (including cube header files, MOLAP aggregation tables, drill-through tables, and ROLAP aggregation tables). Then you must copy the files to the target directories. For important details about these and other tasks that are associated with promoting cubes, see the *SAS OLAP Server: User's Guide*.

---

## Create Directories and Move Physical Files for Prompts

If you promote an object (for example, a stored process or a shared prompt) that includes a **File or directory** prompt, then you must do the following:

- After the promotion is complete, you must edit the prompt to ensure that a valid workspace server is specified.

- If the **File or directory** prompt has a default file or directory value specified, then you must move the associated physical file or directory to the target environment, if it is not already present. After the promotion, you must edit the prompt to associate it with the correct file or directory.

---

## How to Access the Promotion Tools

To access the Export SAS Package and Import SAS Package wizards:

- In SAS Management Console, open the **Folders** tab. Then right-click any folder or object, and select **Export SAS Package** or **Import SAS Package**.
- In SAS Data Integration Studio or SAS OLAP Cube Studio, right-click any folder or object. Then Select **Export SAS Package** or **Import SAS Package**.

For instructions on using the wizards, see the following:

- the product Help for SAS Management Console, SAS Data Integration Studio, or SAS OLAP Cube Studio
- “Using the Export SAS Package and Import SAS Package Wizards: Example Usage Scenario” on page 229

To access the batch import tool (**ImportPackage**) and the batch export tool (**ExportPackage**), go to the following directory:

*SAS-installation-directory*\**SASPlatformObjectFramework\9.2**

For instructions on using the batch tools, see “Using the Batch Export and Import Tools” on page 238.

---

## Using the Export SAS Package and Import SAS Package Wizards: Example Usage Scenario

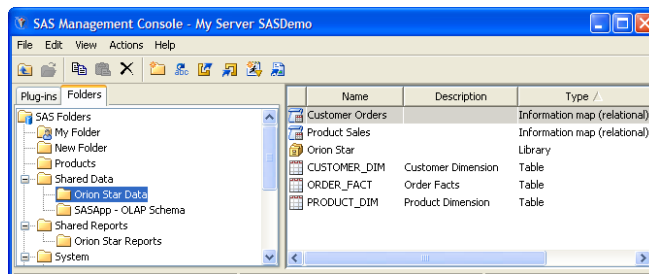
---

### Description of Scenario

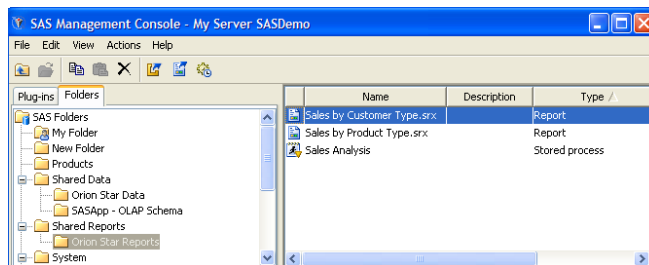
In this example, a set of objects has been developed in Environment 1 and will be promoted to a different metadata server in Environment 2. The objects to be promoted are as follows:

- 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 the library, the tables, and the information maps is stored in the location **SAS Folders\Shared Data\Orion Star Data**, as shown here:



The metadata for the reports and the stored process is stored in the location **SAS Folders\Shared Reports\Orion Star Reports**, as shown here:



*Note:* This example scenario uses SAS Management Console to perform the promotion. If you use SAS Data Integration Studio or SAS OLAP Cube Studio, the procedure is the same except where noted. △

Before promoting these objects, it is important to understand their associations with one another, as well as their associations with other objects and entities that do not appear in the tree. The following table describes these associations. It also describes the preparatory actions that must be taken in the target environment to ensure that the associations are restored following the promotion.

**Table 17.3** 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
Orion Star Reports 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 SAS Content Server)	Ensure that a SAS Content Server exists.
	Customer Orders information map	None
Sales by Product Group report	Sales by Product Group.srx file	Ensure that a SAS Content Server exists.
	Product Sales information map	None
	Sales Analysis stored process	None

Metadata Object or Folder Name	Associated Objects or Content in the Source Environment	Action Needed to Prepare the Target Environment
Orion Star Data folder	Data_ A ACT	Create the Data_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.
	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 **Orion Star Reports** folder and the **Orion Star Data** folder are each associated with an ACT. If the export and import operations include access controls, then the Import SAS Package wizard will expect to find the ACTs already present in the target metadata server.
- Each of the reports is associated with an XML file (with the suffix .srx) on the SAS Content Server. For these to be promoted, the Import SAS Package wizard will expect a SAS Content 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, the Import SAS Package wizard 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. The Import SAS Package wizard 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. If the source code repository definition does not exist, the wizard will enable you to add it.

## Prerequisite Actions

In the example steps that follow, it is assumed that the following preparatory actions have already been completed:

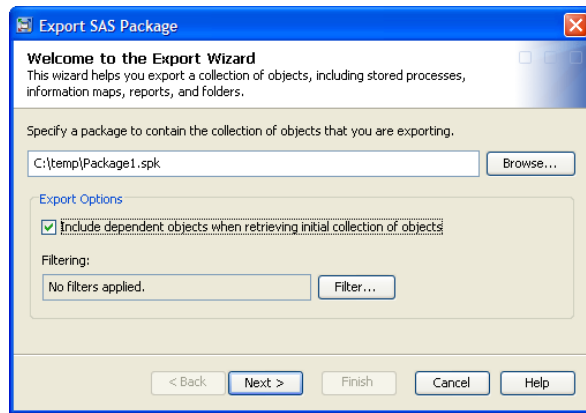
- The following physical items must be created (or already present) in the target environment:
  - a SAS Content Server
  - a file path for the source code file for the stored process
  - physical tables that correspond to the tables in the source environment (or file paths for these tables, if they are being exported along with the metadata)
- The following metadata must be created (or already present) on the target metadata server:
  - ACTs that correspond to, and have the same names as, the ACTs on the source metadata server
  - user group definitions for the groups that the ACTs refer to
  - a server definition for the SAS Content server
  - an application server definition that includes a stored process server and a workspace server

---

## Step 1: Export the Folders and Objects

For the scenario that is described in the previous topics, you would 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 server (that is, the metadata server from which you want to promote the folders and objects).
- 2 Select the **Folders** tab. (This step is not necessary in SAS Data Integration Studio or SAS OLAP Cube Studio.)
- 3 Right-click the **SAS Folders** node, and click **Export SAS Package**. (In SAS Data Integration Studio or SAS OLAP Cube Studio, you would right-click the folder that contains the objects that are to be exported.)
- 4 On the first Export SAS Package 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 dependent objects when retrieving initial collection of objects**, as shown in the following example. This option helps ensure that all objects on which the exported objects depend are included in the export operation.

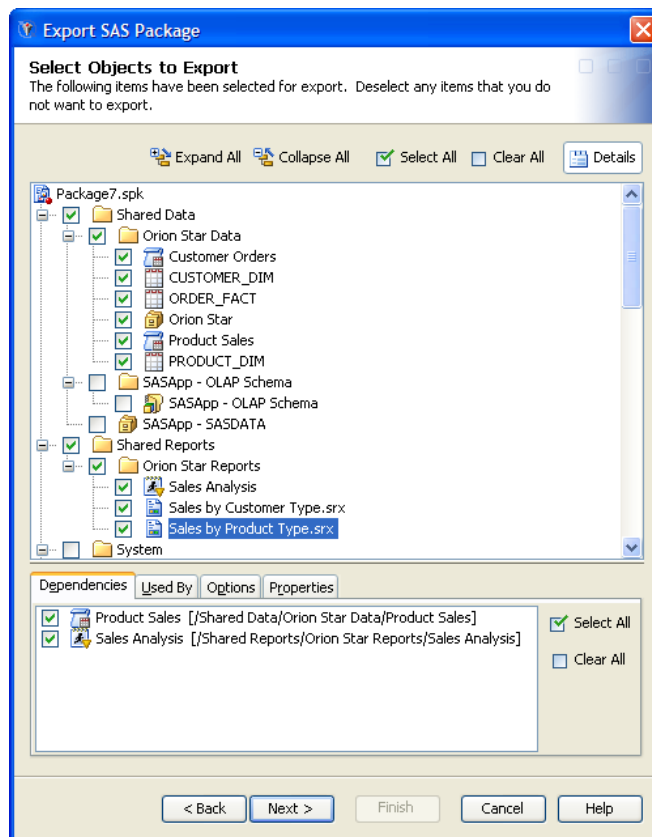


*Note:* On this page, you can also click **Filter** to select objects based on the object name, the object type, or the time period during which the object was created or last modified. (This scenario does not use the filter feature.) △

d Click **Next**.

5 On the next Export SAS Package page, do the following:

- a Select or clear the appropriate check boxes 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.

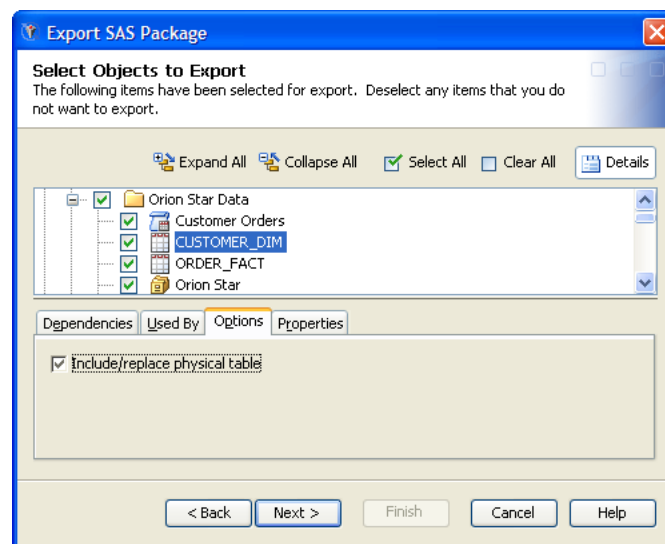
*Note:*

- If you press the CTRL key and highlight multiple objects, the **Dependencies** tab displays dependencies for all of the highlighted objects.
- Dependencies are identified and displayed only for the objects that are initially selected on this wizard page. If you click **Next** to go to the next wizard page, and then you later return to this page and select additional objects, the wizard does not identify the dependencies for the additional 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:*

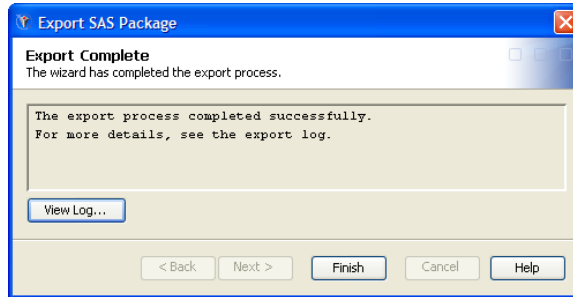
- If you choose the option to export physical tables, you should be aware that large data sets could require lengthy processing times (since data sets are packaged in the SPK file along with the metadata).
- If you press the CTRL key and highlight multiple objects of the same type, the **Options** tab 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 SAS Package page displays a list of the objects and folders that you selected. If the list is correct, click **Next**.



- 7 As the export process begins, the wizard might display a prompt asking you to log on to the SAS Application Server. Be sure to log on as a user who has the appropriate credentials (see “Requirements for Accessing a Standard Workspace Server” on page 37). If you are already logged on to SAS Management Console, SAS Data Integration Studio, or SAS OLAP Cube Studio with the correct credentials, then the prompt does not appear.
- 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 Promotion Tools” on page 253.

*Note:* For complete step-by-step instructions for using the Export SAS Package wizard, see the product Help for SAS Management Console, SAS Data Integration Studio, or SAS OLAP Cube Studio. △

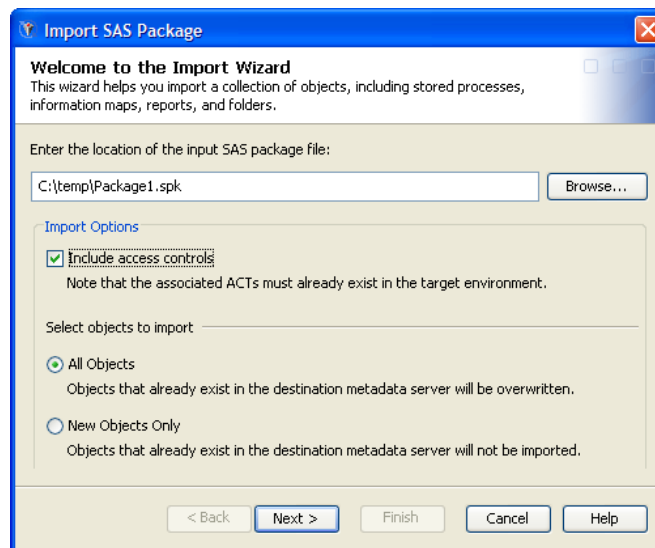
---

## 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.

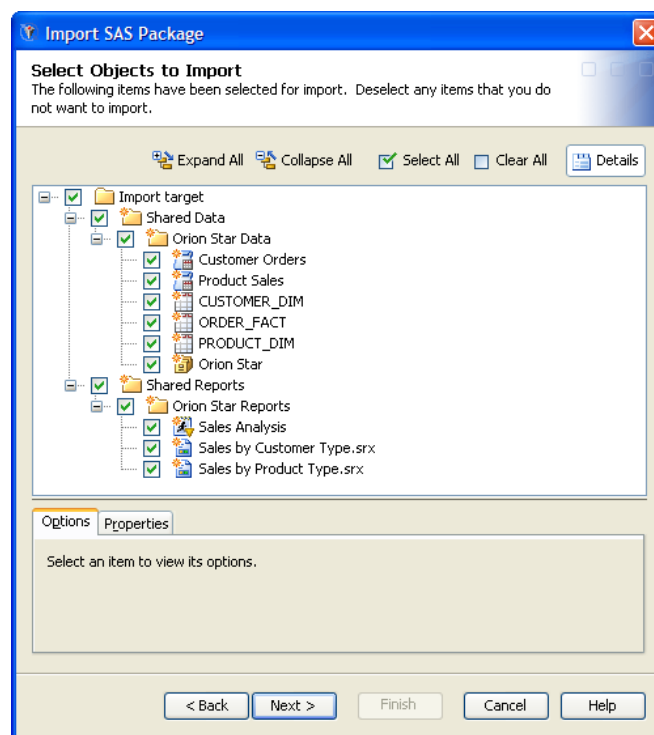
- 1 Locate the SPK file that was created by the Export SAS Package wizard. If necessary, copy it, move it, or e-mail it to a location that is accessible from the target environment.
- 2 Open SAS Management Console, and connect to the target metadata server (that is, the metadata server to which you want to promote the folders and objects).
- 3 Select the folder under which you want to import the folders and objects.
- 4 Right-click to display the menu, and click **Import SAS Package**.
- 5 On the first Import SAS Package 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. This will cause the imported objects to overwrite any objects of the same name that already exist.

*Note:* If you do not want to overwrite objects that already exist, then you would click **New Objects Only**. △



d Click **Next**.

- 6 The next Import SAS Package page displays a tree which shows the objects that are included in the package.



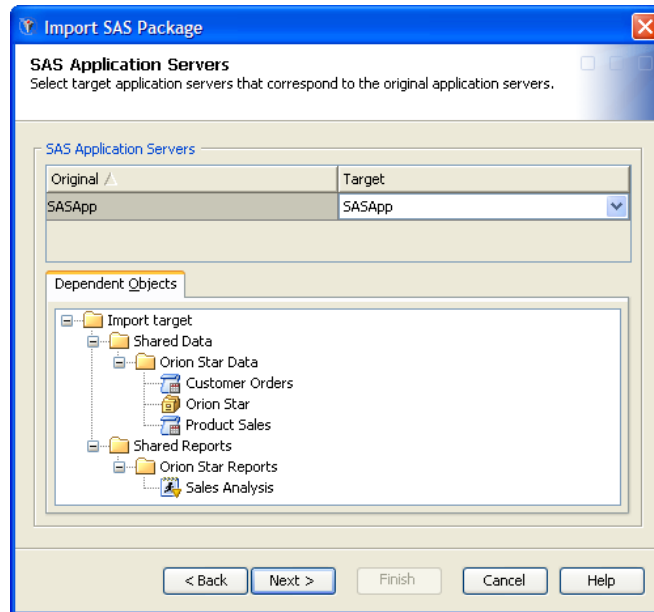
On this page, you can highlight the table objects, and then use the **Options** tab to indicate whether the physical tables are to be imported along with the metadata. You can also edit the table names. (These options are available only if you exported the physical tables.)

Then click **Next**.

- 7 The next Import SAS Package page provides advance information about the metadata values that you will need to select on subsequent windows. For this

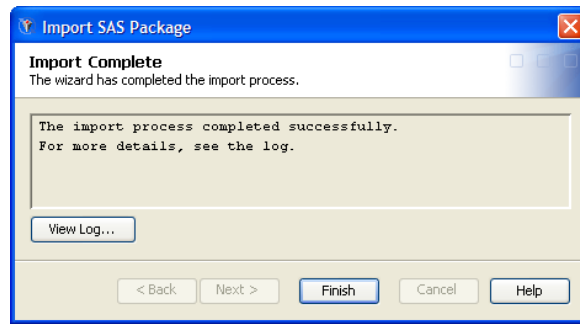
scenario, you will need to select values for an application server, a library directory path, and a source code repository. Click **Next**.

- 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, select the path where the source code file for the stored process should be stored in the target environment. (If the source code repository definition does not yet exist, then click **Add** to create it.) Then click **Next**.
- 10 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**.
- 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 **Next**.
- 12 As the import process begins, the Import SAS Package wizard might display a prompt asking you to log on to the SAS Application Server. Be sure to log on as a user who has the appropriate credentials (see “Requirements for Accessing a Standard Workspace Server” on page 37). If you are already logged on to SAS Management Console, SAS Data Integration Studio, or SAS OLAP Cube Studio with the correct credentials, then the prompt does not appear.
- 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 Promotion Tools” on page 253.

*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 SAS Package wizard, see the product Help for SAS Management Console, SAS Data Integration Studio, or SAS OLAP Cube Studio. △

---

## Using the Batch Export and Import Tools

---

### About the Batch Export and Import Tools

The batch import and export tools enable you to perform promotions from an operating system command line or from a batch script. These tools provide the same capabilities as the Export SAS Package and Import SAS Package wizards, with the following exceptions:

- The wizards provide the option of either including or excluding associated content such as physical tables and source code files. When you use the batch tools, the promotion automatically includes all associated content except tables and external files.
- In the Export SAS Package wizard, you can display a list of the objects on which an object depends and then individually specify which of these objects to export. In the batch export tool, you can select a global option to include all dependent objects.

The batch export and import tools are useful for the following purposes:

- to perform ad hoc promotion jobs from the command line, as an alternative to using the wizards
- to create promotion jobs that are to be run on a scheduled basis or a repeated basis

#### **CAUTION:**

Use of the batch import tool to import SAS 9.1.3 packages can result in a significant number of broken metadata associations, such as tables with no associated library, jobs with missing tables, transformations within jobs with missing mappings, and expressions within mappings with missing columns. As a best practice, you should use the batch import tool to import a SAS 9.1.3 package only if the resources associated with the

imported objects have the same values in the two environments. Otherwise, it is strongly recommended that you use the Import SAS Package wizard in either SAS Management Console or SAS Data Integration Studio. For more information, see “Using the Batch Import Tool to Import a Package that Was Created in SAS 9.1.3” on page 253. △

The batch import tool and export tools are called **ImportPackage** and **ExportPackage** and are located in the following path:

*SAS-installation-directory\SASPlatformObjectFramework\9.2*

You can execute these tools on any machine where the SAS Platform Object Framework is installed.

---

## Preparing to Use the Batch Export Tool or the Batch Import Tool

The batch export tool and import tool require the same special considerations and the same preparation steps as the Export SAS Package and Import SAS Package wizards. (See “Special Considerations for Promoting Metadata” on page 221 and “Preparing to Use the Promotion Tools” on page 226). Make sure that the user who is specified in the options has the appropriate credentials (see “Ensure Appropriate User Credentials” on page 226).

If you are using the batch import or export tool on a UNIX machine, make sure that the DISPLAY variable has been set on the machine. To set the variable, enter a command such as the following:

```
export DISPLAY=machine-name
```

---

## Using the Batch Export Tool

### Syntax for the Batch Export Tool

ExportPackage

```
-host source-host-name
  -port source-port-number
  -user user-id
  -password password
  -domain domain-name
  <-profile connection-profile-name>
  -package package-path-and-filename
  -objects source-location-1 source-location-2 source-location-3 ...
    source-location-n
  <-types object-type-1, object-type-2....object-type-n>
  <-name character-string>
  <-includeDep>
  <-since from-absolute-date | from-relative-date | relative-time-period>
  <-before to-absolute-date | to-relative-date>
  <-created>
  <-modified>
  <-includeEmptyFolders>
  <-subprop>
  <-log log-path | log-path-and-filename | . >
  <-noexecute>
```

<-? | -help>

## Syntax Description for the Batch Export Tool

-host *source-host-name*

specifies the machine name of the metadata server from which the package is being exported.

**Required:** Yes, if **-profile** is not specified.

-port *source-port-number*

specifies the port number of the metadata server from which the package is being exported.

**Required:** Yes, if **-profile** is not specified.

-user *user-id*

specifies the user ID that is to be used to connect to the source metadata server. Make sure that this user has the appropriate credentials (see “Ensure Appropriate User Credentials” on page 226).

**Required:** Yes, if **-profile** is not specified, or if the specified connection profile does not contain user credentials.

-password *password*

specifies the password of the user who is specified in **-user**.

**Required:** Yes, if **-profile** is not specified, or if the specified connection profile does not contain user credentials.

-domain *domain-name*

specifies the SAS authentication domain in which the user ID is to be authenticated.

**Required:** No. You can omit this option if your installation has only one SAS authentication domain. You can also omit this option if you specify **-profile** and if the specified connection profile contains a domain name.

-profile *connection-profile-name*

specifies the name of the connection profile that is to be used to connect to the source metadata server. You can specify this option instead of **-host**, **-port**, **-user**, **-password**, and **-domain**.

The connection profile must exist on the computer where the export command is executed. You can specify any connection profile that has been created for use with SAS Management Console or SAS Data Integration Studio. When you open one of these applications, the available connection profiles are displayed in the drop-down box in the Connect Profile dialog box.

If the connection profile name contains spaces, then use quotation marks. If the connection profile contains a user ID and password, then make sure that the user has the appropriate credentials (see “Ensure Appropriate User Credentials” on page 226).

**Required:** No. You can omit this option if **-host** and **-port** are provided.


**Interaction:** If you specify this parameter, then **-host** and **-port** can be omitted. If the specified connection profile contains a user ID and password, then **-user** and **-password** can also be omitted.

-package *package-path-and-filename*

specifies the path and filename of the package file that you want to create. The objects that you export will be placed in this file.

The filename must have the extension .spk. The path is optional, but if you do specify a path, it must be fully qualified.

**CAUTION:**

**If a file of this name already exists, then it will be overwritten.** If you want to preserve pre-existing packages, then be sure to specify a unique name. 

If the path or filename includes spaces, then use quotation marks.

**Required:** You must specify a filename, but the path is optional.

**Default path:** If you do not specify a path, then the package is placed in the directory where the command was launched.

**-objects** *source-location-1 source-location-2 source-location-3 ... source-location-n*  
specifies the locations of the objects that are to be exported. You can specify any number of locations. Leave a space between each location. If a location includes spaces, then surround the location with quotation marks.

Use the following syntax to specify a location:

*/folder-1/folder-2/...folder-n/<object-name(object-type)>*

The following rules apply to specifying locations:

- ☐ Locations are relative to the **SAS Folders** node. Therefore, the first folder that you specify in a location must be located directly beneath **SAS Folders**.
- ☐ If you specify a folder but you do not specify an object name, then all objects and folders beneath that folder are exported.
- ☐ To export objects from your personal folder, you must specify the actual path (**/Users/user-name/My Folder/**) rather than the shortcut (**/My Folder/**).
- ☐ If you specify an object name, then you must also specify the object's type. For a list of object types, see "Specifying Object Types" on page 244. Object types are case-sensitive.

The following are examples of locations:

**-objects /**

exports the entire folder hierarchy and all of its objects.

**-objects "/Users/sasdemo/My Folder"**

exports all objects and folders that are in the personal folder of the user named **-sasdemo**.

**-objects "/Shared Data/Orion Star Data/Customer Orders(InformationMap)"**

exports the Customer Orders information map, which is located in **/Shared Data/Orion Star Data**.

**-objects "/Shared Data/Orion Star Data/Customer Orders(InformationMap)" "/Shared Data/Orion Star Data/CUSTOMER\_DIM(Table)" "/Shared Data/Orion Star Data/ORDER\_FACT(Table)" "/Shared Reports/Orion Star Reports/Sales by Customer Type(Report)"**

exports the Customer Orders information map, the CUSTOMER\_DIM table, and the ORDER\_FACT table, all of which are located in **/Shared Data/Orion Star Data**; and the Sales by Customer Type report, which is located in **/Shared Reports/Orion Star Reports**.

**Required:** Yes

**Interaction:**

- ☐ To further limit the objects that are exported, you can use the **-types**, **-name**, **-since**, and **-before** options.

- If you want to automatically export all of the objects on which the exported objects depend (instead of having to explicitly list those objects), specify **-includeDep**.

**-types** *object-type-1, object-type-2....object-type-n*

specifies a comma-separated list of the types of objects that are to be exported. If you specify this option, then the tool exports only objects that are of the specified types; all other objects are ignored.

If the string contains spaces, then use quotation marks.

For a list of object types, see “Specifying Object Types” on page 244. Object types are case-sensitive.

If you specify an object type that has one or more subtypes, then the subtypes are automatically included in the export. For example, if you specify **InformationMap**, then **InformationMap.OLAP** and **InformationMap.Relational** are automatically included. If you specify **MQM**, then **MQM.MSMQ** and **MQM.Websphere** are automatically included.

**Required:** No

**Exception:** This option does not apply to folders. Objects that are contained in folders are exported only if their object types are specified. However, the folders themselves are exported regardless of whether they contain the specified object types.

**-includeDep**

specifies that all objects on which the exported objects depend are to be included in the export. If you select this option, the export tool recursively identifies dependencies and includes the identified objects in the export.

For example, if you export an information map, the export tool identifies and exports any libraries, tables, and stored processes that are used by the information map. The process also exports any objects that the identified stored processes depend on.

**CAUTION:**

**If you do not export the objects on which exported objects depend, then the exported objects might not function properly when you import them to the target metadata server.** △

**Required:** No

**-name** *character-string*

specifies a character string that must be present in an object’s name in order for the object to be exported. Objects whose names contain the specified string are exported, and all other objects are ignored.

This option is not case sensitive. For example, if you specify **-name product**, then the tool exports objects with the names **Sales by Product** and **PRODUCT\_DIM**.

If the string contains spaces, then use quotation marks.

**-since** *from-absolute-date | from-relative-date | relative-time-period*

specifies that objects are to be exported only if they were created or modified on or after the specified date, or during the specified time period. You can specify one of the following:

*from-absolute-date*

specifies an absolute date (see “Specifying an Absolute Date (or an Absolute Date and Time)” on page 245)

*from-relative-date*

specifies a date relative to the current date (see “Specifying a Relative Date” on page 245)



*relative-time-period*

specifies a time period relative to the current date (see “Specifying a Relative Time Period” on page 246)

**Required:** No

**Interaction:** You can specify either **-created** or **-modified** with this option. The default setting is **-modified**.

*-before from-absolute-date | from-relative-date*

specifies that objects are to be exported only if they were created or modified on or before the specified date. You can specify one of the following:

*from-absolute-date*

specifies an absolute date (see “Specifying an Absolute Date (or an Absolute Date and Time)” on page 245)

*from-relative-date*

specifies a date relative to the current date (see “Specifying a Relative Date” on page 245)

**Required:** No

**Interaction:**

- ☐ To specify a date range, specify **-since** along with this option.
- ☐ You can specify either **-created** or **-modified** with this option. The default setting is **-modified**.

*-created*

specifies that the **-since** and **-before** options are to be based on each object’s Created date. If **-created** is specified, then objects are exported only if the Created date meets the criteria specified in **-since** and **-before**.


**Required:** No

**Default:** If **-since** or **-before** is specified but **-created** is not specified, then **-modified** is assumed.

**Interaction:** If you specify both **-created** and **-modified**, then **-modified** takes precedence.

*-modified*

specifies that the **-since** and **-before** options are to be based on each object’s Modified date. If **-modified** is specified, then objects are exported only if the Modified date meets the criteria specified in **-since** and **-before**

*Note:* If an object has never been modified, then the object’s Modified date is equal to its Created date. 

**Required:** No

**Default:** If **-since** or **-before** is specified, and if you specify neither **-created** nor **-modified**, then **-modified** is assumed.

**Interaction:** If you specify both **-created** and **-modified**, then **-modified** takes precedence.

*-includeEmptyFolders*

specifies that folders are to be exported even if they do not contain any subfolders or objects.

**Required:** No

**Default:** If this option is not specified, then empty folders are not exported.

*-subprop*

specifies that the export tool is to create an external substitution properties file. By default, the export tool always creates a substitution properties file inside of

the package file. If you specify this option, then the substitution properties file is created externally.

In this file, you can modify values that are associated with the exported objects so that the objects will function properly in the target system. For example, you can modify server names, source code repository paths, OLAP schema names, and cube names to reflect the names that are used on the target system. For detailed instructions, see “Using the Substitution Properties File” on page 247.

**Required:** No

**Default:** If you do not specify this option, then an external substitution properties file is not created. The import process will use the source environment’s settings for server names, source code repository paths, database schema names, and archive paths and will expect those servers, paths, and schemas to exist in the target environment.

A substitution file is still created inside the package file and can be extracted as described in “Using the Batch Import Tool to Import a Package That Was Created by the Export SAS Package Wizard” on page 252.

`-log log-path | log-path-and-filename | .`

specifies one of the following:

- the path (or the path and filename) where the log file is to be written
- that the log file is to be written to the current directory. To specify this option, use a period, as shown here:

`-log .`

The log file contains messages that are generated during the batch tool’s execution.

**Required:** No

**Default:** If you do not specify this option, then the log is directed to the path and filename that are described in “Viewing the Batch Export or Import Log” on page 252.

`-noexecute`

specifies that the package file is not to be created. Instead, a list of all of the objects that are included in the export is displayed, and a log file is created. This option is useful if you want to verify that the command is constructed correctly before you execute it.

**Required:** No

**Default:** If you do not specify this option, then the package is exported to the specified package.

`[-? | -help]`

specifies to display help for the batch export tool.

**Required:** No

## Specifying Object Types

When you use the export batch tool, you might need to specify one or more object types, as follows:

- in the **-objects** option, if you specify individual objects. You must specify the object type in parentheses after each object name.
- in the **-types** option.

The following table provides a list of object type names that you can use with these options.

### CAUTION:

Object types are case sensitive. Be sure to enter them exactly as they are shown here. △

*Note:* You might be able to export additional object types, depending on which applications you have installed. △

**Table 17.4** Valid Object Types

Action	Folder	Project.EG
Application	GeneratedTransform	Project.AMOExcel
BurstDefinition	InformationMap	Project.AMOPowerPoint
Channel	InformationMap.OLAP	Project.AMOWord
Condition	InformationMap.Relational	Prompt
ConditionActionSet	Job	PromptGroup
ContentSubscriber	Library	Report
Cube	MessageQueue	Report.Component
DataExploration	MiningResults	Report.Image
DeployedFlow	MQM.MSMQ (queue manager for MSMQ)	Service.S SOAPGenerated
DeployedJob		StoredProcess
Document	MQM.Websphere (queue manager for WebSphere MQ)	SubscriberGroup.Content
EventSubscriber	Note	SubscriberGroup.Event
ExternalFile	OLAPSchema	Table
	Project	

## Specifying an Absolute Date (or an Absolute Date and Time)

When you use **-since** or **-before** in a batch export command, you can specify an absolute date (or an absolute date and time). Use one of the following formats:

- ddMMMyyyy
- ddMMMyyyy:HH:mm:ss
- ddMMMyy
- ddMMMyy:HH:mm:ss
- MM/dd/yyyy
- MM/dd/yyyy HH:mm:ss
- yyyyMMdd
- yyyyMMdd:HH:mm:ss

*Note:* If you do not specify a time, then the specified date begins at midnight (12:00:00 a.m.) and ends at 11:59:59 p.m. △

## Specifying a Relative Date

When you use **-since** or **-before** in a batch export command, you can specify a date relative to the current date. To specify a relative date, use one of the following values:

### **Today**

is the current date.

### **Yesterday**

is the day just before the current date.

### **"Current day of last year"**

is the same as the current date, except that the year is replaced with the previous year. For example, if the current date is October 12, 2010, then **Current day of last year** is October 12, 2009.

**"Current day of last month"**

is the same as the current date, except that the month is replaced with the previous month. For example, if the current date is October 12, 2010, then

**"Current day of last month"** is September 12, 2010.

**"Current day of last week"**

is seven days previous to the current date. For example, if the current date is October 12, 2010, then **Current day of last week** is October 5, 2010.

**"n days ago"**

is *n* days previous to the current date. When specifying this option, replace *n* with an integer.

*Note:* Dates are assumed to begin at midnight (12:00:00 a.m.) and end at 11:59:59 p.m.  $\triangle$

## Specifying a Relative Time Period

When you use **-since** in a batch export command, you can specify a time period that is relative to the current date. To specify a relative time period, specify one of the following values:

**"Year to date"**

is the period from January 1 of the current year up to and including the current date.

**"Month to date"**

is the period from the first day of the current month up to and including the current date.

**"Week to date"**

is the period from the most recent Monday up to and including the current date.

*Note:* Dates are assumed to begin at midnight (12:00:00 a.m.) and end at 11:59:59 p.m.  $\triangle$

## Example: Batch Export Command

```
ExportPackage -profile "My Profile" -package "c:\SAS Promotion  
Files\Package_1.spk" -objects "/Shared Reports/Orion Star Reports"  
-includeDep -since "Week to date" -modified -subprop
```

This command does the following:

- ☐ exports folders, subfolders, and objects from **/Shared Reports/Orion Star Reports**. Objects are included only if they were created or modified since Monday of the current week.
- ☐ recursively identifies the objects on which the exported objects depend, and exports those objects as well.
- ☐ exports physical content (for example, .srx files) that is associated with the exported objects.
- ☐ creates a package file called **Package\_1.spk** that contains the exported objects and content, and places the file in **C:\SAS Promotion Files**.
- ☐ creates a substitution properties file called **Package\_1.subprop**, and places the file in **C:\SAS Promotion Files**.

## Using the Substitution Properties File

### About the Substitution Properties File

If you specify **-subprop** in a batch export command, then the export tool creates an external substitution properties file. In this file, you can modify values that are associated with the exported objects so that the objects will function properly in the target system. For example, you can modify server names, source code repository paths, database schema names, and archive paths to the values that are appropriate on the target system.

The substitution properties file has the same path and filename that you specify in **-package**, except that it has the filename extension **.subprop** instead of **.spk**. For example, if the package file is named **Package\_1.spk**, then the substitution properties file is named **Package\_1.subprop**.

*Note:* Packages that are created in SAS 9.1.3 do not contain substitution properties files. For more information, see “Using the Batch Import Tool to Import a Package that Was Created in SAS 9.1.3” on page 253. △

### Edit the Substitution Properties File

Before you import a package, make any necessary updates to the substitution properties file. Follow these guidelines:

- Modify only properties that contain the word **Target** (for example, **TargetName=**, **TargetPath=**, and **TargetDirectory=**).
- Update the target properties with the exact names of the resources as they appear on the target system. The names are case sensitive.

Here are examples of properties before the substitution properties file was modified:

```
[Connections: Application Server]
ApplicationServer[1].SourceName=SASAppTest
ApplicationServer[1].TargetName=SASAppTest

[Connections: Source Code Repository]
SourceCodeRepository[1].ApplicationServer=ApplicationServer[1]
SourceCodeRepository[1].SourceDirectory=c:\\StoredProcessesTest
SourceCodeRepository[1].TargetDirectory=c:\\StoredProcessesTest
SourceCodeRepository[1].CreateIfNeeded=false
```

Here are examples of modified properties in a substitution properties file. Note that only the target values have been modified. In addition, note that **CreateIfNeeded** has been changed to **true**. This means that the source code repository definition will be created if it does not exist on the target system.

```
Connections: Application Server]
ApplicationServer[1].SourceName=SASAppTest
ApplicationServer[1].TargetName=SASApp

[Connections: Source Code Repository] SourceCodeRepository[1].ApplicationServer=
ApplicationServer[1]
SourceCodeRepository[1].SourceDirectory=c:\\StoredProcessesTest
SourceCodeRepository[1].TargetDirectory=c:\\StoredProcesses
SourceCodeRepository[1].CreateIfNeeded=true
```

## Apply the Substitution Properties File

To apply the changes that you made to the substitution properties file, specify the **-subprop** option when you run the batch import tool.

---

## Using the Batch Import Tool

### Syntax for the Batch Import Tool

ImportPackage

```
-host target-host-name
  -port target-port-number
  -user user-id
  -password password
  -domain domain-name
  <-profile connection-profile-name>
  -package package-path-and-filename
  -target target-location
  <-types object-type-1, object-type-2....object-type-n>
  <-newOnly>
  <-includeACL>
  <-subProp <substitution-properties-filename>>
  <-preservePaths>
  <-log log-path | log-path-and-filename | . >
  <-noexecute>
  <-? | -help>
```

### Batch Import Tool Syntax Description

-host *target-host-name*  
specifies the machine name of the metadata server to which the package is being imported.

**Required:** Yes, if **-profile** is not specified.

-port *target-port-number*  
specifies the port number of the metadata server to which the package is being imported.

**Required:** Yes, if **-profile** is not specified.

-user *user-id*  
specifies the user ID that is to be used to connect to the target metadata server. Make sure that this user has the appropriate credentials (see “Ensure Appropriate User Credentials” on page 226).

**Required:** Yes, if **-profile** is not specified, or if the specified connection profile does not contain user credentials.

-password *password*  
specifies the password of the user who is specified in **-user**.

**Required:** Yes, if **-profile** is not specified, or if the specified connection profile does not contain user credentials.

**-domain** *domain-name*

specifies the SAS authentication domain in which the user ID is to be authenticated.

**Required:** No. You can omit this option if your installation has only one SAS authentication domain. You can also omit this option if you specify **-profile** and if the specified connection profile contains a domain name.

**-profile** *connection-profile-name*

specifies the name of the connection profile that is to be used to connect to the target metadata server. You can specify this option instead of **-host**, **-port**, **-user**, **-password**, and **-domain**.

The connection profile must exist on the computer where the import command is executed. You can specify any connection profile that has been created for use with SAS Management Console or SAS Data Integration Studio. When you open one of these applications, the available connection profiles are displayed in the drop-down box in the Connect Profile dialog box.

If the connection profile name contains spaces, then use quotation marks. If the connection profile contains a user ID and password, then make sure that the user has the appropriate credentials (see “Ensure Appropriate User Credentials” on page 226).

**Required:** No. You can omit this option if **-host** and **-port** are provided.

**Interaction:** If you specify this parameter, then **-host** and **-port** can be omitted. If the specified connection profile contains a user ID and password, then **-user** and **-password** can also be omitted.

**-package** *package-path-and-filename*

specifies the path and filename of the package file that you want to import. The filename must have the extension .spk. If the path or filename includes spaces, then use quotation marks.

You must specify a package that was created with one of the following methods:

- ☐ the batch export tool
- ☐ the Export SAS Package wizard (see “Using the Batch Import Tool to Import a Package That Was Created by the Export SAS Package Wizard” on page 252 before you import this type of package)
- ☐ the BI Manager Export Wizard in SAS 9.1.3 (see “Using the Batch Import Tool to Import a Package that Was Created in SAS 9.1.3” on page 253 before you import this type of package)

The path is optional, but if you do specify a path, it must be fully qualified.

**Required:** You must specify a filename, but the path is optional.

**Default path:** If you do not specify a path, then the package is assumed to be located in the directory where the command was launched.

**-target** *target-metadata-location*


the location, relative to **SAS Folders**, to which the package is to be imported on the target metadata server. For example:

**-target /**

imports the package to the root level (directly under **SAS Folders**)

**-target "/Shared Data"**

imports the package to the **Shared Data** folder

*Note:* To import objects to your personal folder, you must specify the actual path (**/Users/user-name/My Folder/**) rather than the shortcut (**/My Folder/**). 

**Required:** Yes

**-types** *object-type-1, object-type-2....object-type-n*

specifies a comma-separated list of the types of objects that are to be imported. If you specify this option, then the tool imports only objects that are of the specified types; all other objects are ignored.

For a list of object types, see “Specifying Object Types” on page 244. Object types are case sensitive.

**Required:** No

**Exception:** This option does not apply to folders. Objects that are contained in folders are imported only if their object types are specified. However, the folders themselves are imported regardless of whether they contain the specified object types.

**-newOnly**

specifies that objects are to be imported only if they do not already exist in the specified location on the target metadata server. If objects of the same name and type already exist, then they are not imported.

For example, suppose the package file contains a relational information map called **Customer Orders** in the location **/Shared Data/Orion Star Data**, and that the target metadata server contains a relational information map of the same name in that location. If **-newOnly** is specified, then the information map is not imported.

**Required:** No

**Default:** If you do not specify this option, then objects that already exist on the target metadata server are overwritten by any imported objects that have the same location, name, and type.

**-includeACL**

specifies that access controls are to be imported for all of the objects in the package. If the objects already exist on the target metadata server, then any existing access controls for these objects will be overwritten by the imported access controls.

For details about this option, see “Optional Promotion of Access Controls” on page 221.

**Required:** No

**-subprop** *<substitution-properties-filename>*

specifies that imported objects are to be updated based on properties that are specified in the substitution properties file. You must manually update this file to specify how imported objects are to be mapped to existing servers and directories in the target environment. For instructions, see “Edit the Substitution Properties File” on page 247.

The *substitution-properties-filename* argument is not required with this option. If the filename is omitted, then the name is assumed to be *package-name.subprop*. For example, if your package is named **Package1.spk**, then the import tool expects the substitution properties file to be named **Package1.subprop**. If your properties file has a different name, then you must either rename it or specify the filename in the **-subprop** option. The filename must have the extension **.subprop**.

**CAUTION:**

**If you do not update the properties file or if you do not specify this option, then some imported objects might not function properly.** You might need to log on to the target metadata server after the import is complete and modify servers, paths, and other values that are associated with the imported objects.  $\triangle$

**Required:** No



**-preservePaths**

specifies that objects and folders are to be placed in the same path where they were located in the source environment. If the folders that are included in the path do not exist, they are created.

For example, suppose you export a library from **/Shared Data/Libraries**. If you import the library to the root level (**SAS Folders**) in the target environment, then the import wizard determines whether a **Libraries** folder exists under **Shared Data**. If the folder does not exist, the wizard creates it and places the imported library in that folder.

If you import objects into a folder below the root level, then this option is ignored.

**Required:** This option is required if you import individual objects (independently of the folders that contain them) and you have specified the root folder (**SAS Folders**) as the import target by specifying **-target /**. This option is necessary because the root folder can contain folders, but it cannot contain individual objects.

**-log *log-path* | *log-path-and-filename* | .**

specifies one of the following:

- ☐ the path (or the path and filename) where the log file is to be written
- ☐ that the log file is to be written to the current directory. To specify this option, use a period, as shown here:

**-log .**

The log file contains messages that are generated during the batch tool's execution.

**Required:** No

**Default:** If you do not specify this option, then the log is directed to the path and filename that are described in "Viewing the Batch Export or Import Log" on page 252.

**-noexecute**

specifies that the package is not to be imported to the target metadata server. Instead, a list of all of the objects that are included in the import is displayed, and a log file is created. This option is useful if you want to verify that the command is constructed correctly before you execute it.

**Required:** No

**Default:** If you do not specify this option, then the package is imported to the target metadata server.

**[-? | -help]**

specifies to display help for the batch import tool.

**Required:** No

## Example: Batch Import Command

```
ImportPackage -profile "My Profile" -package "c:\SAS Promotion  
Files\Package_1.spk" -target / -includeACL -subprop
```

This command does the following:

- ☐ imports folders, subfolders, and objects from **Package\_1.spk** and writes them to the root location under **SAS Folders** on the target metadata server
- ☐ updates objects on the target metadata server with values that are specified in the substitution properties file
- ☐ imports access controls for all of the objects in the package

- imports physical content (for example, SRX files) that is associated with the imported objects

---

## Viewing the Batch Export or Import Log

When you run the batch export or import tool, a log file is created in the following path:

- On Windows Vista systems: **C:\Users\user\AppData\Roaming\SAS\Logs**
- On other Windows systems: **C:\Documents and Settings\user\Application Data\SAS\Logs**
- On UNIX systems: *user's-home-directory/SAS/Logs*

*Note:*

- To see the log file on Windows systems, you must enable the option to view hidden files and folders.
- If you want to direct the log to a different location, specify the **-log** option with the import or export command.

△

The log file is named **Export\_yyMMddHHmmss.log** or **Import\_yyMMddHHmmss.log**, where *yyMMddHHmmss* is the date and time that the command was executed.

If the batch import or export command fails, you can use the log file to determine the source of the error. For a list of common error messages, see “Troubleshooting the Promotion Tools” on page 253.

---

## Using the Batch Import Tool to Import a Package That Was Created by the Export SAS Package Wizard

You can use the batch import tool to import a package that was created by the Export SAS Package wizard. When you run the Export SAS Package wizard, the wizard creates a substitution properties file and places it in the package. If you want to provide different values for servers, paths, or other resources, then you must extract the substitution file and modify the properties. Follow these steps:

- 1 Extract the substitution properties file from the package. (Do not save the package file after you perform the extract.)
- 2 Modify the target values in the substitution properties file as needed (see “Using the Substitution Properties File” on page 247).
- 3 Optional) Rename the properties file so that it has the same base filename as the package. For example, if your package is named **Package1.spk**, then rename the properties file **Package1.subprop**.
- 4 Specify **-subprop** in the batch import command. If you did not rename the package (as described in the preceding step), then you must specify the substitution properties filename with this option.

---

## Using the Import SAS Package Wizard to Import a Package that Was Created by the Batch Export Tool

You can use the Import SAS Package wizard to import a package that was created by the batch export tool. The import works the same as it would if the package had been created by the Export SAS Package wizard.


---

## Using the Batch Import Tool to Import a Package that Was Created in SAS 9.1.3

You can use the batch import tool to import a package that was created by the BI Manager export wizard in SAS 9.1.3, subject to the following special considerations:

- The SAS 9.1.3 environment must have the correct software versions. For details, see “Special Considerations for Promoting Metadata” on page 221.
- In SAS 9.1.3, only the following types of objects can be exported: data explorations, documents, deployed jobs, external files, folders, generated transformations, information maps, jobs, job flows, libraries, mining results, notes, reports, stored processes, and tables. In SAS 9.2, a larger number of object types can be imported and exported (see “Specifying Object Types” on page 244).
- For the servers, paths, and other resources that are associated with the imported objects, the batch tool will apply the same values that were used in the SAS 9.1.3 environment. After the import is completed, you must use SAS Management Console (or other SAS applications, as needed) to modify the imported objects’ properties so that their associated servers, paths, and other values are correct for the target environment.

### **CAUTION:**

**Use of the batch import tool to import SAS 9.1.3 packages can result in a significant number of broken metadata associations, such as tables with no associated library, jobs with missing tables, transformations within jobs with missing mappings, and expressions within mappings with missing columns.** As a best practice, you should use the batch import tool to import a SAS 9.1.3 package only if the resources associated with the imported objects have the same values in the two environments. Otherwise, it is strongly recommended that you use the Import SAS Package wizard in either SAS Management Console or SAS Data Integration Studio. 

---

## Troubleshooting the Promotion Tools

The following error messages might appear when you run the Export SAS Package wizard, the Import SAS Package wizard, or the batch import or export tool:

### **Out of memory error occurred during the import (or export) process.**

This error could occur if you export or import a large number of sizable objects, such as information maps or SAS Data Integration Studio jobs. If this error occurs, you can take either of the following actions:

- Instead of exporting all of your objects at the same time, perform multiple exports that include a smaller number of objects.
- Increase the amount of memory that is allocated to the import or export process. For instructions, see “Ensure Adequate Memory” on page 226.

**Duplicate named objects were found within the same folder. These objects will not be exported. View the Details to see the list of objects.**

This error could occur in these circumstances:

- You use the **Include all dependencies** option in the Export SAS Package wizard or the **-includeDep** option with the export batch tool.
- The export includes two or more objects that depend on different objects that have the same name and type. For example:
  - In the Export SAS Package wizard, you select two information maps: Map A and Map B.
  - You select the **Include all dependencies** option.
  - Map A depends on a table called **PRODUCT-DIM** that is located in **/Shared Data/Orion Star Data**, and Map B depends on a different table called **PRODUCT-DIM** that is located in **/Shared Data/Company Data**.

To resolve the error, export the objects' folders along with the objects, as follows:

- If you are using the Export SAS Package wizard:
  - 1 Go back to the previous wizard page.
  - 2 Select a folder that contains both the parent objects and the dependent objects. In the preceding example, you would select the **Shared Data** folder.
  - 3 Within the folder structure, deselect any objects and folders that you do not want to export. However, make sure that the dependent objects that have duplicate names (for example, the two different **PRODUCT-DIM** tables) remain selected, as well as the folders that make up these objects' paths.
- If you are using the batch export tool, modify the **-objects** option so that it specifies the dependent objects that have duplicate names (for example, the two different **PRODUCT-DIM** tables) and their paths.

**The export process has failed. No X11 DISPLAY variable was set, but this program performed an operation which requires it. For more information, view the export log file: *filename***

This error could occur if you submit a batch import or export job on a UNIX machine for which the DISPLAY variable has not been set. To resolve the error, set the DISPLAY variable by entering a command such as the following:

```
export DISPLAY=machine-name
```

**No extraField1 found.**

This error can occur if you try to import a package file whose contents were updated after the package was created. For example, if you open a package file, save any of the files in the package (for example, the export log), and then save the package file, then the package becomes corrupted and cannot be imported.

To resolve the error, perform the export again to re-create the package.

**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.**

This message means that the objects that you are importing depend on objects that are not present on the target metadata server.

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.

**Destination folder has invalid or missing content mapping; cannot import reports.**

If this message appears when you try to run the Import SAS Package wizard, click **OK** to close the message box and click **Cancel** to stop the import process.

Specify content mapping on the target metadata server (see “For Reports, Set Up Content Mapping on the Target Metadata Server” on page 227), and start the import again.

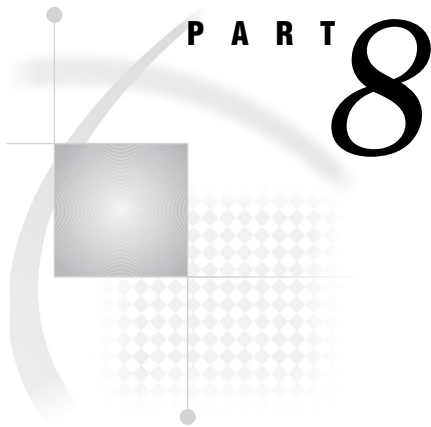
**Could not import "*filename.sas*" in "*path*" on application server "*server name*". Reason: Insufficient authorization to access *path*.**

This message appears in the error log at the end of the import process. It means that the user who ran the import 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 import 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.



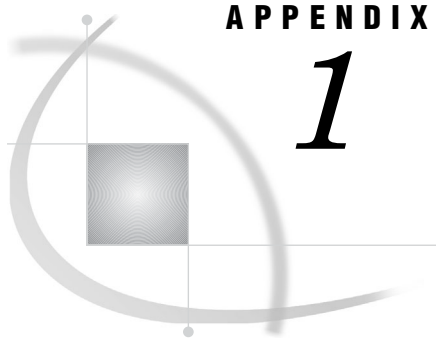


## Appendixes

<i>Appendix 1</i>	<b>Default SAS Ports</b>	259
<i>Appendix 2</i>	<b>Reference: Configuration Directories</b>	263
<i>Appendix 3</i>	<b>Default Locations for Server Logs</b>	269
<i>Appendix 4</i>	<b>Configuration Files</b>	271
<i>Appendix 5</i>	<b>Options for the Metadata Server Invocation Command</b>	289
<i>Appendix 6</i>	<b>Using the SAS Deployment Manager to Update Host Name References</b>	295
<i>Appendix 7</i>	<b>Recommended Reading</b>	319







## APPENDIX

## 1

## Default SAS Ports

*Overview*    **259**

*Default Port Numbers for SAS Servers and Spawners*    **259**

*Default Port Numbers for Third-Party Software*    **260**

*Port Numbering in a Multiple-Level Environment*    **261**

### Overview

The servers in the SAS Intelligence Platform communicate with clients and other servers using TCP/IP. Each server listens on a particular port or ports for incoming requests. During installation, the SAS Deployment Wizard enables you to either accept the default ports or to specify different port numbers.

### Default Port Numbers for SAS Servers and Spawners

The following table shows the default port numbers for SAS servers and spawners that are installed in a **Lev1** environment. The SAS Deployment Wizard allows different port numbers to be specified. Therefore, your site might use different port numbers than the ones that are shown here.

**Table A1.1** Default Port Numbers and Descriptions for SAS Servers and Spawners

Default Port Value	Description
8561	SAS Metadata Server port
8581	SAS object spawner: operator port
8801	SAS object spawner: pooled workspace server port bank 1
8811	SAS object spawner: pooled workspace server port bank 2
8821	SAS object spawner: pooled workspace server port bank 3
8591*	SAS Workspace Server port
8591*	Metadata utilities SAS Workspace Server port
8701	SAS Pooled Workspace Server port
8601	SAS Stored Process Server: bridge connection
8611	SAS Stored Process Server: load balancing connection 1 (MultiBridge)

Default Port Value	Description
8621	SAS Stored Process Server: load balancing connection 2 (MultiBridge)
8631	SAS Stored Process Server: load balancing connection 3 (MultiBridge)
5451	SAS OLAP Server port
8551	SAS/SHARE server port
7551	SAS/CONNECT server port
2171	SAS Table Server port
5091	SAS Remote Services application port
8451	Operating System Services scheduler port
10021	SAS Deployment Tester server port

\* In SAS 9.2, two or more workspace servers can share the same port even if they are running at the same time.

## Default Port Numbers for Third-Party Software

The following table shows the default port numbers for third-party software that is used with the SAS Intelligence Platform. The SAS Deployment Wizard allows different port numbers to be specified. Therefore, your site might use different port numbers than the ones that are shown here.

**Table A1.2** Default Port Numbers and Descriptions for Third-Party Software

Software	Description	Default Port Value
Oracle WebLogic Server	WebLogic NodeManager port	5556
	Non-secure listen port for managed servers	7001, 7101 (additional managed servers increment by 100)
	Secure listen port for managed servers	7002, 7102 (additional managed servers increment by 100)
	Listen port for administration server	7501

Software	Description	Default Port Value
IBM WebSphere Application Server	SOAP port for administrative console	8879
	SOAP port for application servers	8880, 8881 (additional application servers increment by 1)
	Secure HTTPS port for administrative console	9043
	Secure HTTPS ports for application server	9044, 9045 (additional application servers increment by 1)
	Non-secure HTTP port for administrative console	9060
	Non-secure HTTP ports for application server	9080, 9081 (additional application servers increment by 1)
	RMI port for administrative console	9809
JBoss Application Server	RMI ports for application servers	9811, 9812 (additional administrative consoles increment by 1)
	RMI port for managed servers	1099, 1199 (additional managed servers increment by 100)
	Listen port for managed servers	8080, 8180 (additional managed servers increment by 100)
Platform Suite for SAS	Secure listen port for managed servers	8443, 8543 (additional managed servers increment by 100)
	Platform Process Manager port*	1966
	Platform LSF ports	6878, 6881, 6882, 7869, 7870, 7871, 7872
	Platform Grid Management Service port	1976

\* For information about how to change this value, see the Platform Process Manager documentation.

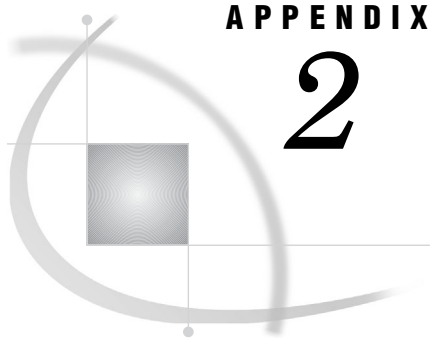
## Port Numbering in a Multiple-Level Environment

When you set up a multiple-level environment (for example, an environment that consists of separate levels for development, test, and production), the SAS Deployment Wizard increments each port number by 1 for each level. For example, a **Lev2** environment uses the port numbers that are shown in the following table.

**Table A1.3** Examples of Incremented Port Numbers in Level 2 of a Multiple-Level Environment

Default Port Value	Description
8562	SAS Metadata Server
8582	SAS object spawner: operator port
8802, 8812, 8822	SAS object spawner: pooled workspace server port banks
8592	SAS Workspace Server port

Default Port Value	Description
8592	Metadata utilities SAS Workspace Server port
8702	SAS Pooled Workspace Server port
8602	SAS Stored Process Server: bridge connection
8612, 8622, 8632	SAS Stored Process Server: load balancing connections
5452	SAS OLAP Server port
8552	SAS/SHARE server port
7552	SAS/CONNECT server port



## APPENDIX

## 2

## Reference: Configuration Directories

<i>Introduction to the Configuration Directory Reference</i>	<b>263</b>
<i>Contents of the Levn Subdirectory</i>	<b>263</b>
<i>Contents of the Metadata Server Subdirectory (SASMeta)</i>	<b>265</b>
<i>Contents of the Application Server Subdirectory (SASApp or SASMain)</i>	<b>266</b>
<i>Contents of the Web Subdirectory</i>	<b>267</b>

### Introduction to the Configuration Directory Reference

This appendix contains details about the contents of the main subdirectories of the SAS Intelligence Platform configuration directory, which exists on each server-tier and middle-tier host machine. The configuration directory, which is referred to in the documentation as *SAS-configuration-directory*, is located in the following path:

- on Windows systems:

*drive*: \SAS\configuration-directory-name

- on UNIX systems:

*installer's-home-directory*/SAS/configuration-directory-name

- on z/OS systems, the directory that is specified in the **CONFIG\_DIR** environment variable

In these paths, *configuration-directory-name* is the directory name that was specified when the deployment plan was created. If no name was specified, then the default name **Config** is used. During installation, the SAS Deployment Wizard enables you to override the configuration directory path with a different path.

For an overview of the configuration directory structure, see “Overview of the Configuration Directory Structure” on page 19.

### Contents of the Levn Subdirectory

The **Levn** subdirectory contains configuration information and other files for a particular installation instance. **Lev1** is generally used for production environments, and additional levels, such as **Lev2** and **Lev3**, can be used for environments that you install for purposes such as development and testing. During installation, the SAS Deployment Wizard enables you to specify the level number.

Depending on what software is installed on the machine, the **Levn** directory can contain the subdirectories and files that are listed in the following table:

**Table A2.1** Contents of the Levn Subdirectory

<b>Subdirectory or File</b>	<b>Description</b>
<b>AppData</b>	Contains data sets for the SAS Content Server repository and data that is installed for the use of specific applications (for example, SAS BI Dashboard).
<b>ConnectSpawner</b>	Contains the management script, configuration files, and logs for the SAS/CONNECT spawner.
<b>Data</b>	Can be used to store user data.
<b>DeploymentTesterServer</b>	Contains files that are used by the Deployment Tester plug-in to SAS Management Console.
<b>Documents</b>	Contains Instructions.html, which contains post-installation configuration instructions; ConfigurationErrors.html; and other application-specific documents.
<b>Logs</b>	Can be used as a common directory for server and spawner logs, if you selected this option during a custom installation. By default, each server has its own separate log directory (see Appendix 3, “Default Locations for Server Logs,” on page 269).
<b>Logs/Configure</b>	Contains logs that are created by the SAS Deployment Wizard.
<b>ObjectSpawner</b>	Contains a management script, configuration files, and logs for the object spawner.
<b>SASApp</b>	For new installations and for multiple-machine installations that were migrated from SAS 9.13, contains management scripts, configuration files, and logs for SAS Application Server components. See “Contents of the Application Server Subdirectory (SASApp or SASMain)” on page 266.
<b>SASMeta</b>	Contains management scripts, configuration files, metadata repositories, logs, and other files for the SAS Metadata Server; and management scripts, configuration files, and logs for components of the metadata utility SAS Application Server. See “Contents of the Metadata Server Subdirectory (SASMeta)” on page 265.
<b>SASMain</b>	For single-machine installations that were migrated from SAS 9.1.3, contains management scripts, configuration files, metadata repositories, logs, and other files for SAS Application Server components. See “Contents of the Metadata Server Subdirectory (SASMeta)” on page 265 and “Contents of the Application Server Subdirectory (SASApp or SASMain)” on page 266.
<b>ShareServer</b>	Contains the management script, configuration information, and log files for the SAS/SHARE server.
<b>SASTS</b>	Contains the management script, configuration information, and log files for the SAS Table Server.
<b>Utilities</b>	Contains XML files that are used as input to the SAS Deployment Wizard and utilities that are associated with this configuration instance.
<b>Web</b>	See “Contents of the Web Subdirectory” on page 267.

Subdirectory or File	Description
generate_boot_scripts.sh	Is a script that is used to regenerate the sas.servers script on UNIX and z/OS. See “Using generate_boot_scripts.sh to Regenerate the sas.servers Script” on page 59.
sas.servers	Is a management script that is used on UNIX and z/OS to start, stop, or restart all servers on the machine in the correct order, or to display the status of all servers on the machine.
sasv9_meta.cfg	Specifies metadata server connection information for the SAS OLAP Server, SAS/SHARE server, and SAS Table Server.

## Contents of the Metadata Server Subdirectory (SASMeta)

On the machines where the metadata server is installed, the **Levn** directory includes a subdirectory for the metadata server’s files and for components of the metadata utility SAS Application Server. **SASMeta** is the name that the SAS Deployment Wizard assigns to the metadata server context and, in turn, to this directory.

The **SASMeta** directory contains the following items:

- a **WorkspaceServer** subdirectory, which contains start-up scripts, configuration files, a logconfig.xml file, and a **Logs** directory for the SASMeta workspace server. This server is used by metadata utilities such as the Backup and Restore Wizard.
- a **BatchServer** subdirectory, which contains configuration files, a logconfig.xml file, and a **Logs** directory for the SASMeta DATA step batch server. This server is used to run scheduled metadata utility jobs, such as jobs created by the Backup Wizard.
- **SASEnvironment** and **Data** subdirectories that can be used by the workspace server.
- several files that control the configuration of the SASMeta application server. For details, see “Reference: Configuration Files for SAS Servers” on page 271.
- a **MetadataServer** subdirectory.

The following table lists the contents of the **MetadataServer** subdirectory. For more information about the metadata server configuration files, see “Configuration Files for the Metadata Server” on page 272.

Contents of the MetadataServer Subdirectory of SASMeta

Subdirectory or File	Description
<b>Journal</b>	Contains the data set for the metadata server journal. For details, see “Configuring Metadata Server Journaling” on page 170.
<b>Logs</b>	Is the default location for metadata server logs.
<b>MetadataRepositories</b>	Contains subdirectories for each metadata repository. These subdirectories contain the SAS data sets that contain your metadata.
<b>rposmgr</b>	Contains data sets for the repository manager.
<b>sasuser</b>	Is a SAS library that can be used to store SAS catalogs that enable you to tailor certain features of SAS to meet specific needs. SAS assigns the SASUSER library at invocation.

Subdirectory or File	Description
adminUsers.txt, trustedUsers.txt, and trustedPeers.xml	Contain security-related configuration information (see “About the Security-Related Configuration Files” on page 23).
backupServer.sas and restoreServer.sas	Are programs that are invoked by MetadataServer.bat or MetadataServer.sh when the backup and restore options are used. These programs use the %OMABAKUP macro to pause the metadata server and copy your metadata server files. For details, see “Running the backupServer.sas and restoreServer.sas Programs” on page 150.
logconfig.xml	Specifies the logging configuration for the metadata server. For details, see Chapter 8, “Administering Logging for SAS Servers,” on page 103.
MetadataServer.bat or MetadataServer.sh,	Is a script that is used to stop, start, pause, resume, or report the status of the metadata server.
MetadataServer.srv	On UNIX and z/OS, contains information that is used by generate_boot_scripts.sh to regenerate the sas.servers script.
MetadataServerSSCU.ini	Configures the metadata server service in the Windows Services Manager.
MetadataServer_shortcuts.ini	Configures shortcuts for the Windows Start menu.
metaparms.sas	Contains user credentials that can be used to execute SAS sessions that include PROC METAOPERATE, PROC METADATA, or &OMABAKUP.
omaconfig.xml	Is the SAS Metadata Server configuration file.
pauseServer.sas, resumeServer.sas, statusServer.sas, and stopServer.sas	Are programs that are invoked by MetadataServer.bat or MetadataServer.sh when the pause, resume, status, and stop options are used.
sasv9.cfg	Specifies SAS options that are put into effect when the metadata server is started.
sasv9_usermods.cfg	Contains user modifications to the SAS options that are specified in sasv9.cfg.

## Contents of the Application Server Subdirectory (SASApp or SASMain)

On machines where components of a SAS Application Server are installed, the **Levn** directory contains an application server subdirectory. A SAS Application Server is a logical entity that contains specific servers (for example, a SAS Workspace Server and a SAS Stored Process Server) that execute SAS code. Application server subdirectories are named as follows:

- **SASApp** is the default name that the SAS Deployment Wizard assigns to directory for the first application server. If you assigned a different name when you ran the SAS Deployment Wizard, then the subdirectory is named accordingly.
- If you configure more than one application server, then additional directories are created that correspond to each of the application server names.



- If you have a single-machine installation that was migrated from SAS 9.1.3, then the application server subdirectory is called **SASMain**.

*Note:* The machine where the metadata server is installed includes an additional SAS Application Server called **SASMeta** that is used to run metadata utilities. See “Contents of the Metadata Server Subdirectory (SASMeta)” on page 265. △

The **SASApp** or **SASMain** subdirectory contains the following items:

- several files that control the application server configuration. For details, see “Reference: Configuration Files for SAS Servers” on page 271.
- some or all of the following subdirectories, depending on which SAS servers have been configured.
  - **BatchServer**
  - **ConnectServer**
  - **GridServer**
  - **OLAPServer**
  - **PooledWorkspaceServer**
  - **StoredProcessServer**
  - **WorkspaceServer**

These subdirectories contain start-up scripts, configuration files, logconfig.xml files to configure logging, and **Logs** directories to store logs. For details about the configuration files, see “Reference: Configuration Files for SAS Servers” on page 271. For details about start-up scripts, see Chapter 5, “Operating Your Servers,” on page 51.

- a **Data** subdirectory, which you can use to store data to be used by components of this application server.
- a **SASEnvironment** subdirectory for storing the following elements of the run-time environment for SAS code that runs on the application server:
  - a **SASCode** subdirectory, which you can use as the deployment directory for SAS Data Integration Studio jobs that are deployed for scheduling. This directory contains a **Jobs** subdirectory to store jobs and a **Steps** subdirectory to store the SAS code for each job step. These directories enable source code to be shared among developers on a project.
  - a **SASFormats** subdirectory where you can store SAS format and informat catalogs for use by applications, such as SAS Data Integration Studio and SAS Information Map Studio, that execute code on the application server.
  - a **SASMacro** subdirectory, where you can store user-created SAS autocall macros that are invoked by SAS code running on the application server.

---

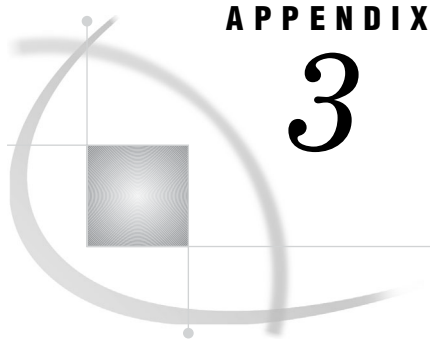
## Contents of the Web Subdirectory

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

- an **Applications** subdirectory, which contains a subdirectory for each of the Web applications and other middle-tier applications that are installed on the machine. These directories contain configuration files, policy files, and other support files for the applications.
- a **Common** subdirectory, which contains configuration files for services that are shared by Web applications.
- a **Logs** subdirectory, which contains logs for Web applications.

- a **Staging** subdirectory, which contains the enterprise archive (EAR) files for your Web applications.
- a **Utilities** subdirectory, which contains utilities related to installed Web applications.

For information about administering middle-tier applications and components, see the *SAS Intelligence Platform: Web Application Administration Guide*.



## APPENDIX

## 3

## Default Locations for Server Logs

*Default Locations for Server Logs* 269

### Default Locations for Server Logs

The following table shows the default location of server logs within the *SAS-configuration-directory*:

*Note:*

- **SASApp** is the default name that the SAS Deployment Wizard assigns to directory for the first application server. If you have a single-machine installation that was migrated from SAS 9.1.3, then the application server subdirectory is called **SASMain**. If you assigned a different name when you ran the SAS Deployment Wizard, then the subdirectory is named accordingly.
- In a custom installation, you can choose to place all of your log files in a single directory. If this option was selected, then your log files are written to the path that was specified during installation (for example, *SAS-configuration-directory\Lev1\Logs*).

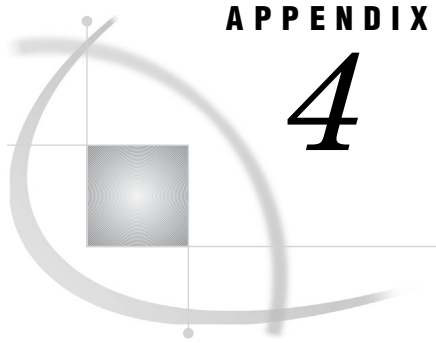
△

**Table A3.1** Location of Server Logs

Server	Log Location within the <i>SAS-configuration-directory</i>
SAS Metadata Server	Windows: <b>Lev1\SASMeta\MetadataServer\Logs</b> UNIX or z/OS: <b>Lev1/SASMeta/MetadataServer/Logs</b>
SAS object spawner	Windows: <b>Lev1\ObjectSpawner\Logs</b> UNIX or z/OS: <b>Lev1/ObjectSpawner/Logs</b>
SAS OLAP Server	Windows: <b>Lev1\SASApp\OLAPServer\Logs</b> UNIX or z/OS: <b>Lev1/SASApp/OLAPServer/Logs</b>
SAS Pooled Workspace Server	Windows: <b>Lev1\SASApp\PooledWorkspaceServer\Logs</b> UNIX or z/OS: <b>Lev1/SASApp/PooledWorkspaceServer/Logs</b>
SAS Stored Process Server	Windows: <b>Lev1\SASApp\StoredProcessServer\Logs</b> UNIX or z/OS: <b>Lev1/SASApp/StoredProcessServer/Logs</b>
SAS Table Server	Windows: <b>Lev1\SASTS\TableServer\Logs</b> UNIX or z/OS: <b>Lev1/SASTS/TableServer/Logs</b>

<b>Server</b>	<b>Log Location within the <i>SAS-configuration-directory</i></b>
SAS Workspace Server	For this server, the initial logging configuration sends log events to the operating system facility rather than to a log file.
SAS/SHARE server	Windows: <b>Lev1\ShareServer\Logs</b> UNIX or z/OS: <b>Lev1/ShareServer/Logs</b>
SAS/CONNECT spawner	Windows: <b>Lev1\ConnectSpawner\Logs</b> UNIX or z/OS: <b>Lev1/ConnectSpawner/Logs</b>
SAS/CONNECT server	For this server, the initial logging configuration sends log events to the operating system facility rather than to a log file.
SAS DATA step batch server	Windows: <b>Lev1\SASApp\BatchServer\Logs</b> UNIX or z/OS: <b>Lev1/SASApp/BatchServer/Logs</b>

For more information about server logging, see Chapter 8, “Administering Logging for SAS Servers,” on page 103.



## APPENDIX

## 4

## Configuration Files

---

<i>Reference: Configuration Files for SAS Servers</i>	<b>271</b>
<i>Introduction to the Configuration File Reference</i>	<b>271</b>
<i>Configuration Files for the Metadata Server</i>	<b>272</b>
<i>Configuration Files for Components of SAS Application Servers</i>	<b>274</b>
<i>Configuration Files for SAS/SHARE Servers and SAS Table Servers</i>	<b>277</b>
<i>Configuration Files for SAS Object Spawners and SAS/CONNECT Spawners</i>	<b>279</b>
<i>Reference Information for omaconfig.xml</i>	<b>280</b>
<i>About the omaconfig.xml File</i>	<b>280</b>
<OMA> Options	<b>281</b>
<RPOSMGR> Options	<b>286</b>
<InternalAuthenticationPolicy> Options	<b>287</b>

---

## Reference: Configuration Files for SAS Servers

---

### Introduction to the Configuration File Reference

This section provides tables that show the name, location, and purpose of the various server configuration files that are used in the SAS Intelligence Platform. Each table in this appendix contains the following columns:

#### Order of Precedence

The numbers in this column indicate the order in which the options in the files are applied. Option settings in files with higher orders of precedence override the settings in files with lower orders of precedence.

“Not applicable” means that the option settings in the file do not affect, and are not affected by, option settings that are specified in other files.

#### Path and Filename

All paths that begin with **Lev1** are located in the SAS configuration directory (see “About the Configuration Directory” on page 19). Within the pathnames, the following variables are used:

- *server-context*, which indicates the name of an application server grouping (for example, **SASApp**, **SASMeta**, or **SASMain**)
- *server-name*, which indicates the name of the logical server or server that the configuration file applies to

#### Purpose

This column describes the purpose of the configuration file, indicates that certain configuration files are not to be modified by users, and provides references to detailed documentation.

## Configuration Files for the Metadata Server

The following table shows the configuration files that affect the metadata server. For an explanation of each columns' contents, see "Introduction to the Configuration File Reference" on page 271.

Configuration Files for the SAS Metadata Server

Order of Precedence	Path and Filename	Purpose
1	Windows: \Lev1\SASMeta\ <b>MetadataServer\sasv9.cfg</b> UNIX and z/OS: /Lev1/SASMeta/ <b>MetadataServer/sasv9.cfg</b>	Specifies server startup options for the metadata server and contains calls to the other files that are listed in the table. Do not modify this file. If you need to make changes, modify the sasv9_usermods.cfg file that is in the same directory.
2	Windows: SAS-install-directory\ <b>SASFoundation\9.2\sasv9.cfg</b> UNIX: SAS-install-directory/ <b>SAS-install-directory/ SASFoundation/9.2/sasv9.cfg</b> z/OS: Not applicable	Specifies system options that apply to all SAS sessions on the host machine.
3	Windows: SAS-install-directory\ <b>SASFoundation\9.2\locale\sasv9.cfg</b> UNIX: SAS-install-directory/ <b>SASFoundation/9.2/locale/sasv9.cfg</b> z/OS: <HLQ>.V92D8ddd.CNTLDSN (BATlocaleW0)	Specifies system options that apply to all SAS sessions on the host machine in the specified locale.
4	Windows: \Lev1\SASMeta\ <b>MetadataServer\sasv9_usermods.cfg</b> UNIX and z/OS: /Lev1/SASMeta/ <b>MetadataServer/sasv9_usermods.cfg</b>	Contains site-specific modifications to the metadata server start-up options. Use this file to change the options that are specified in the sasv9.cfg file in the same directory. See Appendix 5, "Options for the Metadata Server Invocation Command," on page 289.
Not applicable	Windows: \Lev1\SASMeta\ <b>MetadataServer\omacnfig.xml</b> UNIX and z/OS: /Lev1/SASMeta/ <b>MetadataServer/omacnfig.xml</b>	Specifies options for the metadata server and metadata repositories. See "Reference Information for omacnfig.xml" on page 280.
Not applicable	Windows: \Lev1\SASMeta\ <b>MetadataServer\logcnfig.xml</b> UNIX and z/OS: /Lev1/SASMeta/ <b>MetadataServer/logcnfig.xml</b>	Specifies the logging configuration for the server. See Chapter 8, "Administering Logging for SAS Servers," on page 103.

Order of Precedence	Path and Filename	Purpose
Not applicable	Windows: \Lev1\SASMeta\ <b>MetadataServer\adminUsers.txt</b> UNIX and z/OS: /Lev1/SASMeta/ <b>MetadataServer/adminUsers.txt</b>	Defines your system's initial <i>unrestricted user</i> . See “Metadata Server: Unrestricted Role” on page 24.
Not applicable	Windows: \Lev1\SASMeta\ <b>MetadataServer\trustedUsers.txt</b> UNIX and z/OS: /Lev1/SASMeta/ <b>MetadataServer/trustedUsers.txt</b>	Conveys privileged status to a service account to enable that account to act on behalf of other users. See “About the Initial User Accounts” on page 27.
Not applicable	Windows: \Lev1\SASMeta\ <b>MetadataServer\trustedPeers.xml</b> UNIX and z/OS: /Lev1/SASMeta/ <b>MetadataServer/trustedPeers.xml</b>	Can be used to limit the scope of trust for connections from SAS sessions to the metadata server. Initially, the scope is not limited. See the <i>SAS Intelligence Platform: Security Administration Guide</i> .
Not applicable	Windows: Not applicable UNIX and z/OS: /Lev1/SASMeta/ <b>MetadataServer/MetadataServer.srv</b>	On UNIX and z/OS, contains information that is used by <code>generate_boot_scripts.sh</code> to regenerate the <code>sas.servers</code> script.

In z/OS environments, the following additional configuration files affect the metadata server:

Additional Configuration Files for the SAS Metadata Server on z/OS Systems

Order of Precedence	Path and Filename	Purpose
1	z/OS: <HLQ>. <b>TKMVSENV</b> (TKMVSENV)	Contains name/value pairs (environment variables) that apply to all SAS sessions on the machine.
2	z/OS: /Lev1/SASMeta/MetadataServer/ <b>tkmvseenv.cfg</b>	Contains name/value pairs (environment variables) that apply to the metadata server. Do not modify this file. If you need to make changes, modify the <code>tkmvseenv_usermods.cfg</code> file that is in the same directory.
3	z/OS: /Lev1/SASMeta/MetadataServer/ <b>tkmvseenv_usermods.cfg</b>	Contains site-specific modifications to the name/value pairs (environment variables) for the metadata server. Use this file to change the options that are specified in the <code>tkmvseenv.cfg</code> file in the same directory. See the Configuration Guide for SAS 9.2 Foundation for z/OS.

## Configuration Files for Components of SAS Application Servers

The following table shows the configuration files that affect servers that are components of a SAS Application Server. These servers include OLAP servers, workspace servers, pooled workspace servers, stored process servers, and SAS/CONNECT servers. For an explanation of each columns' contents, see "Introduction to the Configuration File Reference" on page 271.

Configuration Files for Components of SAS Application Servers (OLAP Servers, Workspace Servers, Stored Process Servers, and SAS/CONNECT Servers)

Order of Precedence	Path and Filename	Purpose
1	Windows: <code>\Lev1\server-context\server-name\sasv9.cfg</code> UNIX and z/OS: <code>/Lev1/server-context/server-name\sasv9.cfg</code>	Specifies start-up options for the server and contains calls to the other files that are listed in this table. Do not modify this file. If you need to make changes, modify the <code>sasv9_usermods.cfg</code> file that is in the same directory.
2	Windows: <code>\Lev1\server-context\sasv9.cfg</code> UNIX and z/OS: <code>/Lev1/server-context/sasv9.cfg</code>	Specifies start-up options that apply to all components of the application server. Do not modify this file. If you need to make changes, modify the <code>sasv9_usermods.cfg</code> file that is in the same directory. See "System Options for SAS Application Server Components" in the <i>SAS Intelligence Platform: Application Server Administration Guide</i> .
3	Windows: <code>SAS-install-directory\SASFoundation\9.2\sasv9.cfg</code> UNIX: <code>SAS-install-directory/SASFoundation/9.2/sasv9.cfg</code> z/OS: Not applicable	Specifies system options that apply to all SAS sessions on the host machine.
4	Windows: <code>SAS-install-directory\SASFoundation\9.2\locale\sasv9.cfg</code> UNIX: <code>SAS-install-directory/SASFoundation/9.2/locale/sasv9.cfg</code> z/OS: <code>&lt;HLQ&gt;.V92D8ddd.CNTLDSN (BATlocaleW0)</code>	Specifies system options that apply to all SAS sessions on the host machine in the specified locale.



Order of Precedence	Path and Filename	Purpose
5	Windows: \Lev1\server-context\sasv9_usermods.cfg UNIX and z/OS: /Lev1/server-context/sasv9_usermods.cfg	Contains site-specific modifications to the server-startup options that apply to all application server components. Use this file to change the options that are specified in the sasv9.cfg file in the same directory.  See “System Options for Application Servers” in the <i>SAS Intelligence Platform: Application Server Administration Guide</i> .
6	Windows: \Lev1\server-context\autoexec.sas UNIX and z/OS: /Lev1/server-context/autoexec.sas	Contains SAS statements that are executed immediately after SAS initializes on all components of the application server. Do not modify this file. If you need to make changes, modify the autoexec_usermods.sas file that is in the same directory.
7	Windows: \Lev1\server-context\autoexec_usermods.sas UNIX and z/OS: /Lev1/server-context/autoexec_usermods.sas	Contains site-specific modifications to the autoexec.sas file that is located in the same directory.
8	Windows: \Lev1\server-context\server-name\sasv9_usermods.cfg UNIX and z/OS: /Lev1/server-context/server-name/sasv9_usermods.cfg	Contains site-specific modifications to the server’s startup options. Use this file to change the options that are specified in the sasv9.cfg file in the same directory. See “System Options for SAS Application Server Components” in the <i>SAS Intelligence Platform: Application Server Administration Guide</i> .
9	Windows: \Lev1\server-context\server-name\autoexec.sas UNIX and z/OS: /Lev1/server-context/server-name/autoexec.sas	Contains SAS statements that are executed immediately after SAS initializes on the server. Do not modify this file. If you need to make changes, modify the autoexec_usermods.sas file that is in the same directory.
10	Windows: \Lev1\server-context\server-name\autoexec_usermods.sas UNIX and z/OS: /Lev1/server-context/server-name/autoexec_usermods.sas	Contains site-specific modifications to the autoexec.sas file that is located in the same directory.
Not applicable	Windows: \Lev1\server-context\server-name\logconfig.xml UNIX and z/OS: /Lev1/server-context/server-name/logconfig.xml	Specifies the logging configuration for the server. See Chapter 8, “Administering Logging for SAS Servers,” on page 103.

Order of Precedence	Path and Filename	Purpose
Not applicable	Windows: <code>\Lev1\sasv9_meta.cfg</code> UNIX and z/OS: <code>/Lev1/sasv9_meta.cfg</code>	Specifies metadata server connection information for the SAS OLAP Server, SAS/SHARE server, and SAS Table Server.
Not applicable	Windows: Not applicable UNIX and z/OS: <code>/Lev1/server-context/server-name/server-name.srv</code>	On UNIX and z/OS, contains information that is used by <code>generate_boot_scripts.sh</code> to regenerate the <code>sas.servers</code> script.

In z/OS environments, the following additional configuration files affect components of a SAS Application Server:

Additional Configuration Files for Components of SAS Application Servers (OLAP Servers, Workspace Servers, Stored Process Servers, and SAS/CONNECT Servers) on z/OS Systems

Order of Precedence	Path and Filename	Purpose
1	z/OS: <code>&lt;HLQ&gt;.TKMVSENV(TKMVSENV)</code>	Contains name/value pairs (environment variables) that apply to all SAS sessions on the machine.
2	z/OS: <code>/Lev1/server-context/server-name/tkmvsenv.cfg</code>	Contains name/value pairs (environment variables) that apply to this server. Do not modify this file. If you need to make changes, modify the <code>tkmvsenv_usermods.cfg</code> file that is in the same directory.
3	z/OS: <code>/Lev1/server-context/tkmvsenv.cfg</code>	Contains name/value pairs (environment variables) that apply to all components of the application server. Do not modify this file. If you need to make changes, modify the <code>tkmvsenv_usermods.cfg</code> file that is in the same directory.

Order of Precedence	Path and Filename	Purpose
4	z/OS: <b>/Lev1/server-context/ tkmvseenv_usermods.cfg</b>	Contains site-specific name/value pairs (environment variables) that apply to all application server components. Use this file to change the options that are specified in the tkmvseenv.cfg file in the same directory. See the Configuration Guide for SAS 9.2 Foundation for z/OS.
5	z/OS: <b>/Lev1/server-context/server-name/ tkmvseenv_usermods.cfg</b>	Contains site-specific modifications to the server's name/value pairs (environment variables). Use this file to change the options that are specified in the tkmvseenv.cfg file in the same directory. See the Configuration Guide for SAS 9.2 Foundation for z/OS.

## Configuration Files for SAS/SHARE Servers and SAS Table Servers

The following table shows the configuration files that affect SAS/SHARE servers and SAS Table Servers. For an explanation of each column's contents, see "Introduction to the Configuration File Reference" on page 271.

Configuration Files for SAS/SHARE servers and SAS Table Servers

Order of Precedence	Path and Filename	Purpose
1	Windows: <b>\Lev1\server-name\sasv9.cfg</b> UNIX and z/OS: <b>/Lev1/server-name/ sasv9.cfg</b>	Specifies start-up options for the server and contains calls to the other files that are listed in this table. Do not modify this file. If you need to make changes, modify the sasv9_usermods.cfg file that is in the same directory.
2	Windows: <b>\SAS-install-directory\ SASFoundation\9.2\sasv9.cfg</b> UNIX: <b>/SAS-install-directory/ SASFoundation/9.2/sasv9.cfg</b> z/OS: Not applicable	Specifies system options that apply to all SAS sessions on the host machine.
3	Windows: <b>\SAS-install-directory\ SASFoundation\9.2\locale\sasv9.cfg</b> UNIX: <b>/SAS-install-directory/ SASFoundation/9.2/locale/sasv9.cfg</b> z/OS: <b>&lt;HLQ&gt;.V92D8ddd.CNTLDSN (BAT\localeW0)</b>	Specifies system options that apply to all SAS sessions on the host machine in the specified locale.

Order of Precedence	Path and Filename	Purpose
4	Windows: <code>\Lev1\server-name\sasv9_usermods.cfg</code> UNIX and z/OS: <code>/Lev1/server-name/sasv9_usermods.cfg</code>	Contains site-specific modifications to the server's startup options. Use this file to change the options that are specified in the <code>sasv9.cfg</code> file in the same directory. See the <i>SAS/SHARE User's Guide</i> .
5	Windows: <code>\Lev1\server-name\autoexec.sas</code> UNIX and z/OS: <code>/Lev1/server-name/autoexec.sas</code>	Contains SAS statements that are executed immediately after SAS initializes on the server. Do not modify this file. If you need to make changes, modify the <code>autoexec_usermods.sas</code> file that is in the same directory.
6	Windows: <code>\Lev1\server-name\autoexec_usermods.sas</code> UNIX and z/OS: <code>/Lev1/server-name/autoexec_usermods.sas</code>	Contains site-specific modifications to the <code>autoexec.sas</code> file that is located in the same directory.
Not applicable	Windows: <code>\Lev1\sasv9_meta.cfg</code> UNIX and z/OS: <code>/Lev1/sasv9_meta.cfg</code>	Specifies metadata server connection information for the SAS OLAP Server, SAS/SHARE server, and SAS Table Server.
Not applicable	Windows: <code>\Lev1\server-name\logconfig.xml</code> UNIX and z/OS: <code>/Lev1/server-name/logconfig.xml</code>	Specifies the logging configuration for the server. See Chapter 8, "Administering Logging for SAS Servers," on page 103.
Not applicable	Windows: Not applicable UNIX and z/OS: <code>/Lev1/server-name/server-name.srv</code>	On UNIX and z/OS, contains information that is used by <code>generate_boot_scripts.sh</code> to regenerate the <code>sas.servers</code> script.

In z/OS environments, the following additional configuration files affect SAS/SHARE servers and SAS Table Servers:

Additional Configuration Files for SAS/SHARE servers and SAS Table Servers on z/OS Systems

Order of Precedence	Path and Filename	Purpose
1	z/OS: <HLQ>.TKMVSENV(TKMVSENV)	Contains name/value pairs (environment variables) that apply to all SAS sessions on the machine.
2	z/OS: /Lev1/server-name/tkmvsenv.cfg	Contains name/value pairs (environment variables) for the server. Do not modify this file. If you need to make changes, modify the tkmvsenv_usermods.cfg file that is in the same directory.
3	z/OS: /Lev1/server-name/tkmvsenv_usermods.cfg	Contains site-specific modifications to the server's name/value pairs (environment variables). Use this file to change the options that are specified in the tkmvsenv.cfg file in the same directory. See the Configuration Guide for SAS 9.2 Foundation for z/OS.

## Configuration Files for SAS Object Spawners and SAS/CONNECT Spawners

The following table shows the configuration files that affect SAS object spawners and SAS/CONNECT spawners. For an explanation of each columns' contents, see "Introduction to the Configuration File Reference" on page 271.

Configuration Files for the SAS Object Spawner

Order of Precedence	Path and Filename	Purpose
Not applicable	Windows: \Lev1\spawner-name\metaConfig.xml UNIX and z/OS: /Lev1/spawner-name/metaConfig.xml	Specifies metadata server connection information. The spawner must be stopped and restarted to pick up changes to this file.
Not applicable	Windows: \Lev1\ConnectSpawner\ConnectSpawner.bat UNIX: /Lev1/ConnectSpawner/ConnectSpawner.sh z/OS: Not applicable	Contains start-up options for the SAS/CONNECT spawner. See <i>Communications Access Methods for SAS/CONNECT and SAS/SHARE</i> .
Not applicable	Windows: \Lev1\ObjectSpawner\ObjectSpawner.bat UNIX: /Lev1/ObjectSpawner/ObjectSpawner.sh z/OS: Not applicable	Contains start-up options for the object spawner. See "Spawner Invocation Options" in the <i>SAS Intelligence Platform: Application Server Administration Guide</i> .

Order of Precedence	Path and Filename	Purpose
Not applicable	Windows: <code>\Lev1\spawner-name\logconfig.xml</code> UNIX and z/OS: <code>/Lev1/spawner-name/logconfig.xml</code>	Specifies the logging configuration for the spawner. See Chapter 8, “Administering Logging for SAS Servers,” on page 103.
Not applicable	Windows: Not applicable UNIX and z/OS: <code>/Lev1/server-name/server-name.srv</code>	On UNIX and z/OS, contains information that is used by <code>generate_boot_scripts.sh</code> to regenerate the <code>sas.servers</code> script.

## Reference Information for omaconfig.xml

### About the omaconfig.xml File

The omaconfig.xml file specifies changes to standard features of the SAS Metadata Server, the repository manager, and policies related to internal users. The 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)
- journaling options
- the location, the libref, and the engine used by the repository manager
- policies regarding password assignment and management for internal users

For options that are omitted from the omaconfig.xml file, the server uses the option’s default value.

Your omaconfig.xml file might include an option that activates 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 you are instructed to do so by SAS Technical Support.

The omaconfig.xml file is located in the following path:  
*sas-configuration-directory*\Lev1\SASMeta\MetadataServer.

When you change any option in omaconfig.xml, you must stop and restart the metadata server for the change to take effect. For instructions, see Chapter 5, “Operating Your Servers,” on page 51.

#### CAUTION:

**Stopping the metadata server stops other components.** △

The file can contain the following XML elements:

- <OMA>  
specifies options for the metadata server.
- <RPOSMGR>  
specifies options for the repository manager.

## &lt;InternalAuthenticationPolicy&gt;

specifies options for the assignment and management of passwords for internal users. Internal accounts are accounts that exist only in metadata (see “Internal Accounts” in the “Authentication Mechanisms” chapter in the *SAS Intelligence Platform: Security Administration Guide*).

The following topics provide details for each omaconfig.xml option.

---

## <OMA> Options

The <OMA> element specifies the following options for the metadata server:

**ADMINUSERS**="*path-and-filename*"

specifies the name and location of the adminUsers.txt file. A planned installation creates the file in  
*sas-configuration-directory\Lev1\SASMeta\MetadataServer\adminUsers.txt*.

**ALERTEMAIL**="*email-address*"

specifies an e-mail address to which the metadata server will send a notification message when an error occurs that prevents the disk update process for the journal from completing. To specify more than one address, enclose each address with single quotation marks, place a blank space between each address, and enclose the list with parentheses. For example:

```
"('Bill@mycompany.com' 'Susan@mycompany.com')"
```

*Note:* For the ALERTEMAIL, ALERTEMAILATTACHMENT, and ALERTEMAILTEST options to work, the appropriate mail server connection properties (EMAILSYS, EMAILHOST, EMAILPORT, and EMAILID) must be specified in the metadata server's sasv9.cfg or sasv9\_usermods.cfg file. See Appendix 5, “Options for the Metadata Server Invocation Command,” on page 289. △

**ALERTEMAILATTACHMENT**="*path-and-filename*"

specifies the relative path and filename of an existing file to attach to the ALERTEMAIL message. The path should be relative to the directory from which the metadata server is started. For example:

- To attach the metadata server's omaconfig.xml file, which is located in the same directory as the script that starts the server, specify just **omaconfig.xml**.
- to attach the metadata server's sasv9.cfg file, which is located in the same directory as the script that starts the server, specify just **sasv9.cfg**.
- to attach the sasv9.cfg file from the **SASMeta** directory, specify **..\sasv9.cfg**.

**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 that is needed by SAS Information Delivery Portal. This 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.

JNLPRE="1 | Y | T | 0 | N | F"

specifies whether a client identifier is to be included in the preamble of log messages for the metadata server.

If you specify **1**, **Y**, or **T**, then it is not necessary to include the characters **%u** in the conversion pattern for the metadata server log. Doing so would cause the client identifier to appear twice in log messages.

**Default:** 0

*Note:* The default setting is **0** because the default logging configuration for the metadata server includes a conversion pattern with the characters **%u**, which provides the client identifier. △

JOURNALPATH="*path-and-filename*"

specifies a location to create the journal file.

When specifying a location, keep these things in mind:

- For consistency, we recommend that you specify the filename **MetadataJournal.dat**.
- For convenience and ease of management, the standard metadata server configuration creates repositories and the metadata server log on the same file system in which the 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 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.

To change the location of the journal file, follow the detailed instructions in “Move the Journal File to Another File System” on page 171.

- Put the journal file on a native file system, such as the C: (system) drive. The location should be fast and highly reliable. Do not place the file on a network-mounted file system or network appliance.
- If you use the same file system in which the metadata server is started, 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.



If you want to disable journaling, then specify a blank string ("" ) for this option.

*Note:* For optimum performance and to ensure system integrity in the event of a failure, it is strongly recommended that journaling be enabled at all times. △

**Default:** If you do not specify this option, then the journal is created in  
*SAS-configuration-directory\Lev1\SASMeta\MetadataServer\Journal\*  
**MetadataJournal.dat.**

**JOURNALSIZE**="*number-of-bytes*"

specifies a size, in bytes, for the journal file. This space is reserved in the file system and is unavailable to other processes, whether the journal file is full or not.

The size that you assign to the journal file depends on the total disk space available on the SAS Metadata Server host machine. On 32-bit machines and z/OS machine, the default size is 200MB. On 64-bit machines, 500MB is the default setting. The default settings should be sufficient 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 JOURNALSIZE value:

WARNING: insufficient file space available in journal file.

When an entry is added to the 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 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 journal file are safely committed, the first new update is written directly to repositories on disk. Then, journaling resumes.

**Default:** 200MB on 32-bit machines and z/OS machines, and 500MB on 64-bit machines

**JOURNALYIELDCOUNT**="*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 JOURNALYIELDCOUNT 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. JOURNALYIELDCOUNT 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 JOURNALYIELDCOUNT 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.

**MAXACTIVETHREADS**="*number-of-threads*"

specifies the number of threads that are allowed to run concurrently on the metadata server. The number of processors determines the number of concurrent queries that can be made. If the server has only one processor, then the recommended setting is 2. If the server has more than one processor, then the

recommended setting is the number of processors. A setting that is higher or lower than the recommended one can affect performance.

For information about how the metadata server uses threads, and recommendations for setting the MAXACTIVETHREADS option and the related THREADSMIN and THREADSMAX server invocation options, see “Configuring the Number of Threads Used by the Metadata Server” on page 167.

**Default:** By default, the server sets the maximum number of active threads as follows:

- If the host machine has one processor, then the maximum active number of threads is set to 2.
- If the host machine has two or more processors, then the maximum active number of threads is set to the number of processors.

**MAXIMUM\_QUERY\_RETRY\_COUNT**="*number-of-retries*"

specifies the maximum number of times that a metadata query request involving multiple repositories can be restarted because of missing repository locks. If the maximum number of retries (also referred to as the “query retry limit”) is reached before the query request is completed, a repository lock error is returned.

When the metadata server receives a query request, the server does the following to process the request:

- 1 obtains a shared query lock for each repository that is specified in the request.
- 2 as the request is prepared, verifies that a shared query lock has been obtained for each repository that contains records that are needed to complete the request.
- 3 immediately attempts to obtain any shared query locks that were not obtained initially.
- 4 restarts the request if the missing locks are not available. Each restart constitutes a retry.
- 5 repeats the process until the request is complete or until the query retry limit is reached. If the query request cannot be completed within the query retry limit, a repository lock error is returned.

Activity that is related to obtaining repository query locks is recorded in the Metadata.CounterQueryUpdateLock counters. See “Using the Server Performance Counters and Information Fields” on page 96.

**Default:** 7 or the number of repositories that were configured when the metadata server was last started, whichever is higher

**MAXIMUM\_RETRY\_COUNT**="*number-of-retries*"

specifies the maximum number of times that a metadata update request involving multiple repositories can be restarted because of missing repository locks. If the maximum number of retries (also referred to as the “update retry limit”) is reached before the update request is completed, a repository lock error is returned.

When the metadata server receives an update request, the server does the following to process the request:

- 1 obtains an exclusive update lock for each repository that is specified in the request.
- 2 prepares the request, building a transaction of all the attributes, associations, and associated objects that need to be updated, added, or deleted in any repository to successfully complete the request.
- 3 before committing the transaction, verifies that exclusive update locks have been obtained for all repositories that are being updated, and restarts the request if any update locks are missing.

- 4 repeats the process until the request is complete or until the update retry limit is reached. If the update request cannot be completed within the retry limit, a repository lock error is returned.

Activity that is related to obtaining repository update locks is recorded in the Metadata.CounterRetryUpdateLock counters. See “Using the Server Performance Counters and Information Fields” on page 96.

**Default:** 3

SASSEC\_LOCAL\_PW\_SAVE="1 | Y | T | 0 | N | F"

specifies whether users of desktop applications can save their user IDs and passwords in a local metadata connection profile.

If you specify **1**, **Y**, or **T**, then a **Save user ID and password in this profile** check box is available to users when they launch desktop applications. If a user selects this check box, then the user's ID and password are stored in the user's connection profile on the local file system. This profile is used for future connections to the metadata server. Saving the ID and password in the profile allows users to reconnect in the future without entering these values again.

If you do not want to allow IDs and passwords to be stored in client connection profiles, specify **0**, **N**, or **F**.

*Note:*

- After you change the setting for this option and restart the metadata server, each client uses the previous setting for its first connection. The client then discovers the revised setting and conforms to that revised setting for subsequent connections.
- If you change the setting to disallow saved credentials, and credentials are already present in a user's connection profile, those credentials must be manually removed.

△

**Initial setting:** 1

RETURNPASSWORDS="CLEAR | SAS001 | SAS002 | SAS003"

causes passwords to be encoded when they are returned from the metadata server to a requesting client. You can specify one of the following encoding methods:

CLEAR

specifies no encoding of passwords.

SAS001

specifies the use of base64 to encode passwords.

SAS002

specifies the SASProprietary encoding method. This method is included in SAS software and uses a 32-bit key to encode passwords.

SAS003

specifies the Advanced Encryption Standard (AES) encoding method. This method requires that you install SAS/SECURE. If you specify this option but you have not installed SAS/SECURE, then passwords are returned using the SAS002 (SASProprietary) encoding method.

*Note:* SAS/SECURE is an add-on product that requires a separate license. For details about SAS/SECURE, the SASProprietary algorithm, and the AES algorithm, see *Encryption in SAS*. △

**Initial Setting:** SAS002

STOREPASSWORDS="SAS002 | SAS003"

causes passwords to be encoded when they are stored in metadata. You can specify one of the following encoding methods:

**SAS002**

specifies the SASProprietary encoding method. This method is included in SAS software and uses a 32-bit key to encode passwords.

**SAS003**

specifies the Advanced Encryption Standard (AES) encoding method. This method requires that you install SAS/SECURE. If you specify this option but you have not installed SAS/SECURE, then passwords are stored using the SAS002 (SASProprietary) encoding method.

*Note:* SAS/SECURE is an add-on product that requires a separate license. For details about SAS/SECURE, the SASProprietary algorithm, and the AES algorithm, see *Encryption in SAS*. △

**Default:** SAS003

**CAUTION:**

**Passwords that are stored in SAS003 format become unusable and inaccessible if SAS/SECURE is unavailable.** If SAS/SECURE is installed, the default format for stored passwords is SAS003. It is important to keep your SAS/SECURE license current. If you choose to discontinue use of SAS/SECURE, you must revert all stored passwords to SAS002 format before uninstalling the software. To revert passwords, set STOREPASSWORDS="SAS002", restart the metadata server, and use SAS Management Console to re-enter passwords in any logins that need to include passwords. △

**TRUSTEDUSERS="path-and-filename"**

specifies the name and location of the trustedUsers.txt file. The planned installation creates the file in *SAS-configuration-directory\Lev1\SASMeta\MetadataServer*.

---

## <RPOSMGR> Options

The <RPOSMGR> element specifies the following options for the repository manager:

**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. When changing the default value, note that engine values must be specified in uppercase letters.

**PATH="pathname"**

specifies an alternate location for the repository manager. The planned installation creates the repository manager in *SAS-configuration-directory\Lev1\SASMeta\MetadataServer\rposmgr*. *Pathname* can be an absolute reference or a reference relative to the metadata server startup directory.

**OPTIONS="libname-options"**

specifies LIBNAME options for the repository manager. You can specify any option that is valid in a LIBNAME statement (for example, **BLKSZ**=" ").

---

## <InternalAuthenticationPolicy> Options


The <InternalAuthenticationPolicy> element specifies the following options for the assignment and management of passwords for internal users. Internal accounts are accounts that exist only in metadata (see “Internal Accounts” in the “Authentication Mechanisms” chapter in the *SAS Intelligence Platform: Security Administration Guide*).

### CAUTION:

**The option names for <InternalAuthenticationPolicy> are case sensitive and must be entered exactly as they are shown in this list.**   

**ChangeDelayInMinutes**="number-of-minutes"

specifies the number of minutes that must elapse following a password change before an internal user’s password can be changed again.

*Note:* This option does not apply to administrators who are setting passwords for other users.   

**Range:** 0–1440

**Default:** 0

**DigitRequired**="1 | Y | T | 0 | N | F"

specifies whether passwords for internal users must include at least one digit. To enforce this requirement, specify **1**, **Y**, or **T**. To disable this requirement, specify **0**, **N**, or **F**.

**Default:** F

**ExpirationDays**="number-of-days"

specifies the number of days after an internal user’s password is created or changed that the password will expire and need to be changed. A value of 0 specifies that internal user’s passwords will never expire.

**Range:** 0–32767

**Default:** 0

**Note:** You can use SAS Management Console to specify a different number of expiration days for a particular internal user’s password.

**ExpirePasswordOnReset**="1 | Y | T | 0 | N | F"

specifies whether passwords for internal accounts are to be considered already expired when the account is created, so that users are required to change their passwords the first time they log on. To enforce this requirement, specify **1**, **Y**, or **T**. To disable this requirement, specify **0**, **N**, or **F**.

**Interaction:** This option is valid only if ExpirationDays is greater than 0.

**Default:** T

### Exceptions:

- ☐ This option does not apply to administrative users who reset their own passwords in SAS Management Console. When this option is in effect, administrative users who reset their own passwords in SAS Management Console are not required to change their passwords again the next time they log on.
- ☐ You can waive the password change requirement for a specific internal user. To do so, select the **never expire** option in SAS Management Console when you create the user’s account or when you reset the user’s password. Then a password change will not be required the first time the user logs in.

*Note:* The **never expire** option is automatically set for the initial internal users that are created by the SAS Deployment Wizard. △

MinLength="*number-of-bytes*"

specifies the minimum allowed length for the passwords of internal users.

**Range:** 1–32

**Default:** 6

MixedCase="1 | Y | T | 0 | N | F"

specifies whether passwords for internal users must include at least one upper case letter and at least one lower case letter. To enforce this requirement, specify **1**, **Y**, or **T**. To disable this requirement, specify **0**, **N**, or **F**.

**Default:** F

NumPriorPasswords="*number-of-previous-passwords*"

specifies the number of previous passwords that are to be stored in each internal user's password history.

**Range:** 0–5

**Default:** 5

InactiveDaysToSuspension="*number-of-days*"

specifies the number of days after which an internal user's account will be suspended if the account is not used. If you specify 0, then accounts can be inactive indefinitely without being suspended.

**Range:** 0–32767

**Default:** 0

LockoutDurationInMinutes="*number-of-minutes*"

specifies the number of minutes that an internal user must wait before trying to log in again after having been locked out because of excessive login failures (see NumFailuresForLockout).

**Range:** 1–2<sup>31</sup>

**Default:** 60

NumFailuresForLockout="*number-of-failures*"

specifies the number of authentication failures after which an internal user's account will be locked out.

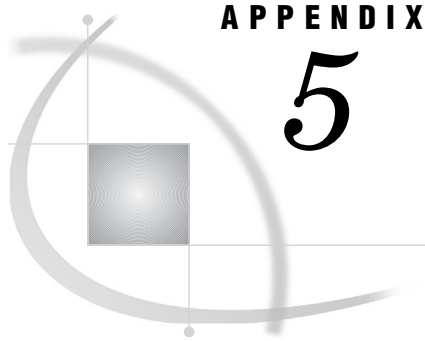
**CAUTION:**

If you specify 0, your system could become vulnerable to password guessing attacks.

△

**Range:** 0–100

**Default:** 3



## APPENDIX

## 5

## Options for the Metadata Server Invocation Command

<i>Modifying the Metadata Server Invocation Options</i>	<b>289</b>
<i>Recommended Parameters</i>	<b>289</b>
<i>Recommended Parameters for All Host Environments</i>	<b>289</b>
<i>Additional Windows Parameters</i>	<b>292</b>
<i>Additional UNIX Parameters</i>	<b>292</b>
<i>Additional z/OS Parameters</i>	<b>292</b>
<i>Optional Parameters</i>	<b>292</b>

### Modifying the Metadata Server Invocation Options

The invocation options for the metadata server are specified in the `sasv9.cfg` file that is located in the following path:

*SAS-configuration-directory*\Lev1\server-context\MetadataServer

During installation, this file is populated with the options that are appropriate for your site. Some of the values are based on information that the installer entered in response to SAS Deployment Wizard prompts.

If you want to specify different values for these options, or if you want to specify additional options, then enter your updates and additions in `sasv9_usermods.cfg`, which is located in the same directory as `sasv9.cfg`. You must restart the server in order for the changes to take effect.

*Note:* In configuration files on z/OS, do not precede system options with a hyphen. △

For information about the options, see:

- “Recommended Parameters” on page 289
- “Optional Parameters” on page 292

### 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 configuration file for Foundation SAS on the host machine.

- the name of the file that contains user modifications to the invocation options.
- 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 library that is 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 journaling-related parameters EMAILSYS, EMAILHOST, EMAILPORT, and EMAILID:

**EMAILSYS SMTP**

specifies that an alert e-mail message is to be 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*).

**EMAILPORT** *port-number*

specifies the port number that is used by the SMTP server that you specified for the EMAILHOST option.

**EMAILID** *"server-email-address"*

specifies the e-mail address that is to appear in the **From** field for alert e-mail messages. The e-mail address can be entered in either of the following forms:

- *"server-name<user-account@domain>"*
- *"<user-account@domain>"*

*Note:* These e-mail parameters need to be specified only if metadata server journaling is enabled. In SAS 9.2, journaling is enabled by default. For more information, see “Configuring Metadata Server Journaling” on page 170. △

- the logging-related parameters LOGCONFIGLOC, PAGESIZE, and LINESIZE:

**LOGCONFIGLOC**=*file-specification*

specifies the name of the XML configuration file that is used to initialize the SAS logging facility for the server. *file-specification* must be a valid filename or a path and filename for your operating environment. If the path contains spaces, enclose *file-specification* in quotation marks.

**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 security-related parameters NETENCRYPTALGORITHM and SSPI:



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.

SSPI

specifies support for the Security Support Provider Interface for single sign on (SSO) connections to IOM servers.

- 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                      specifies that nothing is encrypted. Clients transmit user IDs and passwords as plain text.

CREDENTIALS            specifies that clients encrypt user IDs and passwords.

EVERYTHING            specifies that clients encrypt all client/server communications.

CEL is an alias for CLIENTENCRYPTIONLEVEL.

PROTOCOL=*value*

identifies the network protocol that is 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

**0217E202-B560-11DB-AD91-001083FF6836** identifies a SAS Metadata Server.

TRUSTSASPEER=*pathname*

enables SAS peer sessions from IOM servers to connect as trusted peer sessions. If you specify a blank or empty file, any SAS peer session can connect as a trusted peer. If you specify a file that contains a list of trusted domains, SAS peer sessions are trusted only if they belong to a domain that is listed in your trusted peer file.

*Note:*

- For details about the OBJECTSERVER option, the SSPI option, and the OBJECTSERVERPARMS option and its parameters, see “System Options for Application Servers” in the *SAS Intelligence Platform: Application Server Administration Guide*.
- For details about the other options, see the *SAS Language Reference: Dictionary*.

---

## 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.

`SASINITIALFOLDER` specifies the path to the server's working folder.

For more information, see the *SAS Companion for Windows*.

---

## Additional UNIX Parameters

A UNIX server requires the following parameter in addition to those described in “Recommended Parameters for All Host Environments” on page 289:

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**.

For more information, see the *SAS Companion for UNIX Environments*.

---

## 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 289:

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.

For more information, see the *SAS Companion for z/OS*.

---

## Optional Parameters

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

- The access control parameter `AUTHPROVIDERDOMAIN`:

`AUTHPROVIDERDOMAIN=(provider:domain-name)` specifies an alternative authentication provider. When the `AUTHPROVIDERDOMAIN` parameter is omitted from the metadata server start command, the metadata server uses host authentication to authenticate users. `AUTHPD` is an alias for `AUTHPROVIDERDOMAIN`. For usage information, see “Direct LDAP Authentication” in the “Authentication

Mechanisms” chapter of the *SAS Intelligence Platform: Security Administration Guide*.

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

**-authproviderdomain** = **\(ADIR:MyDomain\)** △

- The OBJECTSERVERPARMS parameters:

**THREADSMIN**=*minimum-number-of-threads*

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

**Default:** By default, the server uses this formula to compute the minimum number of threads:

$$\text{MAX}(5, ((\text{number-of-processors} * 2) + 1))$$

This means that the value is set to either 5 or to  $((\text{number-of-processors} * 2) + 1)$ , whichever is greater.

**THREADSMAX**=*maximum-number-of-threads*

specifies the maximum number of threads in the server’s thread pool. When the number of threads exceeds this number, IOM mechanisms automatically delete threads. TMAX is an alias for THREADSMAX.

*Note:* It is recommended that TMIN=TMAX. △

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

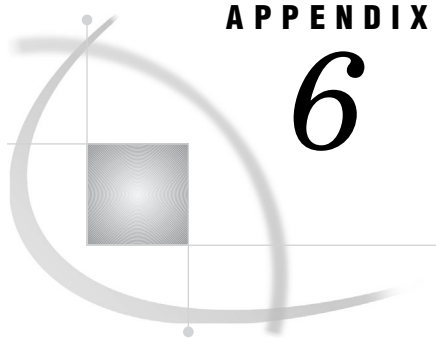
**Default:** By default, if no value is specified for maximum number of threads, it is set to the computed value of THREADSMIN.

*Note:*

- For details about the OBJECTSERVERPARMS option and its parameters, see the *SAS Intelligence Platform: Application Server Administration Guide*.
- For details about the AUTPROVIDERDOMAIN option, see the *SAS Language Reference: Dictionary*.

△





## APPENDIX

## 6

# Using the SAS Deployment Manager to Update Host Name References

<i>Overview of Updating Host Name References</i>	<b>296</b>
<i>About the Update Host Name References Tool in SAS Deployment Manager</i>	<b>296</b>
<i>When to Use the Update Host Name References Tool</i>	<b>297</b>
<i>Use Case 1: An Existing Host Machine Name or Network Domain Name Has Changed</i>	<b>297</b>
<i>Use Case 2: One or More Servers Have Been Cloned to New Machines</i>	<b>298</b>
<i>When Not to Use the Update Host Name References Tool</i>	<b>299</b>
<i>Client Software Components That Require Changes to Host Name References</i>	<b>301</b>
<i>What Modifications Are Performed by the Update Host Name References Tool?</i>	<b>301</b>
<i>Modifications Not Performed by the Update Host Name References Tool</i>	<b>303</b>
<i>Process for Updating Host Name References</i>	<b>304</b>
<i>Summary of the Update Host Name References Process</i>	<b>304</b>
<i>Step 1: Document Your Host Name Changes</i>	<b>304</b>
<i>Step 2: Create a New Account for the Unrestricted User If No Internal Account Exists and the Metadata Server's Host Name Is Changing (Windows Only)</i>	<b>304</b>
<i>Step 3: Change Existing Names at the Operating System Level, or Set Up New Machines to Be Identical to the Current Machines</i>	<b>305</b>
<i>Step 4: Back Up Your Current Deployment and Each New Machine</i>	<b>305</b>
<i>Step 5: Identify the Machines on Which to Run the Tool and Determine the Execution Order</i>	<b>306</b>
<i>Step 6: Run the Update Host Name References Tool on Each Identified Machine</i>	<b>306</b>
<i>Step 7: Complete the Required Manual Steps</i>	<b>311</b>
<i>Step 8: Start Servers and Perform Validation Tasks</i>	<b>311</b>
<i>Step 9: Back Up Each Machine on Which the Update Host Name References Tool Was Executed</i>	<b>312</b>
<i>Troubleshooting the Update Host Name References Tool</i>	<b>312</b>
<i>Example Scenarios for Updating Host Name References</i>	<b>312</b>
<i>Example 1: The Machine Name Has Changed in a Single-Machine Deployment</i>	<b>312</b>
<i>Example 2: The Name of a SAS Application Server Host Machine Has Changed</i>	<b>313</b>
<i>Example 3: The Network Domain Name Has Changed</i>	<b>314</b>
<i>Example 4: A Single-Machine Deployment Has Been Cloned to Another Machine</i>	<b>315</b>
<i>Example 5: A Multiple-Machine Deployment Has Been Cloned to a New Set of Machines</i>	<b>316</b>
<i>Example 6: The Metadata Server Has Been Cloned to a New Machine</i>	<b>317</b>

## Overview of Updating Host Name References

### About the Update Host Name References Tool in SAS Deployment Manager

The SAS Deployment Manager includes an option called Update Host Name References. This option, which is referred to here as the Update Host Name References tool, provides an automated way to update references to server host names in your SAS deployment. You can use this tool if the host name or the network domain name of an existing server machine has changed. You can also use it if you have created an exact clone of one or more servers (or an entire deployment) on another machine or machines.

To use the tool, you launch the SAS Deployment Manager on each machine in your deployment and choose the Update Host Name References option. For each existing machine whose name has changed (or each machine that you have cloned to a new machine), you specify the old host name and the new host name. You can specify all forms of the old and new name, including the fully qualified name, the short name, and the IP address. Or, if you are cloning a machine, you can configure the tool to access the Domain Name System to determine the various forms of the old and new names.

The tool then searches metadata, configuration files, and other files to find occurrences of the old host names and changes them to the new names. When it is finished, the tool produces a report of the changes.

You must run the Update Host Name References tool on every SAS server host machine and every middle-tier host machine in the deployment, even if only one host name has changed. The reason for this requirement is that each machine's configuration files contain references to other machines in the deployment. You must also run the tool on client machines where certain software is installed (see "Client Software Components That Require Changes to Host Name References" on page 301). Otherwise, it is not necessary to run the tool on client machines.

In each execution of the tool, you must enter the complete list of host name changes for the deployment.

*Note:* You can automate running the SAS Deployment Manager when you need to perform the same configuration action on multiple machines in your deployment. The SAS Deployment Manager uses the same record and playback mechanism as the SAS Deployment Wizard to perform a non-interactive, silent configuration. For more information, see "Overview of Automating the SAS Installation on Multiple Machines" in the *SAS Intelligence Platform: Installation and Configuration Guide*. △

After running the Update Host Name References tool, you might still need to manually change some host name references. For example, the tool does not update host names in configuration files for Web application servers. These and other manual steps are specified in the generated report and in "Step 7: Complete the Required Manual Steps" on page 311.

Before using the Update Host Name References tool, be sure to review all of the documentation in this appendix and complete all of the specified prerequisite tasks.

#### **CAUTION:**

The Update Host Name References tool is not supported by certain SAS products and solutions. Refer to SAS Installation Note 39344 to determine whether your installed products and solutions are supported. △

---

## When to Use the Update Host Name References Tool

### Use Case 1: An Existing Host Machine Name or Network Domain Name Has Changed

The Update Host Name References tool can be used when the host name (or network domain name) of one or more existing SAS server or middle-tier host machines has changed. In this situation, you can use the tool to update your SAS deployment to refer to the new host name. The following examples illustrate this use case:

- The name of the machine on which a single-machine deployment is installed has changed, as shown in the following example. You can use the Update Host Name References tool to change all occurrences of the machine name in the deployment to the new name.



Old host name: HOST01.example.com

New host name: HOST99.example.com

- In a multiple-machine deployment, the name of a machine on which a SAS server is installed has changed, as shown in the following example. You can use the Update Host Name References tool to change all occurrences of the server machine's name in the deployment to the new machine name.

SAS Metadata Server  
host machine



MET01.example.com

SAS Application Server  
host machine



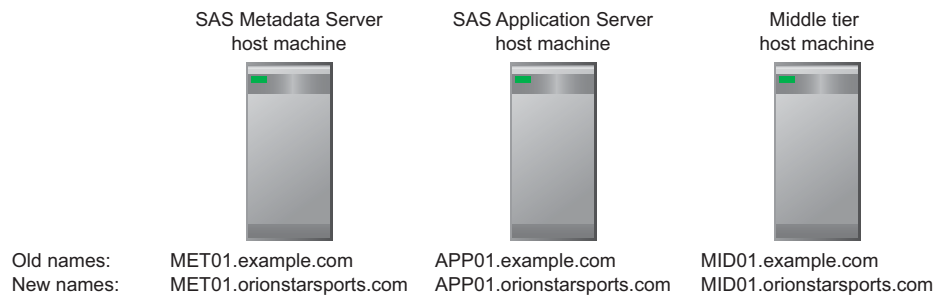
Old name: APP01.example.com  
New name: APP99.example.com

Middle tier  
host machine



MID01.example.com

- The name of the network domain in which a SAS deployment has been installed has changed, as shown in the following example. You can use the Update Host Name References tool to change all occurrences of fully qualified host names throughout the deployment to incorporate the new domain name.



## Use Case 2: One or More Servers Have Been Cloned to New Machines

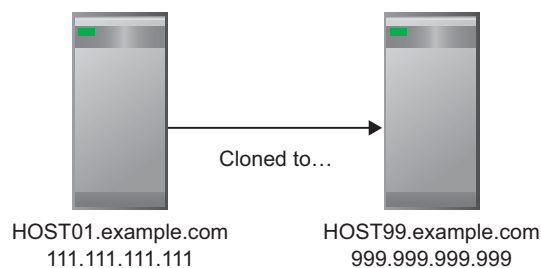
The Update Host Name References tool can be used when you have created an exact clone of one or more SAS servers or middle-tier machines on another machine or machines with different host names. The following examples illustrate this use case:

- You have acquired a new machine and you want to move one of the servers in your deployment to the new machine.
- You want to create an exact clone of your entire deployment on another host machine, on another virtual machine within the same host, or on another set of host machines. For example, you might be creating a test deployment that mirrors an existing production deployment.

To create the clone, you must use cloning, ghosting, or virtualization software to transfer full-system images. On Windows, use a utility such as Norton Ghost from Symantec. On UNIX, use the cloning tool that is appropriate for your flavor of UNIX. After cloning the machine, you can use the Update Host Name References tool to update your SAS deployment with the new machine's host name.

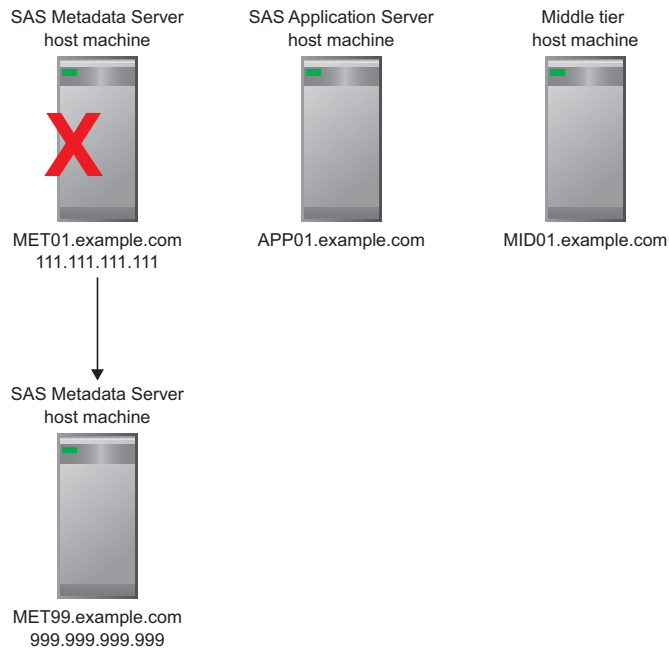
The following are some specific examples of this use case:

- A single-machine deployment has been cloned to another machine, as shown in the following example. You can use the Update Host Name References tool in the new deployment to update all occurrences of the host name to the new host name.

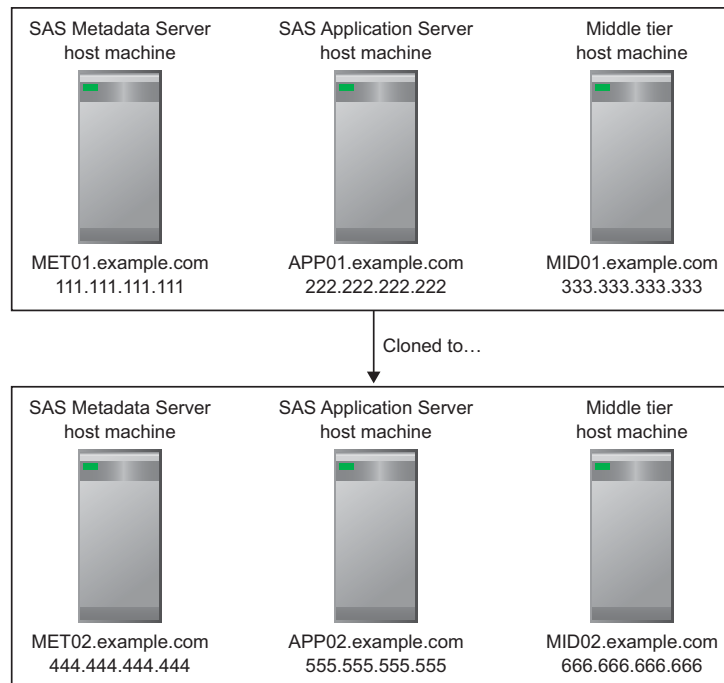


- In a multiple-machine deployment, cloning software has been used to move the metadata server to a new machine, as shown in the following example. You can use the Update Host Name References tool to change all occurrences of the metadata server's host name so that they refer to the new metadata server host machine.





- Cloning software has been used to create a mirror of an entire multiple-machine deployment on a different set of machines, as shown in the following example. You can use the Update Host Name References tool in the new deployment to change all occurrences of each machine name to refer to the new machine names.



## When Not to Use the Update Host Name References Tool

The Update Host Name References tool is not supported by certain SAS products and solutions. Refer to SAS Installation Note 39344 to determine whether your installed products and solutions are supported.

In addition, the Update Host Name References tool is intended only for the use cases that are described in “When to Use the Update Host Name References Tool” on page 297. The tool does not do the following:

- It does not copy software, configuration information, data, or metadata from one machine to another.
- It does not update anything in your deployment except host name references.
- It does not support moving a server to a machine that is of a different operating system or machine type than the existing server machine. It supports only scenarios in which ghosting, cloning, or virtualization software has been used to set up the new machine.

**CAUTION:**

**If you use a method other than ghosting, cloning, or virtualization software to set up a new machine, then the use of the Update Host Name References tool is not supported for that machine.** For example, use of the Update Host Name References tool is not supported if you set up a new machine by running the SAS Deployment Wizard and then copying SAS configuration files from an existing machine.  $\triangle$

The following table lists tools that are available to update various aspects of your deployment. Use this table to make sure that you have selected the correct tool for your situation.

Tools for Updating Your Deployment

Tool Name	Purpose	Documentation Reference
Update Host Name References (in SAS Deployment Manager)	Find references to one or more server or middle-tier host names in a deployment, and replace them with new host names. Use this tool after the name of a SAS server or middle-tier host machine has changed, or after one or more server or middle-tier machines have been cloned to a new machine using ghosting, cloning, or virtualization software.	Appendix 6, “Using the SAS Deployment Manager to Update Host Name References,” on page 295
SAS Deployment Wizard	Install and configure SAS software. Use this tool if you want to move a SAS application server to a different type of machine or a machine with a different operating system (in which case you cannot use ghosting, cloning, or virtualization software).	“Defining Multiple Application Servers” in the <i>SAS Intelligence Platform: Application Server Administration Guide</i>
%OMAPORT macro	Port a metadata repository from one operating system to another.	“Porting a Metadata Repository to a Different Host Environment” on page 189
Promotion tools (Export SAS Package and Import SAS Package)	Promote individual objects or groups of objects from one location to another, either on the same metadata server or a different metadata server. You can also include associated physical content in the promotion.	Chapter 17, “Using the Promotion Tools,” on page 213 and SAS Management Console Help

---

## Client Software Components That Require Changes to Host Name References

You must run the Update Host Name References tool on client machines if any of the following software components are installed:

- ❑ SAS Customer Intelligence Plug-ins for SAS Management Console
- ❑ SAS Customer Intelligence Studio
- ❑ SAS Customer Intelligence Utilities
- ❑ SAS Grid Manager Client Utility
- ❑ SAS Marketing Automation Integration Utilities
- ❑ SAS Marketing Automation Launcher
- ❑ SAS Marketing Optimization Client

If a client machine does not contain any of these components, then you do not need to run the Update Host Name References tool on the machine. However, if the name of the metadata server host machine has changed, you might need to make manual changes to connection profiles on client machines. See “Step 7: Complete the Required Manual Steps” on page 311.

---

## What Modifications Are Performed by the Update Host Name References Tool?

For each existing machine whose name has been changed (or each machine that you are cloning to a new machine), the Update Host Name References tool enables you to specify the old host name and the new host name. The tool searches metadata, configuration files, and other files to find occurrences of the old host names that you specify. It changes these names to the new names that you specify and then produces a report of the changes. Modifications are performed in the following locations, as necessary:

- ❑ files in the SAS configuration directory, such as the following:
  - ❑ SAS server configuration (sasv9.cfg) files.
  - ❑ omaconfig.xml and metaparms.sas, which are used by the metadata server.
  - ❑ the SAS object spawner’s startup script and metadataConfig.xml file.
  - ❑ XML files that are used by SAS Deployment Tester.
  - ❑ properties files and configuration files for remote and local services, if your configuration includes the SAS middle tier.
  - ❑ login configuration files for the Java Authentication and Authorization Service (JAAS), if your configuration includes the SAS middle tier.

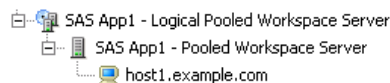
*Note:* These files are modified only in the SAS configuration directory. The report that is produced by the Update Host Name References tool instructs you to manually update the files that are used by your Web application server. △

- ❑ files that contain Windows user IDs that are qualified by a host name (for example, host1\sasadm or sasadm@host1). If you use this type of account for the unrestricted user, trusted user, or other users whose IDs are stored in configuration files, then the host names in the user IDs are updated as needed. These updates affect some of the previously listed files, as well as the following additional files:

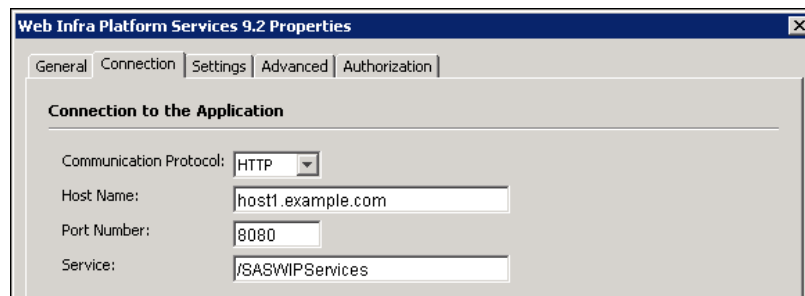
- adminUsers.txt and trustedUsers.txt, which are used by the metadata server.
- sasv9\_meta.cfg, which is used by the SAS OLAP Server, the SAS Table Server, and the SAS/SHARE server.
- the SAS/CONNECT spawner's metadataConfig.xml file.
- JDBC data source definition files, if your configuration includes the middle tier.

*Note:* These files are modified only in the SAS configuration directory. The report that is produced by the Update Host Name References tool instructs you to manually update the files that are used by your Web application server.  $\triangle$

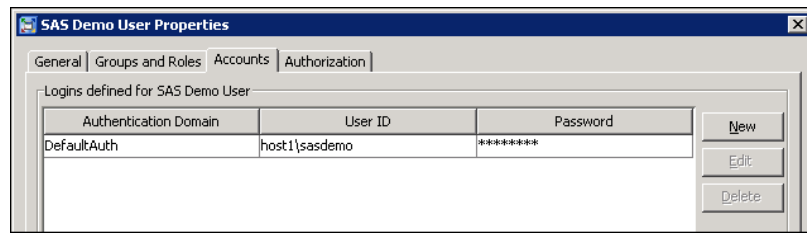
- configuration files for the Stored Process Web application, if your configuration includes the middle tier. The WAR and EAR files for the application are then rebuilt. The Stored Process Web application is redeployed automatically if you use JBoss as your Web application server and if you selected the option to automatically deploy Web applications during installation. Otherwise, the report that is produced by the Update Host Name References tool instructs you to manually redeploy the application.
- metadata properties such as the following:
  - properties in server metadata. For example, host names are stored in the server connection properties for SAS Pooled Workspace Servers, as shown in this example from the Server Manager tree in SAS Management Console:



- properties in metadata for SAS middle-tier services. For example, host names are stored in the metadata properties for SAS Web Infrastructure Platform Services, as shown in this example from the Configuration Manager plug-in for SAS Management Console:



- user IDs for Windows accounts that are stored in metadata and are qualified with a machine name. For example, suppose the user ID for SAS Demo User is stored in metadata as **host1\sasdemo**, as shown in this example from the User Manager plug-in for SAS Management Console:



If host1 is being changed to host2, the user ID would be changed to **host2\sasdemo**.

- the SAS object spawner service. If this service is present on a Windows machine, the tool deletes the service and recreates it using the new host name.

*Note:* The SAS object spawner service is the only service that includes a host name reference. The other SAS services do not need to be recreated. △

Additional files and metadata might also be changed, depending on the software that you have installed. After you run the tool, a report is generated that lists all of the metadata properties and files that were changed.

---

## Modifications Not Performed by the Update Host Name References Tool

The Update Host Name References tool does not modify host names in the following locations in your deployment:

- configuration files for Web application servers. Detailed instructions for changing host name references in these files are provided in the report that is generated when you run the tool. The instructions are tailored for your Web application server.
- log files. It is not necessary to update host names in log files, since these files contain historical information that is not used in processing.
- binary files, temporary files, and SAS data sets. Any files of these types that are found in the SAS configuration directory are ignored.
- users' connection profiles on client machines (unless the client software is listed in "Client Software Components That Require Changes to Host Name References" on page 301). For other client software, connection profiles must be updated manually on each client machine if the metadata server's host name has changed.
- the sas-environment.xml file in the *SAS-configuration-directory/Lev1/Web/Common* on middle-tier machines. This file is not updated automatically because it might contain site-specific customizations that need to be preserved.
- multicast parameters on middle-tier machines. If you have created a new deployment that will be running concurrently with your original deployment, then you might need to modify these parameters to avoid potential conflicts.
- Windows user IDs that are associated with scheduled reports. If the user ID is qualified by a machine name or network domain name that has changed, then you will need to reschedule the report after running the Update Host Name References tool.

For information about making manual changes to client profiles, the sas-environment.xml file, and multicast parameters, see "Step 7: Complete the Required Manual Steps" on page 311.

The report that is generated by the Update Host Name References tool might list other files that were not updated by the tool.

---

## Process for Updating Host Name References

---

### Summary of the Update Host Name References Process

Follow these steps to use the Update Host Name References tool:

- 1 Document your host name changes.
- 2 Create a new account for the unrestricted user if no internal account exists and the metadata server host name is changing (Windows only).
- 3 Change existing names at the operating system level, or set up new machines to be identical to the current machines.
- 4 Back up your current deployment and each new machine.
- 5 Identify the machines on which to run the tool, and determine the execution order.
- 6 Run the Update Host Name References tool on each identified machine.
- 7 Complete the required manual steps.
- 8 Start your servers and perform validation tasks.
- 9 Perform another backup of each machine on which the Update Host Name References tool was executed.

If you plan to run any of the other SAS Deployment Manager functions on a new machine in addition to running the Update Host Name References tool, be sure to run the Update Host Name References tool first. For example, if you want to run both the Update Host Name References tool and the Update Passwords tool, you must run the Update Host Name References tool first.

---

### Step 1: Document Your Host Name Changes

For each machine whose machine name is to change (or for each machine that you are cloning), write down the current name and the new name. List all forms of each name, including the following:

- the fully qualified host name (for example, P4001.example.com)
- the short host name (for example, P4001)
- the IP name (for example, 111.111.111.111)

---

### Step 2: Create a New Account for the Unrestricted User If No Internal Account Exists and the Metadata Server's Host Name Is Changing (Windows Only)

When you run the Update Host Name References tool, you will be prompted for a user ID and password to connect to the SAS Metadata Server. Before assigning a new name to the metadata server's host machine (or before cloning the metadata server to a new machine) on Windows, you need to make sure that you have an unrestricted user account that can successfully connect to the server.

If the unrestricted user has an internal account (for example, sasadm@saspw) or a network account, then you will be able to use that account to connect to the metadata server even if its machine name has changed.

If the unrestricted user has only an external account that is qualified by a local machine name that is changing, then take one of the following actions:

**CAUTION:**

**You must take one of these actions before you change the metadata server machine name at the operating system level. If you are cloning the metadata server or the entire deployment, you must take these actions before you perform the cloning process. △**

- In the User Manager plug-in, add an internal account to the unrestricted user's definition. Use the internal account when you run the Update Host Name References tool.
- In the User Manager plug-in, add another external account to the unrestricted user's definition. Qualify the account with the metadata server's new host name, and use this account when you run the Update Host Name References tool.

*Note:* If you rename the metadata server at the operating system level (or clone the metadata server to a new machine) before you have created the new account, it is still possible to connect to the server as an unrestricted user. You can do so by specifying the appropriate local account in the adminUsers.txt file, and then using that account to connect. △

---

### Step 3: Change Existing Names at the Operating System Level, or Set Up New Machines to Be Identical to the Current Machines

If you are renaming one or more existing server or middle-tier machines (or renaming a network domain), then you must change the names at the operating system level before you run the Update Host Name References tool.

If you are creating a clone of one or more SAS servers or middle-tier machines (or of an entire SAS deployment), you must use cloning, ghosting, or virtualization software to transfer full-system images. On Windows, use a utility such as Norton Ghost from Symantec. On UNIX, use the cloning tool that is appropriate for your flavor of UNIX. After the clone is created, check to make sure that the cloned machine is identical to the source machine (except for the machine name). For example, the SAS configuration directory and its contents, the SAS installation directory and its contents, user accounts, and third-party software must be identical.

**CAUTION:**

**If you use a method other than ghosting, cloning, or virtualization software to set up a new machine, then the use of the Update Host Name References tool is not supported for that machine.** For example, use of the Update Host Name References tool is not supported if you set up a new machine by running the SAS Deployment Wizard and then copying SAS configuration files from an existing machine. △

After you have finished the cloning process, do not change anything on the machine before you run the Update Host Name References tool. For example, do not change any directory names or user accounts.

---

### Step 4: Back Up Your Current Deployment and Each New Machine

Make a complete backup of the deployment, including the current machines and any new machines. The backup should include the metadata repositories, as well as the file systems on each machine. To ensure that you make a correct backup of the metadata server, see Chapter 9, “Best Practices for Backing Up and Restoring Your System,” on page 123.

---

## Step 5: Identify the Machines on Which to Run the Tool and Determine the Execution Order

Next, identify the machines on which you need to run the Update Host Name References tool, and then determine the order in which you need to execute the tools on those machines. The requirements are as follows:

- The tool must be run on every SAS server host machine and every middle-tier host machine in the deployment, even if only one host name has changed. The reason for this requirement is that each machine's configuration files contain references to other machines in the deployment.
- The tool must also be run on client machines where certain client software is installed. See "Client Software Components That Require Changes to Host Name References" on page 301. Otherwise, it is not necessary to run the tool on client machines.
- The tool must be run first on the SAS Metadata Server machine, then on SAS application server machines, then on the middle-tier machine, and then on any required client machines.

For example:

- If the name of one or more existing machines has changed (or the name of the network domain in which the deployment is installed has changed), then run the tool on each machine in the deployment. Run the tool on the metadata server host machine first, then on the SAS application servers host machines, and then on the middle-tier host machine.
- If you have cloned an entire deployment to one or more new machines, then run the tool on each machine in the new deployment. In a multiple-machine deployment, run the tool on the new metadata server host machine first, then on the new SAS application server host machines, and then on the new middle-tier host machine.
- If you have cloned just part of a multiple-machine deployment to new machines, then run the tool on each of the new machines as well as on any machines that were not cloned to new machines.

For example, if you have cloned the metadata server host to a new machine, then begin by running the tool on the new metadata server host machine. Then run the tool on the existing application server host machines, and then on the existing middle-tier host machine.

- If you have identified client machines on which you need to run the tool, run it on those machines last, after the server and middle-tier machines.

For further guidance in identifying the machines and determining the required order, see "Example Scenarios for Updating Host Name References" on page 312.

---

## Step 6: Run the Update Host Name References Tool on Each Identified Machine

Before running the Update Host Name References tool, make sure you have completed all of the preceding steps. You should have planned your host name changes, renamed the machines or network domain at the operating-system level (or correctly set up any new machines), and then backed up your current deployment and any new machines. In addition, make sure that you have identified the machines on which to run the tool and determined the correct execution order. It might have also been necessary to add a new account for the unrestricted user.



**CAUTION:**

You must run the tool on every SAS server host machine and middle-tier machine in the deployment, even if only one machine name has changed. Run the tool on the metadata server first, then on SAS application server machines, then on the middle-tier machine, and then on any required client machines. △

On each machine, follow these steps to run the Update Host Name References tool:

- 1 Make sure that the metadata server is running.

*Note:* If you have cloned the metadata server (or the entire deployment) to a new machine, then start the new metadata server, not the old one. You will be able to start it on the new machine even though the server's host name references have not been updated in the deployment. △

- 2 Make sure that all other server processes on all machines in the deployment are stopped. These include SAS server processes as well as middle-tier processes, including the Web application server and the SAS Services Application (Remote Services). If you are using an IBM WebSphere Application Server or an IBM WebSphere Application Server Network Deployment, then be sure to stop your deployment manager server and node manager server as well as your application servers.

**CAUTION:**

If you have cloned the middle-tier machine (or the entire deployment) to a new machine, then you must also stop the middle-tier processes on the old middle-tier machine (or in the old deployment). If the two middle tiers will be running concurrently, then you might need to modify the multicast parameters before restarting these processes, as described in “Step 7: Complete the Required Manual Steps” on page 311. △

- 3 Log on to the machine with the same user account that was used to deploy SAS 9.2, and start the SAS Deployment Manager on the machine. To do so, navigate to *SAS-installation-directory/SASDeploymentManager/9.2* and launch **config.exe** (Windows), **config.sh** (UNIX) or **config.rexx** (z/OS). On Windows, you can use the shortcut on the **Start** menu.
- 4 On the first page of the SAS Deployment Manager, select **Update Host Name References**.
- 5 On the Update Host Name References Information page, review the information that is displayed, which is a summary of the information in this documentation.
- 6 On the Select Configuration Directory/Level page, select the configuration directory for which you are running the tool.

If the configuration directory is not listed, select **Enter Configuration Directory and Level** and use the **Browse** button to find the directory. Then select the appropriate configuration level from the drop-down box.

- 7 On the Specify Connection Information page, enter the following information:
  - ☐ the metadata server host machine and port number.

If the name of the metadata server machine has changed, or if you have cloned the metadata server (or the entire deployment) to a new machine, enter the new metadata server machine name at the **Host Name** prompt.

- ☐ the user ID and password of an unrestricted user (that is, a user who is in the **SAS Metadata Server: Unrestricted** role).

*Note:* If the metadata server has been cloned or renamed, and if the unrestricted user has only a local machine account, see “Step 2: Create a New Account for the Unrestricted User If No Internal Account Exists and the Metadata Server's Host Name Is Changing (Windows Only)” on page 304. △

8 On Specify Host Information page, make the following entries:

**Old Host Name(s)**

Enter a comma-delimited list of the host names for the machines whose names have changed, or for the machines that have been cloned to new machines. Be sure to include all machines in the deployment whose names have changed (or that have been cloned to new machines). Since the tool is not case-sensitive, you can enter the names using either uppercase or lowercase letters.

**New Host Name(s)**

Enter a comma-delimited list of the new host names for the machines whose names have changed (or the machines on which you have created clones of existing machines). Be sure to include the same forms of the machine names and to enter the names in the same order that you entered them in the **Old Host Name(s)** field.

**Perform DNS lookup**

For a given host machine, different files and metadata in your deployment might contain different forms of the host name, including the fully qualified name, the short name, and the IP address. If you are cloning one or more servers to a new machine, then you can select **Perform DNS lookup** option, which accesses the Domain Name System to determine all of the forms of each host name.

If the name of an existing host machine (or network domain name) has changed, then you must deselect **Perform DNS lookup**. The Domain Name System does not contain information about machine names or network domains that no longer exist.

**CAUTION:**

**You must complete the preceding fields in such a way that all applicable forms of the host name are updated correctly throughout your deployment.** To make sure that all applicable forms of the machine names are updated correctly, refer to the following table. △

## Determining the Correct Format for Host Name Entries

Scenario	Format for Entering Old and New Host Names	Perform DNS Lookup?
One or more existing SAS server or middle-tier host machines have changed.	Enter both the fully qualified name and the short name for each machine whose name has changed. It is not necessary to enter IP addresses, since the IP address stays the same in this scenario.	Deselect <b>Perform DNS lookup</b> .
The name of a network domain name has changed.	Enter fully qualified host names for the machines that contain the network domain name that is changing. It is not necessary to enter short names or IP addresses, since these forms of the name stay the same in this scenario.	Deselect <b>Perform DNS lookup</b> .
One or more servers (or an entire deployment) have been cloned to a new machine or machines.	Enter fully qualified host names for the machines that have been cloned to new machines. If you are selecting <b>Perform DNS lookup</b> (which is strongly recommended), then you do not need to enter the machines' IP addresses or short names.	As a best practice, select <b>Perform DNS lookup</b> . Otherwise, you must enter all forms of the machine names.

For example entries, see “Example Scenarios for Updating Host Name References” on page 312.

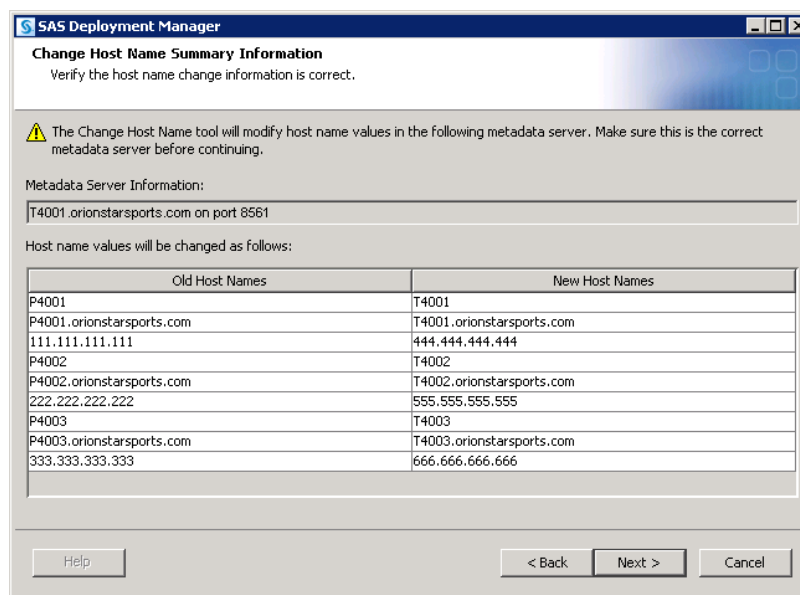
If you select **Perform DNS lookup**, then the lookup is performed when you click **Next**. The results of the lookup appear on the next page (Update Host Name References Summary).

If the specified names are not found in the Domain Name System, the following message appears: **Unable to resolve the DNS host name for machine-name. Enter a different value for the host name or clear the option to perform a DNS lookup of the name.** You can then correct your entry and click **Next** to repeat the lookup. If you are changing the name of an existing machine or network domain, then you must deselect **Perform DNS lookup**. The lookup cannot be performed in these scenarios.

- 9 On the Update Host Name References Summary Information page, review the list of host name changes that will be made, and make sure that they match the old and new names that you identified in “Step 1: Document Your Host Name Changes” on page 304. The current host names are listed in the first column, and the new host names are listed in the second column.

If you did not select the **Perform DNS lookup** option, this page lists all host name changes exactly as you specified them on the previous page.

If you selected the **Perform DNS lookup** option, this page lists all forms of the host name that were found during the DNS lookup, along with the values that will replace those names. For example, if you specified P4001.orionstarsports.com, P4002.orionstarsports.com, and P4003.orionstarsports.com as the old host names and T4001.orionstarsports.com, T4002.orionstarsports.com, and T4003.orionstarsports.com as the new host names, this page might display a list similar to the following:



In the preceding example, the DNS lookup option found all other forms of the machine names for the old machine, including the short names and the IP addresses. These names are listed in the left column. The DNS lookup option also found the corresponding names for the new machines and listed them in the right column.

Review the information carefully to ensure the following:

- ☐ The correct metadata server is specified in the Metadata Server Information.
- ☐ The correct host name changes are listed.

If the information is correct, click **Next** to proceed. Otherwise, click **Back** to make the necessary corrections.

- 10 The Summary page lists the main execution steps, including a list of the products on the machine whose configuration information will be changed. If you are sure you are ready for the host names to be changed, click **Start** on this page.

**CAUTION:**

**The Update Host Name References tool does not have an “undo” feature. The results of running the tool can be undone only by restoring the machine from a backup.**

Therefore, you should click **start** only if you are sure you have set up the execution correctly. △

- 11 The next page displays the status of each step as it is executed.
- 12 When execution is finished, the tool creates a document called `ChangeHostName_date-and-time.html`. This document lists all of the files and metadata properties that were changed and provides instructions for any additional steps to be performed manually. It might also report warnings if issues were encountered during the execution.

If a browser is installed on your machine, the report is displayed automatically. Otherwise, you can retrieve the report from the **Documents** directory within your configuration directory.

Repeat this process on each machine that you identified in “Step 1: Document Your Host Name Changes” on page 304, being sure to specify exactly the same old and new host names that you specified in the first execution. If you have multiple SAS configuration directories (for multiple SAS deployments) on the same machine, you must run the Update Host Name References tool separately for each directory.

If any part of the execution fails, see “Troubleshooting the Update Host Name References Tool” on page 312.

---

## Step 7: Complete the Required Manual Steps

In addition to listing the files and metadata profiles that were updated, the `ChangeHostName_date-and-time.html` report provides instructions for steps to be performed manually. The report is stored in the **Documents** directory within your configuration directory. The manual steps might include the following:

- updating configuration files for Web application servers. The instructions that are provided are tailored for your Web application server.
- redeploying the Stored Process Web application, if the application’s configuration files contain Windows user IDs that are qualified by a host name.
- restarting SAS servers.

You might need to perform the following additional steps:

- If the metadata server’s host name has changed, then you must update host names in users’ connection profiles on each machine where client software is installed (unless the client software is listed in “Client Software Components That Require Changes to Host Name References” on page 301). Be sure to update the profiles for SAS Management Console, which is typically installed on server and middle-tier machines.

For instructions, see “Administering Connection Profiles for Desktop Applications” in the *SAS Intelligence Platform: Desktop Application Administration Guide*.

- If you have created a new deployment that will be running concurrently with your original deployment, and if the deployments include a middle tier, then you might need to modify the multicast parameters to avoid potential conflicts. See “Administering Multicast Options” in the *SAS Intelligence Platform: Web Application Administration Guide*.
- If your deployment includes a middle tier, then you might need to update host names in the `sas-environment.xml` file. This file is located on middle-tier machines in the following path: *SAS-configuration-directory/Lev1/Web/Common*. The file is not updated automatically because it might contain site-specific customizations that need to be preserved. For more information, see “Configuring the SAS Environment File” in the *SAS Intelligence Platform: Web Application Administration Guide*.
- If you have previously scheduled a report that is associated with a machine or domain-qualified user ID on Windows, and the machine name or network domain name has changed, then you will need to reschedule the report.

---

## Step 8: Start Servers and Perform Validation Tasks

After you have completed the necessary manual updates, start all of your servers as specified in `ChangeHostName_date-and-time.html`. Also start any servers that are not specifically mentioned, such as the SAS Services Application (Remote Services) and Web application servers.

### CAUTION:

If the unrestricted user or the SAS Trusted User has an external account that is qualified by a host name that has changed (for example, `host1\sasadm` or `host1\sastrust`), then you must stop and start the metadata server before you start any of the other servers. Doing so will ensure that the metadata server runs with the correct values for those accounts. △

After the servers have been started, validate your deployment. For details, see:

- Chapter 5, “Operating Your Servers,” on page 51
- “Validate the SAS 9.2 Servers” in the *SAS Intelligence Platform: Installation and Configuration Guide*

You can also refer to your Instructions.html document, which is located in the **Documents** directory within your configuration directory. If you use this document, you will need to substitute new machine names where applicable.

---

## Step 9: Back Up Each Machine on Which the Update Host Name References Tool Was Executed

After you have validated your servers, it is strongly recommended that you make a complete backup of each machine on which the Update Host Name References tool was executed. The backup should include the metadata repositories, as well as the file systems on each machine. To ensure that you make a correct backup of the metadata server, see Chapter 9, “Best Practices for Backing Up and Restoring Your System,” on page 123.

---

## Troubleshooting the Update Host Name References Tool

The Update Host Name References tool creates a log for each software component whose metadata or configuration information needs to be updated. The logs are named *software-component\_changeHost\_date-and-time.log* and are written to *SAS-configuration-directory/Levn/Logs/Configure*. Refer to these logs if the update process for a particular software component completes with errors.

The SAS Deployment Manager also creates a log called SDM.log, which can be used for general troubleshooting for the SAS Deployment Manager. The log is written to the following path:

- On Windows Vista systems:  
**C:\Users\user\AppData\Local\SAS\SASDeploymentWizard\9.2**
- On other Windows systems: **C:\Documents and Settings\user\Local Settings\Application Data\SAS\SASDeploymentWizard\9.2**
- On UNIX systems: *user's-home-directory/.SASAppData/SASDeploymentWizard/9.2*

---

## Example Scenarios for Updating Host Name References

---

### Example 1: The Machine Name Has Changed in a Single-Machine Deployment

In this example, a single-machine deployment is installed on HOST01.example.com, and the name of the machine has been changed to HOST99.example.com. All occurrences of the machine's fully qualified name and short name need to be updated throughout the deployment to refer to the new name.



Old host name: HOST01.example.com  
New host name: HOST99.example.com

In this scenario, you would use the Update Host Name References tool as follows:

- 1 Create a complete system backup of the machine, including a correct backup of the metadata server.
- 2 Start the metadata server and stop all of the other servers.
- 3 Run the Update Host Name References tool on the machine.

In the Specify Host Names dialog box, you must enter both the fully qualified host names and the short host names. (You cannot perform a DNS lookup when the name of an existing machine has changed.) You do not need to enter the IP address, because it remains the same in this scenario. Specify the following:

- ☐ **Old host name:** HOST01.example.com, HOST01
  - ☐ **New host name:** HOST99.example.com, HOST99
  - ☐ **Perform DNS lookup:** Not selected
- 4 Perform the necessary manual steps, as listed in the report. Because the metadata server host name has changed, update the connection profiles for client applications where necessary.
  - 5 Start the servers and perform the validation steps.
  - 6 Create a complete system backup of the machine, including a correct backup of the metadata server.

---

## Example 2: The Name of a SAS Application Server Host Machine Has Changed

In this example, The SAS Metadata Server, SAS application server, and SAS middle tier are installed on separate machines. The SAS application server is installed on a machine called APP01.example.com, and the name of this machine has been changed to APP99.example.com. All occurrences of the application server machine's fully qualified name and short name need to be updated throughout the deployment to refer to the new name.

SAS Metadata Server  
host machine



MET01.example.com

SAS Application Server  
host machine



Old name: APP01.example.com  
New name: APP99.example.com

Middle tier  
host machine



MID01.example.com

In this scenario, you would use the Update Host Name References tool as follows:

- 1 Create a complete system backup of each machine in the deployment, including a correct backup of the metadata server.
- 2 Start the metadata server, and stop all of the other servers.
- 3 Run the Update Host Name References tool on MET01.example.com.

In the Specify Host Names dialog box, you must enter both the fully qualified host names and the short host names. (You cannot perform a DNS lookup when the name of an existing machine has changed.) You do not need to enter the IP address, because it remains the same in this scenario. Specify the following:

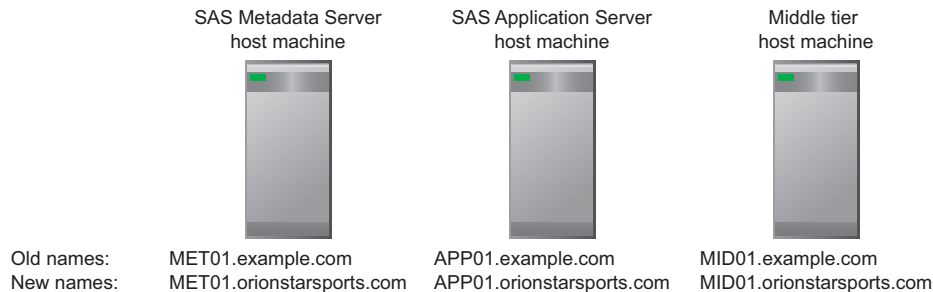
- ☐ **Old host name:** APP01.example.com,APP01
- ☐ **New host name:** APP99.example.com,APP99
- ☐ **Perform DNS lookup:** Not selected

- 4 Run the Update Host Name References tool on APP99.example.com. In the Specify Host Names dialog box, make exactly the same entries that you made in step 3.
- 5 Run the Update Host Name References tool on MID01.example.com. In the Specify Host Names dialog box, make exactly the same entries that you made in step 3.
- 6 Perform the necessary manual steps, as listed in the report.
- 7 Start the servers and perform the validation steps.
- 8 Create a complete system backup of the deployment, including a correct backup of the metadata server.

---

### Example 3: The Network Domain Name Has Changed

In this example, a SAS deployment is installed in a network domain called example.com, and the domain name has been changed to orionstarsports.com. All occurrences of fully qualified host names in the deployment need to be updated to incorporate the new domain name.



In this scenario, you would use the Update Host Name References tool as follows:

- 1 Create a complete system backup of each machine in the deployment, including a correct backup of the metadata server.
- 2 Start the metadata server, and stop all of the other servers.
- 3 Run the Update Host Name References tool on MET01.orionstarsports.com.

In the Specify Host Names dialog box, you need to enter only the fully qualified host names, because the short host names and the IP addresses remain the same in this scenario. You cannot perform a DNS lookup when the names of existing machines have been changed. Specify the following:

- ☐ **Old host name:** MET01.example.com,APP01.example.com,MID01.example.com

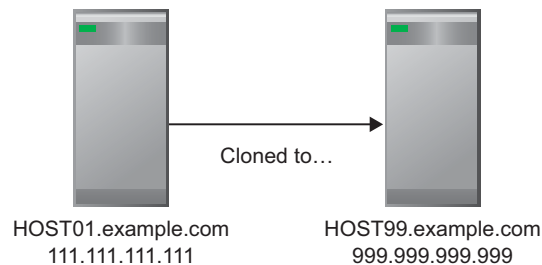


- ☐ **New host name:** MET01.orionstarsports.com,  
APP01.orionstarsports.com,MID01.orionstarsports.com
  - ☐ **Perform DNS lookup:** Not selected
- 4 Run the Update Host Name References tool on APP01.orionstarsports.com. In the Specify Host Names dialog box, make exactly the same entries that you made in step 3.
  - 5 Run the Update Host Name References tool on MID01.orionstarsports.com. In the Specify Host Names dialog box, make exactly the same entries that you made in step 3.
  - 6 Perform the necessary manual steps, as listed in the report. Because the fully qualified host name for the metadata server has changed, update the connection profiles for client applications where necessary.
  - 7 Start the servers and perform the validation steps.
  - 8 Create a complete system backup of the deployment, including a correct backup of the metadata server.

---

### Example 4: A Single-Machine Deployment Has Been Cloned to Another Machine

In this example, a single-machine deployment is installed on HOST01.example.com, and the deployment has been cloned to another machine called HOST99.example.com. In the new deployment, all references to the fully qualified host name, short host name, and IP address need to be updated to refer to the new host machine.



In this scenario, you would use the Update Host Name References tool as follows:

- 1 Create a complete system backup of HOST01.example.com, including a correct backup of the metadata server.
  - 2 Set up the new machine (HOST99.example.com) by using cloning, ghosting, or virtualization software. For details, see “Step 3: Change Existing Names at the Operating System Level, or Set Up New Machines to Be Identical to the Current Machines” on page 305.
  - 3 Start the metadata server on the new machine (HOST99.example.com). Stop all of the other servers on the new machine, and stop the middle-tier processes on the old machine (HOST01.example.com).
  - 4 Run the Update Host Name References tool on HOST99.example.com.
- In the Specify Host Names dialog box, you need to enter only the fully qualified host names. The DNS lookup identifies all other forms of the host names so that they can be updated correctly. Specify the following:

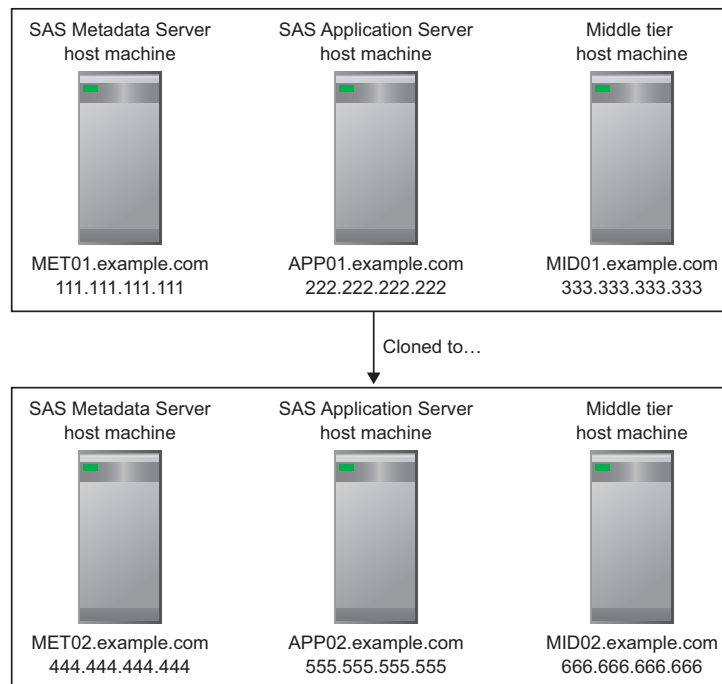
- ☐ **Old host name:** HOST01.example.com
- ☐ **New host name:** HOST99.example.com

□ **Perform DNS lookup:** Selected

- 5 Perform the necessary manual steps, as listed in the report. Because the metadata server host name has changed, update the connection profiles for client applications where necessary. If the deployment includes a middle tier, and if the new deployment will be running concurrently with the old deployment, then the multicast parameters might also need to be modified.
- 6 Start the servers and perform the validation steps.
- 7 Create a complete system backup of the new machine, including a correct backup of the metadata server.

## Example 5: A Multiple-Machine Deployment Has Been Cloned to a New Set of Machines

In this example, a multiple-machine deployment is installed on MET01.example.com, APP01.example.com, and MID01.example.com. Cloning software has been used to create a mirror of the deployment on a different set of machines called MET02.example.com, APP02.example.com, and MID02.example.com. In the new deployment, all occurrences of each machine's fully qualified name, short name, and IP address need to be updated to refer to the new machines.



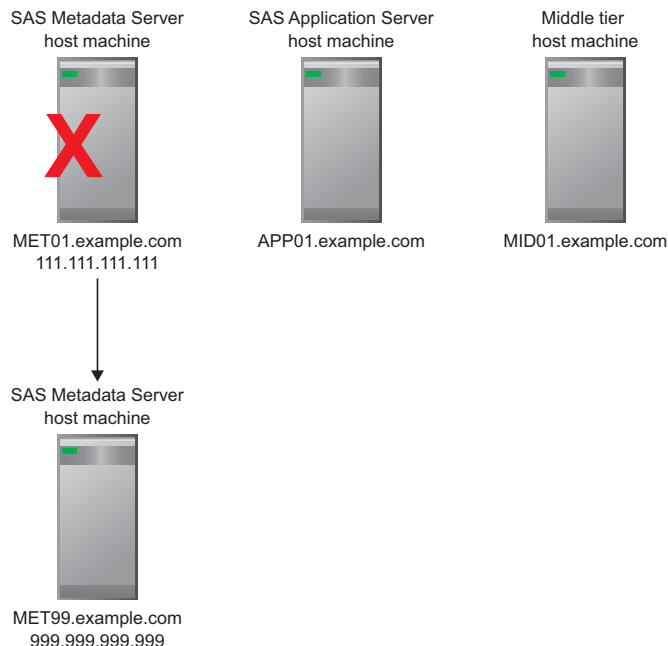
In this scenario, you would use the Update Host Name References tool as follows:

- 1 Create a complete system backup of each machine in the original deployment, including a correct backup of the metadata server.
- 2 Set up the new machines (MET02.example.com, APP02.example.com, and MID02.example.com) by using cloning, ghosting, or virtualization software. For details, see “Step 3: Change Existing Names at the Operating System Level, or Set Up New Machines to Be Identical to the Current Machines” on page 305.

- 3 Start the metadata server on the new metadata server machine (MET02.example.com). Stop all of the other servers in the new deployment, and stop the middle-tier processes in the old deployment.
- 4 Run the Update Host Name References tool on MET02.example.com.  
In the Specify Host Names dialog box, you need to enter only the fully qualified host names. The DNS lookup identifies all other forms of the old and new host names so that they can be updated correctly. Specify the following:
  - ☐ **Old host name:**  
**MET01.example.com, APP01.example.com, MID01.example.com**
  - ☐ **New host name:**  
**MET02.example.com, APP02.example.com, MID02.example.com**
  - ☐ **Perform DNS lookup:** Selected
- 5 Run the Update Host Name References tool on APP02.example.com. In the Specify Host Names dialog box, make exactly the same entries that you made in step 4.
- 6 Run the Update Host Name References tool on MID02.example.com. In the Specify Host Names dialog box, make exactly the same entries that you made in step 4.
- 7 Perform the necessary manual steps, as listed in the report. Because the metadata server host name has changed, update the connection profiles for client applications where necessary. If the new deployment will be running concurrently with the old deployment, then the multicast parameters might also need to be modified.
- 8 Start the servers and perform the validation steps.
- 9 Create a complete system backup of the new deployment, including a correct backup of the metadata server.

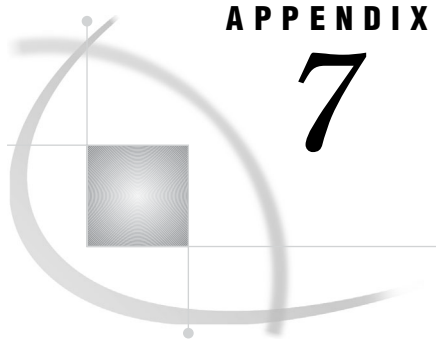
## Example 6: The Metadata Server Has Been Cloned to a New Machine

In this multiple-machine deployment example, cloning software has been used to move the metadata server from a machine called MET01.example.com to a new machine called MET99 example.com. All occurrences of the qualified name, short name, and IP address for the metadata server host machine need to be updated throughout the deployment to refer to the new host machine.



In this scenario, you would use the Update Host Name References tool as follows:

- 1 Create a complete system backup of the deployment, including a correct backup of the metadata server.
- 2 Set up the new machine (MET99.example.com) by using cloning, ghosting, or virtualization software. For details, see “Step 3: Change Existing Names at the Operating System Level, or Set Up New Machines to Be Identical to the Current Machines” on page 305.
- 3 Start the metadata server on the new machine (MET99.example.com). Stop all of the other servers in the deployment.
- 4 Run the Update Host Name References tool on MET99.example.com.  
In the Specify Host Names dialog box, you need to enter only the fully qualified host names. The DNS lookup identifies all other forms of the host names so that they can be updated correctly. Specify the following:
  - ☐ **Old host name:** MET01.example.com
  - ☐ **New host name:** MET99.example.com
  - ☐ **Perform DNS lookup:** Selected
- 5 Run the Update Host Name References tool on APP01.example.com. In the Specify Host Names dialog box, make exactly the same entries that you made in step 4.
- 6 Run the Update Host Name References tool on MID01.example.com. In the Specify Host Names dialog box, make exactly the same entries that you made in step 4.
- 7 Perform the necessary manual steps, as listed in the report. Because the metadata server host name has changed, update the connection profiles for client applications.
- 8 Start the servers and perform the validation steps.
- 9 Create a complete system backup of the deployment, including a correct backup of the metadata server.



## APPENDIX

## 7

## Recommended Reading

---

*Recommended Reading* 319

---

## Recommended Reading

Here is the recommended reading list for this title:

- ❑ *SAS Intelligence Platform: Overview*
- ❑ *SAS Intelligence Platform: Installation and Configuration Guide*
- ❑ *SAS Intelligence Platform: 9.1.3 to 9.2 Migration Guide*
- ❑ *SAS Intelligence Platform: Security Administration Guide*
- ❑ *SAS Intelligence Platform: Application Server Administration Guide*
- ❑ *SAS Intelligence Platform: Desktop Application Administration Guide*
- ❑ *SAS Intelligence Platform: Data Administration Guide*
- ❑ *SAS Intelligence Platform: Web Application Administration Guide*
- ❑ *Grid Computing in SAS*
- ❑ *SAS Companion for UNIX Environments*
- ❑ *SAS Companion for Windows*
- ❑ *SAS Companion for z/OS*
- ❑ *SAS Interface to Application Response Measurement (ARM): Reference*
- ❑ *SAS Language Interfaces to Metadata*
- ❑ *SAS Language Reference: Concepts*
- ❑ *SAS Language Reference: Dictionary*
- ❑ *SAS Logging: Configuration and Programming Reference*
- ❑ *SAS/CONNECT User's Guide*
- ❑ *SAS/SHARE User's Guide*
- ❑ *Scheduling in SAS*

For a complete list of SAS publications, go to **support.sas.com/bookstore**. If you have questions about which titles you need, please contact a SAS Publishing Sales Representative at:

SAS Publishing Sales  
 SAS Campus Drive  
 Cary, NC 27513  
 Telephone: 1-800-727-3228  
 Fax: 1-919-531-9439

E-mail: **[sasbook@sas.com](mailto:sasbook@sas.com)**

Web address: **[support.sas.com/bookstore](http://support.sas.com/bookstore)**

Customers outside the United States and Canada, please contact your local SAS office for assistance.

# Glossary

---

**access control template**

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. Short form: ACT.

**ACT**

See access control template.

**aggregation**

a summary of detail data that is stored with or referred to by a cube.

**ancestor logger**

a logger that is at a higher level in relation to other loggers in the logger hierarchy. For example, the Audit logger is an ancestor logger of Audit.Meta and Audit.Authentication.

**appender**

a named entity that represents a specific output destination for messages. Destinations include fixed files, rolling files, operating system facilities, and client applications.

**Application Response Measurement**

the name of 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. Short form: ARM.

**archive path**

in the Publishing Framework, the complete path of the location to which a user or program can publish an archive package.

**ARM**

See Application Response Measurement.

**authentication**

the process of verifying the identity of a person or process within the guidelines of a specific authorization policy.

**authentication domain**

a SAS internal category that pairs logins with the servers for which they are valid. For example, an Oracle server and the SAS copies of Oracle credentials might all be classified as belonging to an OracleAuth authentication domain.

**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. 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.

**client application**

an application that runs on a client machine.

**client-side pooling**

a configuration in which the client application maintains a collection of reusable workspace server processes.

**connection profile**

a client-side definition of where a metadata server is located. The definition includes a computer name and a port number. In addition, the connection profile can also contain user connection information.

**conversion pattern**

an expression that specifies an appender definition's pattern layout. A conversion pattern consists of a combination of user-supplied literal text and conversion specifiers.

**credentials**

the user ID and password for an account that exists in some authentication provider.



**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**

an optional metadata store for a SAS Metadata Server that can be configured in addition to the foundation repository. Custom repositories are useful for physically segregating metadata for storage or security purposes.

**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 management system**

a software application that enables you to create and manipulate data that is stored in the form of databases. Short form: DBMS.

**DBMS**

See 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.

**descendant logger**

a logger that is at a lower level in relation to other loggers in the logger hierarchy. For example, Audit.Meta and Audit.Authentication are descendant loggers of the Audit logger.

**development environment**

a computing environment in which application developers use software tools to write, compile, and debug programs.

**diagnostic level**

the degree of severity of a log event that can occur during SAS program processing. Examples of levels are trace, debug, informational, warning, error, and fatal. Short form: level.

**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".

**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.

**fileref**

a name that is temporarily assigned to an external file or to an aggregate storage location such as a directory or a folder. The fileref identifies the file or the storage location to SAS.

**format**

See SAS format.

**foundation repository**

the required metadata store for a SAS Metadata Server. Each SAS Metadata Server has one foundation repository that is created by default when the metadata server is configured.

**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.

**grid computing**

a type of computing in which large computing tasks are distributed among multiple computers on a network.

**home directory**

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 provides a user-friendly view of a data source. When you use an information map to query data for business needs, you do not have to understand the structure of the underlying data source or know how to program in a query language.

**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. Short form: IOM.

**Integrated Object Model server**

a SAS object server that is launched in order to fulfill client requests for IOM services. Short form: IOM server.

**Integrated Windows authentication**

a Microsoft technology that facilitates use of authentication protocols such as Kerberos. In the SAS implementation, all participating components must be in the same Windows domain or in domains that trust each other.

**internal account**

a SAS account that you can create as part of a user definition. Internal accounts are intended for metadata administrators and some service identities; these accounts are not intended for regular users.

**IOM**

See Integrated Object Model.

**IOM server**

See Integrated Object Model server.

**Java application**

a stand-alone program that is written in the Java programming language.

**job**

a collection of SAS tasks that can create output.

**libref**

a SAS name that is associated with the location of a SAS library. For example, in the name MYLIB.MYFILE, MYLIB is the libref, and MYFILE is a file in the SAS library.

**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 English\_USA.

**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.

**logger**

a named entity that identifies a message category. Logger names have a hierarchical format that enables you to configure logging at a broad or a fine-grained level.

**logging configuration**

an XML file or a set of SAS program statements that determines how log events are processed. You use the logging configuration to assign thresholds to loggers, to configure appenders, and to specify which categories and levels of log events are written to each appender.

**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 SAS copy of information about an external account. Each login includes a user ID and belongs to one SAS user or group. Most logins do not include a password.

**metadata**

data about data. For example, metadata typically describes resources that are shared by multiple applications within an organization. These resources can include software, servers, data sources, network connections, and so on. Metadata can also be used to define application users and to manage users' access to resources. Maintaining metadata in a central location is more efficient than specifying and maintaining the same information separately for each application.

**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 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.

**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 repository**

a collection of related metadata objects, such as the metadata for a set of tables and columns that are maintained by an application. A SAS Metadata Repository is an example.

**metadata server**

a server that provides metadata management services to one or more client applications. A SAS Metadata Server is an example.

**middle tier**

in a SAS business intelligence system, the architectural layer in which Web applications and related services execute. The middle tier receives user requests, applies business logic and business rules, interacts with processing servers and data servers, and returns information to users.

**migrate**

to populate a new deployment of SAS software with the content, data, or metadata (or a combination of these) from an existing deployment. Migrating might include upgrading to a new software release, converting data or metadata, or other changes to ensure compatibility.

**migration**

the process of moving data to a different server or storage device, converting the data for use in a different operating environment, or both. In a broader sense, the migration process can also include installing the newer software and converting applications for use with that software.

**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**

See online analytical processing.

**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.

**online analytical processing**

a software technology that enables users to dynamically analyze data that is stored in multidimensional database (MDDb) tables. Short form: OLAP.

**planned deployment**

a method of installing and configuring a SAS business intelligence system. This method requires a deployment plan that contains information about the different

hosts that are included in the system and the software and SAS servers that are to be deployed on each host. The deployment plan then serves as input to an installation and configuration tool called the SAS Deployment Wizard.

**plug-in**

a file that modifies, enhances, or extends the capabilities of an application program. The application program must be designed to accept plug-ins, and the plug-ins must meet design criteria specified by the developers of the application program. In SAS Management Console, a plug-in is a JAR file that is installed in the SAS Management Console directory to provide a specific administrative function. The plug-ins enable users to customize SAS Management Console to include only the functions that are needed.

**pool**

a group of server connections that can be shared and reused by multiple client applications. A client-side pool consists of one or more puddles.

**pooling**

the act or process of creating a pool.

**portlet**

a Web component that is managed by a Web application and that is aggregated with other portlets to form a page within the application. Portlets can process requests from the user and generate 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.

**project repository**

a metadata repository that serves as an individual work area or playpen. Project repositories are available for SAS Data Integration Studio only. In general, each user who participates in change management has his or her own project repository.

**promote**

to copy selected metadata and associated content within or between planned deployments of SAS software. This promotion process is repeatable for a particular deployment.

**promotion**

the process of copying selected metadata and associated content within or between planned deployments of SAS software. This process is repeatable for a particular deployment.

**publish**

to deliver electronic information, such as SAS files (including SAS data sets, SAS catalogs, and SAS data views), other digital content, and system-generated events to one or more destinations. These destinations can include e-mail addresses, message queues, publication channels and subscribers, WebDAV-compliant servers, and archive locations.

**puddle**

a group of servers that are started and run using the same login credentials. Each puddle can also allow a group of clients to access the servers.

**resource template**

an XML file that specifies the information that is needed for creating a metadata definition for a SAS resource.

**root logger**

the highest-level logger in the logger hierarchy. In a logging configuration, all other loggers inherit the root logger's attributes.

**SAS catalog**

a SAS file that stores many different kinds of information in smaller units called catalog entries. A single SAS catalog can contain different types of catalog entries.

**SAS configuration directory**

the location where configuration information for a SAS deployment is stored. The configuration directory contains configuration files, logs, scripts, repository files, and other items for the SAS software that is installed on the machine.

**SAS Content Server**

a server that stores digital content (such as documents, reports, and images) that is created and used by SAS client applications. To interact with the server, clients use WebDAV-based protocols for access, versioning, collaboration, security, and searching.

**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.

**SAS Deployment Manager**

a cross-platform utility that manages SAS deployments. The SAS Deployment Manager supports functions such as updating passwords for your SAS deployment, rebuilding SAS Web applications, and removing configurations.

**SAS Deployment Wizard**

a cross-platform utility that installs and initially configures many SAS products. Using a SAS installation data file and, when appropriate, a deployment plan for its initial input, the wizard is designed to prompt the customer for all the remaining input at the start of the session so that the customer does not have to monitor an entire deployment.

**SAS format**

a type of SAS language element that applies a pattern to or executes instructions for a data value to be displayed or written as output. Types of formats correspond to the data's type: numeric, character, date, time, or timestamp. The ability to create user-defined formats is also supported. Examples of SAS formats are BINARY and DATE. Short form: format.

**SAS Foundation**

the superset of all SAS software that is installable with the Base SAS installation. For a particular customer, the SAS Foundation is a collection of software consisting of the members of that superset required to support the deployment that the customer desires.

**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.

**SAS library**

a collection of one or more files that are recognized by SAS and that are referenced and stored as a unit. Each file is a member of the library.

**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 Pooled Workspace Server**

a SAS Workspace Server that is configured to use server-side pooling. In this configuration, the SAS object spawner maintains a collection of workspace server processes that are available for clients.

**SAS Stored Process**

a SAS program that is stored on a server and defined in metadata, and which can be executed by client applications. Short form: stored process.

**SAS Stored Process Server**

a SAS IOM server that is launched in order to fulfill client requests for SAS Stored Processes.

**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.

**SAS Table Server**

a data access server that supports the capabilities provided by SAS Shared Services for the SAS BI Dashboard, SAS Web Report Studio, and other SAS solutions.

**SAS token authentication**

a process in which the metadata server generates and verifies SAS identity tokens to provide single sign-on to other SAS servers. Each token is a single-use, proprietary software representation of an identity.

**SAS Workspace Server**

a SAS IOM server that is launched in order to fulfill client requests for IOM workspaces.

**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 libraries.

**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 Pooled 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.

**server-side pooling**

a configuration in which a SAS object spawner maintains a collection of reusable workspace server processes that are available for clients. The usage of servers in this pool is governed by the authorization rules that are set on the servers in the SAS metadata.

**source metadata server**

the metadata server from which metadata is promoted.

**spawner**

See object spawner.

**stored process**

See SAS Stored Process.

**subscribe**

to sign up to receive electronic content that is published to a SAS publication channel.



**target metadata server**

the metadata server to which metadata is promoted.

**thread**

a single path of execution of a process that runs on a core on a CPU.

**threshold**

the lowest event level that is processed. Log events whose levels are below the threshold are ignored.

**transformation**

in data integration, an operation that extracts data, transforms data, or loads data into data stores.

**trusted user**

a privileged service account that can act on behalf of other users on a connection to the metadata server.

**UNC**

See Universal Naming Convention.

**Universal Naming Convention**

a naming standard that is used to specify the file or directory path for networked computing resources such as servers, databases, and peripheral devices. UNC uses the format *\\server-name\shared-resource-pathname*. Short form: UNC.

**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**

an application that is accessed via a Web browser over a network such as the Internet or an intranet. SAS Web applications are Java Enterprise Edition (JEE) applications that are delivered via Web application archive (WAR) files. The applications can depend on Java and non-Java Web technologies.

**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. Short form: WebDAV.

**WebDAV**

See Web Distributed Authoring and Versioning.

**WebDAV repository**

a collection of files that are stored on a Web server so that authorized users can read and edit them.

**workspace**

in the IOM object hierarchy for a SAS Workspace Server, an object that represents a single session in SAS.



# Index

---

## Numbers

- 32-bit Windows systems
  - metadata server performance and 174

## A

- absolute date 245
- absolute date and time 245
- access controls
  - promoting 221
- access modes
  - metadata repository 77, 182
- access state
  - checking with METADATA procedure 80
  - checking with SAS Management Console 79
  - metadata repositories 77, 79, 80, 182
  - repository manager 77, 78, 80
  - SAS Metadata Server 76
  - SAS Metadata Server, checking with METAOPERATE procedure 76
  - SAS Metadata Server, checking with SAS Management Console 76
  - SAS Metadata Server, effects on metadata repositories 78
- accessibility features 4
- Administration access mode 183
- Administration state 53
- administration tasks 39
  - first-priority setup tasks 40
  - ongoing tasks 46
  - optional setup tasks 44
  - standard setup tasks 43
- adminUsers.txt file 23
- AppData subdirectory 20
- appenders 104
- application server subdirectory (SASApp or SASMain) 21
  - contents of 266
- ARM log information 44
  - collecting in SAS OLAP Server 115
  - configuring DATA step batch servers, Windows or UNIX 114
  - configuring DATA step batch servers, z/OS 114
  - for SAS Data Integration Studio batch jobs 44
  - for SAS Data Integration Studio jobs 113
- ARM logging facility 110
- associations
  - among objects in SAS folders 202
  - restoring metadata associations 220
- auditing
  - server logging level changes 116
- authentication
  - workspace server access and 37
- Authorization Manager plug-in 8

## B

- Backup Wizard 128, 137
  - defining backup jobs and corresponding restore jobs 140
  - managing backup and restore jobs 142
  - options and restrictions 138
  - REORG option 130
  - RUNANALYSIS option 130
  - running backup or restore jobs 143
  - scheduling backup jobs 145
  - summary of procedures for 138
  - troubleshooting 144
  - user credentials for 139
- BACKUPMETADATAJOURNALFILE= option
  - %OMABAKUP macro 159
- backups 123
  - See also* %OMABAKUP macro
  - alternating destinations program, Windows 154
  - alternating locations 130
  - before and after major changes 129
  - best practices for 127
  - data stored in file systems 132
  - defining jobs 140
  - destination for 129
  - full backup program (example) 153
  - full metadata server backup 151
  - managing jobs 142
  - metadata repositories 128
  - operating system commands for SAS Metadata Server 126
  - optimizing SAS Metadata Server memory usage 130
  - process for 40
  - reclaiming unused disk space 130
  - regularly scheduled full backups 128
  - repository manager 128
  - restoring repositories and repository manager 155
  - running jobs 143
  - SAS Content Server 131
  - SAS Metadata Server utilities for 124
  - SAS Table Server 131
  - scheduling jobs 145
  - specific SAS folders 126
  - synchronizing metadata backups 132
  - when metadata server demand is low 129
- backupServer.sas program 150

- batch export command 246
- batch export tool 214, 238
  - batch export command 246
  - importing packages created by 253
  - preparing to use 239
  - specifying absolute date or date and time 245
  - specifying object types 244
  - specifying relative date 245
  - specifying relative time period 246
  - substitution properties file 247
  - syntax 239
  - syntax description 240
  - troubleshooting 253
  - using 239
  - viewing log 252
- batch import command 251
- batch import tool 214, 238
  - batch import command 251
  - importing metadata from SAS 9.1.3 225
  - importing packages created by Export SAS Package wizard 252
  - importing packages created in SAS 9.1.3 253
  - preparing to use 239
  - syntax 248
  - syntax description 248
  - troubleshooting 253
  - using 248
  - viewing log 252
- batch jobs
  - ARM log information for 44
- BatchServer subdirectory 21
- best practices
  - associations among objects in SAS folders 202
  - backups 127
  - folder management 201
  - folders 47
  - restores 134
  - system integrity 47
- BI Lineage plug-in 203
  - granting permission to view 205
  - running and scheduling scans 205
  - uses 208
  - viewing scan results 206
- BILineage repository 182, 205
- BMC Performance Manager 101
- boot script
  - installing sas.servers script as 60

## C

- cache
  - clearing credentials cache 37
- capturing
  - metadata server management events in system facility appender 117
  - XML information in SAS Metadata Server log 117
- centralized administration 33
- change analysis 187
- change management
  - for SAS Data Integration Studio 44
- client applications
  - auditing server logging level changes 116
  - configuration files 24
- clients
  - concurrent 174
- Clients tab
  - Server Manager 91
- concurrent queries/clients 174
- configuration
  - concurrent queries/clients 174
  - DATA step batch servers, Windows or UNIX 114
  - DATA step batch servers, z/OS 114
  - initial logging configuration for SAS servers 107
  - logging XML information 118
  - metadata server journaling 170
  - metadata server performance and 167
  - modifying for processing servers 44
  - modifying server logging configurations 110
  - number of threads for metadata server 167
  - SAS Metadata Server initial configuration 14
  - securing 40
- configuration directory 263
  - AppData subdirectory 20
  - application server subdirectory (SASApp or SAS-Main) 21, 266
  - default paths 19
  - Levn subdirectory 20, 263
  - metadata server subdirectory (SASMeta) 21, 265
  - protections on UNIX and z/OS 42
  - protections on Windows 40
  - structure 19
  - Web subdirectory 22, 267
- configuration files 22, 271
  - client applications 24
  - components of SAS Application Servers 274
  - object spawners 279
  - omaconfig.xml 22, 280
  - SAS/CONNECT spawners 279
  - SAS Metadata Server 22, 272
  - SAS servers 23
  - SAS/SHARE servers 277
  - security-related 23
  - table servers 277
- Configuration Manager plug-in 8
- connectivity 43
- content
  - promoting along with metadata 218
- content mapping 198
  - setting up for promotion tools 227
- copying metadata 185
- credential-based authentication
  - workspace server access and 37
- credentials
  - for Backup Wizard 139
  - for %OMABAKUP macro 149
  - monitoring SAS servers with SAS Management Console 89
  - operating SAS servers with SAS Management Console 65
  - promotion tools 226
- credentials cache
  - clearing 37, 65
- cubes
  - creating directories and moving physical files for 228
  - promoting from SAS 9.1.3 224
- custom folder structures 208
  - examples 210
  - guidelines for creating 208
- custom repositories 13, 181
  - creating 183

**D**

- daemons
  - running servers as UNIX daemons 61
- Data Library Manager plug-in 8
- data sources
  - establishing connectivity to 43
- DATA step batch servers
  - collecting ARM log information, Windows or UNIX 114
  - collecting ARM log information, z/OS 114
- Data subdirectory 21
- DataFlux Integration Server 17
- dates
  - absolute date 245
  - absolute date and time 245
  - relative date 245
- default SAS ports 259
  - port numbers for SAS servers and spawners 259
  - port numbers for third-party software 260
- default server log locations 269
- delegated administration 33
- dependencies
  - between repositories 13
  - server dependencies 52, 55
- desktop applications
  - Java heap memory for 44
- DESTINATIONPATH option
  - %OMARUNAN macro 177
  - %OMABAKUP macro 157
- development environment 44
- directories
  - See also* configuration directory
  - creating for cubes 228
  - creating for physical files on target system 228
  - creating for prompts 228
- DISABLECONFIGURATIONFILEBACKUP= option
  - %OMABAKUP macro 158
- DISABLOFFLINEBACKUP= option
  - %OMABAKUP macro 159
- DISABLEREADONLYBACKUP= option
  - %OMABAKUP macro 158
- disk space
  - reclaiming unused space 130

**E**

- error logging
  - for %OMABAKUP macro 151
- Export SAS Package wizard 214
  - accessing 229
  - backing up specific SAS folders 126
  - example usage scenario 229
  - importing packages created by 252
  - troubleshooting 253
- exporting metadata 185, 187

**F**

- file systems
  - backing up data stored in 132
- first-priority setup tasks 40
- folder structure 197
  - customizing 208
  - initial structure for migrated systems 200
  - initial structure for new installations 198
  - purpose of 197

- folders 47
  - associations among objects in 202
  - backing up specific SAS folders 126
  - best practices 47, 201
  - home folders 199
  - metadata folder structure 43
  - moving SAS 9.1.3 tables and libraries to 224
  - personal folders 199
  - setting up for promotion 227
  - setting up security for promotion tools 227
- Folders tab 198
  - SAS Management Console 9, 47
- foundation repositories 13, 14, 181
  - replicating 185
- Foundation Services Manager plug-in 8

**G**

- generate\_boot\_scripts.sh script 59
  - regenerating sas.servers script 59
- groups
  - adding users to SAS Administrators 36
  - in Metadata Manager plug-in 34
  - in Server Manager plug-in 35

**H**

- heap memory 44
- Hobbit 101
- home folders 199
- host authentication
  - workspace server access and 37
- host environment
  - porting metadata repository to different host 189
  - recommended parameters for metadata server invocation options 289
- host name references, updating 296
  - See also* Update Host Name References tool
  - client software components requiring changes to host name references 301
  - example scenarios 312
  - process for 304
- HP OpenView 101

**I**

- IBM Tivoli 101
- IBM WebSphere Application Server
  - starting or stopping, Windows or UNIX 70
- IBM WebSphere Application Server Network Deployment
  - starting or stopping, Windows or UNIX 70
  - starting or stopping, z/OS 71
- Import SAS Package wizard 214
  - accessing 229
  - backing up specific SAS folders 126
  - example usage scenario 229
  - importing packages created by batch export tool 253
  - troubleshooting 253
- importing
  - metadata 185, 187
  - metadata from SAS 9.1.3 225
  - packages 252, 253
  - packages created in SAS 9.1.3 253
  - SPK file 235
- information fields 96
- initial user accounts

- See* user accounts, initial
- input/output balancing 173
- Integrated Windows authentication (IWA)
  - workspace server access and 37
- invocation options
  - See* SAS Metadata Server invocation options

## J

- Java heap memory 44
- JBoss Application Server
  - starting or stopping 69
- jobs
  - collecting ARM log information for 113
  - defining backup jobs and corresponding restore job 140
  - managing backup and restore jobs 142
  - running backup or restore jobs 143
  - scheduling 44
- journaling 14, 170
  - changing journaling options 170
  - configuring metadata server journaling 170
  - moving journal file 171

## L

- Levn subdirectory 20
  - contents of 263
- libraries
  - moving SAS 9.1.3 libraries to folders 224
- log files
  - creating for SAS Workspace Server troubleshooting 112
- Log tab
  - Server Manager 94
- logconfig.xml file 23
- loggers 104
  - SAS Metadata Server loggers 104
- Loggers tab
  - Server Manager 93
- logging 44
  - See also* ARM log information
  - See also* server logging
  - error logging for %OMABAKUP macro 151
- logging facility 103
- logging levels
  - adjusting 91
  - auditing changes in 116
  - changing for individual loggers 93
  - returning to normal levels 96
- logical servers 15
- logs
  - default locations for server logs 269
- Logs directories 21, 22
- LSF Administrator 31
- LSF Services group 27
- lsfuser account 32

## M

- Management Console: Advanced role 26
- Management Console: Content Management role 26
- Map Service Manager plug-in 8
- MAXACTIVETHREADS option 167
  - specifying values for 169
- memory
  - for promotion tools 226
  - how optimization works 175

- input/output balancing 173
- Java heap memory 44
- optimizing metadata server usage with
  - %OMABAKUP 175
- optimizing metadata server usage with
  - %OMARUNAN 175
- optimizing usage for SAS Metadata Server 130
- recovering quickly on metadata server 177
- setting MEMSIZE parameter for metadata server 173
  - when to optimize usage for metadata server 175
- MEMSIZE parameter
  - setting for metadata server 173
- metadata
  - copying 185
  - exporting 185, 187
  - importing 185, 187
  - importing from SAS 9.1.3 225
  - promoting 185, 221
  - promoting content along with 218
  - promoting from SAS 9.1.3 to 9.2 223
  - setting up folder structure 43
  - synchronizing backups 132
  - user groups initially defined in 26
- metadata associations
  - restoring 220
- Metadata Manager Export Wizard 187
- Metadata Manager Import Wizard 187
- Metadata Manager plug-in 8
  - group and role memberships 34
  - SAS Metadata Server and 11
  - who can do what 34
- METADATA procedure
  - checking access state of metadata repositories 80
  - checking access state of repository manager 80
  - creating and running programs 80
  - example output 81
- metadata repositories 181
  - See also* repositories
  - access modes 77, 182
  - access states 77
  - access states, and SAS Metadata Server 78
  - access states, checking with METADATA procedure 80
  - access states, checking with SAS Management Console 79
  - backing up 128
  - changing the properties of 184
  - copying metadata 185
  - custom repositories 181, 183
  - deleting 193
  - exporting metadata 185
  - foundation repositories 181
  - importing metadata 185
  - metadata server and access to 47
  - moving to new location on same metadata server 188
  - porting to different host environment 189
  - project repositories 182
  - promoting metadata 185
  - registering 191
  - tools for managing 185
  - unregistering 192
- metadata server
  - See* SAS Metadata Server
- Metadata Server: Operation role 25
- metadata server subdirectory (SASMeta) 21
  - contents of 265
- Metadata Server: Unrestricted role 24

- Metadata Server: User Administration role 25
- MetadataRepositories subdirectory 22, 47
- MetadataServer subdirectory 22
- METAOPERATE procedure
  - checking access state of SAS Metadata Server 76
  - pausing metadata server to Offline state 133
  - resuming metadata server 134
- Microsoft Office Manager 101
- middle-tier hosts
  - configuration directory 19
- migration 215
  - initial folder structure for migrated systems 200
- multiple-level environment
  - port numbering in 261
- My Folder 198, 199

## N

- NEWRPOS option
  - %OMAPORT macro 190
- NULL option
  - %OMAPORT macro 190

## O

- object spawner
  - See* SAS object spawner
- objects
  - associations among objects in SAS folders 202
  - promoting from SAS 9.1.3 to 9.2 224
  - security for promotion tools 227
  - specifying type for batch export tool 244
- Offline access mode 183
- Offline state 53, 133
- OLAP cubes
  - creating directories and moving physical files for 228
  - promoting from SAS 9.1.3 224
- OLAP servers
  - See* SAS OLAP Server
- OLDRPOS option
  - %OMAPORT macro 189
- %OMABAKUP macro 128, 147
  - alternating destinations program, Windows 154
  - error logging 151
  - full backup program (example) 153
  - full metadata server backup 151
  - functionality of 148
  - optimizing metadata server memory usage 175
  - options 157
  - reference 156
  - REORG option 130
  - restoring all repositories 155
  - restoring repository manager 155
  - RUNANALYSIS option 130
  - running backupServer.sas 150
  - running restoreServer.sas 150
  - server connection statement 156
  - user credentials for 149
  - warning messages 155
  - writing custom programs 151
- %OMABAKUP macro statement 157
- omaconfig.xml file 22, 280
  - InternalAuthenticationPolicy options 287
  - OMA options 281
  - RPOSMGR options 286

- %OMAPORT macro
  - details 190
  - porting metadata repository to different host 189
  - reference 189
  - syntax 189
- %OMARUNAN macro
  - optimizing metadata server memory usage 175, 176
  - reference 177
  - syntax 177
- ongoing system administration tasks 46
- Online access mode 183
- Online state 53
- operating system
  - configuration directory protections, UNIX and z/OS 42
  - configuration directory protections, Windows 40
  - user groups initially defined in 26
- operating system commands
  - backing up SAS Metadata Server 126
- operating system facilities
  - logging 110
- optional setup tasks 44
- Options tab
  - Server Manager 93
- Oracle WebLogic Server
  - starting or stopping 72

## P

- packages 216
  - importing 252, 253
- pattern layouts 104
- pausing
  - SAS Metadata Server 65
  - SAS object spawner 66
  - SAS OLAP Server 67
- performance
  - configuration settings for metadata server 167
  - configuring concurrent queries/clients 174
  - configuring number of threads used by metadata server 167
  - input/output balancing 173
  - metadata server on 32-bit Windows systems 174
  - optimizing for metadata server 44
  - setting MEMSIZE parameter for metadata server 173
  - Web applications 44
- performance counters 96
- personal folders 199
- physical files
  - moving for cubes 228
  - moving for prompts 228
  - synchronizing metadata backups 132
- plug-ins 7
  - steps for using 9
- pooled workspace servers
  - See* SAS Pooled Workspace Servers
- ports 19
  - default port numbers for SAS servers and spawners 259
  - default port numbers for third-party software 260
  - default SAS ports 259
  - numbering in multiple-level environment 261
- processes
  - controlling on servers 68
- processing servers
  - modifying configuration of 44
- Products folder 199
- project repositories 13, 182

promotion 214  
     *See also* promotion tools  
     access controls 221  
     content along with metadata 218  
     ensuring correct software version when exporting from SAS 9.1.3 224  
     importing metadata from SAS 9.1.3 225  
     metadata 185, 221  
     metadata, from SAS 9.1.3 to 9.2 223  
     moving SAS 9.1.3 tables and libraries to folders 224  
     object types for 215  
     OLAP cubes from SAS 9.1.3 224  
     process of 215  
     prompts from SAS 9.1.3 224  
     restoration of metadata associations 220  
     selective 217  
 promotion tools 214  
     accessing 229  
     creating directories and moving physical files for cubes 228  
     creating directories and moving physical files for prompts 228  
     creating directories for physical files on target system 228  
     credentials for 226  
     memory for 226  
     preparing for use 226  
     setting up content mapping on target metadata server 227  
     setting up folders on target metadata server 227  
     setting up security for folders or objects 227  
     setting up servers 226  
     starting spawners 226  
     troubleshooting 253  
 prompts  
     creating directories and moving physical files for 228  
     promoting from SAS 9.1.3 224  
 properties  
     changing metadata repository properties 184  
 PUBLIC user group 27  
 Publishing Framework plug-in 8

## Q

queries  
     concurrent 174  
     viewing 101  
 quiescing  
     SAS object spawner 66  
     SAS OLAP Server 67

## R

ReadOnly access mode 183  
 refreshing  
     SAS object spawner 66  
     SAS OLAP Server 67  
 registering  
     metadata repositories 191  
     unregistering metadata repositories 192  
 relative date 245  
 relative time period 246  
 REORG option  
     reclaiming unused disk space 130  
     %OMABAKUP macro 158  
 replication 185  
 Replication Wizard 185

reports  
     scheduling 44  
     setting up content mapping for 227  
 repositories 12  
     *See also* metadata repositories  
     *See also* SAS Metadata Repositories  
     dependencies between 13  
     example program to restore all repositories 155  
 repository manager 12, 181  
     access state 78  
     access state, checking with METADATA procedure 80  
     backing up 128  
     example program to restore 155  
 required servers 52  
 resetting  
     SAS Metadata Server 65  
 restarting all servers  
     UNIX or z/OS 58  
 RESTORE= option  
     %OMABAKUP macro 159  
 restores 123  
     *See also* Backup Wizard  
     *See also* %OMABAKUP macro  
     all repositories and repository manager 155  
     best practices for 134  
     defining jobs 140  
     full restores 134  
     managing jobs 142  
     manually restoring unresponsive metadata server 161  
     of metadata associations 220  
     running jobs 143  
     SAS Metadata Server utilities for 124  
     unresponsive metadata server 135  
 restoreServer.sas program 150  
 resuming  
     SAS Metadata Server 65  
     SAS object spawner 66  
     SAS OLAP Server 67  
 roles 24  
     in Metadata Manager plug-in 34  
     in Server Manager plug-in 35  
     Management Console: Advanced 26  
     Management Console: Content Management 26  
     Metadata Server: Operation 25  
     Metadata Server: Unrestricted 24  
     Metadata Server: User Administration 25  
     user roles for system administration 24  
 rolling log files 108  
 rposmgr subdirectory 22  
 RPOSMGRPATH= option  
     %OMABAKUP macro 158  
 rpsomgr subdirectory 47  
 RUNANALYSIS option  
     %OMABAKUP macro 175  
     optimizing metadata server memory usage 130  
     %OMABAKUP macro 159

## S

SAS 9.1.3  
     correct software version when exporting from 224  
     importing metadata from 225  
     importing packages with batch import tool 253  
     moving tables and libraries to folders 224  
     promoting metadata to 9.2 223  
     promoting OLAP cubes from 224



- promoting prompts from 224
- SAS 9.2
  - promoting metadata from 9.1.3 223
- SAS Administrator user account 28
- SAS Administrators group 27
  - adding users to 36
- SAS Anonymous Web User 31
- SAS Application Server 15
  - components 16
  - configuration files for components of 274
  - hierarchy 15
- SAS batch servers 16
- SAS/CONNECT servers 16
  - configuration files for 274
- SAS/CONNECT spawner 17
  - configuration files for 279
  - operating with individual script 63
- SAS Content Server 17
  - backing up 131
  - synchronizing metadata backups with 132
  - validating 83
- SAS Data Integration Studio
  - ARM log information for batch jobs 44
  - ARM log information for jobs 113
  - change management for 44
- SAS Deployment Manager 10
  - Update Host Name References tool 296
- SAS Deployment Tester server 18
  - operating with individual script 63
- SAS Deployment Wizard
  - enabling server logging 106
- SAS First User 30
- SAS folder structure
  - See* folder structure
- SAS folders
  - See* folders
- SAS Folders 198
- SAS General Servers group 27
- SAS Grid Servers 16
- SAS In-Process Services 17
- SAS Installer account 29
- SAS Intelligence Platform 3
- SAS Management Console 7
  - BI Lineage plug-in 203
  - checking access state of metadata repositories 79
  - checking access state of SAS Metadata Server 76
  - connecting to servers 89
  - controlling processes on servers 68
  - creating custom repositories 183
  - displaying server monitoring information 89
  - Folders tab 9, 47
  - logging 109
  - logging levels 91, 93, 96
  - monitoring SAS servers 88
  - obtaining monitoring information 91
  - operating SAS Metadata Server 65
  - operating SAS object spawner 66
  - operating SAS OLAP Server 67
  - operating SAS servers 65
  - plug-ins 7
  - plug-ins, steps for using 9
  - SAS OLAP Server Monitor 100
  - server management features 88
  - Server Manager tabs 91
  - workspace server access 37
- SAS Metadata Bridges 187
- SAS Metadata Repositories 12
  - See also* metadata repositories
  - BI lineage repository 182, 205
  - custom repositories 13
  - foundation repositories 13
  - project repositories 13
- SAS Metadata Server 11
  - See also* SAS Metadata Server invocation options
  - access state 76
  - access state, and metadata repositories 78
  - access state, checking with METAOPERATE procedure 76
  - access state, checking with SAS Management Console 76
  - backing up with operating system commands 126
  - backup and restore programs for 150
  - backup and restore utilities 124
  - backups when demand is low 129
  - capturing management events 117
  - capturing XML information in log 117
  - configuration files for 22, 272
  - configuring concurrent queries/clients 174
  - configuring journaling 170
  - configuring number of threads used by 167
  - custom repositories 13
  - foundation repositories 13
  - full backups 151
  - how it works 13
  - initial configuration 14
  - input/output balancing 173
  - loggers 104
  - manually restoring when unresponsive 161
  - Metadata Manager plug-in and 11
  - metadata repository access and 47
  - moving repositories to new location on same server 188
  - operating with individual script 61
  - operating with SAS Management Console 65
  - optimizing memory usage 130
  - optimizing memory usage with %OMABAKUP 175
  - optimizing memory usage with %OMARUNAN 175
  - optimizing performance 44
  - optimizing performance and reliability 44
  - pausing 65
  - pausing to Offline state 133
  - performance on 32-bit Windows systems 174
  - performance-related configuration settings 167
  - project repositories 13
  - recovering memory quickly 177
  - repository manager 12
  - resetting 65
  - restoring repositories and repository manager when unresponsive 135
  - resuming 65, 134
  - SAS Metadata Repositories 12
  - SASMeta server context and 12
  - setting MEMSIZE parameter 173
  - states 53
  - stopping 65
  - validating 82
  - when to optimize memory usage 175
- SAS Metadata Server invocation options 289
  - additional UNIX parameters 292
  - additional Windows parameters 292
  - additional z/OS parameters 292
  - modifying 289
  - optional parameters 292

- recommended parameters 289
- recommended parameters for all host environments 289
- SAS object spawner 17
  - checking status of 82
  - configuration files for 279
  - detailed logging for troubleshooting 113
  - information fields 96
  - operating with individual script 61
  - operating with SAS Management Console 66
  - pausing 66
  - performance counters 96
  - quiescing 66
  - refreshing 66
  - resuming 66
  - stopping 66
- SAS OLAP Server 16
  - collecting ARM log information in 115
  - configuration files for 274
  - operating with individual script 61
  - operating with SAS Management Console 67
  - pausing 67
  - quiescing 67
  - refreshing 67
  - resuming 67
  - stopping 67
  - validating 82
- SAS OLAP Server Monitor 100
  - accessing 100
  - refreshing server connections 101
  - refreshing server list 101
  - viewing sessions and queries 101
- SAS Open Metadata Architecture 11
- SAS Pooled Workspace Servers 16
  - configuration files for 274
  - validating 82
- SAS Server Users group 26
- SAS servers
  - See also* server logging
  - configuration files for 23, 271
  - default port numbers 259
  - information fields 96
  - initial logging configuration 107
  - monitoring 87
  - monitoring with SAS Management Console 88
  - monitoring with third-party products 101
  - operating individually with scripts 61
  - operating with SAS Management Console 65
  - performance counters 96
  - start-up parameters 53
  - starting and stopping, z/OS 57
- SAS Services Application
  - starting or stopping with script 64
- SAS/SHARE server 17
  - configuration files for 277
  - operating with individual script 63
- SAS Spawned Servers account 30
- SAS Stored Process Servers 16
  - configuration files for 274
  - validating 82
- SAS System Services group 27
- SAS Table Server 17
  - backing up 131
  - configuration files for 277
  - operating with individual script 61
  - synchronizing metadata backups with 132
- SAS Trusted User 29
- sas user group 26
- SAS Web Administration Console 10
- SAS Workspace Servers 16
  - configuration files for 274
  - creating log files for troubleshooting 112
  - requirements for accessing 37
  - validating 82
- SASApp subdirectory 21
  - contents of 266
- SASBackup directory 22
- SASEnvironment subdirectory 21
- SASGRP user group 26
- SASMain subdirectory 21
  - contents of 266
- SASMeta DATA step batch server 12
- SASMeta server context 12
- SASMeta subdirectory 21
  - contents of 265
- SASMeta workspace server 12
- sas.servers script 58
  - displaying server status, UNIX or z/OS 58
  - installing as boot script 60
  - regenerating 59
  - restarting all servers, UNIX or z/OS 58
  - starting or stopping all servers, UNIX 58
  - starting or stopping all servers, UNIX or z/OS 58
  - starting or stopping all servers, z/OS 58
- SASUSERS user group 27
- sasv9.cfg file 23
- sasv9\_usermods.cfg file 23
- Schedule Manager plug-in 8
  - scheduling backup jobs created by Backup Wizard 145
- scheduling
  - backup jobs created by Backup Wizard 145
- scripts
  - displaying status of server or spawner 76
  - generate\_boot\_scripts.sh 59
  - operating SAS/CONNECT spawner 63
  - operating SAS Deployment Tester server 63
  - operating SAS Metadata Server 61
  - operating SAS object spawner 61
  - operating SAS OLAP Server 61
  - operating SAS servers individually 61
  - operating SAS/SHARE server 63
  - operating SAS Table Server 61
  - operating servers with 61
  - sas.servers 58, 60
  - starting or stopping SAS Services Application 64
- security
  - best practices for system integrity 47
  - configuration directory protections, UNIX and z/OS 42
  - configuration directory protections, Windows 40
  - configuration files 23
  - first-priority setup tasks 40
  - for system administration tasks 33
  - optional setup tasks 44
  - setting up for promotion tools 227
  - standard setup tasks 43
- selective promotion 217
- server connection statement
  - %OMABAKUP macro 156
- server context 15
- server dependencies 52
  - running servers as Windows services 55

- server logging 44, 103
  - ARM log information for SAS Data Integration Studio jobs 113
  - ARM log information in SAS OLAP Server 115
  - auditing logging level changes 116
  - capturing metadata server management events 117
  - capturing XML information in SAS Metadata Server log 117
  - default locations for server logs 269
  - detailed logging for SAS object spawner troubleshooting 113
  - detailed logging for server troubleshooting 111
  - enabling 106
  - initial configuration for SAS servers 107
  - log files for SAS Workspace Server troubleshooting 112
  - modifying configurations 110
  - SAS Metadata Server loggers 104
- Server Manager plug-in 8
  - Clients tab 91
  - group and role memberships 35
  - Log tab 94
  - Loggers tab 93
  - Options tab 93
  - SASMeta server context 12
  - Sessions tab 92
  - Spawned Server Activity tab 92
  - tabs 91
  - who can do what 35
- server-name.srv files 60
- server performance counters 96
- server scripts
  - displaying status of server or spawner 76
- server status 75
  - access modes 182
  - access state of metadata repositories 77
  - access state of repository manager 77
  - access state of SAS Metadata Server 76
  - displaying with server script 76
  - object spawner status 82
  - UNIX or z/OS 58
  - validating SAS Content Server 83
  - validating servers 82
- server-tier hosts
  - configuration directory 19
- servers 6, 15, 17
  - See also* server status
  - connecting to, with SAS Management Console 89
  - controlling processes 68
  - default port numbers 259
  - operating SAS servers individually with scripts 61
  - operating SAS servers with SAS Management Console 65
  - operating with scripts 61
  - operation of 52
  - required servers 52
  - restarting all servers, UNIX or z/OS 58
  - running as UNIX daemons 61
  - running as Windows services 55
  - SAS Metadata Server states 53
  - setting up for promotion tools 226
  - start-up parameters 53
  - starting and stopping SAS servers, z/OS 57
  - starting and stopping Web application servers 69
  - starting in correct order 52
  - starting or stopping all servers, UNIX or z/OS 58
  - starting or stopping with sas.servers script 58
  - stopping in correct order 53
  - summary of procedures for server operation 54
- SERVERSTARTPATH= option
  - %OMABAKUP macro 158
- sessions
  - viewing 101
- Sessions tab
  - Server Manager 92
- setup tasks
  - first-priority 40
  - optional 44
  - standard 43
- Shared Data folder 199
- Spawned Server Activity tab
  - Server Manager 92
- spawners 17
  - See also* SAS object spawner
  - default port numbers 259
  - displaying status with server script 76
  - setting up for promotion tools 226
- SPK files
  - importing 235
- standard setup tasks 43
- start-up order for servers 52
- starting
  - all servers, UNIX or z/OS 58
  - IBM WebSphere Application Server, Windows or UNIX 70
  - IBM WebSphere Application Server Network Deployment, z/OS 71
  - JBoss Application Server 69
  - Oracle WebLogic Server 72
  - SAS Deployment Tester server 64
  - SAS servers, z/OS 57
  - Web application servers 69
- state of the system 6
- states
  - of SAS Metadata Server 53
- status of servers
  - See* server status
- stop order for servers 53
- Stopped state 54
- stopping
  - all servers, UNIX or z/OS 58
  - IBM WebSphere Application Server, Windows or UNIX 70
  - IBM WebSphere Application Server Network Deployment, Windows or UNIX 70
  - IBM WebSphere Application Server Network Deployment, z/OS 71
  - JBoss Application Server 69
  - Oracle WebLogic Server 72
  - SAS Deployment Tester server 64
  - SAS Metadata Server 65
  - SAS object spawner 66
  - SAS OLAP Server 67
  - SAS servers, z/OS 57
  - Web application servers 69
- stored process servers
  - See* SAS Stored Process Servers
- substitution properties file 247
  - applying 248
  - editing 247
- synchronizing metadata backups 132
- system, state of 6

- system administration
  - centralized administration 33
  - delegated administration 33
  - ongoing tasks 46
  - security for 33
  - user roles 24
- system facility appender
  - capturing metadata server management events 117
- system integrity
  - best practices for 47

## T

- table server
  - See* SAS Table Server
- Table Server Administrators group 27
- Table Server Manager plug-in 8
- tables
  - moving SAS 9.1.3 tables to folders 224
- tasks
  - See* administration tasks
- test environment 44
- third-party products
  - monitoring SAS servers with 101
- third-party software
  - default port numbers for 260
- threads
  - configuring for metadata server performance 167
- THREADSMAX option 167
  - specifying values for 169
- THREADSMIN option 167
  - specifying values for 169
- time
  - absolute date and time 245
  - relative time period 246
- token authentication
  - workspace server access and 37
- troubleshooting
  - Backup Wizard 144
  - detailed logging for SAS object spawner 113
  - detailed logging for servers 111
  - log files for SAS Workspace Server 112
  - promotion tools 253
  - Update Host Name References tool 312
- trustedPeers.xml file 23
- trustedUsers.txt file 23

## U

- UNIX
  - collecting ARM log information 114
  - configuring DATA step batch servers 114
  - displaying server status 58
  - metadata server invocation option parameters 292
  - operating system protections 42
  - restarting all servers 58
  - running servers as daemons 61
  - starting or stopping all servers 58
  - starting or stopping IBM WebSphere Application Server 70
  - starting or stopping IBM WebSphere Application Server Network Deployment 70
  - summary of procedures for server operation 55
- unregistering metadata repositories 192
- unrestricted user 23
- Update Host Name References tool 296

- See also* host name references, updating
  - modifications not performed by 303
  - modifications performed by 301
  - troubleshooting 312
  - when not to use 299
  - when to use 297
- updating host name references
  - See* host name references, updating
- user accounts, initial 27
  - LSF Administrator 31
  - lsfuser 32
  - SAS Administrator 28
  - SAS Anonymous Web User 31
  - SAS First User 30
  - SAS Installer 29
  - SAS Spawned Servers account 30
  - SAS Trusted User 29
- user credentials
  - for Backup Wizard 139
  - for %OMABAKUP macro 149
  - promotion tools 226
- user groups
  - initial groups 26
  - initially defined in metadata 26
  - initially defined in operating system 26
- User Manager plug-in 8
- user roles
  - for system administration 24
- users
  - access management 43
  - adding 43
  - adding to SAS Administrators group 36
- Users folder 199

## V

- Validate feature 82, 83

## W

- warning messages
  - generated by %OMABAKUP macro 155
- Web application servers
  - starting and stopping 69
- Web applications
  - optimizing performance 44
- Web subdirectory 22
  - contents of 267
- Windows
  - 32-bit systems and metadata server performance 174
  - backup program using alternating destinations 154
  - collecting ARM log information 114
  - configuring DATA step batch servers 114
  - metadata server invocation option parameters 292
  - operating system protections 40
  - starting or stopping IBM WebSphere Application Server 70
  - starting or stopping IBM WebSphere Application Server Network Deployment 70
  - summary of procedures for server operation 55
- Windows services
  - running servers as 55
- workspace servers
  - See* SAS Workspace Servers
- WorkspaceServer subdirectory 21

**X**

## XML information

capturing in SAS Metadata Server log 117

XML log entries 119

**Z**

## z/OS

collecting ARM log information 114

configuring DATA step batch servers 114

displaying server status 58

metadata server invocation option parameters 292

operating system protections 42

restarting all servers 58

starting and stopping SAS servers 57

starting or stopping all servers 58

starting or stopping IBM WebSphere Application Server

Network Deployment 71

summary of procedures for server operation 55



# Your Turn

---

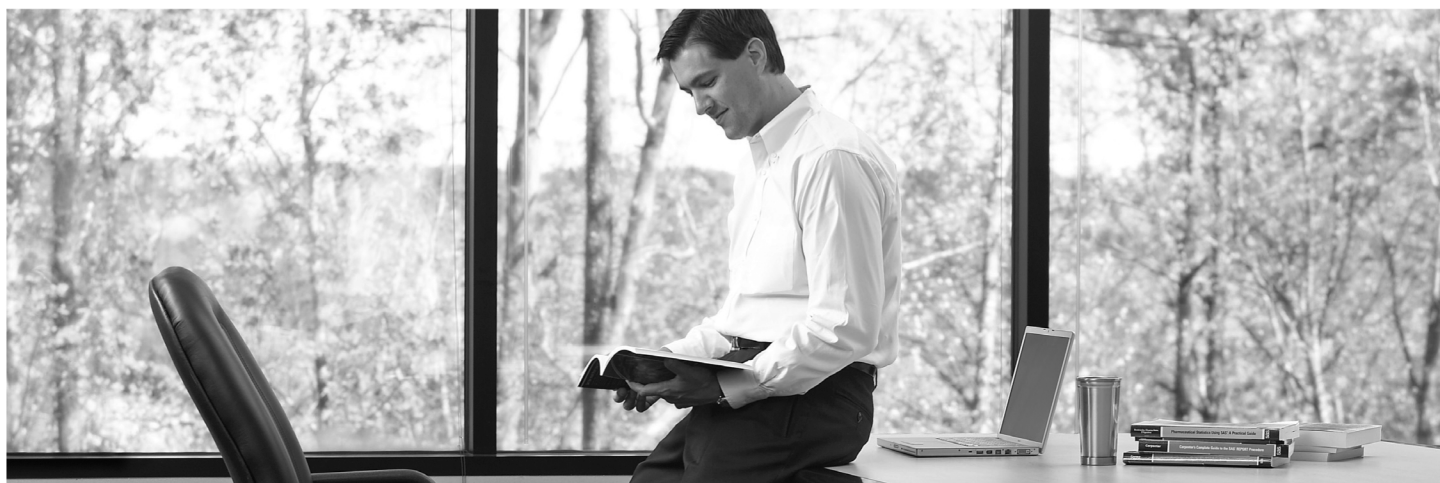
We welcome your feedback.

- 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 Delivers!



SAS Publishing provides you with a wide range of resources to help you develop your SAS software expertise. Visit us online at [support.sas.com/bookstore](http://support.sas.com/bookstore).

## SAS® PRESS

SAS Press titles deliver expert advice from SAS® users worldwide. Written by experienced SAS professionals, SAS Press books deliver real-world insights on a broad range of topics for all skill levels.

[support.sas.com/saspress](http://support.sas.com/saspress)

## SAS® DOCUMENTATION

We produce a full range of primary documentation:

- Online help built into the software
- Tutorials integrated into the product
- Reference documentation delivered in HTML and PDF formats—free on the Web
- Hard-copy books

[support.sas.com/documentation](http://support.sas.com/documentation)

## SAS® PUBLISHING NEWS

Subscribe to SAS Publishing News to receive up-to-date information via e-mail about all new SAS titles, product news, special offers and promotions, and Web site features.

[support.sas.com/spn](http://support.sas.com/spn)

## SOCIAL MEDIA: JOIN THE CONVERSATION!

Connect with SAS Publishing through social media. Visit our Web site for links to our pages on Facebook, Twitter, and LinkedIn. Learn about our blogs, author podcasts, and RSS feeds, too.

[support.sas.com/socialmedia](http://support.sas.com/socialmedia)



**THE  
POWER  
TO KNOW®**