SAS® 9.3 Intelligence Platform

System Administration Guide

Second Edition

SAS® Documentation

The correct bibliographic citation for this manual is as follows: SAS Institute Inc. 2012. *SAS® 9.3 Intelligence Platform: System Administration Guide, Second Edition*. Cary, NC: SAS Institute Inc.

SAS® 9.3 Intelligence Platform: System Administration Guide, Second Edition

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

Electronic book 1, August 2012

Contents

PART 8 Appendixes 305

What's New in System Administration for the SAS 9.3 Intelligence Platform

Overview

The SAS Intelligence Platform includes the following enhancements and changes for SAS 9.3:

- changes to SAS server operation, monitoring, and logging
- a new backup and recovery facility for the SAS Metadata Server
- changes to metadata server administration
- enhancements to SAS Folders
- enhancements to the promotion tools (including the Export SAS Package and Import SAS Package wizards and the batch export and import tools.
- SAS 9.3 to SAS 9.3 migration, which enables you to use the SAS Migration Utility along with the SAS Deployment Wizard to create a copy of your SAS 9.3 deployment
- deprecated features
- documentation enhancements

Changes to SAS Server Operation, Monitoring, and Logging

SAS 9.3 includes the following changes to operation, monitoring, and logging for SAS servers:

- The new SAS Framework Data Server is the default location for middle-tier data such as alerts, comments, and workflows, as well as data for the SAS Content Server and SAS Service Parts Optimization. The server is provided as an alternative to using a third-party DBMS. (The server cannot be used as a general-purpose data store.)
- Audit.Meta.Updates is a new logging category for the SAS Metadata Server that enables you to capture events that are performed on metadata objects. The logged events include adding, updating, deleting, checking out, checking in, and fetching objects, as well as undoing checkouts. See “About Metadata Server Loggers” on page 121.

- Alternative logging configuration files called logconfig.apm.xml are provided for use with the SAS Enterprise BI Audit and Performance Measurement package. This package is available for download from `http://support.sas.com/rnd/emi`. The logging configuration files are customized for your deployment.
- In SAS Management Console, you can now use the Server Manager **Options** tab to dynamically update the JNLLINEMAX and JNLSTRMAX object server parameters. These parameters pertain to the SAS Metadata Server, SAS OLAP Server, and SAS object spawner, as well as to processes on SAS Workspace Servers, SAS Pooled Workspace Servers, and SAS Stored Process Servers. This feature enables you to capture generated XML entries in the metadata server log without stopping the server. See "Capture XML Information in the SAS Metadata Server Log" on page 134 and "Use the Options Tab in the Server Manager" on page 109.
- The second maintenance release for SAS 9.3 includes new files that can be used to customize the following shell scripts: level_env.sh, appservercontext_env.sh, sas.sh, connectspawner.sh, metadataserver.sh, olapserver.sh, shareserver.sh, and sasbatch.sh. Use of these files, which have the suffix _usermods.sh, prevents script customizations from being overwritten when a new SAS release is installed.

New Backup and Recovery Facility for the SAS Metadata Server

The SAS 9.3 Metadata Server includes a new server-based facility that performs metadata server backups automatically on a scheduled basis. This facility can also be used to perform ad hoc backups and to perform roll-forward recovery. The facility replaces the %OMABAKUP macro, which was available in SAS 9.1 and 9.2, and the Backup and Restore Wizard in SAS Management Console, which was available in SAS 9.2.

The backup and recovery facility provides the following features:

- A schedule of daily backups is configured by the SAS Deployment Wizard. Therefore, daily backups are taken without the need for administrator intervention. In addition, unscheduled backups are launched automatically in certain situations, such as after the SAS Deployment Wizard configures the SAS Metadata Server.
- Administrators can use SAS Management Console to change the schedule and the configuration options, including the backup directory location and the backup retention policy. Backups can also be run on an ad hoc (unscheduled) basis from SAS Management Console, from the operating system command line, from SAS, or through third-party scheduling software.
- When repositories are restored in the event of a metadata server failure, transactions recorded in the journal file can be used to perform roll-forward recovery to a specified point in time after the last backup was taken.
- SAS Management Console displays detailed backup and recovery history. The history includes all backups that are run using the server's backup facility, regardless of how the backup was invoked (from the server's own scheduling facility, from SAS Management Console, from the command line, from SAS, or from third-party scheduling software). Status icons indicate which backups are active and which backups have been deleted based on the retention policy. When you view the properties of a backup, the backup files are validated, and the status icon is updated to indicate whether the backup is usable.

- The backup process is a “hot” backup that executes in a separate thread while the metadata server is running. Therefore, the server does not need to be paused during the backup and all transactions, including updates, continue to be processed.

  If journaling has been disabled or if the Reorganize Repositories backup option has been selected, the server is paused for read-only use, so that queries (but not updates) can continue to be processed.

- The metadata server runs its own backup and recovery processes. These processes do not rely on any other servers, such as workspace servers, DATA step batch servers, or scheduling servers. Because the metadata server has permission to access all of the needed files, there is no need to set up a host account with special operating system rights and permissions.

For details, see “Backing Up and Recovering the SAS Metadata Server” on page 167.

Changes to SAS Metadata Server Administration

The following changes have been made to administration for the SAS Metadata Server:

- In the Metadata Manager in SAS Management Console, the Active Server Properties dialog box includes a new **Statistics** tab that displays metadata server statistics including CPU seconds, memory utilization, thread utilization, and I/O activity.
- A new feature has been added to the Metadata Manager in SAS Management Console that enables you to test alert e-mails without stopping the metadata server. See “Managing Alert E-mail Options for the SAS Metadata Server” on page 191.
- Configuration options are available to enable authentication to the e-mail server. These options can be specified in the SAS Deployment Wizard if the custom installation option is selected. They can also be specified in the metadata server invocation command. See “Options for the Metadata Server Invocation Command” on page 339.
- A new option, -optimizeIMDB, has been added for the metadata server command. This option helps optimize the memory usage of the metadata server by analyzing the handling of character variables. It replaces the %OMARUNAN macro and the RUNANALYSIS option of %OMABAKUP, which were available in previous releases. See “Running the optimizeIMDB Command Option” on page 198.
- A new configuration option, OMA JOURNALTYPE=, has been added for the metadata server. This option specifies the type of journal file that is created. The SAS Deployment Wizard sets this option to ROLL_FORWARD during installation or during migration from SAS 9.1.3 or 9.2. The ROLL_FORWARD setting creates a linear journal file that permanently stores all transactions that have occurred since the most recent backup. The journal file is written to the same location as the associated backup files. Transactions recorded in the journal file can be used to perform roll-forward recovery to a specified point in time after the last backup was taken. For more information, see “Configuring Metadata Server Journaling” on page 190.
- For Metadata Manager functions such as creating a new repository and registering a repository, the **Browse** button is now enabled for navigation to the directory path for metadata repositories.

Enhancements to SAS Folders

The following enhancements have been implemented for the SAS Folders tree in SAS Management Console:

- On the new **Search** tab, you can search for metadata objects that meet specified search criteria. You can search based on object names, locations, descriptions, object types, creation or modification dates, keywords, and responsibilities. You can save any search to a Search folder. The search is rerun whenever the folder is opened.
- You can create a Favorites folder in which to save references to often-used metadata definitions. Within the Favorites folder, you can create favorites groups to organize these references.
- The name of the parent folder that contains users' home folders has been changed from Users to User Folders. If you use the SAS Migration Utility to migrate to SAS 9.3 from a previous version, then the folder name Users is retained. If you use the promotion tools (instead of the SAS Migration Utility) to move your content to SAS 9.3, see "Promoting Personal Folders from SAS 9.2 to SAS 9.3" on page 299.
- Virtual folders called Servers and Security are displayed under the System folder in the SAS Folders tree for use in promoting server and security objects. See "Promoting Security Objects and Server Objects" on page 292.

For more information, see "Working with SAS Folders" on page 217.

Enhancements to Promotion Tools

The following enhancements have been added to the promotion tools. These tools include the Export SAS Package and Import SAS Package wizards in SAS Management Console, SAS Data Integration Studio, and SAS OLAP Cube Studio, as well as the batch import and export tools.

- In SAS Management Console, support has been added for promoting additional object types including security objects (users, user groups, roles, and access control templates) and server objects (SAS Application Servers and their component servers, SAS object spawners, and SAS/CONNECT spawners). See "Promoting Security Objects and Server Objects" on page 292.

  In the second maintenance release of SAS 9.3, promotion support was added for additional server types including database servers, Hadoop servers, DataFlux Federation Servers, message queue polling servers, operating system scheduling services, Platform Process Manager scheduling servers, SAS/SHARE servers, and SAS Scalable Performance Data (SPD) servers.
- Advanced filtering options enable you to identify objects for export based on their keywords, notes, extended attributes, and responsibilities fields.
- As part of the promotion process, you can reorganize content by specifying new folder locations for individual objects that you are importing. The connections among objects are maintained. See "Move Objects to Different Folders When Using the Import Wizard" on page 258 and "Use the Substitution Properties File to Specify New Folder Locations for Objects" on page 278.

- When tables or external files are overwritten during an import, you can use the Change Analysis feature to identify and resolve differences in column metadata. See “Use the Import Wizard’s Change Analysis Feature to Resolve Column Differences in Tables and External Files” on page 259. (This feature is not available in the batch import tool.)

SAS 9.3 to SAS 9.3 Migration

You can use the SAS Migration Utility along with the SAS Deployment Wizard to create a second SAS 9.3 deployment that contains the same SAS content and the same configuration as an existing SAS 9.3 deployment. For details, see “Performing SAS 9.3 to SAS 9.3 Migration” on page 373.

Deprecated Features

The following features have been removed:

- The following options for the metadata server configuration file (omaconfig.xml) have been deprecated: OMA ALERTEMAILATTACHMENT and OMA JNLPRE. Attachments to alert e-mails for the metadata server are no longer supported.
- The %OMARUNAN macro has been deprecated. It has been replaced by the -optimizeIMDB option of the metadata server command.
- The Backup and Restore Wizard has been removed from SAS Management Console, and the %OMABAKUP macro has been deprecated. These features have been replaced by the new metadata server backup and recovery facility.

Documentation Enhancements

The following enhancements have been made to the *SAS Intelligence Platform: System Administration Guide*:

- A new chapter about using the Deployment Tester plug-in for SAS Management Console has been added. See “Using the Deployment Tester” on page 89.
- A new topic about using the Metadata Analyze and Repair utility in SAS Management Console has been added. See “Analyzing and Repairing Metadata” on page 212.
- The documentation for promotion tools has been expanded and reorganized into four separate chapters.
- In the second edition, expanded documentation has been provided for same-version (SAS 9.3 to SAS 9.3) migration. See “Performing SAS 9.3 to SAS 9.3 Migration” on page 373.
- In the second edition, instructions for setting global options have been added to the documentation for the batch export and import tools. These options can be used to change the default behavior of the batch tools when exporting and importing stored

processes and tables. See “Set Global Options for the Batch Export and Import Tools” on page 265.

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: 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: Middle-Tier 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 Logging: Configuration and Programming Reference*
- *SAS System Options: Reference*
- *SAS/CONNECT User's Guide*
- *SAS/SHARE User's Guide*
- *Scheduling in SAS*
- SAS offers instructor-led training and self-paced e-learning courses to help you administer the SAS Intelligence Platform. For more information about the available courses, see support.sas.com/admintraining.

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:

SAS Publishing Sales
SAS Campus Drive

Cary, NC 27513-2414
Phone: 1-800-727-3228
Fax: 1-919-677-8166
E-mail: sasbook@sas.com
Web address: support.sas.com/bookstore

Part 1

Getting Started

Chapter 1

Introduction to the SAS Intelligence Platform: System Administration Guide

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

Building on the technologies in the SAS Intelligence Platform, SAS offers solutions for industries such as financial services, life sciences, health care, retail, and manufacturing, as well as line-of-business solutions in areas such as customer management, enterprise risk management, and financial management. 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
  - analyzing and repairing metadata
  - working with SAS Folders
  - promoting metadata and related physical content

The appendixes of this guide provide reference information about log files, configuration files, and default ports. They also provide information about utilities that you can use to modify host name references and to migrate content to another SAS 9.3 deployment.

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
- middle-tier administration
- Web application administration
- desktop application administration
- scheduling
- logging

To access these documents, go to **`http://support.sas.com/93administration`**.

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/93administration`**.

Accessibility Features in the SAS Intelligence Platform Products

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

In order for assistive technologies that run in Windows to interoperate with a SAS application that is based on Java technology, the Java Access Bridge from Oracle must be installed and configured in the application's installation directory. For information about downloading and installing this software, go to **`http://www.sas.com/industry/government/java.html`**.

Please contact **`accessibility@sas.com`** if you need this document in an alternative digital format.

Chapter 2
Understanding the State of Your System

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.

- other SAS servers that might have been installed in your environment, including a SAS/SHARE server, a SAS Content Server, and a SAS Framework Data 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.

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.

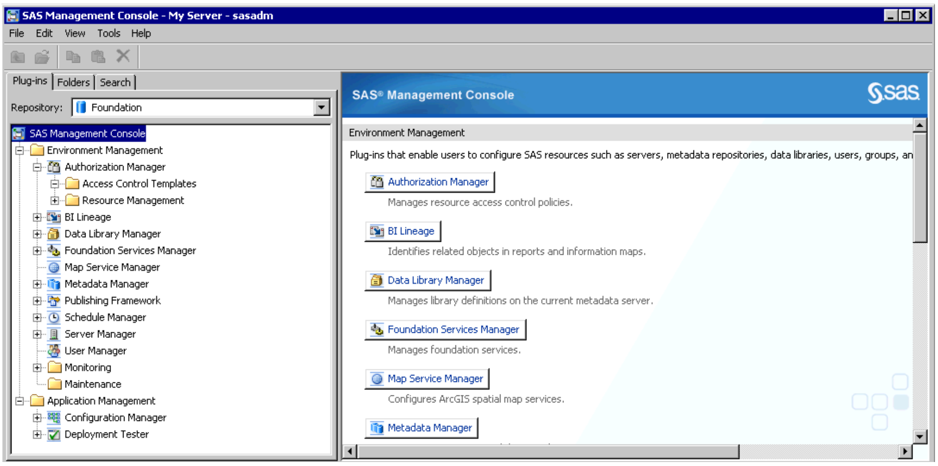
Whenever the SAS Metadata Server is running, you can use SAS Management Console to connect to the SAS Metadata Server and view and manage the metadata objects that are stored in the server's metadata repositories. The SAS Management Console user interface includes a **Plug-ins** tab, a **Folders** tab, and a **Search** tab that you can use to access and manage 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 25.

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:



Plug-ins that enable users to configure SAS resources such as servers, metadata repositories, data libraries, users, groups, an

Authorization Manager
Manages resource access control policies.

BI Lineage
Identifies related objects in reports and information maps.

Data Library Manager
Manages library definitions on the current metadata server.

Foundation Services Manager
Manages foundation services.

Map Service Manager
Configures ArcGIS spatial map services.

Metadata Manager

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 *SAS Intelligence Platform: Security Administration Guide*.
BI Lineage	Identify connections between BI objects in the SAS Folders tree. For details, see “Using the BI Lineage Plug-in” on page 224 and the SAS Management Console Help.
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 *SAS Intelligence Platform: Web Application Administration Guide*.

Plug-in Name	Plug-in Description
Data Library Manager	Create and maintain definitions for SAS libraries and database schemas. For details, see the *SAS Intelligence Platform: Data Administration Guide*.
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 BI Dashboard, and SAS Web Report Studio. For details, see the *SAS Intelligence Platform: Web Application Administration Guide*.
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 *SAS Intelligence Platform: Web Application Administration Guide*.
Metadata Manager	Perform administration tasks related to the SAS Metadata Server, including the following: • stop, pause, resume, or reset the metadata server • check the status of the metadata server and the metadata repositories • create and manage metadata repositories • back up or restore your metadata repositories • analyze and repair metadata
Publishing Framework	Create and maintain definitions for channels, package subscribers, group subscribers, and event subscribers. For details, see the *SAS Intelligence Platform: Web Application Administration Guide*.
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 *Scheduling in SAS*.
Server Manager	Perform administration tasks related to SAS servers, including the following: • check the status of servers and validate them to ensure that they are configured correctly • stop, pause, quiesce, resume, or refresh the SAS object spawner or the SAS Framework Data Server; and stop, pause, resume, or quiesce a process on a SAS OLAP Server, SAS Workspace Server, or SAS Stored Process Server • connect to the metadata server, to components of SAS Application Servers, to the SAS object spawner, or to a SAS Framework Data Server to perform monitoring activities Details about the preceding tasks are provided in this guide. You can also use the Server Manager plug-in to create and maintain definitions for servers. For details, see the *SAS Intelligence Platform: Application Server Administration Guide*.
User Manager	Create and maintain definitions for users, groups, and roles. For details, see the *SAS Intelligence Platform: Security Administration Guide*.

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.

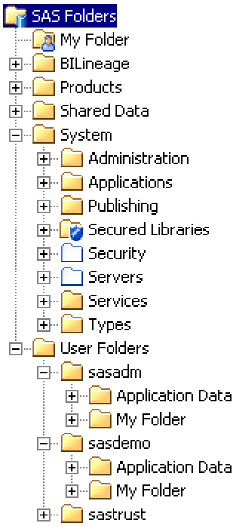
About the SAS Management Console Folders Tab

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 217.

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:
  - update passwords for the service accounts that were configured when you ran the SAS Deployment Wizard. For details, see "Managed Passwords" in Chapter 2 of *SAS Intelligence Platform: Security Administration Guide*.
  - rebuild Web applications. For details, see "Rebuilding the SAS Web Applications" in Chapter 7 of *SAS Intelligence Platform: Middle-Tier Administration Guide*.
  - uninstall SAS software from the local machine. For details, see Appendix 4, "Uninstalling Your SAS Software," in *SAS Intelligence Platform: Installation and Configuration Guide*.
  - remove one or more components of a SAS Intelligence Platform configuration from your environment. For details, see Appendix 3, "Removing a SAS Configuration," in *SAS Intelligence Platform: Installation and Configuration Guide*.
  - renew your software license for some SAS solutions that depend on a SAS middle tier. For details, see Appendix 3, "Updating the SID File in Metadata for SAS Solutions," in *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 "Using the SAS Deployment Manager to Update Host Name References" on page 346.
  - apply downloaded hot fixes to your SAS software. For details, see the *SAS 9.3 Guide to Software Updates*.

- 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; view the current configuration of Web applications; view audit reports of logon and logoff activity; and dynamically adjust logging levels for some Web applications. For details, see “Using the SAS Web Administration Console ” in Chapter 5 of *SAS Intelligence Platform: Middle-Tier 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 SAS applications in your environment and can support hundreds of concurrent users. This architecture enables the following:

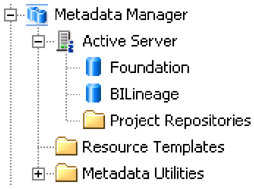
- the exchange of metadata between applications, so that applications can work together more easily.
- centralized management of metadata resources. Because there is a common framework for creating, accessing, and updating metadata, it is easier to manage the applications that rely on this metadata.

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:

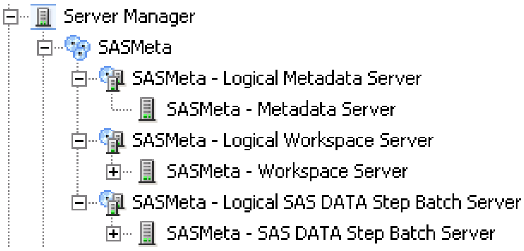


You can use this plug-in to stop, pause, resume, or reset the metadata server; to check the status of the metadata server and the metadata repositories; and to create and manage metadata repositories. You can also access utilities such as the metadata server backup and recovery facility.

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:


Server Manager
SASMeta
SASMeta - Logical Metadata Server
SASMeta - Metadata Server
SASMeta - Logical Workspace Server
SASMeta - Workspace Server
SASMeta - Logical SAS DATA Step Batch Server
SASMeta - SAS DATA Step Batch Server

As shown in the preceding display, the SASMeta server context also includes a workspace server and DATA step batch server. These servers are visible only to users who are in the SAS Administrators group and are reserved for specialized uses.

CAUTION:

The SASMeta workspace server is intended only for use in running certain metadata utilities. Do not use it for other SAS processing.

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.

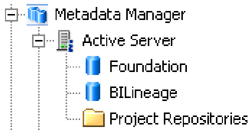
A project repository enables a user to check out and lock metadata from a foundation repository or a custom repository so that the metadata can be modified and tested in a separate area. When the metadata is ready, it is checked in and unlocked so that it is available to other users.

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 and SAS 9.3, 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 9.3 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 20.
- Effective with SAS 9.3, automatic backups of the metadata server are scheduled to occur daily (seven days a week) at 1:00 a.m. server local time. For details, see "Overview of the SAS Metadata Server Backup and Recovery Facility" on page 167.
- Roll-forward journaling has been enabled. This option creates a linear journal file that permanently stores all transactions that have occurred since the most recent backup. The journal file is written to the same location as the associated backup files.
- 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 123.

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 reserved for specialized uses.

logical servers

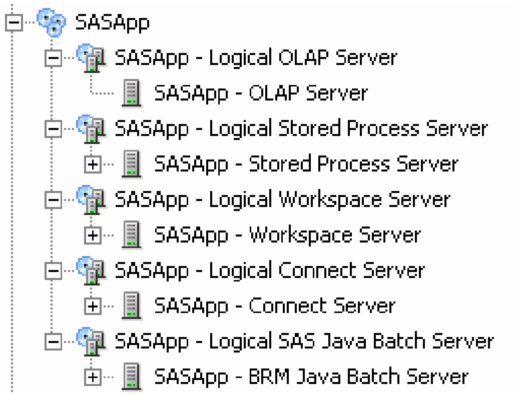
a logical framework within SAS Application Server metadata that contains definitions for one or more similar servers. A logical server is referenced when an application requests a specific launch mode, such as interactive or batch.

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:



- SASApp
  - SASApp - Logical OLAP Server
    - SASApp - OLAP Server
  - SASApp - Logical Stored Process Server
    - SASApp - Stored Process Server
  - SASApp - Logical Workspace Server
    - SASApp - Workspace Server
  - SASApp - Logical Connect Server
    - SASApp - Connect Server
  - SASApp - Logical SAS Java Batch Server
    - SASApp - BRM Java Batch Server

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 *SAS Intelligence Platform: Application Server Administration Guide*.
SAS Pooled Workspace Server	SAS Pooled Workspace Servers are workspace servers that use server-side pooling. 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. 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. For details, see the *SAS Intelligence Platform: Application Server Administration Guide*.
SAS Stored Process Servers	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. Stored process servers are configured to be load balanced. By default, the object spawner balances workloads across three stored process server processes. For details, see the *SAS Intelligence Platform: Application Server Administration Guide*.
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 *SAS/CONNECT User's Guide*.
SAS OLAP Server	SAS OLAP Servers provide access to multidimensional data. For details, see the *SAS Intelligence Platform: Application Server Administration Guide*.
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 *Scheduling in SAS*.
SAS Grid Server	A SAS Grid Server is configured if you have the SAS Grid Manager software application. 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 *Grid Computing in SAS*.

Overview of Other SAS Servers and Spawners

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

Table 2.3 *Other SAS Servers and Spawners*

Server or Spawner Name	Server or Spawner Description
SAS Content Server	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. The SAS Content Server is used by SAS Information Delivery Portal and SAS Web Report Studio, as well as other SAS products and solutions. The server's repositories are stored in the **`AppData`** subdirectory of your configuration directory. For more information, see the *SAS Intelligence Platform: Middle-Tier Administration Guide*.
DataFlux Data Management Server	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. For information about configuring this server, see Chapter 5, “Administering SAS Data Integration Studio,” in *SAS Intelligence Platform: Desktop Application Administration Guide*.
SAS In-Process Services	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 *Scheduling in SAS*.
SAS object spawner	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. 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. For details, see the *SAS Intelligence Platform: Application Server Administration Guide*.
SAS Framework Data Server	Is the default location for middle-tier data such as alerts, comments, and workflows, as well as data for the SAS Content Server and SAS Service Parts Optimization. The server is provided as an alternative to using a third-party DBMS. (The server cannot be used as a general-purpose data store.)

Server or Spawner Name	Server or Spawner Description
SAS/CONNECT spawner	Listens for SAS/CONNECT client requests and invokes SAS sessions on the remote computer. For details, see *Communications Access Methods for SAS/CONNECT and SAS/SHARE*.
SAS/SHARE server	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 *SAS/SHARE User's Guide*.

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 "Using the Deployment Tester" on page 89 and the SAS Management Console Help.

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 "Default SAS Ports" on page 307. 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:

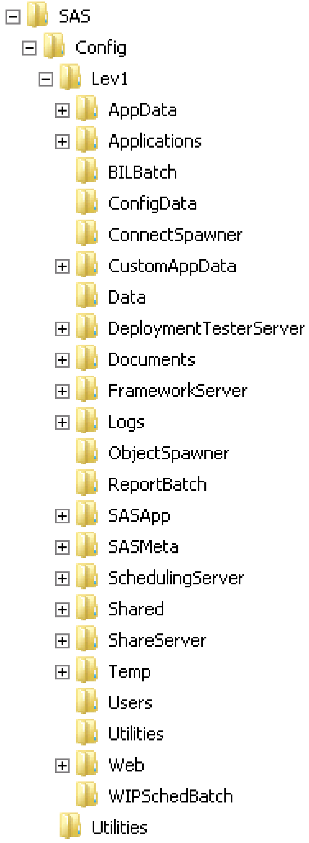
  `usr/local/`*deployment-plan-name* or `usr/local/sas`

  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:



```
⊟ SAS
  ⊟ Config
    ⊟ Lev1
      ⊞ AppData
      ⊞ Applications
        BILBatch
        ConfigData
        ConnectSpawner
      ⊞ CustomAppData
        Data
      ⊞ DeploymentTesterServer
      ⊞ Documents
      ⊞ FrameworkServer
      ⊞ Logs
        ObjectSpawner
        ReportBatch
      ⊞ SASApp
      ⊞ SASMeta
      ⊞ SchedulingServer
      ⊞ Shared
      ⊞ ShareServer
      ⊞ Temp
        Users
        Utilities
      ⊞ Web
        WIPSchedBatch
      Utilities
```

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

About the Levn Subdirectory

The **`Lev`***n* 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 **`Lev`***n* 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.
- a subdirectory for the metadata server.
- subdirectories for other SAS servers and spawners, including the SAS object spawner, the SAS Deployment Tester Server, the SAS Framework Data Server, the SAS/CONNECT spawner, and the SAS/SHARE server.
- a Web subdirectory for files related to Web applications.
- logs and documents that were created during the installation and configuration process

For more details about the **Lev***n* directory, see “Contents of the Levn Subdirectory” on page 313.

About the Application Server Subdirectory

On machines where components of a SAS Application Server are installed, the **Lev***n* 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 the SAS Deployment Wizard creates additional directories 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 317. For information about the server configuration files that are contained in this subdirectory, see “Configuration Files for Components of SAS Application Servers” on page 324.

About the SASMeta Subdirectory

On the machines where the metadata server is installed, the `Lev`*n* 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 reserved for specialized activities.
- `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.
  - a script to back up the metadata server (for details, see "SAS Metadata Server Backup Tasks" on page 171).
  - 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 called `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 "Best Practices for Backing Up and Restoring Your SAS Content" on page 155.
  - a `Backups` directory that is the default destination for metadata server backups and journal files.

For more information about the metadata server configuration files, see "Configuration Files for the Metadata Server" on page 322. For a complete list of the contents of the MetadataServer subdirectory, see "Contents of the Metadata Server Subdirectory (SASMeta)" on page 315.

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 318. For information about administering middle-tier applications and components, see the *SAS Intelligence Platform: Web Application*

Administration Guide and the *SAS Intelligence Platform: Middle-Tier 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
- alert e-mail options

For details about the omaconfig.xml file, see "Reference Information for omaconfig.xml" on page 329.

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 "Administering Logging for SAS Servers" on page 119.

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 321.

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 25.

Note: If you migrated your system from SAS 9.1.3, then you might have additional users in this file. In SAS 9.3, 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 28.

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 Connections" 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 Role
- Metadata Server: User Administration Role
- Metadata Server: Operation Role
- Management Console: Advanced Role
- Management Console: Content Management Role

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 "Who Can Do What: Credential Requirements for SAS Management Console Tasks" on page 35.
- 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 28.

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 "Who Can Do What: Credential Requirements for SAS Management Console Tasks" on page 35.

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)
- administer the metadata server backup and recovery facility

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 Spawned 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.

LSF Services
: a group whose members can schedule jobs in the LSF component of Platform Suite for SAS. 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 33

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. 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
- SAS Trusted User
- SAS Installer

- SAS Spawned Servers account
- SAS First User
- SAS Anonymous Web User
- LSF Administrator
- lsfuser

These user accounts might have been assigned different names at your site.

SAS Administrator

Table 2.4 *SAS Administrator Characteristics*

Type of Installation	Default User Name	Default User ID	Required?	Location of Account
Default settings	SAS Administrator	sasadm@saspw	Yes	Metadata
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 26. In addition, the SAS Administrator account is initially a member of the SAS Administrators group.

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 9.3.

In default installations of SAS 9.3, 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

Table 2.5 SAS Trusted User Characteristics

Type of Installation	Default User Name	Default User ID	Required?	Location of Account
Default settings	SAS Trusted User	sastrust@saspw	Yes	Metadata
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.

- 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 default installations of SAS 9.3, 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

Table 2.6 SAS Installer Characteristics

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 Spawned Servers Account

Table 2.7 *SAS Spawned Servers Account*

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

The SAS Spawned 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.

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

Table 2.8 *SAS First User Characteristics*

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.

SAS Anonymous Web User

Table 2.9 SAS Anonymous Web User Characteristics

Type of Installation	Default User Name	Default User ID	Required?	Location of Account
Default settings	SAS Anonymous Web User	webanon@saspw	No	Metadata
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, 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

Table 2.10 LSF Administrator Characteristics

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

Table 2.11 *lsfuser*

Default User ID	Required?	Location of Account
lsfuser	Yes, if Platform Suite for SAS is installed and is used to schedule SAS Web Report Studio 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.

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

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
: 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
: 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*

	Requirements*	
Actions	**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 **Management Console: Advanced** role.
Create, register, and delete custom repositories Change a repository's properties, including the access mode, the location of users' folders, and mapping for notes and documents	Membership in the SAS Administrators group	Membership in the **Management Console: Advanced** and **Metadata Server: Operation** roles, and WriteMetadata permission on the Root folder of the SAS Folders tree.
Stop, pause, or resume the metadata server Administer metadata server backups Test metadata server alert e-mails	Membership in the SAS Administrators group	Membership in the **Management Console: Advanced** and **Metadata Server: Operation** roles.
Continue to use all functions of SAS Management Console when the metadata server is paused to the Administration state Analyze, repair, or upgrade metadata Initialize or unregister a custom repository	Membership in the **Metadata Server: Unrestricted** role	
Add, initialize, register, unregister, and delete a foundation repository.**	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	

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

** 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 38.

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*

	Requirements*	
Actions	**Centralized Administration Approach**	**Delegated Administration Approach**
Create or update server metadata (except the SASMeta workspace server and the SASMeta DATA step batch server) Validate servers (except the SASMeta workspace server and the SAS Framework Data Server) Connect to servers and spawners and view the following tabs**: • Metadata server **Connections** and **Options** tabs • Object spawner **Options** tab • OLAP server and framework data server **Connections**, **Sessions**, and **Options** tabs	Membership in the SAS Administrators group Validation of a standard (non-pooled) workspace server also requires credentials to start a workspace server.	Membership in the **Management Console: Advanced** role Validation of a standard (nonpooled) workspace server also requires credentials to start a workspace server.
Connect to servers and spawners (except the metadata server, the SASMeta workspace server, and the framework data server) and view all tabs** Adjust the logging level for a server, spawner, or server process (except the metadata server and the framework data server) Stop, pause, resume, or quiesce a server, spawner, or server process (except the metadata server, the SASMeta workspace server, and the framework data server) Refresh the object spawner	Membership in the SAS Administrators group	Membership in the **Management Console: Advanced** role, and Administer permission on server metadata
Connect to the metadata server and view the **Clients**, **Loggers**, and **Logs** tabs** Adjust the metadata server's logging levels Stop, pause, or resume the metadata server	Membership in the SAS Administrators group	Membership in the **Management Console: Advanced** and **Metadata Server: Operation** roles
Validate the SASMeta workspace server Connect to the SASMeta workspace server and view tabs** Stop, pause, resume, or quiesce a process on the SASMeta workspace server	Membership in the SAS Administrators group	Membership in the **Management Console: Advanced** role; and ReadMetadata and Administer permission on the SASMeta workspace server

	Requirements[*]	
Actions	**Centralized Administration Approach**	**Delegated Administration Approach**
Validate the SAS Framework Data Server Connect to the framework data server and view tabs[**] Stop, pause, resume, or quiesce the framework data server	Specification as a system user in the framework data server's configuration file (dfs_serv.xml, located in ***SAS-configuration-directory*/Lev1/ FrameworkServer/dffedsvrcfg**). By default, this file specifies the SAS Administrator as a system user.	

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

[**] 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 "About Adding Users to the SAS Administrators Group" on page 38.

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 **Management Console: 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 Chapter 4 of *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 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 special metadata server administration tasks.

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

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 System Administration 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.	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. 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 42. These recommendations assume that your SAS servers and spawners run as services under the Local System account. 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 44.
Establish a formal, regularly scheduled backup process.	Establish a formal, regularly scheduled backup process that includes your metadata repositories as well as the associated physical files. The SAS 9.3 Metadata Server includes a new server-based facility that performs metadata server backups automatically on a scheduled basis. By default, a schedule of daily backups is configured by the SAS Deployment Wizard. As a best practice, you should modify your backup configuration to specify a storage device other than the device that is used to store the metadata repositories and the server configuration files, and you should be sure to include this backup location in your regular system backups. See "Backing Up and Recovering the SAS Metadata Server" on page 167. 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. For guidance in setting up a backup process, see "Best Practices for Backing Up and Restoring Your SAS Content" on page 155.

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
SAS-configuration-directory	SYSTEM and Administrators	Full Control
	All other users	List Folder Contents, Read
SAS-configuration-directory`\Lev1`	SYSTEM and Administrators	Full Control
	SAS Spawned Servers (sassrv)	On Windows Vista, Windows 7, and Windows Server 2008: Special Permissions to read and execute* On Windows XP: Read and Execute*
	All other users	List Folder Contents, Read
`Lev1\SASApp`	SYSTEM and Administrators	Full Control
	SAS Spawned Servers (sassrv)	Windows Vista, Windows 7, and Windows Server 2008 only: Special Permissions to read and execute*
	All other users	List Folder Contents, Read
`Lev1` subdirectories: `Documents`, `ReportBatch`, `SASMeta`, `Utilities`, `Web`	SYSTEM and Administrators	Full Control
	All other users	List Folder Contents, Read
`Lev1` subdirectories: • `ConnectSpawner` • `Logs` • `ObjectSpawner` • `SASApp\OLAPServer` • `SASMeta\MetadataServer` • `FrameworkServer` • `ShareServer`	SYSTEM and Administrators	Full Control Remove all other users and groups
`SASApp` subdirectories : `PooledWorkspaceServer`, `StoredProcessServer`	SYSTEM, Administrators	Full Control
	SAS Spawned Servers (sassrv)	On Windows Vista, Windows 7, and Windows Server 2008 only: Read & Execute, List Folder Contents, and Read*
	All other users	No access

Directories	Users	Recommended Permissions
SASApp subdirectories : **PooledWorkspaceServer\Logs**, **StoredProcessServer\Logs**	SYSTEM, Administrators, and SAS Spawned Servers (sassrv)	Full Control*
SASApp subdirectories: • **ConnectServer\Logs** • **Data\wrsdist** • **Data\wrstemp** • **PooledWorkspaceServer\sasuser** • **StoredProcessServer\sasuser** • **WorkspaceServer\Logs** **SASMeta\WorkspaceServer\Logs**	SYSTEM, Administrators, and SAS Spawned Servers (sassrv)	Full Control
sasv9_meta.cfg file	SYSTEM and Administrators	Read and Write Remove all other users and groups

* Effective with the second maintenance release for SAS 9.3, the SAS Deployment Wizard automatically sets these permissions for sassrv.

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 129).

For details about the configuration directory structure, see "Overview of the Configuration Directory Structure" on page 20.

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
SAS-configuration-directory *SAS-configuration-directory*/`Lev1` `Lev1` subdirectories: `Documents`, `ReportBatch`, `SASApp`, `SASMeta`, `Utilities`, `Web`	SAS Installer	Read, Write, and Execute
	All other users	Read and Execute
`Lev1` subdirectories: • `ConnectSpawner` • `Logs` • `ObjectSpawner` • `SASApp/OLAPServer` • `SASMeta/MetadataServer` • `FrameworkServer` • `ShareServer`	SAS Installer	Read, Write, and Execute
	All other users	No access
`SASApp` subdirectories : `PooledWorkspaceServer`, `StoredProcessServer`	SAS Installer	Read, Write, and Execute
	sas group	Read and Execute
`SASApp` subdirectories : • `ConnectServer/Logs` • `Data/wrsdist` • `Data/wrstemp` • `PooledWorkspaceServer/Logs` • `PooledWorkspaceServer/sasuser` • `StoredProcessServer/Logs` • `StoredProcessServer/sasuser` • `WorkspaceServer/Logs` `SASMeta/WorkspaceServer/Logs`	SAS Installer	Read, Write, and Execute
	sas group	Read, Write, and Execute
sasv9_meta.cfg file	SAS Installer	Read and Write
	All other users	no access

Note:

- Make sure that the SAS Spawned 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 Spawned 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 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 Read, Write, and Execute permission on the log directory. (See "Create a Log File for Workspace Server Troubleshooting" on page 129).
- For details about the configuration directory structure, see "Overview of the Configuration Directory Structure" on page 20.

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.	Standard security setup activities include the following: • creating SAS administrators and regular SAS users • managing access to metadata, data, and application functionality For details, see "Selected Tasks" in Chapter 4 of *SAS Intelligence Platform: Security Administration Guide*.
Establish connectivity to your data sources.	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 Chapter 2 of *SAS Intelligence Platform: Data Administration Guide*.
Set up your metadata folder structure.	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 "Working with SAS Folders" on page 217. *Note:* Be sure to secure access to the folders as described in the *SAS Intelligence Platform: Security Administration Guide*.

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
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 SAS servers will start automatically when you start the machine. The comments in the sas.servers script contain installation instructions that are specific to your operating environment. For details, see "Using the sas.servers Script on UNIX or z/OS to Start or Stop All Servers" on page 59.
Optimize the performance of the metadata server.	To optimize the performance of the metadata server, you might want to adjust the maximum number of threads used by the server. See "Configuring the Number of Threads Used by the Metadata Server" on page 188. For other performance-related tasks, see "Managing Metadata Server Memory" on page 195.
Modify the configuration of your processing servers.	The following are some server configuration changes that you might want to make for performance optimization or other reasons: • 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 Chapter 7 of *SAS Intelligence Platform: Application Server Administration Guide*. • Modify load balancing or algorithms for stored process servers and pooled workspace servers. For details, see "Understanding the Load-Balancing Algorithms" in Chapter 7 of *SAS Intelligence Platform: Application Server Administration Guide*. • 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 *SAS Intelligence Platform: Application Server Administration Guide*. • 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 *SAS Companion for z/OS*.
Optimize Web application performance.	For information about configuring middle-tier components for better efficiency and performance, see "Best Practices for Middle-Tier Configuration" in Chapter 2 of *SAS Intelligence Platform: Middle-Tier Administration Guide*.
Adjust server logging.	You can change your server configurations to obtain additional logging information for troubleshooting, performance management, or other purposes. For details, see "Administering Logging for SAS Servers" on page 119 and *SAS Logging: Configuration and Programming Reference*.
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 *Scheduling in SAS*.
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. see "Managing the Java Heap Used By Desktop Applications" in Chapter 2 of *SAS Intelligence Platform: Desktop Application Administration Guide*.
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 Chapter 5, "Administering SAS Data Integration Studio," in *SAS Intelligence Platform: Desktop Application Administration Guide*.

Task	Description
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 "Collecting ARM Log Information for SAS Data Integration Studio Jobs" on page 130.

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.	"Operating Your Servers" (p. 54)
Check the status of a server of or of a metadata repository.	"Checking the Status of Servers" (p. 77)
Monitor the activity of servers.	"Monitoring the Activity of SAS Servers" (p. 103)
Perform regular full backups of the metadata server.	"Best Practices for Backing Up and Restoring Your SAS Content" (p. 155)
Use server logs and configure logging options.	"Administering Logging for SAS Servers" on page 119 and *SAS Logging: Configuration and Programming Reference*
Create a new metadata repository.	"Managing SAS Metadata Repositories" (p. 203)
Promote individual metadata objects or groups of objects.	"Promotion Tools Overview" (p. 237)

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*. You can use Appendix 2, "Checklist for a More Secure Deployment," in *SAS Intelligence Platform: Security Administration Guide* 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: Middle-Tier 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/93administration`**.

Best Practices for Ensuring the Integrity of Your System

Allow Only the Metadata Server to Access Your Metadata Repositories

The **`MetadataRepositories`** and **`rposmgr`** 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 **`rposmgr`** directories. These data sets should be accessed only by metadata server processes.
- Do not place other files in these directories.
- Maintain a daily schedule of metadata server backups, as described in "Backing Up and Recovering the SAS Metadata Server" on page 167.

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 "Working with SAS Folders" on page 217. If you need to move or copy the contents of these folders, use the procedures that are provided in "Promotion Tools Overview" on page 237.

Use Usermods Files When Customizing Autoexec Files, Server Configuration Files, and Startup Scripts

If you need to customize a configuration file, autoexec file, or start-up script for a SAS server, do not directly modify the file or script. Instead, add your customizations to the corresponding file that is named **`server-name`**_usermods.cfg, autoexec_usermods.sas, **`server-name`**_usermods.bat, or **`server-name`**_usermods.sh.

Use of these files prevents your customizations from being overwritten when a new SAS release is installed.

Part 2

Starting, Stopping, and Checking the Status of Servers

Chapter 5
Operating Your Servers

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 Framework Data 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:

Table 5.1 *Server Dependencies*

Server	Server Dependencies
SAS Metadata Server	None
SAS OLAP Server	SAS Metadata Server
SAS object spawner	SAS Metadata Server
SAS Framework Data Server	None
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 is an example of a start-up order that 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 Framework Data Server.
7. Start the SAS Services Application (Remote Services).
8. Start the SAS Deployment Tester server (optional).
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 321.

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 55.

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 "Who Can Do What: Credential Requirements for SAS Management Console Tasks" on page 35.

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

Table 5.2 *Summary of Server Operation Procedures*

Server Type	Available Methods for Server Operation	Documentation Location for Start-up Parameters
SAS Metadata Server, SAS OLAP Server, and SAS object spawner	Windows service * z/OS started task ** sas.servers script *** (UNIX or z/OS) Individual server script (or Windows shortcut) SAS Management Console (except start-up)	"Options for the Metadata Server Invocation Command" on page 339 "Overview of System Options for SAS Application Server Components" in Chapter 14 of *SAS Intelligence Platform: Application Server Administration Guide* "Spawner Invocation Options" in Chapter 12 of *SAS Intelligence Platform: Application Server Administration Guide*
SAS Workspace Server, SAS Pooled Workspace Server, and SAS Stored Process Server	• Started by the object spawner • SAS Management Console (except start-up)	"Overview of System Options for SAS Application Server Components" in Chapter 14 of *SAS Intelligence Platform: Application Server Administration Guide*

Server Type	Available Methods for Server Operation	Documentation Location for Start-up Parameters
SAS Framework Data Server	Windows service [*] z/OS started task [**] sas.servers script [***] (UNIX or z/OS) Individual server script (or Windows shortcut) SAS Management Console (except start-up)	
SAS Services Application (Remote Services)	Windows service [*] z/OS started task [**] sas.servers script [***] (UNIX or z/OS) Remote Services script (or Windows shortcut)	*SAS Intelligence Platform: Middle-Tier Administration Guide*
Web application server	Server console or script Windows service (JBoss and Oracle WebLogic only)[*]	*SAS Intelligence Platform: Middle-Tier Administration Guide*
SAS Content Server	Starts automatically when the Web application server is started	*SAS Intelligence Platform: Middle-Tier Administration Guide*
SAS/CONNECT spawner, SAS/SHARE server, and SAS Deployment Tester server	Windows service [*] z/OS started task [**] sas.servers script [***] (UNIX or z/OS) Individual server script (or Windows shortcut)	*Communications Access Methods for SAS/CONNECT and SAS/SHARE* *SAS/SHARE User's Guide*

[*] On Windows systems, this is the recommended method.

[**] On z/OS systems, this is the recommended method.

[***] On UNIX systems, this is the recommended method.

Refer to the following topics for information about operating servers and services:

- "Running Servers as Windows Services" on page 56
- "Running Servers as UNIX Daemons" on page 62
- "Using the sas.servers Script on UNIX or z/OS to Start or Stop All Servers" on page 59
- "Starting and Stopping SAS Servers on z/OS" on page 58
- "Using Scripts to Operate SAS Servers Individually" on page 63
- "Using SAS Management Console to Operate SAS Servers" on page 67
- "Starting and Stopping Web Application Servers" on page 71

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 Framework Data Server, and the

SAS/SHARE 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 and the SAS Deployment Tester server might also be installed as a services. These services are set up to be started manually.

SAS provides 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*`/Lev`*n*`/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:

Table 5.3 *Service Dependencies and Actions*

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 Framework Data Server	None	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-Lev1] Object Spawner
SAS [Config-Lev1] Remote Services
SAS [Config-Lev1] SASApp - OLAP Server
SAS [Config-Lev1] SASMeta - Metadata Server
SAS [Config-Lev1] FrameworkServer - SAS Framework Data 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. 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, SAS/CONNECT spawners, and SAS/SHARE servers.

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 Metadata Server <(port-number)>, State, starting
SAH011999I SAS Metadata Server <(port-number)>, State, running
SAH019001I SAS Metadata Server <(port-number)>, State, stopping
SAH019999I SAS 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 63.
- 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 59.

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 essential SAS servers and spawners on the machine in the correct order by using a single command
- display the status of all the essential SAS servers and spawners on the machine by using a single command

Note:

- Effective with SAS 9.3, the script does not include the SAS Deployment Tester server.
- 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 58.

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 Framework Data Server

- SAS Services Application (Remote Services)

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.

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 essential 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 Framework Data Server, SAS Remote Services.

Note: The OLAP server, object spawner, SAS/SHARE server, and SAS/CONNECT spawner 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 Remote Services, SAS Framework Data 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 SAS Framework Data 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 runningSAS 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.

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 start-up 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 specific 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. For more information, see "Using the sas.servers Script on UNIX or z/OS to Start or Stop All Servers" on page 59. On z/OS systems, SAS recommends operating servers as started tasks. See "Starting and Stopping SAS Servers on z/OS" on page 58.

For information about the start-up parameters for each server, see the documentation references that are provided in "Summary of Procedures for Server Operation" on page 55.

Note: If you need to customize a server script, do not directly update the script. Instead, add your customizations to the corresponding file that is named **`server-name`**_usermods_.bat or **`server-name`**_usermods.sh. Use of these files prevents your customizations from being overwritten when a new SAS release is installed.

Use Individual Scripts to Operate the SAS Metadata Server, SAS Object Spawner, SAS OLAP Server, and SAS SAS Framework Data Server

You can use scripts to start, stop, pause, or resume the metadata server, an object spawner, an OLAP server, or a framework data 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 following

- The metadata server must be running before you can start an object spawner or an OLAP server.
- Before using scripts to operate the metadata server, see "About SAS Metadata Server States" on page 78.

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 5.4 Scripts for Operating Servers and Spawners

Server or Spawner	Script Name and Location
SAS Metadata Server	Windows: **`SASMeta\MetadataServer\MetadataServer.bat`**[*] UNIX and z/OS: **`SASMeta/MetadataServer/MetadataServer.sh`**[*]
SAS object spawner	Windows: **`ObjectSpawner\ObjectSpawner.bat`** UNIX and z/OS: **`ObjectSpawner/ObjectSpawner.sh`**
SAS OLAP Server	Windows: **`SASApp\OLAPServer\OLAPServer.bat`**[**] UNIX and z/OS: **`SASApp/OLAPServer/OLAPServer.sh`**[**]
SAS Framework Data Server	Windows: **`FrameworkServer\dffedsvrcfg\dffedsvrcfg.bat`** UNIX and z/OS: **`FrameworkServer/dffedsvrcfg/dffedsvrcfg.sh`**

[*] In single-machine installations that were migrated from SAS 9.1.3, the directory name might be *SASMain* instead of *SASMeta*.

[**] Your installation might have a different name for the *SASApp* directory.

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 or the OLAP 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, SAS/CONNECT spawners, and SAS/SHARE servers.
- 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 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`
uses the metadata server's backup facility to copy all of your metadata repositories, the repository manager, and the contents of the metadata server's configuration directory to the configured backup location. The backup process executes in a separate thread while the metadata server is running. Therefore, the server does not need to be paused. For more information, see "Backing Up and Recovering the SAS Metadata Server" on page 167.

CAUTION:
For production backups, it is highly recommended that you follow the best practices that are provided in this guide. See "Best Practices for Backing Up and Restoring Your SAS Content" on page 155.

`-optimizeIMDB`
optimizes the usage of memory on the metadata server. This option analyzes the handling of character variables, checking for items such as unnecessary indexes, long character variables that can be stored as variable-length strings, and duplicate strings. To apply the results of the analysis, you must pause and resume (or restart) the server after executing this command option. For more details, see "Using optimizeIMDB to Optimize and Report on Metadata Server Memory Usage" on page 197.

Use Individual Scripts to Operate the SAS/CONNECT Spawner, SAS/SHARE Server, and SAS Deployment Tester Server

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 5.5 Scripts for Operating Servers and Spawners

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

See Also

- *Communications Access Methods for SAS/CONNECT and SAS/SHARE*
- *SAS/SHARE User's Guide*
- “Using the Deployment Tester” on page 89

Starting or Stopping the SAS Services Application (Remote Services)

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 Remote Services as a Windows service.

   `-remove`
   removes the Remote Services service from Windows.

   `-start`
   starts Remote Services.

   `-stop`
   stops Remote Services.

   `-restart`
   restarts Remote Services.

   `-status`
   displays the current status of Remote Services.

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 37.

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 78.

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, SAS/CONNECT spawners, and SAS/SHARE servers.
- To start the metadata server after stopping it, you must use a script or Windows services. If you want to be able to resume processing from SAS Management Console, 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 56 and "Using Scripts to Operate SAS Servers Individually" on page 63. For information about metadata server start-up parameters, see "Options for the Metadata Server Invocation Command" on page 339.

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 to pick up and apply any changes that have been made to the server or spawner configurations. As part of the refresh process, the spawner quiesces any servers that it has started. These servers shut down when their clients have completed their work.

   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 56 and “Using Scripts to Operate SAS Servers Individually” on page 63. For information about start-up options for the object spawner, see “Spawner Invocation Options” in Chapter 12 of *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.

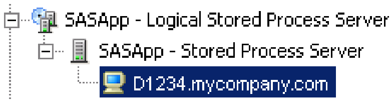
   Note: To start the OLAP server after quiescing it, you must use a script or Windows services. If you want to be able resume processing from SAS Management Console, use **Pause** instead.

To start the OLAP server, see “Running Servers as Windows Services” on page 56 and “Using Scripts to Operate SAS Servers Individually” on page 63. For information about OLAP server start-up parameters, see “Overview of System Options for SAS Application Server Components” in Chapter 14 of *SAS Intelligence Platform: Application Server Administration Guide*.

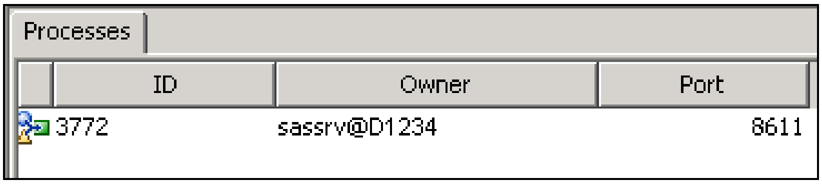
Use SAS Management Console to Control Processes on Servers

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.


SASApp - Logical Stored Process Server
SASApp - Stored Process Server
D1234.mycompany.com

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:

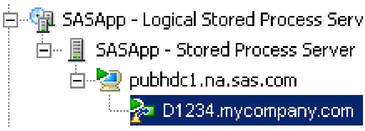


Processes

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:



SASApp - Logical Stored Process Server
SASApp - Stored Process Server
pubhdc1.na.sas.com
D1234.mycompany.com

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 37. For details about the tabs, see "Use the Server Manager Tabs to Obtain Monitoring Information and to Adjust Logging Levels (If Needed)" on page 107.

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 (Remote Services)” on page 66.
2. Use one of the following scripts:
   - On Windows:

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

   `-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 56.

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:
   - If the deployment manager 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 deployment manager:
     - On Windows:

       WAS_INSTALL_ROOT`\bin\startManager.bat -profileName "SASDmgr01"`
     - 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:
   - 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"
       ```

   SAS*host-name*01 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.

   SAS*host-name*01 is the default profile name on Windows and UNIX. If multiple servers are installed, then the default names for the additional profiles are SAS*host-name*02, SAS*host-name*03, 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 start-up 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 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: SAS 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 56.

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 start-up scripts.

Chapter 6
Checking the Status of Servers

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
- determine whether the SAS Framework Server is listening on the specified host and port and that can process a submitted query

To run more extensive validation tests on the SAS Content Server, you can use the SAS Deployment Tester plug-in in SAS Management Console or the SAS Deployment Tester batch clients. See “Using the Deployment Tester” on page 89 or the SAS Management Console Help. 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*.

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 63.

On UNIX and z/OS, you can use the **-status** option of the sas.servers script to display the status of the 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 60.

Checking the Access State and Other Properties of the SAS Metadata Server

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.

The Offline state is 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, with the following exceptions
  - users who are in the **Metadata Server: Unrestricted** role can continue to perform all functions in SAS Management Console.
  - users who are in both the **Metadata Server: Operation** and the **Management Console: Advanced** role (or who are in the SAS Administrators group) can stop, pause, resume, and reset the metadata server.

  In all other situations, 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

In addition, the server is automatically placed in the Administration state in the following situations:

- When you execute the Metadata Analyze and Repair tools.
- If a problems occur during server initialization when the server is started or resumed. When the server is in the Administration state, the administrator can log on to SAS Management Console, examine the server log to determine the source of the problem, make any necessary corrections, and resume the 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.

Use SAS Management Console to Check the Access State and Other Properties of the SAS Metadata Server

To use SAS Management Console to check the access state and other properties 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 Active Server Properties dialog box appears.

   In this dialog box, the **General** tab displays the following:

   - the server's current state and, if the server is paused, any comments that were entered by the administrator
   - information about whether a backup is currently running
   - the server's locale, version, and alert e-mail address
   - a button for testing alert e-mails
3. Select the **Statistics** tab to see server statistics including CPU seconds, memory utilization, thread utilization, and I/O activity.

See Also

"Use SAS Management Console to Stop, Pause, Resume, or Reset the SAS Metadata Server" on page 67

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 log:

```
NOTE: Server dXXXX.us.company.com SAS Version is 9.03.01M0P040611
NOTE: Server dXXXX.us.company.com SAS Long Version is 9.03.01M0P040611.
NOTE: Server dXXXX.us.company.com Operating System is X64_S08R2.
NOTE: Server dXXXX.us.company.com Operating System Family is WIN.
NOTE: Server dXXXX.us.company.com Operating System Version is 6.1.
NOTE: Server dXXXX.us.company.com Client is userid@DOMAIN.
NOTE: Server dXXXX.us.company.com Metadata Model is Version 12.04.
NOTE: Server dXXXX.us.company.com is RUNNING on 09May11:10:31:26.
```

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

```
NOTE: Server dXXXX.us.company.com SAS Version is 9.03.01M0P040611
NOTE: Server dXXXX.us.company.com SAS Long Version is 9.03.01M0P040611.
NOTE: Server dXXXX.us.company.com Operating System is X64_S08R2.
NOTE: Server dXXXX.us.company.com Operating System Family is WIN.
NOTE: Server dXXXX.us.company.com Operating System Version is 6.1.
NOTE: Server dXXXX.us.company.com Client is userid@DOMAIN.
NOTE: Server dXXXX.us.company.com Metadata Model is Version 12.04.
NOTE: Server dXXXX.us.company.com is PAUSED on 09May11:10:31:26.
```

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 25.

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, as described in "About SAS Metadata Repository Access Modes" on page 205. 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 or 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.

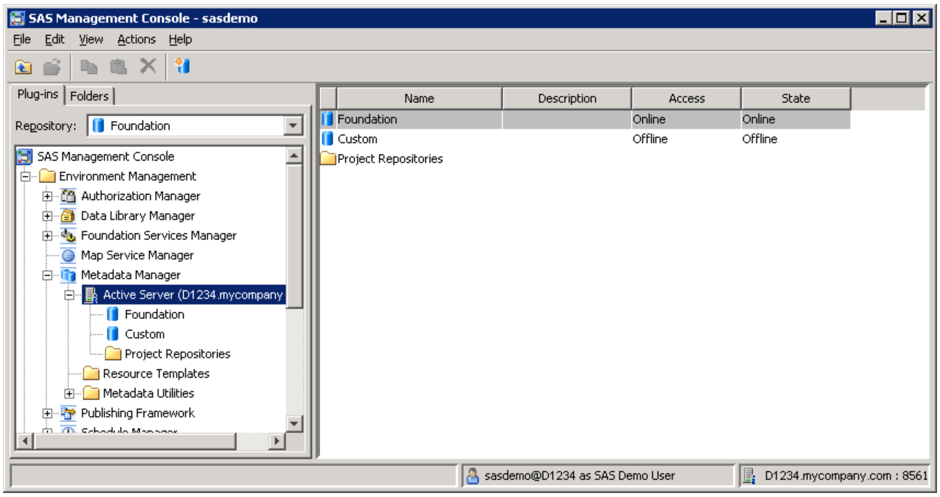
Use SAS Management Console to Check the Access State of a SAS Metadata Repository

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:



Name	Description	Access	State
Foundation		Online	Online
Custom		Offline	Offline
Project Repositories			

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 82.

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, as described in “How the Access State of the Metadata Server Affects the Access States of Metadata Repositories” on page 82. 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="12"
         Access="OMS_FULL"
         CurrentAccess="OMS_FULL"
         PauseState=""
         Path=”rposmgr”/>
      <Repository Id="A00000001.A50Q21KN"
         Name="Foundation"
         Desc=""
         DefaultNS="SAS"
         RepositoryType="FOUNDATION"
         RepositoryFormat="12"
         Access="OMS_FULL"
         CurrentAccess="OMS_FULL"
         PauseState=""
         Path=”MetadataRepositories/Foundation”/>
      <Repository Id="A00000001.A5IPLNM1"
         Name="Test Repository"
         Desc=""
         DefaultNS="SAS"
         RepositoryType="CUSTOM"
         RepositoryFormat="12"
         Access="OMS_READONLY"
         CurrentAccess="OMS_READONLY"
```

```
PauseState="READONLY"
Path=”MetadataRepositories/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 37.
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 37.

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.

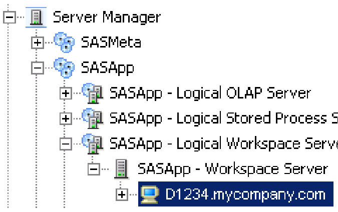
   b. Select **Validate**.

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:



Server Manager
SASMeta
SASApp
SASApp - Logical OLAP Server
SASApp - Logical Stored Process Server
SASApp - Logical Workspace Server
SASApp - Workspace Server
D1234.mycompany.com

   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 Appendix 2, "Troubleshooting the SAS Server Tier," in *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 "Using the Deployment Tester" on page 89 or the SAS Management Console Help.

Validate the SAS Content Server and the SAS Framework Data Server

If your installation includes a SAS Content Server and SAS Framework Data Server, you can use the Validate feature to determine whether these servers is set up correctly, as follows:.

Note: The user who validates the framework data server must be specified as a system user in the server's configuration file (dfs_serv.xml, located in ***`SAS-configuration-directory`*****`/Lev1/FrameworkServer/dffedsvrcfg`**). By default, this file specifies the SAS Administrator as a system user.

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

   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.
3. Expand the node for the logical SAS Framework Data Server. Then right-click the **FrameworkServer** node and select **Validate**.

   SAS Management Console attempts to connect to the server and submit a query. If the connection and the query are successful, the following message is displayed:

   ```
   Validation Successful!
   ```

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

Note: To run more extensive validation tests on the SAS Content Server, you can use the SAS Deployment Tester plug-in in SAS Management Console or the SAS Deployment Tester batch clients. See "Using the Deployment Tester" on page 89 or the SAS Management Console Help.

Chapter 7
Using the Deployment Tester

Deployment Tester Overview

What Is the Deployment Tester?

The Deployment Tester is a diagnostic tool used for assessing a SAS deployment. After an installation or upgrade, you can use the Deployment Tester to ensure that your SAS software and critical server components have been installed and configured correctly.

After making changes to your SAS deployment or to the underlying hardware or operating environment, you can use the Deployment Tester to ensure that SAS components are still in working order. The Deployment Tester is also useful for ongoing monitoring of the health of your deployment, as well as for troubleshooting when problems occur.

The Deployment Tester consists of the following components:

- the Deployment Tester server. This server is installed on each server-tier machine in a SAS deployment and on each middle-tier machine that is running Remote Services.
- the Deployment Tester plug-in for SAS Management Console, which enables you to run deployment tests interactively from a single interface. You can also view test results and manage test result files from the plug-in.
- the Deployment Tester batch client, which enables you to run deployment tests from a command line or on a scheduled basis. The batch client is installed on each machine where Foundation SAS is installed.
- the following test suites:
  - the SAS Installation Qualification Tool (SAS IQ) test suite, which demonstrates that your SAS software has been installed and maintained according to specifications. SAS IQ is also delivered as stand-alone application that can be run independently of the Deployment Tester.
  - the SAS Operational Qualification Tool (SAS OQ) test suite, which demonstrates that your Base SAS software is operational. SAS OQ is also delivered as stand-alone application that can be run independently of the Deployment Tester.
  - suites to test the functionality of SAS OLAP Servers, SAS Stored Process Servers, SAS Content Servers, and Remote Services
  - suites to check SAS licenses and their expiration dates on SAS Pooled Workspace Server and SAS Workspace Server host machines

You can use SAS OQ to create additional tests for your SAS software, and additional test suites are available for download. As tests are developed by SAS or by your organization, you can add them to the Deployment Tester.

When to Use the Deployment Tester

The validation tools that are described in “Checking the Status of Servers” on page 77 are useful when you want to perform a quick check of your SAS servers. The Deployment Tester supplements the validation tools by providing more extensive testing. As a best practice, you should use the Deployment Tester at the following times to ensure that your SAS deployment is functional and operational:

- after you have used the SAS Deployment Wizard to install new software, to upgrade your SAS software, or to configure a new SAS Application Server component
- after you have made any type of change to the configuration of Remote Services, the SAS Content Server, the SAS Framework Data Server, or a SAS Application Server component
- after you have used SAS Deployment Manager to uninstall SAS software, to remove an existing configuration, to update host name references, or to apply a hot fix
- after you have promoted server metadata from one metadata repository to another
- after you have used the SAS Migration Utility to migrate your SAS content from one SAS 9.3 deployment to another

- after you have recovered metadata repository files from a backup

When your site experiences a problem with the operation of SAS software, you can use the Deployment Tester as a troubleshooting tool to help identify the cause. If necessary, you can save the test results and send them to SAS Technical Support for troubleshooting assistance.

Using the Deployment Tester on a Regularly Scheduled Basis

You might want to run the Deployment Tester on a regular schedule (for example, daily or once a week) to check the health of your system. This feature can be helpful in complying with regulatory policies. To run tests on a scheduled basis, use the batch client. See "Using the Deployment Tester Batch Client" on page 96.

Prerequisites for Using the Deployment Tester

Installing JUnit

To run certain test suites, the Deployment Tester requires JUnit, which is an open-source framework for writing and running tests in the Java programming language. Before your SAS software was installed, JUnit should have been installed on each Deployment Tester server machine, as specified on the pre-installation checklist.

If JUnit is not found when you attempt to run a test that requires it, an error message appears. The message includes information about the path in which to install the JAR file for the JUnit software.

The JUnit software can be downloaded from `http://support.sas.com/thirdpartysupport`. On the Third-Party Software Requirements page, select the **Other Third Party Software Products** link for your SAS software release. On the SAS 9.3 Support for Additional Software Products page, note the version of JUnit that is required. Then click the **Download** link for JUnit and follow the instructions to download the software.

After downloading the software, move the JAR file to the appropriate directory. If the name of the JAR file includes a version number, you must rename the file to junit.jar.

Starting the Deployment Tester Server

The Deployment Tester server is installed on each server-tier machine in a SAS deployment and on each middle-tier machine that is running Remote Services. Before executing a Deployment Tester test suite, you must make sure that the Deployment Tester server is running on the host machine for the process that you are testing.

Because the Deployment Tester server is an optional server, it set up to be started manually. It is set up as a manual service on Windows machines, and it is not included in the sas.servers script on UNIX and z/OS machines. When you finish installing and configuring SAS 9.3, the server is automatically started on each server-tier and middle-tier machine so that you can test the newly installed components. After you perform these initial tests, it is recommended that you stop the server so that the resources will be available for other processes.

When you need to use the Deployment Tester plug-in or batch client, use one of the following procedures to start the Deployment Tester server:

- On Windows: Start the Deployment Tester server Windows service; start the server from the Windows **Start** menu; or go to ***`SAS-configuration-directory`***`\Lev1\DeploymentTesterServer` and run the command **`DeploymentTesterServer.bat -start`**. When you are finished using the Deployment Tester, stop the server by using the same command with the **`-stop`** option.
- On UNIX: Go to ***`SAS-configuration-directory`***`/Lev1/DeploymentTesterServer` and run the command **`DeploymentTesterServer.sh -start`**. When you are finished using the Deployment Tester, stop the server by using the same command with the **`-stop`** option.
- For z/OS, see "Starting and Stopping SAS Servers on z/OS" on page 58.

Deployment Tester Test Suites: Descriptions and Requirements

Deployment Tester Test Suites and Their Requirements

The following table describes the Deployment Tester test suites that are installed with the SAS Intelligence Platform and indicates whether each test suite requires JUnit or special credentials.

Test Suite	Test Description	JUnit Required?	Required User Credentials
Content Server Tests	Runs basic unit tests for validating the SAS Content Server.	Yes	Unrestricted user account (such as sasadm@saspw)
OLAP Server Tests	Runs basic functionality tests for the SAS OLAP Server.	No	Unrestricted user account (such as sasadm@saspw) or an account that is a member of the SAS Administrators group
Pooled Workspace Server setinit Test	Creates a report of the SAS software that is installed on the SAS Pooled Workspace Server host machine. The report shows the license expiration date for each product.	No	Regular user account
Remote Services Tests	Runs basic unit tests for validating the configuration, deployment, and usage of Remote Services.	Yes	Unrestricted user account (such as sasadm@saspw)
SAS OQ	Runs the SAS Operational Qualification Tool (SAS OQ) test suite, which demonstrates that your Base SAS software is operational.*	No	No credentials required

Test Suite	Test Description	JUnit Required?	Required User Credentials
SAS IQ	Runs the SAS Installation Qualification Tool (SAS IQ) test suite, which demonstrates that your SAS software has been installed and maintained according to specifications.*	No	No credentials required
Stored Process Server Test	Runs basic tests for the SAS Stored Process Server.	Yes	Regular user account
Workspace Server setinit Test	Creates a report of the SAS software that is installed on the SAS Workspace Server host machine. The report shows the license expiration date for each product.	No	Regular user account (must have ReadMetadata permission on the server metadata). On Windows, the user might need to have a host account with rights to "Logon as a batch job."

* For details about SAS IQ and SAS OQ, see the *SAS Qualification Tools User's Guide* at **http://support.sas.com**.

Adding Tests to the Deployment Tester

Adding SAS OQ Tests

The primary input files for the SAS OQ test suite are test tables, which are stored in files with the extension .tab. These files define which tests to run, how to run the tests, and how to determine whether the tests worked correctly. You can add more SAS OQ tests to the test suite by creating new test tables or adding new tests to the existing tables. For details, see the *SAS Qualification Tools User's Guide* at **http://support.sas.com**.

Adding the AIX System Tuning Test Suite

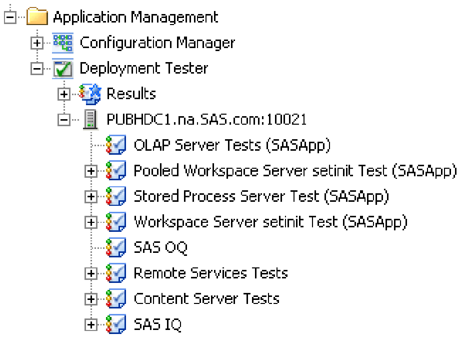
Administrators can use the AIX System Tuning test suite for SAS 9.3 Deployment Tester to optimize the performance of AIX environments. The test suite compares the AIX configuration to recommended best practices. To download the test suite, go to the Enterprise Management Integration page at **http://support.sas.com/rnd/emi/DeployTest/index.html**.

Using the Deployment Tester Interactively from SAS Management Console

Launching Deployment Tester Tests from SAS Management Console

To run the Deployment Tester interactively from SAS Management Console:

1. Start the Deployment Tester server on the host machine for each component that you want to test.
2. Log on to SAS Management Console as a user who is a member of the **Management Console: Advanced** role or the SAS Administrators group. On the **Plug-ins** tab, expand the **Application Manager** node and the **Deployment Tester** node.
3. In the Deployment Tester tree, locate the node for the Deployment Tester server where you want to run tests. The server is identified by the host machine name and the port (for example, server.mycompany.com:10021). Expand the server node to see the list of available tests, as shown in the following example:


Application Management
Configuration Manager
Deployment Tester
Results
PUBHDC1.na.SAS.com:10021
OLAP Server Tests (SASApp)
Pooled Workspace Server setinit Test (SASApp)
Stored Process Server Test (SASApp)
Workspace Server setinit Test (SASApp)
SAS OQ
Remote Services Tests
Content Server Tests
SAS IQ

TIP If no tests appear, this might mean that the Deployment Tester server has not been started.

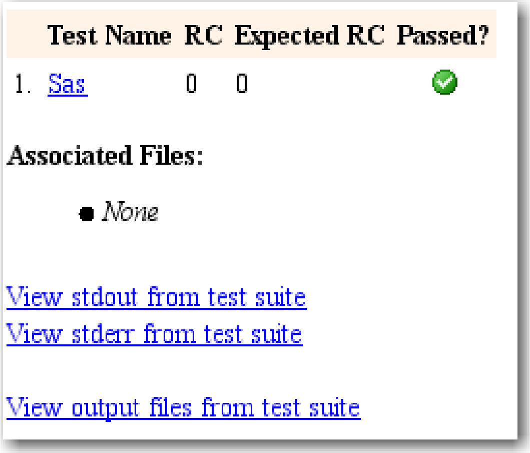
4. Right-click the test that you want to run, and select **Run Test Suite**. (Or if you want to run all of the tests that are available on this machine, right-click the node for the Deployment Tester server and select **Run All Test Suites**.)
5. Some tests require special user credentials. If you are prompted for credentials, enter the appropriate user ID and password. The password will be encrypted using SAS proprietary 32-bit (sas002) encryption. For details about required credentials, see "Deployment Tester Test Suites: Descriptions and Requirements" on page 92.

Test suites run as background processes, so you can perform other work while a test suite is running. When the tests are completed, the results are displayed in the right panel.

When you are finished executing tests, it is recommended that you stop the Deployment Tester server.

View Deployment Tester Results in SAS Management Console

When you run a Deployment Tester test suite interactively from SAS Management Console, the test results appear as soon as the test is completed. Each result is listed in the Results tree under the Deployment Tester plug-in. When you select a result in the tree, the Test Suite Results report appears in the right pane, as shown in the following example:



	Test Name	RC	Expected RC	Passed?
1.	Sas	0	0	✓

Associated Files:

- *None*

View stdout from test suite
View stderr from test suite

View output files from test suite

A green icon with a check mark (✓) indicates that the test completed successfully.

CAUTION:

> **If you right-click a test result in the Results tree and select “Close Results,” the result files are deleted from the tree and the associated documents are deleted from the file system.**

To view the output reports for a test, click the links on the results panel. Depending on which test you ran, you can view results by clicking the test name, **View stdout from test suite**, **View stderr from test suite**, or **View output files from test suite**.

If you need assistance in interpreting test results, contact SAS Technical Support.

Where Deployment Tester Result Files Are Stored When Tests Are Run from SAS Management Console

The result files that you can view from SAS Management Console are saved in the following paths in the file system:

- Windows Vista, Windows 7, and Windows Server 2008 systems: `C:\Users\user\AppData\Roaming\SAS\SASDeploymentTesterSASMCPlugin\2.1\Reports\yyyymmdd\report_test-suite-name`
- Other Windows systems: `C:\Documents and Settings\user\Application Data\SAS\SASDeploymentTesterSASMCPlugin\2.1\Reports\yyyymmdd\report_test-suite-name`
- UNIX: `user's-home-directory/SAS/SASDeploymentTesterSASMCPlugin/2.1/Reports/yyyymmdd/report_test-suite-name`

Note: When you right-click a test result in the Results tree and select **Close Results**, the result files are deleted from the tree and from the paths in the preceding list.

Permanently Saving Deployment Tester Test Results

To create a permanent record of tests that have been run, you can save the results to a location other than the user's home directory. To save the results of a test, right-click the test in the Results tree and make one of the following selections:

- Select **Save Results to Folder** if you want to save the results as a collection of loose files. In the dialog box that appears, navigate to the path where the results are to be saved. Then create a folder in which to store the files, select the new folder, and click **Save**.
- Select **Save Results as Zip File** if you want to save the results in a compressed file. This feature is helpful if you want to send the test results to another party such as SAS Technical Support.

  In the dialog box that appears, you might want to navigate to a different directory before you save the file. To do so, select **All Files** in the **Files of type** field. After navigating to the correct directory, change **Files of type** back to **Zip files (*.zip)**. Then enter a filename and click **Save**.

Deleting (Closing) Deployment Tester Test Results

Periodically, you might want to delete old Deployment Tester results from the Results tree in SAS Management Console and from your home directory. If there is a chance that you will need to refer to the results in the future, be sure to first save the results to another location, as described in the preceding topic.

To delete Deployment Tester results from the Results tree and from the users's home directory, select one or more results in the tree. Then right-click the mouse and select **Close Results**.

Using the Deployment Tester Batch Client

About Running the Deployment Tester in Batch Mode

The Deployment Tester can be run as a batch client, which enables you to automate the testing process. To regularly check on the health of your system, you can set up your scheduling system to run the batch client daily, weekly, or on any other schedule.

The batch client can be run from any machine where the Deployment Tester batch client is installed. Before running the batch client, be sure to start the Deployment Tester server on the host machine for each component that you want to test.

Syntax for the Batch Deployment Tester Command

```
bclient.sh | bclient.bat
    -contract SAS-installation-directory/SASDeploymentTesterClient/2.1/
      Contract/SASTestSuite_contract.xml
      -Vserver=dt-server
      -Vdir=contract-file-dir
      -VcontractFile=contract-file-name
```

Syntax Description for the Batch Deployment Tester Command

bclient.sh | bclient.bat

specifies the command for starting the batch Deployment Tester. Use -bclient.sh for UNIX or -bclient.bat for Windows.

Requirement: Required

-contract *SAS-installation-directory*/SASDeploymentTesterClient/2.1/Contract/SASTestSuite_contract.xml

specifies the name and location of the launcher contract file, SASTestSuite_contract.xml. This file calls other test-specific contract files. By default, the SAS Deployment Wizard installs this file in ***`SAS-installation-directory`*`/SASDeploymentTesterClient/2.1/Contract`**.

In ***`SAS-installation-directory`***, specify the path where your SAS software is installed on the local machine.

Requirement: Required

Example: **`-contract C:/Program Files/SASHome/SASDeploymentTesterClient/2.1/Contract/SASTestSuite_contract.xml`**

-Vserver=*dt-server*

specifies the host machine and port for the Deployment Tester server that is to run the test. Use the format *host-name:port*. The value that you specify depends on which SAS component you are testing. If you are running a test suite for a SAS server, then specify the SAS Deployment Server that is installed on the host machine for the server that you are testing. If you are running a test suite for Remote Services, then specify the SAS Deployment Server on the middle-tier machine where Remote Services is installed. If you are running SAS OQ or SAS IQ, then specify the SAS Deployment Server on the machine that contains the SAS software that you want to test.

If you are running only the SAS OQ or SAS IQ test suite, and if the SAS software that you are testing is on the same machine as the batch client, you can either omit this parameter or specify a value of **`server-local`**.

Requirement: This parameter is optional in the following situation: if you are running only the SAS OQ or SAS IQ test suite, and if the SAS software that you are testing is on the same machine as the batch client. In all other situations, the parameter is required.

Tip: The host machine names and ports for each instance of the Deployment Tester server are displayed under **Application Management** ⇨ **Deployment Tester** in the **Plug-ins** tab of SAS Management Console.

Example: **`-Vserver=server.mycompany.com:10021`**

-Vdir=*contract-file-dir*

specifies the directory path for the contract file that is specified in **`-Vserver=`***contract-file-name*. The path varies depending on which test suite you are running. To determine the correct path, see "Filenames and Locations for Deployment Tester Test Suite Contracts" on page 98.

Requirement: Required

-VcontractFile=*contract-file-name*

specifies the name of the contract file for the test suite that you want to run. To determine the correct filename, see "Filenames and Locations for Deployment Tester Test Suite Contracts" on page 98.

Requirement: Required

-Vuserid=*my_userid*
specifies a valid user ID to access the component that is being tested.

Requirement: Required for the following test suites: Content Server Tests, OLAP Server Tests, Pooled Workspace Server setinit Test, Remote Services Tests, Stored Process Server Test, and Workspace Server setinit Test.

See: For details about credential requirements, see "Deployment Tester Test Suites and Their Requirements" on page 92.

-Vpassword=*my_password*
specifies the password associated with the user ID that is specified in -Vuserid=. The password should be encrypted using SAS proprietary 32-bit encryption. To obtain the encrypted password, use PROC PWENCODE. If you do not encrypt the password, it will be transmitted using plain text.

Requirement: Required for the following test suites: Content Server Tests, OLAP Server Tests, Pooled Workspace Server setinit Test, Remote Services Tests, Stored Process Server Test, and Workspace Server setinit Test.

Filenames and Locations for Deployment Tester Test Suite Contracts

The following table provides default directory paths and filenames for the contract files that are used for each Deployment Tester test suite. These values are used in the -Vdir= and -VcontractFile parameters of the batch command.

Note: The directory paths that are shown here are default locations. The paths in your deployment might be different.

Test Suite	Default Directory Path for Contract File for -Vdir=	Contract Filename for -VcontractFile
Content Server Tests	***SAS-configuration-directory*/Lev1/Web/Applications/SASWIPServices9.3/dtest/SASContentServer**	ContentServerTest_contract.xml
OLAP Server Tests	***SAS-configuration-directory*/Lev1/SASApp/OLAPServer/dtest/OLAPServer**	olap_contract.xml
Pooled Workspace Server setinit Test	***SAS-configuration-directory*/Lev1/SASApp/PooledWorkspaceServer/dtest/PooledWorkspaceServer**	SetinitTest_contract.xml
Remote Services Tests	***SAS-configuration-directory*/Lev1/Web/Applications/RemoteServices/dtest/RemoteServices**	RSTests_contract.xml
SAS IQ	***SAS-installation-directory*InstallMisc/utilities/installqual/9.3**	sasiq_contract.xml
SAS OQ	***SAS-installation-directory*/dtest/sasoq**	sasoq_contract.xml
Stored Process Server Test	***SAS-configuration-directory*/Lev1/SASApp/StoredProcessServer/dtest/StoredProcessServer**	STPTest_2_contract.xml

Test Suite	Default Directory Path for Contract File for -Vdir=	Contract Filename for -VcontractFile
Workspace Server setinit Test	***SAS-configuration-directory*/Lev1/SASApp/ WorkspaceServer/dtest/WorkspaceServer**	SetinitTest_contract.xml

SAS Deployment Tester Batch Command: UNIX Example

The following is an example of a batch command issued on a UNIX machine to test a SAS OLAP Server that is located on a Windows machine:

```
cd $SASHOME/SASDeploymentTesterClient/2.1
bin/bclient.sh -contract Contract/SASTestSuite_contract.xml \
"-Vserver=server.sas.com:10021" \
"-Vdir C:\SAS\EBIServer\Lev1\SASApp\OLAPServer\dtest\OLAPServer" \
-VcontractFile=olap_contract.xml
"-Vuserid=domain\user123"
"-Vpassword={sas002}0123456789"
```

SAS Deployment Tester Batch Command: Windows Example

The following is an example of a batch command issued on a Windows machine to test a SAS OLAP Server that is located on a different Windows machine:

```
cd C:\Program Files\SASHome\SASDeploymentTesterClient\2.1
bin\bclient.bat -contract Contract\SASTestSuite_contract.xml
"-Vserver=server.sas.com:10021"
"-Vdir=C:\SAS\EBIServer\Lev1\SASApp\OLAPServer\dtest\OLAPServer"
-VcontractFile=olap_contract.xml
"-Vuserid=domain\user123"
"-Vpassword={sas002}0123456789"
```

Viewing Results of Tests That Are Run from the Deployment Tester Batch Client

If you run tests on a batch basis, you can access the test results by opening the following file:

- On Windows Vista, Windows 7, and Windows Server 2008 systems: **C:\Users \*user*\AppData\Roaming\SAS\SASDeploymentTesterClient \2.1\Reports\yyyymmdd\report_*test-suite-name*\index.html**
- On other Windows systems: **C:\Documents and Settings\*user* \Application Data\SAS\SASDeploymentTesterClient\2.1\Reports \yyyymmdd\report_*test-suite-name*\index.html**
- On UNIX: ***user's-home-directory*/.SASAppData/ SASDeploymentTesterClient/2.1/Reports/*yyyymmdd*/report_*test-suite-name*/index.html**

Part 3

System Monitoring and Logging

Chapter 8
Monitoring the Activity of SAS Servers

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, framework data 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 104.
- third-party monitoring tools. For details, see "Using Third-Party Products to Monitor SAS Servers" on page 118.

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

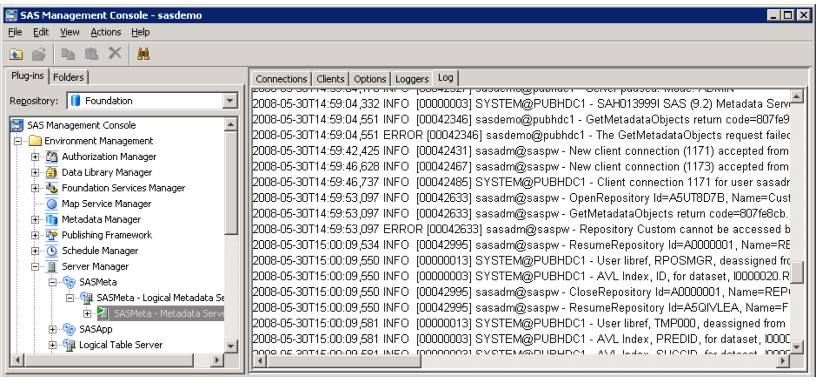
Using SAS Management Console to Monitor SAS Servers

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, framework data servers, and workspace servers. You can view the following types 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:



```
2008-05-30T14:59:04,332 INFO  [00000003] SYSTEM@PUBHDC1 - SAH013999I SAS (9.2) Metadata Serv
2008-05-30T14:59:04,551 INFO  [00042346] sasdemo@pubhdc1 - GetMetadataObjects return code=807fe9
2008-05-30T14:59:04,551 ERROR [00042346] sasdemo@pubhdc1 - The GetMetadataObjects request failec
2008-05-30T14:59:42,425 INFO  [00042431] sasadm@saspw - New client connection (1171) accepted from
2008-05-30T14:59:46,628 INFO  [00042467] sasadm@saspw - New client connection (1173) accepted from
2008-05-30T14:59:46,737 INFO  [00042485] SYSTEM@PUBHDC1 - Client connection 1171 for user sasadr
2008-05-30T14:59:53,097 INFO  [00042633] sasadm@saspw - OpenRepository Id=A5UT8D7B, Name=Cust
2008-05-30T14:59:53,097 INFO  [00042633] sasadm@saspw - GetMetadataObjects return code=807fe8cb.
2008-05-30T14:59:53,097 ERROR [00042633] sasadm@saspw - Repository Custom cannot be accessed b
2008-05-30T15:00:09,534 INFO  [00042995] sasadm@saspw - ResumeRepository Id=A0000001, Name=RE
2008-05-30T15:00:09,550 INFO  [00000013] SYSTEM@PUBHDC1 - User libref, RPOSMGR, deassigned frc
2008-05-30T15:00:09,550 INFO  [00000003] SYSTEM@PUBHDC1 - AVL Index, ID, for dataset, I0000020.R
2008-05-30T15:00:09,550 INFO  [00042995] sasadm@saspw - CloseRepository Id=A0000001, Name=REP
2008-05-30T15:00:09,550 INFO  [00042995] sasadm@saspw - ResumeRepository Id=A5QIVLEA, Name=F
2008-05-30T15:00:09,581 INFO  [00000013] SYSTEM@PUBHDC1 - User libref, TMP000, deassigned from
2008-05-30T15:00:09,581 INFO  [00000003] SYSTEM@PUBHDC1 - AVL Index, PREDID, for dataset, I000C
```

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 step:

1. Obtain the appropriate user credentials and clear cached credentials (if necessary).
2. Use SAS Management Console to connect to servers and display server monitoring information.
3. Use the Server Manager tabs to obtain monitoring information and to adjust logging levels (if needed).

4. 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 37.

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 37.
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 framework data 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 framework data servers, a **Sessions** tab also appears.

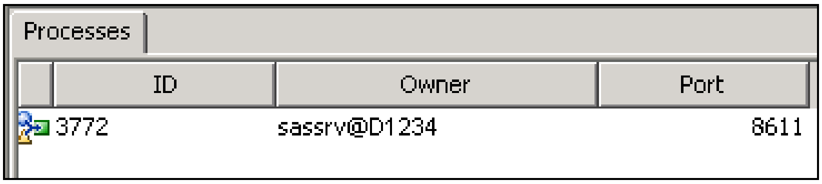
      Note: If some tabs are disabled, then you need additional user credentials. See "Who Can Do What in the Server Manager Plug-in" on page 37.

      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 107.
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:



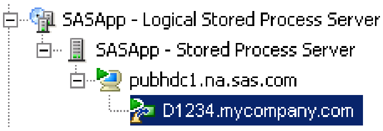
Processes

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:



SASApp - Logical Stored Process Server
SASApp - Stored Process Server
pubhdc1.na.sas.com
D1234.mycompany.com

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 37.

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 107.

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 37.

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 107.

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.

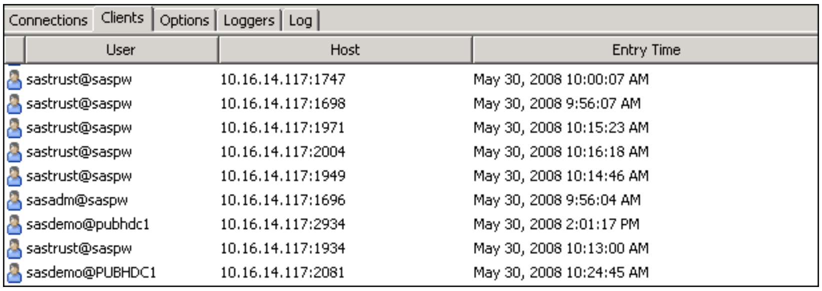
Depending on which server component you are monitoring, SAS Management Console might display the following tabs:

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

Note: If some tabs are disabled, then you might need to connect to the server or you might need additional user credentials.

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:



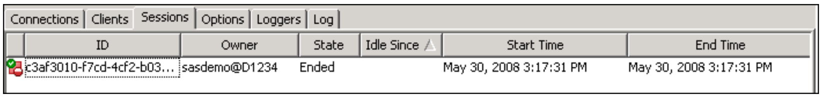
User	Host	Entry Time
sastrust@saspw	10.16.14.117:1747	May 30, 2008 10:00:07 AM
sastrust@saspw	10.16.14.117:1698	May 30, 2008 9:56:07 AM
sastrust@saspw	10.16.14.117:1971	May 30, 2008 10:15:23 AM
sastrust@saspw	10.16.14.117:2004	May 30, 2008 10:16:18 AM
sastrust@saspw	10.16.14.117:1949	May 30, 2008 10:14:46 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
- framework data 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:



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.

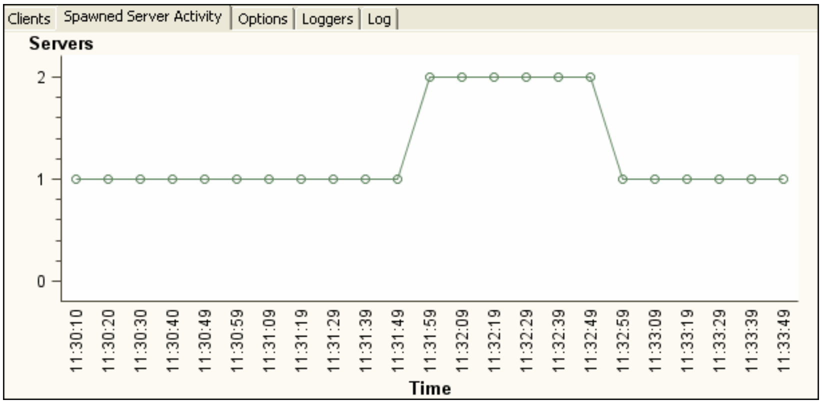
If terminated sessions are displayed, you can remove them by right-clicking the selected server node or process node in the server tree and selecting **Refresh**.

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

- OLAP servers
- stored process server processes
- framework data servers

Use the Spawned Server Activity Tab in the Server Manager

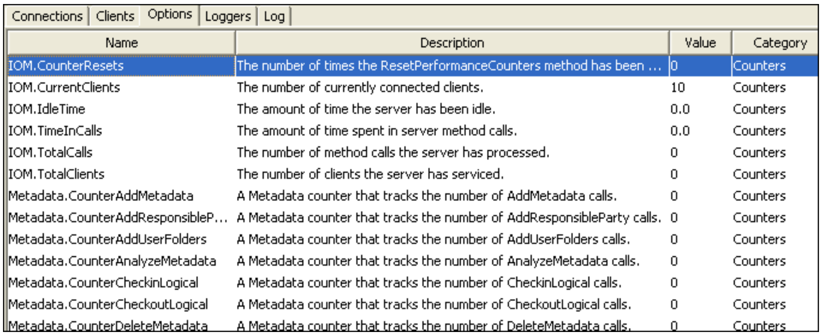
The **Spawned Server Activity** tab in the Server Manager displays a graph that shows server activity associated with the currently selected object spawner connection. The graph shows how many spawned workspace servers, pooled workspace servers, and stored process servers are active at a given point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



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 113.

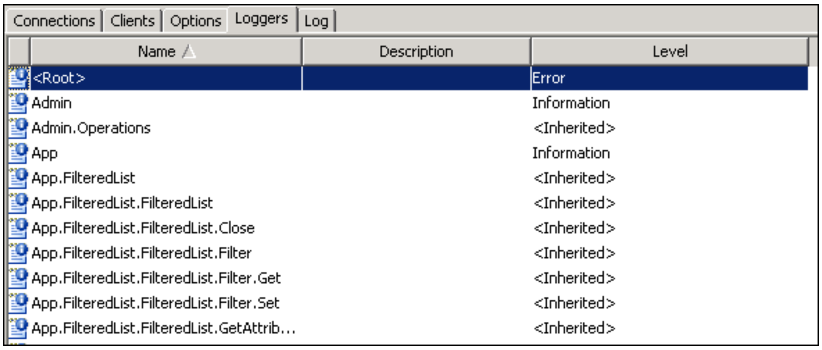
Some types of properties (for example, the JNLLINEMAX and JNLSTRMAX object server parameters) can be modified in this tab. To change the value of a property, right-click the property entry and click **Edit**. The changes remain in effect until the server component or process stops. When the server component is restarted, the options revert to those that are defined in the server's configuration file.

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
- framework data 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:



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.

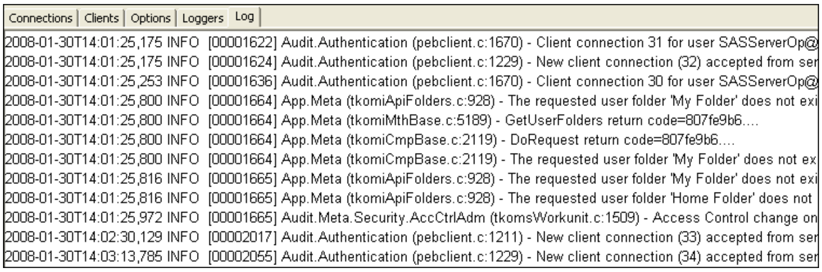
The **Loggers** tab is displayed for the following server components:

- metadata server
- object spawner connections
- OLAP servers
- pooled workspace server processes
- stored process server processes
- framework data servers
- workspace server processes

For more information, see "Administering Logging for SAS Servers" on page 119 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:



```
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 ex
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 **OK** 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 **OK** 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. 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
- framework data servers

For more information, see "Administering Logging for SAS Servers" on page 119 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, 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.

   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.
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 104.

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 118.

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 about 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.Counter`*method-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.CounterIO`*transaction-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.CounterJournal`*counter-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.CounterJournalDecompDataAvailable`
provides the raw number of bytes for the Metadata.CounterJournalDataAvailable counter, before compression.

`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.CounterJournalDecompDataAvailableMax`
provides the raw number of bytes for the Metadata.CounterJournalDataAvailableMax counter, before compression.

`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.CounterJournalDecompDataWritten`
provides the raw number of bytes for the Metadata.CounterJournalDataWritten counter, before compression.

`Metadata.CounterJournalMaxFileEntrySize`
indicates the maximum number of bytes of data in a single journal entry in the journal.

`Metadata.CounterJournalMaxDecompEntrySize`
provides the raw number of bytes for the Metadata.CounterJournalMaxFileEntrySize counter, before compression.

`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.CounterJournalTransQueue`d
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 334.

`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 MAXIMUM_QUERY_RETRY_COUNT on page 334.

`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.
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 Chapter 6 of *SAS Intelligence Platform: Middle-Tier Administration Guide*.

Chapter 9
Administering Logging for SAS Servers

About Server Logging

The SAS 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, Audit.Meta.Mgmt, and Audit.Meta.Updates. 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, and FATAL. 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 104.

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 134.

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).

Audit.Meta.Updates
: (new with SAS 9.3) is the parent logger for events that are performed on public objects. The logged events include adding, updating, deleting, checking out, checking in, and fetching objects, as well as undoing checkouts.

  Audit.Meta.Updates.PublicObjects logs events about public objects in foundation and custom repositories. Audit.Meta.Updates.Project and Audit.Meta.Updates.Project.PublicObjects log information about public objects in project repositories.

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.A300000O.

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 113.

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' start-up 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 “Initial Logging Configuration for SAS Servers” on page 123.

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.

Table 9.1 *Initial Logging Configuration for SAS Servers*

	Log Destination		
Server	**Rolling Log File and SAS Management Console Log Tab**	**Operating System Facility**	**ARM Logging Facility**
SAS Metadata Server	• Admin, App, Audit, and IOM events (INFO, WARN, ERROR, and FATAL) • all other events with a level of ERROR or FATAL	None	None

	Log Destination		
Server	**Rolling Log File and SAS Management Console Log Tab**	**Operating System Facility**	**ARM Logging Facility**
SAS object spawner	• App, Audit, and IOM events (INFO, WARN, ERROR, and FATAL) • all other events with a level of ERROR or FATAL	Admin events (INFO, WARN, ERROR, or FATAL)	None
SAS OLAP Server	• Admin events (ERROR and FATAL) • 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 132.
SAS Pooled Workspace Server	• Admin events (ERROR and FATAL) • App, Audit, and IOM events (INFO, WARN, ERROR, and FATAL) • App.Program events (ERROR and FATAL) • all other events with a level of ERROR or FATAL Events are sent only to the rolling log file.	None	None
SAS Stored Process Server	• Admin events (ERROR and FATAL) • App, Audit, and IOM events (INFO, WARN, ERROR, and FATAL) • App.Program events (FATAL) • all other events with a level of ERROR or FATAL		
SAS Framework Data Server	• Admin events (ERROR or FATAL) • App, Audit, and IOM events (INFO, WARN, ERROR, and FATAL) • 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)

	Log Destination		
Server	Rolling Log File and SAS Management Console Log Tab	Operating System Facility	ARM Logging Facility
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 130.
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 framework data 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 "Default Locations for Server Logs" on page 319.

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 start-up 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 129.

SAS Management Console (**Log** tab)

For metadata servers, object spawners, OLAP servers, stored process servers, and framework data 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 107.

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 132 and "Collecting ARM Log Information for SAS Data Integration Studio Jobs" on page 130.

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 104.
- 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"
  - "Create a Log File for Workspace Server Troubleshooting" on page 129
  - "Enable More Detailed Logging for SAS Object Spawner Troubleshooting" on page 129
- If you have installed the SAS Enterprise BI Audit and Performance Measurement package, you can use the alternative logging configuration files (called logconfig.apm.xml) that are provided for use with the package. The package is available for download from `http://support.sas.com/rnd/emi`.

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 130

- "Collecting ARM Log Information in SAS OLAP Servers" on page 132
- "Audit Server Logging Level Changes That Originate in Client Applications" on page 133
- "Capture Metadata Server Management Events in the System Facility Appender" on page 134
- "Capture XML Information in the SAS Metadata Server Log" on page 134

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, 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 129.

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, 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 104.

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.

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 104.

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.

After you have configured the batch server to collect ARM log information, you can use the SAS Data Integration Studio job statistics reporting facility to analyze the log events. For details, see "Configuring and Running Job Statistics Reports for SAS Data Integration Studio" on page 139.

Note: 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 37.
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*, specify 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 123.

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 about 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">
```

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 104 .

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. 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 (or in a separate log file) and used for troubleshooting. To capture the entries, use the one of the following procedures (or a combination of these procedures) as appropriate:

- "Capture XML Information in the Metadata Server Log for the Current Server Session Only"
- "Capture XML Information in the Metadata Server Log across Multiple Server Sessions"

Note: For more information about the call interfaces, see *SAS Language Interfaces to Metadata*.

Capture XML Information in the Metadata Server Log for the Current Server Session Only

If you only want to capture XML information during the current metadata server session, follow these steps:

1. Open the Server Manager in SAS Management Console and connect to the metadata server.
2. On the **Options** tab, change the settings for the **IOM.JnlStrMax** and **IOM.JnlLineMax** properties to a higher value (for example, 1000000). The parameters are explained in the following table:

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: **`*truncated to 500 characters (jnlStrMax)`**.
JNLLINEMAX	Specifies the maximum journal line length. The default is 1000 characters. If this parameter is omitted, a maximum of 1000 characters of the XML string is logged, and the following message is appended to the XML: **`*truncated to 1000 characters (jnlLineMax)`**.

3. On the **Loggers** tab, increase the logging level for the App.OMI logger to TRACE.

   The XML information will be captured in the metadata server log for the duration of the current server session, or until you return the App.OMI logger to its default level (Inherited). To avoid adverse effects on performance, you should reset the logger when you are finished capturing XML information.

Capture XML Information in the Metadata Server Log across Multiple Server Sessions

If you want to capture XML information across multiple metadata server sessions, follow these steps:

1. Update the object server parameters in the metadata server's configuration file as follows:

   a. Open the sasv9.cfg and sasv9_usermods.cfg files in a text editor. The files are located in *SAS-configuration-directory*`/Lev1/SASMeta/MetadataServer`.

   b. In sasv9.cfg, find the line that specifies object server parameters. The line begins with **`-objectserverparms`** (or **`objectserverparms`** for z/OS). Copy this line to sasv9_usermods.cfg.

   c. In sasv9_usermods.cfg, add JNLSTRMAX and JNLLINEMAX to the existing object server parameters. Enclose the entire list of parameters in quotation marks, as shown in the following example:

      ```
      -objectserverparms "jnlStrMax=1000000 jnlLineMax=1000000 cel=credentials"
      ```

      CAUTION:
      The typical configuration file contains more object server parameters than are shown in this example. Make sure not to delete or overwrite the existing parameters.

      You can specify different values for JNLSTRMAX and JNLLINEMAX than those that are shown in the example.

2. Open the server's logconfig.xml file (located in *SAS-configuration-directory*`/Lev1/SASMeta/MetadataServer`), and add the following lines:

   ```
   <logger name="App.OMI">
        <level value="trace"/>
        <appender-ref ref="IOMCALLTRACE"/>
   </logger>
   ```

   If you want to capture the XML information n a separate log file (instead of in the metadata server log), insert the following lines before the App.OMI logger element:

   ```
   <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>
   ```

   If you want, you can specify different values (for example, a different log filename) for the FileAppender parameters. For details, see the FileAppender topic in the *SAS Logging: Configuration and Programming Reference*.

3. Restart the metadata server.

To avoid adverse effects on performance, you should remove the App.OMI logger from logconfig.xml and restart the server when you are finished capturing XML information.

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:

```
2010-11-06T15:37:55,815 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:

```
2010-11-06T15:37:55,815 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="26Sep2010:16:12:48" MetadataUpdated="02Nov2010:18:10:37"
Options=""
Path="C:\SAS\Config\Lev1\SASMeta\MetadataServer\MetadataRepositories\Foundation"
RepositoryFormat="11" RepositoryType="FOUNDATION"/><RepositoryBase
Id="A0000001.A5PJCXVL" Name="Custom1" Access="0" Desc="" Engine="BASE"
MetadataCreated="26Sep2010:16:14:28" MetadataUpdated="02Nov2010:18:10:37"
Options=""
Path="C:\SAS\Config\Lev1\SASMeta\MetadataServer\MetadataRepositories\Custom1"
RepositoryFormat="11" RepositoryType="CUSTOM"/><RepositoryBase
Id="A0000001.A5FIU3O4" Name="Project1" Access="0" Desc="" Engine="BASE"
MetadataCreated="26Sep2010:16:15:53" MetadataUpdated="02Nov2010:18:10:37"
Options=""
Path="C:\SAS\Config\Lev1\SASMeta\MetadataServer\MetadataRepositories\Project1"
RepositoryFormat="11" RepositoryType="PROJECT"/></Objects><ns>REPOS</ns>
<Flags>264</Flags><options/></GetMetadataObjects> retval=0
```

Chapter 10
Configuring and Running Job Statistics Reports for SAS Data Integration Studio

About SAS Data Integration Studio Job Statistics Reports

The SAS Data Integration Studio job statistics reports enable you to use SAS BI reporting tools to analyze the performance of deployed jobs. The reporting package parses the contents of logs generated by the deployed jobs and generates statistical data sets summarized by job step. This data is used to create a variety of reports that enable you to track and analyze the following:

- job performance based on the contents of ARM logs

- error and warning messages
- source and target table names
- SAS metadata user identities

To use the job statistics reporting package, you must first install it on the server tier and customize it for your environment. You can then run the parsing programs and build the OLAP cubes (if applicable) on a regularly scheduled basis to produce detailed and summary job statistics data. A variety of predefined reports based on this data are provided. The reports can be accessed in different ways, depending on which SAS applications you have installed:

- You can access the reports in SAS Web Report Studio if you have it installed. The SAS Web Report Studio reports can also be accessed through the SAS Information Delivery Portal.
- If you have not installed SAS Web Report Studio, then reports are generated from SAS Stored Processes. You can view the output of the stored processes in the SAS Stored Process Web Application or the SAS Information Delivery Portal.

Configure Job Statistics Reporting for SAS Data Integration Studio

Summary of Steps for Configuring the Job Statistics Reporting Package

The job statistics reporting package is delivered with the SAS Data Integration Studio client application. Before you can use the package, complete the following steps to install it on your server tier and configure it for your environment:

- Create directories on the batch server host machine.
- Copy the reporting components to the batch server host machine.
- Customize the properties file, job logs parsing program, and batch script for your environment.
- Add the Java Runtime Environment executable to the PATH environment variable.
- Import the BI reporting objects from DIStudioJobStats.spk. This step is necessary only if you will be using SAS Web Report Studio to generate the job statistics reports.
- Import the stored processes and customize them for your environment. This step is optional if you will be using SAS Web Report Studio to generate the job statistics reports.
- Enable ARM logging and run-time statistics collection for deployed SAS Data Integration Studio jobs.
- Apply the required permissions.

Details are provided in the following topics.

Step 1: Create Directories on the Batch Server Host Machine

First, you must create directories on the SAS DATA step batch server host machine. Follow these steps on the host machine that runs the jobs for which you want to report statistics:

1. Create a directory to store the reporting package's programs, scripts, and configuration information. For example:
   - On Windows: `C:\SASDIStudioJobReports`
   - On UNIX: `/usr/local/SASDIStudioJobReports`
2. Create a directory to be used as the library location for the job statistics data sets. For example:
   - On Windows: `C:\SASDIStudioJobStats`
   - On UNIX: `/usr/local/SASDIStudioJobStats`

   Give Write access to this directory for the user who will build the OLAP cubes for job statistics reporting. (This step is required only if you will be using SAS Web Report Studio to generate reports).
3. Create a directory to contain the source code for stored processes that produce job statistics reports. (This step is optional if you will be using SAS Web Report Studio to generate reports.) Here are examples:
   - On Windows: `C:\SASDIStudioJobStoredProcesses`
   - On UNIX: `/usr/local/SASDIStudioJobStoredProcesses`

   Give Write access to this directory for the user who will import the stored process objects. (See "Step 6: Import Stored Processes and Customize Them for Your Environment" on page 144.)

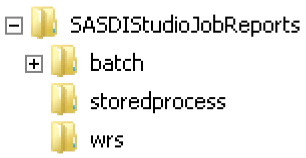
   Note: In the subsequent configuration steps, the directories are referred to using the example names.

Step 2: Copy Reporting Components to the Batch Server Host Machine

On the machine where SAS Data Integration Studio client software is installed, navigate to the following directory:

`SAS-installation-directory``/SASDataIntegrationStudio/4.3/jobreports`

Copy the contents of the `jobreports` directory from the client machine to the `SASDIStudioJobReports` directory on the batch server machine. The resulting directory should have the following structure:



SASDIStudioJobReports
batch
storedprocess
wrs

Step 3: Customize the Properties File, Job Logs Parsing Program, and Batch Script for Your Environment

On the batch server machine, follow these steps to customize the properties file, job logs parsing program, and batch scripts for your environment:

1. In the path **`SASDIStudioJobReports/batch`**, locate the job statistics properties file for your operating system environment (for example, DIStudioJobStats.properties.Windows). Copy this file to DIStudioJobStats.Properties.

   Then open DIStudioJobStats.Properties in a text editor and make the following changes:

   a. Find the line that contains **`DIStudioJobStats.logdir=`**. Update this property to point to the location of the logs for your batch server. For example:

   ```
   DIStudioJobStats.logdir=C\:\\SAS\\Config\\Lev1\\SASApp\\BatchServer\\Logs
   ```

   Note that in Windows environments, each colon (:) and backslash (\) must be preceded by an additional backslash character.

   Note: For best results, the batch server logs should be stored in their own directory rather than in a directory that is shared by other server logs. To determine the configured location, open the batch server's logconfig.xml file (located in ***`SAS-configuration-directory`*****`/Lev1/SASApp/BatchServer`**). In this file, check the FileNamePattern parameter of the RollingFileAppender configuration. If a unique location is not specified in this parameter, specify a location such as ***`SAS-configuration-directory`*****`/Lev1/SASApp/BatchServer/Logs`**.

   b. Find the line that contains **`DIStudioJobStats.user=`**. This property should specify the user ID for the SAS General Server account (sometimes referred to as the SAS Spawned Server account) that is defined in your operating system.

   Typically, the user ID for this user is sassrv. If a different user is specified, update this property with the appropriate user ID.

   c. Find the line that contains **`DIStudioJobStats.password=`**. Update this property with the password of the user that is specified in **`DIStudioJobStats.user=`**. The password should be encrypted using SAS proprietary 32-bit encryption. To obtain the encrypted password, use PROC PWENCODE. Specify the encrypted password as shown in the following example:

   ```
   DIStudioJobStats.password={sas002}0123456789
   ```

   d. Find the line that contains **`DIStudioJobStats.librefs=`**. Update this property with the directory location that you created for the job statistics data sets. For example:

   ```
   DIStudioJobStats.librefs=C\:\\SASDIStudioJobStats
   ```

   e. Find the line that contains **`DIStudioJobStats.sqlurl`**. Update this property with the host name and port for the SAS Workspace Server. For example:

   ```
   DIStudioJobStats.sqlurl=jdbc:sasiom://HOST01.example.com:8591
   ```

   f. Find the line that contains **`DIStudioJobStats.overwriteTables=`**. Make sure that this property is set to **`true`**.

Note: Later, after you run the log parsing program for the first time, this setting should be changed to **`false`**.

Save the file.

2. In **`SASDIStudioJobReports/batch`**, locate the file joblogsFlow.sas. Open this file in the SAS Program Editor or a text editor, and find the LIBNAME statement. Update the LIBNAME statement to specify the directory location for the job statistics data sets. This should be the same path that you specified in step 1.c on page 142. For example:

   ```
   libname jobstats 'C:\SASDIStudioJobStats';
   ```

   Save the file.

3. In the path **`SASDIStudioJobReports/batch`**, locate the file joblogsFlow.bat or joblogsFlow.sh. Open this file in a text editor and make the following changes:

   - Edit the path to the batch server command to specify the correct path for your environment. For example:

     ```
     C:\SAS\Config\Lev1\SASApp\BatchServer\sasbatchjob
     ```

   - Edit the value for the -log parameter to specify an appropriate path and filename for the parsing program's log file. For example:

     ```
     -log "C:\SASDIStudioJobReports\batch\logs\joblogsFlow.log"
     ```

   - Edit the value for the -sysin parameter to specify the location of the joblogsFlow.sas program. For example:

     ```
     -sysin "C:\SASDIStudioJobReports\batch\joblogsFlow.sas"
     ```

   Save the file.

Step 4: Add the Java Runtime Environment Executable to the PATH Environment Variable.

The SAS Data Integration Studio job statistics reporting package requires version 1.5 or higher of the Java Runtime Environment (JRE). This software should have been installed on your batch server machine when you installed your SAS software.

Update the PATH environment variable on your machine to include the path to the JRE executable.

Step 5: Import BI Reporting Objects from DIStudioJobStats.spk

The package file DIStudioJobStats.spk contains BI objects (including information maps, cubes, SAS Web Report Studio reports, table definitions, and a library) that are used to produce SAS Data Integration Studio job statistics reports.

Note: Import this package only if you will be using SAS Web Report Studio to generate the job statistics reports. Otherwise, follow the instructions in "Step 6: Import Stored Processes and Customize Them for Your Environment" on page 144.

To import DIStudioJobStats.spk, follow these steps:

1. Log on to SAS Management Console on a machine that has access to the directories on the DATA step batch server host machine.
2. On the **Folders** tab, right-click the root folder (**`SASFolders`**) or another folder where you want to install the package. Then select **Import SAS Package**.

3. On the first Import SAS Package wizard page, click **Browse** and navigate to `SASDIStudioJobReports/wrs` in the file system of the batch server machine. Select the file DIStudioJobStats.spk to import.

   If you are importing to the root folder (`SASFolders`), deselect the **Preserve source path information for objects in the package** check box. Click **Next**.

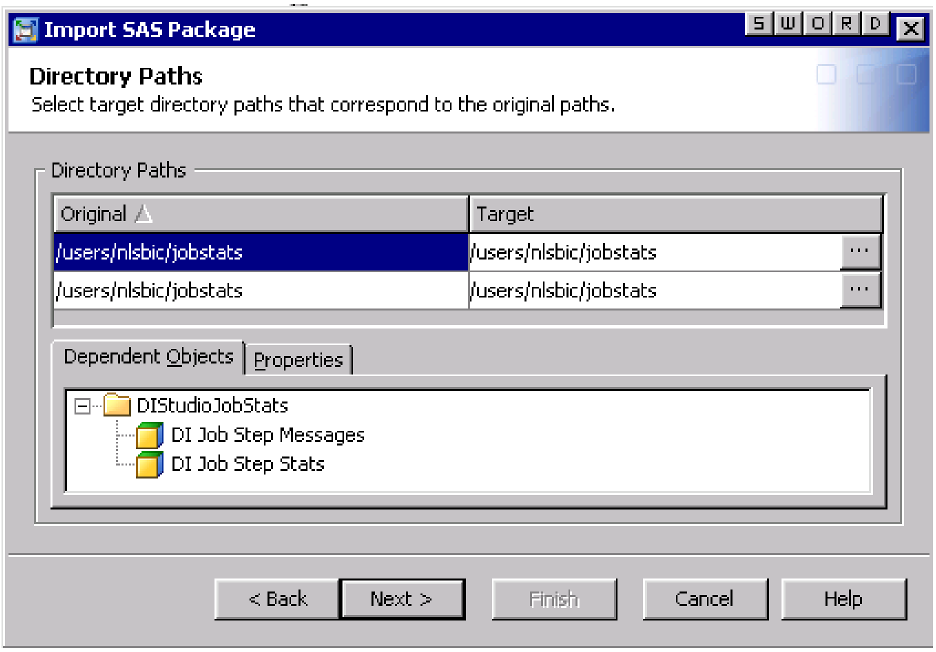
4. On the second wizard page, click **Next** to accept the defaults. If a warning is displayed about directly applied access control entries, click **Yes** to proceed.

5. The third wizard page displays information about required metadata connections for the imported objects. Click **Next**.

6. The next several wizard pages prompt for metadata connections such the SAS Application Server and OLAP schema. Specify values that are appropriate for your environment.

7. When you are prompted for report objects, click **Next** to accept the defaults.

8. The next wizard page prompts for a directory path, as shown here:



Import SAS Package

Directory Paths

Select target directory paths that correspond to the original paths.

Directory Paths

Original	Target
/users/nlsbic/jobstats	/users/nlsbic/jobstats
/users/nlsbic/jobstats	/users/nlsbic/jobstats

Dependent Objects | Properties

- DIStudioJobStats
  - DI Job Step Messages
  - DI Job Step Stats

   In both rows of the Target column, browse to the path for your job statistics data library. Specify the same path that you specified in DIStudioJobStats.Properties and joblogsFlow.sas (for example, `C:\SASDIStudioJobStats`).

9. The next wizard page provides a summary of the objects to be imported. Click **Next** to begin the import.

The import creates a metadata folder called `DIStudioJobReports`. The folder contains tables, reports, jobs, maps, cubes, and a library.

Step 6: Import Stored Processes and Customize Them for Your Environment

Import DIStudioJobStatsStoredProcess.spk

The package file DIStudioJobStatsStoredProcess.spk contains stored processes that are used to produce SAS Data Integration Studio job statistics reports.

Note: The stored processes are provided for sites that do not have SAS Web Report Studio. If you will be using SAS Web Report Studio to produce the reports, this step is optional.

To import the file DIStudioJobStatsStoredProcess.spk, follow these steps:

1. Make sure that you have Write permission on the directory that was created for the stored process source code (for example, `/SASDIStudioJobStoredProcesses`).
2. On the **Folders** tab of SAS Management Console, right-click the `DIStudioJobStats` folder, which was created in "Step 5: Import BI Reporting Objects from DIStudioJobStats.spk". Then select **Import SAS Package**.
3. On the first Import SAS Package wizard page, click **Browse** and navigate to `SASDIStudioJobReports/storedprocess` in the file system of the batch server machine. Select the file DIStudioJobStatsStoredProcess.spk to import. Click **Next**.
4. On the second wizard page, click **Next** to accept the defaults.
5. The third wizard page displays information about required metadata connections for the imported objects. Click **Next**.
6. The fourth wizard page prompts for a metadata connection to a SAS Application Server. Specify the value that is appropriate for your environment, and click **Next**.
7. The fifth wizard page prompts for a directory path for source code repositories. Specify the directory as follows:
   a. On the **Target source code repositories** tab, click **Add**.
   b. In the Add Source Code Repository dialog box, specify the source code directory that you created in "Step 1: Create Directories on the Batch Server Host Machine" on page 141 (for example, `C:\SASDIStudioJobStoredProcesses`). Click **OK**.

   The directory should appear in the Target column on the wizard page. Click **Next**.
8. The sixth wizard page provides a summary of the objects to be imported. Click **Next** to begin the import, and then click **Finish** on the final wizard page.

The `DIStudioJobStats` folder should now contain three stored processes. In addition, the source code directory in the file system should contain three SAS programs.

Customize the Stored Process Source Code for Your Environment

After you import DIStudioJobStatsStoredProcess.spk, you must customize the stored process source code for your environment, as follows:

1. In the file system of the batch server host machine, navigate to the directory that contains the source code for the stored processes (for example, `C:\SASDIStudioJobStoredProcesses`).
2. If you are using a UNIX machine, make sure that you have Edit permission (chmod 664) for the three stored process programs in the directory (disjobmesgsreport.sas, disjobparammesgs.sas, and disjobperformancereport.sas).
3. Open each of the three programs that are stored in the directory, and locate the following lines in the programs:
   - In disjobmesgsreport.sas:

     ```
     %isExist(libpath=d:\jobstats)
     ```

```
%preProcessData(libpath=d:\jobstats)
```

- In disjobparammesgs.sas:

```
%prepareData(libpath=d:\jobstats,jobname=,jobnumber=)
```

- In disjobperformancereport.sas:

```
%isExist(libpath=d:\jobstats)
```

4. In each of these lines, edit the value for LIBPATH to specify the path for your job statistics data library. Specify the same path that you specified in DIStudioJobStats.Properties and joblogsFlow.sas (for example, **`C:\SASDIStudioJobStats`**).

Step 7: Enable ARM Logging and Run-Time Statistics Collection for Deployed SAS Data Integration Studio Jobs

To enable the collection of data for job statistics reporting for deployed SAS Data Integration Studio jobs, follow these steps:

1. Set up ARM logging and run-time statistics collection by following the instructions in “About Collecting ARM Log Information for SAS Data Integration Studio Jobs” on page 130.
2. To enable the collection of statistics for deployed jobs, select **Tools** ⇨ **Options** in SAS Data Integration Studio. Then deselect the **Override and disable performance statistics when a job is deployed** check box, and click **OK**.

Step 8: Apply Required Permissions

To enable the creation and viewing of SAS Data Integration Studio job statistics reports, you must apply the following permissions to files, directories, metadata folders, and objects:

Directory, File, Metadata Folder, or Object	User	Required Permission
Directory that stores programs, scripts, and configuration information for the reporting package For example: **`/SASDIStudioJobReports`**	User who is specified in the DIStudioJobStats.user property of DIStudioJobStats.Properties, and who runs joblogsFlow.bat or joblogsFlow.sh For example: sassrv	Write
Library location for job statistics data sets For example:**`/SASDIStudioJobStats`**	User who is specified in the DIStudioJobStats.user property of DIStudioJobStats.Properties, and who runs joblogsFlow.bat or joblogsFlow.sh (for example, sassrv) User who builds cubes *	Write
DIStudioJobStats.sh and joblogsFlow.sh	User who is specified in the DIStudioJobStats.user property of DIStudioJobStats.Properties, and who runs joblogsFlow.bat or joblogsFlow.sh For example: sassrv	Execute (chmod +x)

Directory, File, Metadata Folder, or Object	User	Required Permission
OLAP schema object For example: SASApp - OLAP Schema	User who builds cubes*	Read
Metadata folder that contains the DI Job Step Messages cube and the DI Job Step Stats cube For example: `DIStudioJobStats`	User who builds cubes*	ReadMetadata
DI Job Step Messages cube, DI Job Step Stats cube, DI Job Step Messages cube job, and DI Job Step Stats cube job	User who builds cubes*	WriteMetadata
Metadata folder that contains information maps for the reporting package For example: `DIStudioJobStats`	User who accesses the SAS Web Report Studio reports*	Read and ReadMetadata

* Required only if you will be using SAS Web Report Studio to generate reports.

Configuration of the SAS Data Integration Studio job statistics reporting package is now complete. The following topics provide instructions for the ongoing production of job statistics reports.

Creating Job Statistics for the SAS Data Integration Studio Job Statistics Reporting Package

Summary of Steps for Creating Job Statistics

To create job statistics for use in SAS Data Integration Studio job statistics reports, follow these steps:

- Run deployed jobs on the SAS DATA step batch server.
- Parse the batch server logs to create job statistics data sets.
- Run the second parsing program to create job statistics summary data sets.
- Build the job statistics cubes. This step is necessary only if you will be using SAS Web Report Studio to generate the job statistics reports.

You might want to use a scheduling facility system to run the parsing programs and build the cubes on a regular basis (for example, nightly). Doing so will ensure that up-to-date reports are always available.

Details for each step are provided in the following topics.

Step 1: Run Deployed Jobs on the SAS DATA Step Batch Server

The job statistics reporting package collects statistics only for jobs that have been deployed on the SAS DATA step batch server where the reporting package has been installed. Be sure to deploy your SAS Data Integration Studio jobs on this server. See “Deploying Jobs” in the *SAS Data Integration Studio: User's Guide*.

Step 2: Parse the Batch Server Logs to Create Job Statistics Data Sets

After the batch server log directory has been populated by logs, you can execute the log parsing program. This program extracts information from the batch server logs to create job statistics data sets. The extracted information includes ARM information, error and warning messages, source and target table names, and SAS metadata user identities. Follow these steps to run the parsing program:

1. On the batch server host machine, go to the subdirectory that contains the batch scripts for SAS Data Integration Studio job statistics reporting (for example, **`SASDIStudioJobReports/batch`**). This directory was set up in "Step 1: Create Directories on the Batch Server Host Machine" on page 141.

2. Execute the following script:

   - On Windows: DIStudioJobStats.bat

   - On UNIX: DIStudioJobStats.sh

3. When the program is finished executing, do the following to make sure that it completed successfully:

   - Open the library location for the job statistics data sets (for example, **`C:\SASDIStudioJobStats`**) and make sure that the following data sets have been created: dijobstepmesg.sas7bdat, dijobstepstatspre.sas7bdat, dijobstepstatspre.sas7bndx, dijobsteptable.sas7bdat, and logfilehistory.sas7bdat.

   - Examine DIStudioJobStats.log, which is located in **`SASDIStudioJobReports/batch/logs`**.

4. After you run the parsing program for the first time, update the file DIStudioJobStats.Properties (located in **`SASDIStudioJobReports/batch`**) as follows:

   a. Find the line that contains **`DIStudioJobStats.overwriteTables=`**. Change the value of this property to **`false`**.

   b. Save the file.

Step 3: Run the Second Parsing Program to Create Summary Data Sets

The second parsing program creates summary data sets from the detailed data. To run this program, follow these steps on the batch server host machine:

1. Log on as the user who is specified in the DIStudioJobStats.user parameter of the DIStudioJobStats.Properties (for example, sassrv).

2. From the subdirectory that contains the batch scripts (for example, **`SASDIStudioJobReports/batch`**), execute the following script:

   - On Windows: joblogsFlow.bat

   - On UNIX: joblogsFlow.sh

3. When the program is finished executing, do the following to make sure that it completed successfully:

   - Make sure that the following data sets have been created in the library location for the job statistics data sets: dijobmesgs.sas7bdat, dijobstepstats.sas7bdat, dijobstepstats.sas7bndx, dijobsummary.sas7bat, djobsummarylist.sas7bdat, and dijobtables.sas7bdat.

- Examine joblogsFlow.log, which is located in **`SASDIStudioJobReports/batch/logs`**.

Step 4: Build the Job Statistics Cubes for SAS Web Report Studio Reports

The next step is to access the summary data sets and build the physical cubes that are used to generate job statistics reports.

Note: This step is necessary only if you will be using SAS Web Report Studio to generate the job statistics reports.

The user who builds the cubes must have the appropriate permissions as specified in "Step 8: Apply Required Permissions" on page 146.

To build the physical cubes, follow these steps:

1. Log on to SAS Data Integration Studio. On the **Inventory** tab, expand the **Cube** node.
2. Right-click the **DI Job Step Stats** cube, and select **Build Physical Files**.
3. Right-click the **DI Job Step Messages** cube, and select **Build Physical Files**
4. The SAS log from each job is displayed. Examine the logs to ensure that the jobs ran successfully.

   Note: When you build the physical cube for DI Job Step Messages, the following errors are displayed if your batch server log files do not contain any warning or error messages:

   ```
   ERROR: The input data set does not contain any observations.

   ERROR: The input data set cannot be opened.

   ERROR: Cube "DI Job Step Messages" cannot be created.
   ```

   If these messages are displayed, the job statistics reports that analyze error and warning messages will be unavailable. You do not need to take any actions to resolve this issue.

After the cubes are created, you can access the SAS Data Integration Studio job statistics reports.

Accessing the SAS Data Integration Studio Job Statistics Reports

After you have created job statistics as described in the preceding topic, you can access the SAS Data Integration Studio job statistics reports as follows:

- In SAS Web Report Studio, click **File** ⇨ **Open** and navigate to the **`DIStudioJobStats`** folder. Within the folder, the following reports are available:
  - DI Job Message Destination Reports
  - DI Job Message Reports

  - DI Job Performance Reports
  - DI Job Step Transform Reports
  - DI Job Summary Reports
  - DI Runtime Job Summary Reports

  The user who accesses the reports must have the appropriate permissions as specified in "Step 8: Apply Required Permissions" on page 146.

- In the SAS Stored Process Web Application, select **Search for Stored Processes and Reports**. On the search page, enter `DISJob` in the **Find** field, and click **Search**. The following reports are available: DISJobMesgsReport and DISJobPerformanceReport.

  Note: DISJobMesgParamReport is used internally by the other stored processes and is not intended to be accessed directly by users.

The reports and stored processes can also be accessed in the SAS Information Delivery Portal.

Removing Historical Data from the Job Statistics Library

Steps for Removing Historical Data from the Job Statistics Library

You might want to occasionally remove historical data from your SAS Data Integration Studio job statistics data sets. Otherwise, data will continue to be added to the data sets indefinitely. To remove historical data, follow these steps:

1. First, you might want to copy the current contents of the job statistics library (for example, `C:\SASDIStudioJobStats`) to an archive location. This step ensures that the historical data will be available if it is needed in the future.
2. Open one of the following scripts in a text editor:
   - On Windows: DIStudioJobStatsDataClean.bat
   - On UNIX: DIStudioJobStatsDataClean.sh

   The scripts are located in `SASDIStudioJobReports/batch` on the batch server host machine.
3. Edit the options in the script to specify the data that you want to remove from the detailed data sets. Refer to "Syntax for the CleanHistoricalDataApp Script" on page 151 and "Syntax Description for the CleanHistoricalDataApp Script" on page 151.
4. Run the updated script.
5. After the script has run successfully, execute joblogsFlow.bat or joblogsFlow.sh as described in "Step 3: Run the Second Parsing Program to Create Summary Data Sets" on page 148. This script creates new summary data sets from the detailed data.

   When you view the job statistics reports, the reports will reflect the removal of the historical data.

Syntax for the CleanHistoricalDataApp Script

CleanHistoricalDataApp <-data> <-log> <-duration *YEAR|MONTH|DAY number*> <-timescope FromDate(*MM/dd/yyyy*) ToDate(*MM/dd/yyyy*)>

Syntax Description for the CleanHistoricalDataApp Script

-data
specifies that data is to be removed from the tables that contain data from logs.

-log
specifies that data is to be removed from the tables that contain log filenames.

-duration *YEAR|MONTH|DAY number*
specifies that data is to be removed if it is older than the specified number of years, months, or days.

-timescope FromDate(*MM/dd/yyyy*) ToDate(*MM/dd/yyyy*)
specifies that data that was collected during the specified period of time is to be removed.

Part 4

Backing Up and Restoring Your SAS Content

Chapter 11

Best Practices for Backing Up and Restoring Your SAS Content

About Backups and Restores

To ensure the integrity of the content that is created and managed by the SAS Intelligence Platform, you should establish a formal, regularly scheduled backup process. It is important to back up all of the following content items at the same point in time so that related information will be synchronized if a restore becomes necessary:

- all of your metadata repositories, the repository manager, and the files in the metadata server configuration directory. The SAS Metadata Server includes a server-based facility (new with SAS 9.3) that creates correct backups of these files.

- the SAS Content Server, the SAS Framework Data Server, and the physical content that is stored in files and databases. Because the metadata server's backup facility does 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 some situations it might be appropriate to back up specific objects or folders in the metadata folders (**`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 "Promotion Tools Overview" on page 237.

Before you back up your SAS content, read the following information:

- "About the Metadata Server Backup Facility" on page 156
- "Using Operating System Commands to Back Up the Metadata Server" on page 157
- "About Using the Export SAS Package Wizard to Back Up Specific SAS Folders" on page 157
- "Best Practices for Backing Up Your SAS Content" on page 158
- "Best Practices for Restoring Your SAS Content" on page 165

CAUTION:

If you choose to use operating system commands to back up your metadata repositories and the repository manager, instead of using the metadata server's backup facility, 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.

About the Metadata Server Backup Facility

The SAS 9.3 Metadata Server includes a new server-based facility that performs metadata server backups automatically, on a scheduled basis, without the need for administrator intervention. This facility can also be used to perform ad hoc backups and to perform roll-forward recovery. The facility replaces the %OMABAKUP macro, which was available in SAS 9.1 and 9.2, and the Backup and Restore Wizard in SAS Management Console, which was available in SAS 9.2.

The backup facility backs up all registered metadata repositories, the repository manager, and the metadata server's configuration directory. By default, backups are scheduled to run at 1:00 a.m. daily. Administrators can change this schedule as needed. Backups can also be launched from SAS Management Console, from the operating system command line, from SAS, or through third-party scheduling software.

The metadata server backup facility provides the following advantages:

- The backup facility executes in a separate thread while the metadata server is running. Therefore, the server does not need to be paused during backups unless certain options are selected. When repositories are restored in the event of a metadata server failure, transactions recorded in the journal file can be used to perform roll-forward recovery to a specified point in time after the last backup was taken.
- The metadata server runs its own backup and recovery processes and does not rely on any other servers. Because the metadata server has permission to access all of the needed files, there is no need to set up a host account with special operating system rights and permissions.
- The backup facility automatically includes all of the metadata server's data sets, including the repository manager, all of your registered metadata repositories, and

the individual files that are stored in the metadata server configuration directory (***SAS-configuration-directory*/Lev1/SASMeta/MetadataServer**).

- SAS Management Console displays a detailed history of backups that were performed using the backup facility. Status icons indicate which backups are active and which backups have been deleted based on the retention policy. When you view the properties of a backup, the backup files are validated, and the status icon is updated to indicate whether the backup is usable.
- When repositories are restored in the event of a metadata server failure, transactions recorded in the journal file can be used to perform roll-forward recovery to a specified point in time after the last backup was taken.

See Also

- "Backing Up and Recovering the SAS Metadata Server" on page 167

Using Operating System Commands to Back Up the Metadata Server

Instead of using the metadata server backup facility, 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 164
- "Example of a PROC METAOPERATE Program That Resumes the Metadata Server" on page 164

You can also pause and resume the server by using SAS Management Console or a server management script. See "Operating Your Servers" on page 54. 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 addition to performing regular full backups, it might be appropriate in certain situations 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.

See Also

For details about using the Export SAS Package and Import SAS Package wizards and the batch import and export tools, see "Promotion Tools Overview" on page 237.

Best Practices for Backing Up Your SAS Content

Summary of Best Practices for Backing Up Your SAS Content

The following are the recommended best practices for backing up your SAS content:

- Always use the metadata server backup facility to back up the repository manager and the metadata repositories.
- Perform regularly scheduled full backups.
- Perform backups before and after major changes.
- Specify a reliable backup destination that is included in daily system backups.
- Occasionally use the Reorganize Repositories option to reclaim unused disk space.
- Back up the SAS Content Server.
- Back up the SAS Framework Data Server and its databases.
- Back up data that is stored in file systems.
- Synchronize metadata backups with SAS Content Server, SAS Framework Data Server, and physical file backups,

The following topics describe these best practices.

Always Use the Metadata Server Backup Facility 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, and the metadata server's configuration directory is to use the metadata server backup facility. By default, the backup facility runs scheduled backups at 1:00 a.m. daily. Administrators can change this schedule as needed. Your daily system backup process can then copy the files from the metadata server's backup destination.

The advantages of the metadata server backup facility are described in "About the Metadata Server Backup Facility" on page 156.

CAUTION:

> **If you choose to use operating system commands to back up your metadata repositories and the repository manager, instead of using the metadata server's backup facility, then you must be sure to pause the metadata server before performing the backup. Otherwise, the backup files will not be usable.**

Perform Regularly Scheduled Full Backups

By default, the metadata server backup facility takes daily backups of your metadata repositories, the repository manager, and the files in the metadata server's configuration directory. To safeguard the integrity of your metadata, it is important to maintain this backup schedule. Daily use of the metadata server backup facility will enable you to recover the server to a particular point in time in the event of a server failure, while correctly maintaining associations between metadata.

To maintain system integrity, you must also back up the SAS Content Server, the SAS Framework Data Server, and the physical files that are associated with your metadata. See the following topics:

- "Back Up the SAS Content Server" on page 161
- "Back Up the SAS Framework Data Server and Its Databases" on page 161
- "Back Up Data That Is Stored in File Systems" on page 162
- "Synchronize Metadata Backups with SAS Content Server, SAS Framework Data Server, and Physical File Backups" on page 163

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 update your SAS software to a new release
- before and after you configure a new SAS product or remove the configuration of an existing product
- before and after you configure a new tier in your SAS deployment
- after you add a new repository
- before you delete a repository

The best way to perform an ad hoc (unscheduled) backup of the metadata server is to use the metadata server backup facility. You can launch this facility from SAS Management Console, from the operating system command line, from a SAS session, or through a third-party scheduling system.

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 Framework Data 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 157.

In unscheduled backup situations where the metadata has changed but no physical files are involved, it is sufficient to back up just the metadata server. 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.

Specify a Reliable Backup Destination That Is Included in Daily System Backups

By default, the metadata server backup facility writes backup files and journal files to the **`Backups`** subdirectory of the metadata server configuration directory. Within the backup location, each set of backup files (along with the associated journal file) is stored in a directory that is named based on the date and time that the backup is started.

As a best practice, you should modify your backup configuration to specify a storage device other than the device that is used to store the metadata repositories and the server configuration files. Specifying a separate device will ensure that backup files and their associated journal files (including the most current journal file) are available in the event of a disk failure.

Make sure that the **`Backups`** directory (or the backup destination that you specify) is included in your regular system backups.

See Also

"Modifying the Metadata Server Backup Configuration (Backup Location and Retention Policy)" on page 172

Occasionally Use the Reorganize Repositories 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 metadata server backup facility, you can use the Reorganize Repositories option to reclaim unused disk space that is left from previously deleted metadata objects. The default backup schedule specifies a weekly reorganization. It is not necessary to reorganize the repositories more frequently than once a week, except in extraordinary situations such as the deletion of a large amount of metadata.

The repository reorganization process affects disk space only. It does not affect the memory usage of the metadata server.

See Also

"About the Reorganize Repositories Backup Option" on page 170

Back Up the SAS Content Server

If your deployment includes SAS Web applications or the SAS Web Infrastructure Platform, 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 it is sufficient to stop only the SAS Content Server application. If you are using a JBoss Application Server, then you must stop both the SAS Content Server application and the JBoss Application 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 Framework Data Server and Its Databases

If your deployment includes a SAS Framework Data Server, then it is important to back up the server and its databases. The server might contain the following databases:

- SYSCAT.TDB, which is a transactional database file that stores configuration information for the server.
- SHAREDSERVICES.FDB, which is the default storage location for middle-tier data such as alerts, comments, and workflows. (During installation, a different type of database might have been specified for this data.)
- LINEAGESERVICE.FDB, which is the default storage location for metadata connection information that is used by SAS Environment Manager. (During installation, a different type of database might have been specified for this data.)

To back up all of the SAS Framework Server files including scripts, logs, configuration files, and databases, follow these steps:

1. Stop the SAS Framework Data 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/FrameworkServer**

If necessary, you can back up the SYSCAT.TDB and SHAREDSERVICES.FDB databases while the server is running. To do so, use the following command:

```
dffedfire -gbak -b database-name -mode read_only -user sysdba -pass
     masterkey backup-filename
```

If you need to restore a database from a backup, stop the SAS Framework Data Server and then use following command to perform the restore:

```
dffedfire -gbak -r backup-filename -user sysdba -pass masterkey
     database-name
```

The **dffedfire** command and the databases are typically located in ***SAS-configuration-directory*/Lev1/FrameworkServer/dffedsvrcfg**.

SHAREDSERVICES.FDB might have been assigned a different location when SAS 9.3 was installed. For more information about this database, see the *SAS Intelligence Platform: Middle-Tier 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.

You must use operating system commands or third-party tools to back up the physical data, which might include the following:

- 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
  - RDBMS 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 Framework Data Server, and Physical File Backups

Suggested 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 Framework Data 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 Framework Data Server, and the physical files concurrently (that is, in the same backup window).
- Back up the SAS Content Server, the SAS Framework Data 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. If you are running the backup on a batch basis (for example, as part of a daily schedule), then you can do the following to implement this approach

  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 164.

     You can use this program to pause the metadata server while you back up the SAS Content Server, the SAS Framework Data 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 164. 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, the SAS Framework Data 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 to prevent clients from updating metadata while you are backing up the associated data:

  1. Use the metadata backup facility to back up the metadata server. Then immediately use SAS Management Console to pause the metadata server. As an alternative, you can use SAS Management Console to temporarily change the registered access mode of the repositories to ReadOnly.
  2. Back up the SAS Content Server, the SAS Framework Data Server, and the physical data.
  3. When you are finished backing up the SAS Content Server, the SAS Framework Data 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 metadata server backup facility, it is not necessary to pause the server while you back it up. However, you can use this program to pause the metadata server while you back up the SAS Content Server, SAS Framework Data 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 Framework Data 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 SAS Content

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 (including omaconfig.xml, adminUsers.txt, trustedUsers.txt, and logconfig.xml). If you use the metadata server backup facility to create your backups, then you can use the metadata server recovery facility in the event of a server failure. The recovery facility automatically restores the repository manager and all of the repositories that are included in the backup. You can also choose to restore the contents of the metadata server's configuration directory and to perform roll-forward recovery to a specified point in time after the last backup was taken.
- the SAS Content Server, the SAS Framework Data Server, and the physical files that are associated with the metadata. The metadata server recovery facility does 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 161 and "Back Up Data That Is Stored in File Systems" on page 162.

Restoring an Unresponsive Metadata Server

The metadata server recovery facility 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 "What to Do If the SAS Metadata Server Is Unresponsive" on page 183.

Chapter 12
Backing Up and Recovering the SAS Metadata Server

About the SAS Metadata Server Backup and Recovery Facility

Overview of the SAS Metadata Server Backup and Recovery Facility

The SAS 9.3 Metadata Server includes a new server-based facility that performs metadata server backups automatically on a scheduled basis. This facility can also be used to perform ad hoc backups and to perform roll-forward recovery. The facility replaces the %OMABAKUP macro, which was available in SAS 9.1 and 9.2, and the Backup and Restore Wizard in SAS Management Console, which was available in SAS 9.2.

The backup and recovery facility provides the following features:

- A schedule of daily backups is configured by the SAS Deployment Wizard. Therefore, daily backups are taken without the need for administrator intervention. In addition, unscheduled backups are launched automatically in certain situations, such as after the SAS Deployment Wizard configures the metadata server.
- Administrators can use SAS Management Console to change the schedule and the configuration options, including the backup directory location and the backup retention policy. Backups can also be run on an ad hoc (unscheduled) basis from SAS Management Console, from the operating system command line, from SAS, or through third-party scheduling software.
- When repositories are restored in the event of a metadata server failure, transactions recorded in the journal file can be used to perform roll-forward recovery to a specified point in time after the last backup was taken.
- SAS Management Console displays detailed backup and recovery history. The history includes all backups that are run using the server's backup facility, regardless of how the backup was invoked (from the server's own scheduling facility, from SAS Management Console, from the command line, from SAS, or from third-party scheduling software).
- The backup process is a "hot" backup that executes in a separate thread while the metadata server is running. Therefore, the server does not need to be paused during the backup, and all transactions, including updates, continue to be processed. If journaling has been disabled or if the Reorganize Repositories backup option has been selected, the server is paused for Read-Only use, so that queries (but not updates) can continue to be processed.
- The metadata server runs its own backup and recovery processes. These processes do not rely on any other servers, such as workspace servers, DATA step batch servers, or scheduling servers. Because the metadata server has permission to access all of the needed files, there is no need to set up a host account with special operating system rights and permissions.

Note: When you migrate to SAS 9.3, the SAS Deployment Wizard removes all SAS 9.1.3 and SAS 9.2 backup and restore jobs, backup schedules, and backup programs that were registered in SAS Management Console. The SAS 9.1.3 and SAS 9.2 metadata server backup utilities, including %OMABAKUP and the Backup and Restore Wizard, are not supported in SAS 9.3.

About the Automatically Configured Daily Backups of the Metadata Server

When you migrate to or install SAS 9.3, the SAS Deployment Wizard configures a schedule of automatic daily backups of the SAS Metadata Server. The automatic backups are configured as follows:

- Backups are performed daily (seven days a week) at 1:00 a.m. server local time.
- The backup that is performed on Monday uses the Reorganize Repositories option.
- Backups and their associated journal files are stored in the following location: ***SAS-configuration-directory*/Lev1/SASMeta/MetadataServer/Backups**. (If you select the Custom option when installing SAS 9.3, the SAS Deployment Wizard enables you to specify a different location.) Within the backup location, each set of backup files is stored in a directory that is named based on the date and time that the backup was started (for example, `2011-06-06T01_00_01-04_00`). Do not change this directory name, because the recovery facility requires the name to be in this format.

TIP As a best practice, you should modify your backup configuration to specify a storage device other than the device that is used to store the metadata repositories and the server configuration files. Specifying a separate device will ensure that backup files and their associated journal files (including the most current journal file) are available in the event of a disk failure.

TIP Be sure to include this backup location in your regular system backups.

- Backups are retained for a period of seven days. Each time a backup is successfully completed, all previously created backup files that are more than seven days old are deleted from the backup directory. A record of the deleted backups is retained in backup history. (When a backup is attempted but not competed successfully, no backup files are deleted.)

The backup facility automatically backs up the following files:

- all of your metadata repositories, regardless of their access state (Online, Read-Only, Administration, or Offline).
- the repository manager.
- all of the files in ***SAS-configuration-directory*/Lev1/SASMeta/MetadataServer**, including metadata server configuration files, INI files, SAS programs, scripts, and backup history files, as well as any additional files that you might have stored in this directory. These files are backed up to a **ConfigurationFiles** subdirectory within the backup location.

See Also

- "Modifying the Metadata Server Backup Schedule" on page 171
- "Modifying the Metadata Server Backup Configuration (Backup Location and Retention Policy)" on page 172

About Unscheduled Backups of the Metadata Server

In addition to running scheduled backups, the metadata server automatically backs itself up in certain unscheduled situations. Unscheduled backups use the same server-based backup facility and the same backup configuration options that are used for scheduled backups.

- A backup is run automatically after the SAS Deployment Wizard configures the SAS Metadata Server.
- A backup is run automatically after you complete a successful recovery of the metadata server. (Backups that are completed in this situation are not deleted when they are older than the retention period.)
- If you change the JOURNALTYPE option in the omaconfig.xml file to NONE or SINGLE (which is not recommended), and you later change the option back to ROLL_FORWARD, a metadata server backup is run automatically when you restart the server.

As a best practice, you should run an ad hoc backup before and after making other types of major changes to your system. For example, you should run a backup before and after the SAS Deployment Wizard configures the middle and client tiers of your deployment, before and after you change the host name of an existing server, before you delete a repository, and after you add a new repository.

See Also

- "Using SAS Management Console to Run an Ad Hoc Metadata Server Backup" on page 173
- "Using the MetadataServer Command or backupServer.sas to Run Ad Hoc Metadata Server Backups" on page 174

How Journal Options Affect Metadata Server Backup and Recovery

The metadata server journal is a high-speed disk file in which transactions are recorded until they can be applied to repositories. The JOURNALTYPE option of the metadata server configuration file (omaconfig.xml) specifies whether metadata journaling is enabled and, if so, which type of journal file is created:

- The SAS Deployment Wizard sets the JOURNALTYPE option to ROLL_FORWARD. This option creates a linear journal file that permanently stores all transactions that have occurred since the most recent backup. The journal file is written to the same location as the associated backup files. Each time a new backup is executed, journaling stops and a new journal file is started in the new backup location.

  CAUTION:

  As a best practice, it is strongly recommended that you keep the default setting of ROLL_FORWARD for the JOURNALTYPE option. ROLL_FORWARD must be specified if you want to be able to use roll-forward recovery in the event of a metadata server failure.

- If you change the JOURNALTYPE option to SINGLE (which is not recommended), the roll-forward option is unavailable in the event of a metadata server failure. This setting creates a single circular journal file of a fixed size at the location that is specified in JOURNALPATH. Journal entries are appended to the file until the file grows to the size that is specified in JOURNALSIZE. Subsequent entries overwrite previously written entries starting at the beginning of the file.
- If you change the JOURNALTYPE option to NONE (which is not recommended), then no journal file is created and the roll-forward option is unavailable in the event of a metadata server failure. In addition, the metadata server cannot be backed up while the server is in an ONLINE state. The server will pause itself to a READONLY state each time a backup is executed, and any update transactions that are issued during the backup will fail.

See Also

- "About Metadata Server Journaling" on page 190
- "JOURNALTYPE="NONE | SINGLE | ROLL_FORWARD"" on page 332

About the Reorganize Repositories Backup Option

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. When you schedule or run a metadata server backup, you can select the Reorganize Repositories option to reclaim unused disk space left from the previously deleted metadata objects.

The default backup schedule specifies a weekly reorganization at 1:00 a.m. each Monday. It is not necessary to reorganize the repositories more frequently than once a week, except in extraordinary situations such as the deletion of a large amount of metadata.

If the Reorganize Repositories option is selected, the backup process does the following:

- pauses the server, placing it in a READONLY state
- copies the metadata server files to the backup destination
- re-creates the repository data sets in place, eliminating unused space in the process
- resumes the server to an ONLINE state

CAUTION:

> **The Reorganize Repositories option should be used only during times of little or no user activity. The metadata server is paused during the reorganization process, and any update transactions that are issued during this process will fail.**

SAS Metadata Server Backup Tasks

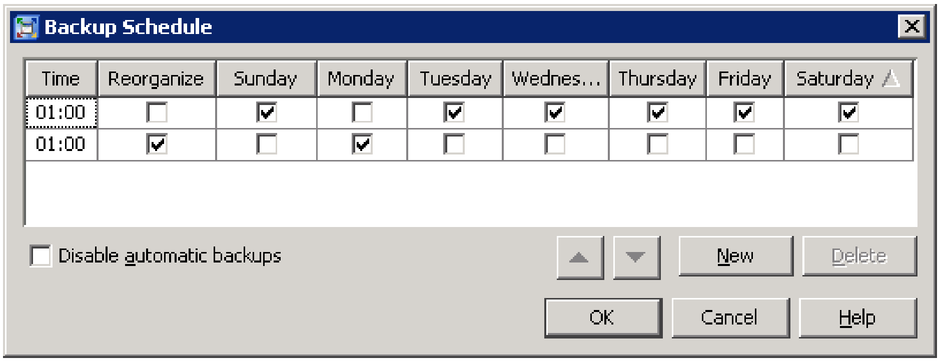
Credential Requirements for Administering Metadata Server Backups

To administer metadata server backups in SAS Management Console, you must be an unrestricted user, a user who is in the SAS Administrators group, or a user who is a member of the **Metadata Server: Operation** role. The tasks that require these credentials include modifying the backup schedule or the backup configuration, running ad hoc backups, and executing a metadata server recovery.

Modifying the Metadata Server Backup Schedule

If the default backup schedule for the metadata server is not appropriate for your site, you can change it using SAS Management Console. On the Plug-ins tab, navigate to **Environment Management** ⇨ **Metadata Manager** ⇨ **Metadata Utilities** ⇨ **Server Backup**. Then right-click **Server Backup** and select **Backup Schedule**.

The following dialog box appears:



Time	Reorganize	Sunday	Monday	Tuesday	Wednes...	Thursday	Friday	Saturday
01:00	☐	☑	☐	☑	☑	☑	☑	☑
01:00	☑	☐	☑	☐	☐	☐	☐	☐

☐ Disable automatic backups

The dialog box displays a row for each time of day that backups are scheduled to run, with check marks indicating the days of the week. The **Time** column displays the time in a 24-hour time format; for example, `1:00` means 1:00 a.m., and `23:30` means 11:30 p.m. The backup runs when the local time of the metadata server host machine is equal to the specified time. Backups that include the Reorganize Repositories option are listed on separate lines with a check mark in the **Reorganize** column.

You can modify the schedule as follows:

- To change the time for an existing row in the schedule, select the **Time** field and enter the new time using a 24-hour format in server local time.
- To specify different days of the week in an existing row, select or clear the appropriate check boxes in the columns **Sunday** through **Saturday**.
- To specify whether the repositories are to be reorganized during the backups, select or clear the check box in the **Reorganize** column. (In most cases, it is not necessary to reorganize the repositories more frequently than once a week.)
- To add another row to the schedule, click **New**. Enter the time using a 24-hour time format in server local time. Then select the days of the week that the backup is to be run, and select the **Reorganize** option if needed. You must select at least one day on each line.
- To delete an entire row of the schedule, click any cell in the row and then click **Delete**.
- To prevent the scheduled backups from running, select the **Disable automatic backups** check box. The scheduled backups will not run until you clear this check box.

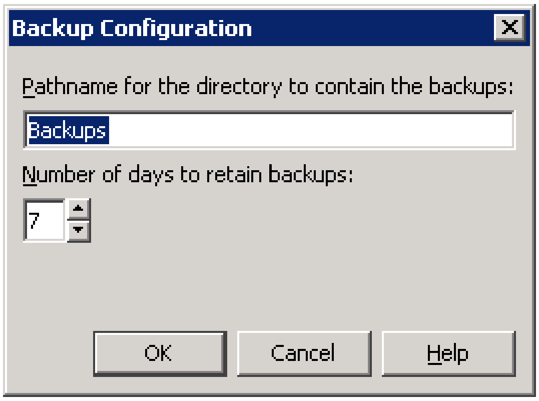
When you are finished, click **OK** to save the changes.

TIP If you schedule multiple backups per day, be sure to leave enough time for each backup job to complete before the next scheduled backup starts.

Modifying the Metadata Server Backup Configuration (Backup Location and Retention Policy)

SAS Management Console enables you to modify the backup location as well as the backup retention policy. On the Plug-ins tab, navigate to **Environment Management** ⇨ **Metadata Manager** ⇨ **Metadata Utilities** ⇨ **Server Backup**. Then right-click **Server Backup** and select **Backup Configuration**.

The following dialog box appears:



Backup Configuration

Pathname for the directory to contain the backups: Backups

Number of days to retain backups: 7

You can modify the configuration as follows:

- To change the backup location, specify either a complete path or a path relative to the MetadataServer directory. If the lowest-level directory in the path does not exist, it will be created. Directories that are higher in the path must already exist.

  TIP As a best practice, you should specify a storage device other than the device that is used to store the metadata repositories and the server configuration files. Specifying a separate device will ensure that the backup files and their associated journal files (including the most current journal file) are available in the event of a disk failure.

  Follow these guidelines:

  - You can specify any network location that is accessible from the metadata server. If you want to write the backup files to a path that is remote to the metadata server, then specify a UNC path (for example,`\\D2345\C$\SASBackup`).
  - If your metadata server is installed on a Windows host, do not specify a mapped drive.
  - Make sure that the process owner for the metadata server has credentials to write to the directory. To protect the integrity of the backups, you might want to deny access for other users.
  - Be sure to include the directory in your regular system backups.

- To change the number of days that backup files are retained in the backup location, select the appropriate number in the **Number of days to retain backups** selection box. Each time a successful backup is completed, previous backups that are older than the specified number of days are automatically deleted. The backup history automatically displays the offline status icon (⚠) for the deleted backups.

  If you do not want backup files to be deleted automatically based on a retention policy, then select `0` in the selection box. If you make this selection, you will need to manually delete files from the backup location on a regular basis to ensure disk space availability.

  Note: The offline status icon (⚠) is not displayed automatically for backups that you delete manually. To update the status icon for a manually deleted backup, you must open the backup's Properties dialog box.

When you are finished, click **OK** to save the changes.

Using SAS Management Console to Run an Ad Hoc Metadata Server Backup

After certain types of events, the metadata server automatically backs itself up. See "About Unscheduled Backups of the Metadata Server" on page 169. As a best practice, you should run an ad hoc backup before and after making other types of major changes to your system. For example, you should run a backup before and after the SAS Deployment Wizard configures the middle and client tiers of your deployment, before and after you change the host name of an existing server, before you delete a repository, and after you add a new repository.

Ad hoc backups use the same server-based backup facility and the same backup options that are used for scheduled backups. To run an ad hoc metadata server backup from SAS Management Console:

1. On the Plug-ins tab, navigate to **Environment Management** ⇨ **Metadata Manager** ⇨ **Metadata Utilities** ⇨ **Server Backup**.

2. Right-click **Server Backup** and select **Run Backup Now**.
3. In the dialog box that appears, you can enter a comment that will appear in the backup history; for example, you might want to state the reason for the ad hoc backup. (The comment is not displayed to clients of the metadata server.)

   You can also select the check box to **Reorganize repositories**.

   TIP The Reorganize Repositories option should be used only during times of little or no user activity. The metadata server is paused during the reorganization process, and any update transactions that are issued during this process will fail.

The backup appears in backup history immediately after the backup process is complete.

Using the MetadataServer Command or backupServer.sas to Run Ad Hoc Metadata Server Backups

You can also run an ad hoc backup by using one of the following methods. Backups that are run using these methods use the same server-based backup facility and the same backup options that are used for scheduled backups.

- From the directory ***`SAS-configuration-directory`*****`/Lev1/SASMeta/MetadataServer`**, issue one of the following commands:

  - On Windows: **`MetadataServer backup`**
  - On UNIX or z/OS: **`MetadataServer.sh backup`**

  These commands execute the program backupServer.sas.

- Directly run the program backupServer.sas. The program is located in ***`SAS-configuration-directory`*****`/Lev1/SASMeta/MetadataServer`**.

The backup appears in backup history in SAS Management Console's after you select **View** ⇨ **Refresh**. The following comment appears with the backup:

```
Started by user-ID with the MetadataServer command
```

Note: You cannot reorganize repositories when running a backup with the MetadataServer command or backupServer.sas.

CAUTION:
The backupServer.sas. program uses the metaparms.sas file, which contains the user ID and the encoded password for the SAS Administrator user. Be sure to protect this file from unauthorized access.

Using Third-Party Scheduling Software to Run Metadata Server Backups

Instead of using the backup scheduling facility that is provided with the metadata server, you can choose to use a third-party scheduling system, as follows:

1. Use SAS Management Console to disable the metadata server's scheduled backups, as follows:

   a. On the Plug-ins tab, navigate to **Environment Management** ⇨ **Metadata Manager** ⇨ **Metadata Utilities** ⇨ **Server Backup**. Then right-click **Server Backup** and select **Backup Schedule**.

   b. In the Backup Schedule dialog box, select the **Disable automatic backups** check box, and click **OK**.

2. Set up your scheduling system to run the **`MetadataServer backup`** command or to directly execute backupServer.sas on a daily basis.

Troubleshooting Metadata Server Backup Jobs

If a backup does not finish successfully, an e-mail with the subject line `Metadata Server Backup Failed` is sent to the designated recipients for metadata server alerts. The e-mail specifies the server's host name, the path to the server's configuration directory, and the backup name. For details about the failure, refer to the metadata server log and search for the backup name. The log is typically located in the Logs subdirectory of the server's configuration directory.

When a backup fails, an "x" icon (☒) also appears in the Status column of the backup and recovery history panel in SAS Management Console. To determine why the backup failed or why the backup files have become invalid, right-click the backup and select **Properties**. In the Properties dialog box, the **Error Summary** field provides high-level information to help resolve the problem. If more details are available, then the **Full Report** button is activated. Click this button to obtain a text report that you can browse, copy, and paste.

The ConfigurationFiles subdirectory of each backup contains files called MetadataServerBackupHistory.xml, MetadataServerBackupConfiguration.xml and MetadataServerRecoveryManifest.xml. In addition, the backup directory contains the MetadtaServerBackupManifest.xml file. These files might be useful to SAS Technical Support if you require troubleshooting assistance.

See Also

"Managing Alert E-mail Options for the SAS Metadata Server" on page 191

Recovering the SAS Metadata Server

About Metadata Server Recovery

You can use SAS Management Console to recover the SAS Metadata Server. The following options are available:

- You can recover just the metadata repositories and the repository manager, or you can choose to recover the configuration files as well.
- You can recover from a backup that is listed in the backup history panel, or you can recover from backup files that are stored in an alternate network-accessible location.
- You can use the roll-forward recovery to apply metadata updates that are stored in the journal file.

The recovery facility provides safeguards to ensure the integrity of the backup files from which you are recovering. The recovery operation checks to make sure that the backup directory contains all of the correct files and that the files have the correct name and file sizes. In addition, each backup file contains a universal unique identifier that is used to make sure that you are recovering files for the correct metadata server. If any problems exist, the recovery is not started and a warning message is displayed.

During recovery operations, the metadata server is automatically paused to a RECOVERY state. This state is similar to an OFFLINE state but more restrictive.

External systems such as Windows Services Manager report the server as paused. After the recovery, the metadata server automatically takes a new backup. If the recovery was successful, the server is automatically returned to the state that it was in before the recovery process.

Note: In SAS 9.3, recoveries must be run from SAS Management Console. They cannot be run from a script or a command.

See Also

"What to Do If the SAS Metadata Server Is Unresponsive" on page 183

About Roll-Forward Recovery

When recovering the metadata server, you can choose to use roll-forward recovery to update the recovered metadata repository files with transactions from the journal file. Roll-forward recovery uses the journal file that is located in the same directory as the backup files. The updates that are applied include additions, modifications, and deletions of metadata, as well as additions and deletions of metadata repositories. You can choose to apply all transactions that are in the journal file, or you can apply all transactions through a specified date and time.

Roll-forward recovery can be used only if the JOURNALTYPE option of the metadata server configuration file (omaconfig.xml) was set to ROLL_FORWARD at the time that the metadata server updates were made. (The ROLL_FORWARD option is enabled automatically by the SAS Deployment Wizard.)

CAUTION:
If you recover the metadata server without selecting the roll-forward option, then any update or delete transactions that occurred after the backup was taken will be lost. In addition, any repositories that were after the backup was taken will be deleted.

Executing a Metadata Server Recovery

To recover the SAS metadata server from a backup, either with or without roll-forward recovery, follow these steps:

1. Log on to SAS Management Console as an unrestricted user, a user who is in the SAS Administrators group, or a user who is a member of the **Metadata Server: Operation** role. On the Plug-ins tab, navigate to **Environment Management** ⇨ **Metadata Manager** ⇨ **Metadata Utilities** ⇨ **Server Backup**.

2. As a best practice, it is recommended that you pause the server to the ADMINISTRATION state.

   If you start the recovery while the server in the ADMINISTRATION state, the server changes the state to RECOVERY, performs the recovery, and then returns the server to the ADMINISTRATION state. You can then verify system readiness before clients reconnect. When readiness has been verified, you can manually resume the server.

   If you start the recovery while the server is ONLINE, the recovery process pauses the server to the RECOVERY state and then returns it to the ONLINE state after a successful recovery.

3. Use one of the following methods to start the recovery process:

- If you want to recover using backup files that are in the configured backup location, follow these steps:

  1. On the backup history list in the right panel, find the backup from which you want to recover. To determine its status, either double-click the backup entry or right-click and select **Properties**. If the files are usable for a recovery, the **Verification** field of the Properties dialog box displays this message: `All files in this backup are currently correct.`

     You can also review the detailed information about the backup on the **General** and **Repositories** tabs.

  2. When you are sure that you have identified the correct backup, right-click the backup entry in the backup history and select **Recover from This Backup**.

- If you want to recover from a different location (for example, the location that contains your system backups), right-click the **Server Backup** node (located on the **Plug-ins** tab under **Metadata Utilities**), and select **Recover from Alternate Location**. In the Recover from Alternate Location dialog box, enter the complete pathname for the directory that contains the backup. You can specify any network location that is accessible from the metadata server. If the backups are remote to the metadata server, then specify a UNC path, as in the following example:

  ```
  \\D2345\C$\SASMetadataServer\2011-01-14T01_00_01-05_00
  ```

  Note: If your metadata server is installed on a Windows host, do not specify a mapped drive.

4. After you select **Recover from This Backup** (or after you enter a pathname in the Recover from Alternate Location dialog box and click **OK**), the server accesses the backup directory and attempts to validate the backup files. It checks to make sure that all of the backup files are present and that they have the correct filenames, file sizes, and universal unique identifier.

   Note: If the validation is not successful, one of the following messages is displayed:

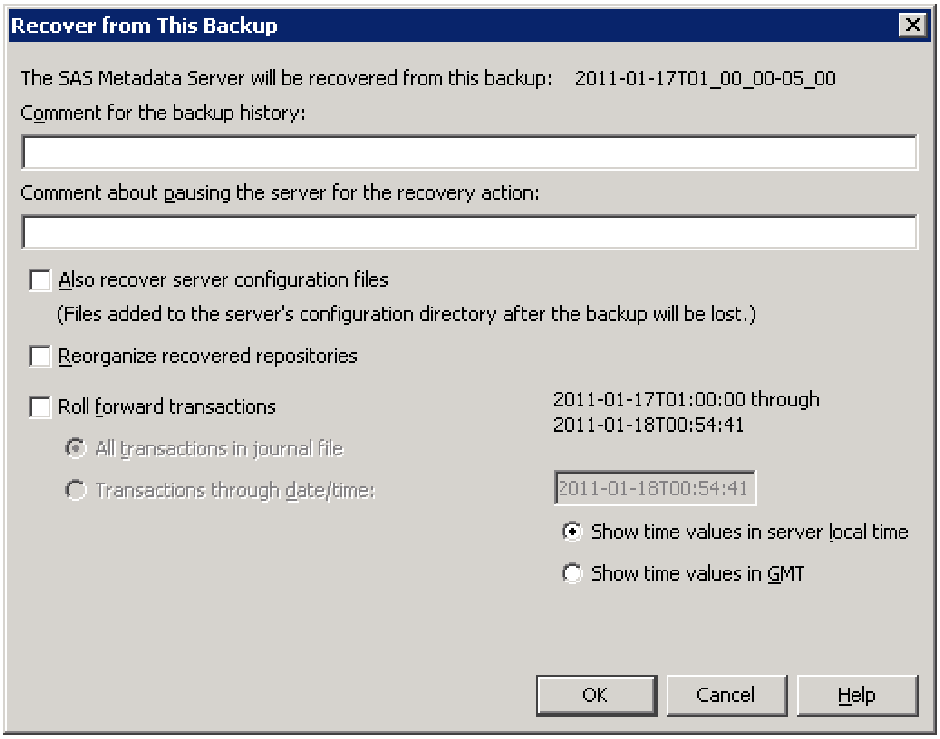
   - `A Recover cannot be run because this Backup is offline and not available.` If this message appears, you will need to recover from a different set of backup files.

   - `This Backup does not match what was backed up. Are you sure you want to recover from this Backup?` If this message appears, select **Details** to determine which files in the backup directory are incorrect.

     If the configuration file backups are incorrect but the repository backups are correct, you can click **OK** to resume the recovery. However, make sure that you have not selected the option to **Also recover server configuration files**.

     If the repository backups are incorrect, click **Cancel** and select a different backup from which to recover.

   If the validation is successful (or if you resumed the recovery after receiving an error message), the Recover from This Backup dialog box appears:



The SAS Metadata Server will be recovered from this backup: 2011-01-17T01_00_00-05_00

Comment for the backup history:

Comment about pausing the server for the recovery action:

☐ Also recover server configuration files
(Files added to the server's configuration directory after the backup will be lost.)

☐ Reorganize recovered repositories

☐ Roll forward transactions 2011-01-17T01:00:00 through 2011-01-18T00:54:41

● All transactions in journal file

○ Transactions through date/time: 2011-01-18T00:54:41

● Show time values in server local time

○ Show time values in GMT

5. In the first text field of the Recover from This Backup dialog box, enter a comment that will appear in the backup history panel with this recovery. For example, you might want to state the reason that the server is being recovered.

6. In the second text field, enter a comment that will be displayed to users while the server is paused during the recovery.

7. If appropriate, select the check boxes for the following options:

   Also recover server configuration files
   This option does the following:

   - deletes all files from the server's configuration directory (for example, ***SAS-configuration-directory*/Lev1/SASMeta/MetadataServer**)
   - from the CongurationFiles subdirectory in the backup location, copies all files (with the exception of MetadataServerBackupConfiguration.xml, MetadataServerBackupHistory.xml, and MetadataServerRecoveryManifest.xml) to the server's configuration directory

   CAUTION:
   If you select "Also recover server configuration files," you will lose any additions or modifications to files in the MetadataServer directory that have been made since the backup was taken. As a best practice, you should select this option only if your configuration files have become corrupted.

   Reorganize recovered repositories
   This option reorganizes the metadata repositories when they are backed up following the recovery. The reorganization reclaims any unused disk space that is left from previously deleted metadata objects. See "About the Reorganize Repositories Backup Option" on page 170.

   Roll forward transactions
   In addition to restoring the repository files from the backup, this option recovers transactions that were recorded in the metadata server journal file after the

backup was created. If you select this option, the following additional options become available:

All transactions in journal file
This option recovers all transactions that are recorded in the journal file. The time period that is covered by the journal file is displayed on the right side of the dialog box in server local time. (If you want to display the time period in Greenwich Mean Time (GMT), select **Show time values in GMT**.)

Transactions through date/time
This option recovers all transactions through a date and time that you specify. This option is useful if you want to recover the server to the state that it was in just before a specific problematic transaction. You can use the metadata server log to determine the exact time that the transaction occurred. See "Use the Log Tab in the Server Manager" on page 111.

Enter the roll-forward ending date and time in the format *yyyy-mm-dd*`T`*hh:mm:ss*. You can either edit the default values that are displayed in the field, or you can cut and paste the date and time from the appropriate event in the metadata server log. Most log events use server local time.

The dialog box enables you to specify the time in either server local time or GMT, depending on which of the following radio buttons is selected:

- **Show time values in server local time** displays the journal file's default starting and ending time in server local time. You can override the ending date and time. Be sure to enter the time in server local time if this radio button is selected.

- **Show time values in GMT** displays the journal file's default starting and ending times in Greenwich Mean Time. You can override the ending date and time. Be sure to enter the time in GMT if this radio button is selected.

CAUTION:
The preceding radio buttons do not convert your entry to server local time or to GMT. If you select or deselect one of these radio buttons after entering a date and time, your entry remains unchanged.

8. Before starting the recovery operation, the server pauses itself to a RECOVERY state. This state is similar to an OFFLINE state but more restrictive.

   Note: For example, the RECOVERY state prevents clients from obtaining a list of registered repositories. When the server is in the RECOVERY state, external systems such as Windows Services Manager report the server as paused.

9. When the recovery is completed, a backup of the newly recovered server is taken automatically, regardless of whether the recovery completed successfully.

   When the post-recovery backup is completed, all previous backups are retained (that is, no backups are deleted based on the retention policy).

10. If the recovery completed without errors, the server is returned to the state that it was in prior to the recovery process.

    If a recovery fails, the server is placed in the ADMINISTRATION state. The administrator can then access SAS Management Console and try to resolve the errors or recover the server from a different backup.

Troubleshooting Metadata Server Recovery Jobs

If a recovery does not finish successfully, an e-mail with the subject line `Metadata Server Recovery Failed` is sent to the designated recipients for metadata server alerts. The e-mail specifies the server's host name, the path to the server's configuration directory, and the recovery name. For details about the failure, refer to the metadata server log and search for the recovery name. The log is typically located in the **`Logs`** subdirectory of the server's configuration directory.

When a recovery fails, an "x" icon (⊗) also appears in the Status column of the backup and recovery history panel in SAS Management Console. To determine why the recovery failed, right-click the recovery and select **Properties**. On the Properties dialog box, the **Error Summary** field provides high-level information to help resolve the problem. If more details are available, then the **Full Report** button is activated. Click this button to obtain a text report that you can browse, copy, and paste.

The ConfigurationFiles subdirectory of each backup contains files called MetadataServerBackupHistory.xml, MetadataServerBackupConfiguration.xml and MetadataServerRecoveryManifest.xml. These files might be useful to SAS Technical Support if you require troubleshooting assistance.

Note: In a Windows environment, the recovery operation cannot update a READONLY file. If you have manually set any of the metadata server files to READONLY, recovery errors will be reported. In this situation, the recovery might still be valid if no changes have been made to the READONLY file since the backup was taken.

See Also

"Managing Alert E-mail Options for the SAS Metadata Server" on page 191

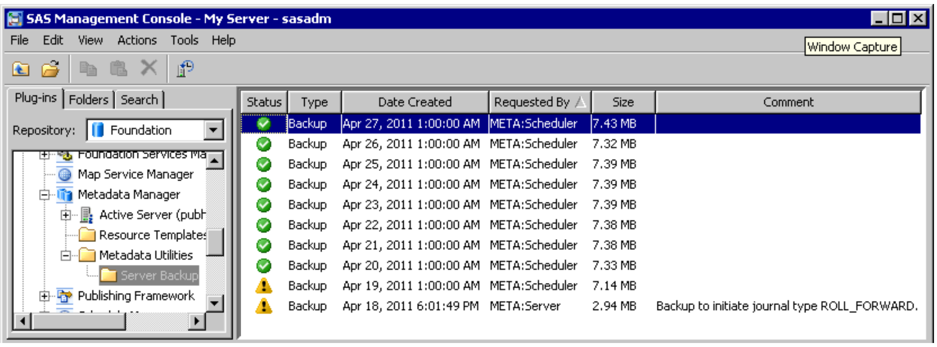
Viewing Backup and Recovery History

SAS Management Console provides a list of all backups and recoveries that have been run using the server's backup and recovery facility. All backups performed by this facility are listed regardless of how the process was invoked (from the server's own scheduling facility, from SAS Management Console, from the command line, from SAS, or from third-party scheduling software).

To view the backup and recovery history list:

1. Log on to SAS Management Console. On the **Plug-ins** tab, navigate to **Environment Management** ⇨ **Metadata Manager** ⇨ **Metadata Utilities** ⇨ **Server Backup**.

2. Select the **Server Backup** node. The backup and recovery history appears in the right pane, as shown in the following example:



Status	Type	Date Created	Requested By	Size	Comment
✓	Backup	Apr 27, 2011 1:00:00 AM	META:Scheduler	7.43 MB	
✓	Backup	Apr 26, 2011 1:00:00 AM	META:Scheduler	7.32 MB	
✓	Backup	Apr 25, 2011 1:00:00 AM	META:Scheduler	7.39 MB	
✓	Backup	Apr 24, 2011 1:00:00 AM	META:Scheduler	7.39 MB	
✓	Backup	Apr 23, 2011 1:00:00 AM	META:Scheduler	7.39 MB	
✓	Backup	Apr 22, 2011 1:00:00 AM	META:Scheduler	7.38 MB	
✓	Backup	Apr 21, 2011 1:00:00 AM	META:Scheduler	7.38 MB	
✓	Backup	Apr 20, 2011 1:00:00 AM	META:Scheduler	7.33 MB	
⚠	Backup	Apr 19, 2011 1:00:00 AM	META:Scheduler	7.14 MB	
⚠	Backup	Apr 18, 2011 6:01:49 PM	META:Server	2.94 MB	Backup to initiate journal type ROLL_FORWARD.

CAUTION:

> **The icons that are initially displayed do not necessarily report the most recent status of a backup. To obtain the most recent status, you must right-click the backup and select Properties.**

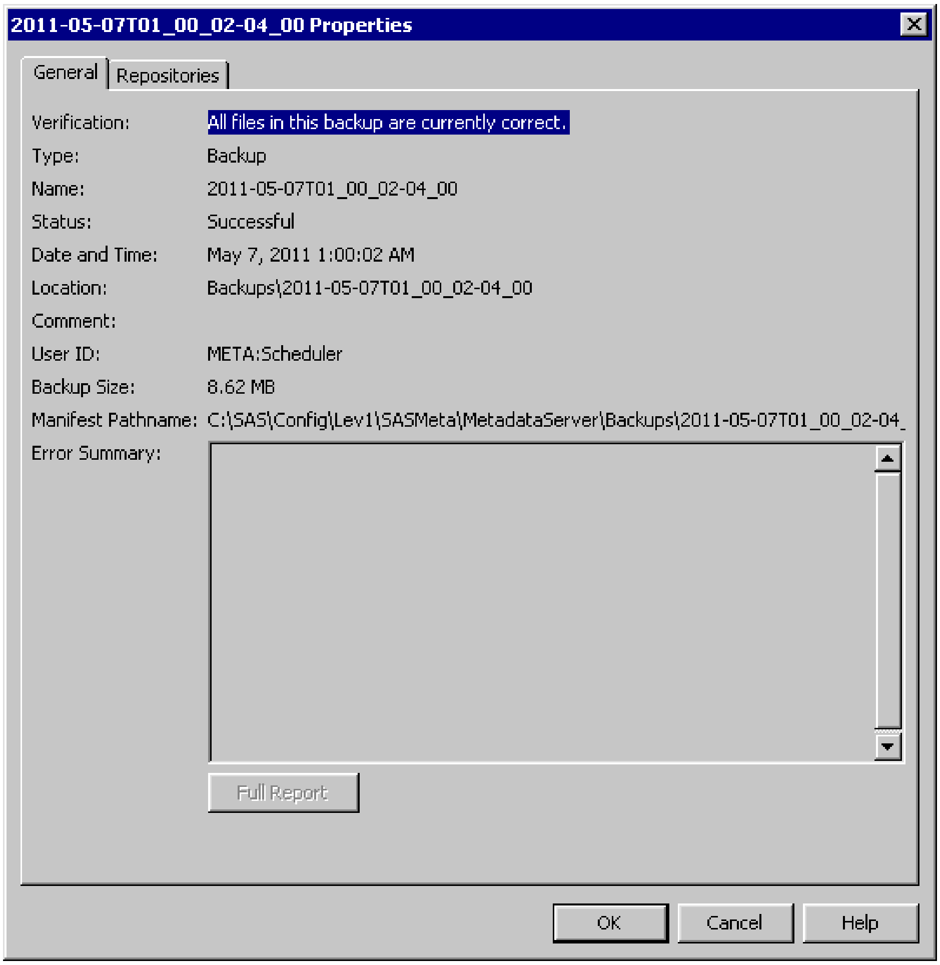
Note: If you migrated to SAS 9.3 from a previous SAS version, the backup history does not display any backup history prior to the migration. The SAS Deployment Wizard removes all SAS 9.1.3 and SAS 9.2 backup and restore jobs, backup schedules, and backup programs that are registered in SAS Management Console.

3. To obtain more information about a backup or a recovery, and to update the status icon for a backup, right-click the history entry and select **Properties**. The Properties dialog box appears, displaying a text description of the status and other information about the backup. Explanatory error messages are displayed if the backup was unsuccessful or if it is found to be invalid.

Icons in the history list indicate the most recently reported status for each backup or recovery. The status icons have the following meanings:

- The checkmark icon (✓) means that the backup or recovery was successful. For backups, this icon also means that the backup was determined to be valid the last time the files were checked. A backup is considered valid if all of the files are present in the backup location, all of the files have the correct GUID, and all filenames and file sizes are correct.
- The exclamation point icon ⚠ means that the backup was successful, but the backup files are currently offline (that is, they are not present in the backup location). The files might have been deleted automatically based on the retention policy, or they might have been moved or deleted manually. This icon appears automatically when backups are deleted based on the retention policy.
- The “x” icon ✖ indicates one of the following situations:
  - the backup or recovery was not successful
  - the backup was successful, but when the files were last checked, they were found to be invalid (that is, one or more files is missing, or one or more files has an incorrect GUID, incorrect name, or incorrect size)

The properties dialog box provides details about the backup or recovery, as shown in the following example:



2011-05-07T01_00_02-04_00 Properties

Verification: All files in this backup are currently correct.
Type: Backup
Name: 2011-05-07T01_00_02-04_00
Status: Successful
Date and Time: May 7, 2011 1:00:02 AM
Location: Backups\2011-05-07T01_00_02-04_00
Comment:
User ID: META:Scheduler
Backup Size: 8.62 MB
Manifest Pathname: C:\SAS\Config\Lev1\SASMeta\MetadataServer\Backups\2011-05-07T01_00_02-04_
Error Summary:

The **Repositories** tab lists the repositories that were included in the backup or recovery. (The **Repositories** tab does not appear if the backup is offline. It also does not appear for recoveries other than the most recent recovery.)

The Error Summary field displays any errors that occurred during the backup or recovery or during the verification of a backup. If more details are available, then the **Full Report** button is activated. Click this button to obtain a text report that you can browse, copy, and paste.

Chapter 13
What to Do If the SAS Metadata Server Is Unresponsive

What to Do If the SAS Metadata Server Is Unresponsive

If the metadata server is not responding to client requests, follow these steps:

1. Use one of the following methods to stop the server:

   - Use the Metadata Manager in SAS Management Console, or use the metadata server script. In this situation, these are the preferred methods for stopping the server.
   - (Windows only) If you cannot stop the server using the Metadata Manager or the script, then stop the Windows service. If you cannot stop the service, then use the Windows Task Manager to stop the server process.
   - (UNIX only) If you cannot stop the server using the Metadata Manager or the script, use one of the following commands to stop the server process:

     ```
     kill -2 server-process-id
     ```

     ```
     kill -15 server-process-id
     ```

     If the process fails to stop, use the following command:

     ```
     kill -9 server-process-id
     ```

     For more information, see “Exiting or Interrupting Your SAS Session in UNIX Environments” in the *SAS Companion for UNIX Environments*.

   Verify that the process has successfully stopped.

2. Review the entries in the metadata server log to determine the cause of the problem. For example, the log might indicate an issue in your operating environment.

   By default, the log is located in ***SAS-configuration-directory*/Lev1/SASMeta/MetadataServer/Logs**. Your site might have selected the option to write all server logs to a single directory, such as ***SAS-configuration-directory*/Lev1/Logs**.

   If the log identifies an issue in your environment, resolve the issue, if possible.

3. Start the server as described in “Operating Your Servers” on page 54.

4. If you still cannot start the server, review the log again and resolve any issues, if possible.

5. If you are unable to resolve the issue and start the server, you will need to use manual steps to start it. You will then be able to use the metadata server recovery facility to restore the server from a backup. For instructions, see SAS Note 43649.

   CAUTION:

   If the server cannot be started, do not attempt to re-initialize the server or perform a recovery without reading SAS Note 43649.

Part 5

Administering the SAS Metadata Server

Chapter 14
Metadata Server Configuration Tasks

Overview of SAS Metadata Server Configuration

The following configuration settings can be used to manage the SAS 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 188.
- the metadata server journaling options. See “Configuring Metadata Server Journaling” on page 190.
- the alert e-mail options. See “Managing Alert E-mail Options for the SAS Metadata Server” on page 191.

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: **`MAX(5,((`*`number-of-processors`*` * 2) + 1))`**

This means that the value is set to either 5 or to ((*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 188.

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

(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 (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 (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 329. 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 339. 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

Journaling is enabled by default for the SAS Metadata Server. For best performance, it is recommended that journaling be enabled at all times. In addition, journaling must be properly configured in order for roll-forward recovery to be available in the event that you need to restore the 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.

The metadata server is initially set up to write journal entries to a journal file that is stored in the same directory as the most recent metadata server backup. If problems occur that prevent the repository data sets from being updated from the journal, an alert e-mail message is generated.

See Also

"Managing Alert E-mail Options for the SAS Metadata Server" on page 191

About the Journal Options

Journaling is controlled by options that are set in the omaconfig.xml file when you install or migrate your system. In most cases, there is no need to change the initial settings.

The OMA JOURNALTYPE option specifies the type of journal file that is created. By default, the SAS Deployment Wizard sets the value of this option to ROLL_FORWARD, which creates a linear journal file that permanently stores all transactions that have occurred since the most recent backup. The journal file is written to the same location as the associated backup files. Each time a new backup is executed, journaling stops and a new journal file is started in the new backup location.

CAUTION:

> **ROLL_FORWARD must be specified if you want to be able to use roll-forward recovery in the event of a metadata server failure.**

Other allowable values for OMA JOURNALTYPE are SINGLE and NONE. These values are not recommended. If you specify one of these values, then roll-forward recovery is not available.

Additional options in omaconfig.xml that apply to journaling include the following:

- JOURNALSIZE and JOURNALPATH, which are applicable only if you specify SINGLE as the OMAJOURNALTYPE.
- JOURNALYIELDCOUNT, which specifies the number of records that the update process can apply before yielding to allow other internal processing tasks. Do not change this setting without consulting with SAS Technical Support.

See Also

- "Reference Information for omaconfig.xml" on page 329
- "How Journal Options Affect Metadata Server Backup and Recovery" on page 170

Managing Alert E-mail Options for the SAS Metadata Server

About Metadata Server Alerts

The metadata server sends alert e-mails in the following situations:

- An error occurs during a metadata server backup or recovery.
- A problem occurs that prevents the repository data sets from being updated from the journal.

Alert e-mails that are generated by the metadata server are sent to the addresses that are specified in the OMA ALERTEMAIL option in the omaconfig.xml file. The generated e-mail has `Metadata Server Alert` in the subject line. The body of the message specifies the error that occurred, the name of the metadata server host machine, the metadata server port, and the location of the metadata server log.

How Alert E-mail Options for the Metadata Server Are Configured

To generate alert e-mails, the metadata server refers to the following configuration options:

- OMA ALERTEMAIL in omaconfig.xml specifies one or more e-mail addresses to which alerts are to be sent.
- EMAILSYS, EMAILHOST, and EMAILPORT options in the metadata server's sasv9.cfg file provide information about the e-mail protocol and e-mail server.
- The EMAILID option in the metadata server's sasv9.cfg file provides information that appears in the From address of alert e-mails. If the e-mail server requires authentication, this option also provides the user ID that is used for authentication.
- The EMAILAUTHPROTOCOL and EMAILPASSWORD options in the metadata server's sasv9.cfg file provide additional information to authenticate to the e-mail server, if authentication is required.

The initial settings for these options are based on information that is specified in the SAS Deployment Wizard. Values for EMAILAUTHPROTOCOL, EMAILID, and EMAILPASSWORD can be specified in the wizard only if the custom installation option is selected.

See Also

- To update alert e-mail addresses, see "Reference Information for omaconfig.xml" on page 329.
- To modify e-mail server or authentication information, see "Modifying the Metadata Server Invocation Options" on page 339.
- To update the e-mail user ID or password, see "Update a Managed Password" in Chapter 4 of *SAS Intelligence Platform: Security Administration Guide*.

Testing E-mail Alerts for the SAS Metadata Server

After your SAS system has been initially deployed, you should run a test to make sure that alert e-mails are working properly. You should repeat this test after you make any changes to your alert e-mail configuration.

To test e-mail alerts for the metadata server:

1. Log on to SAS Management Console as a user who has the appropriate credentials. (See "Who Can Do What in the Metadata Manager Plug-in" on page 36.)
2. In the Plug-ins tab, expand the **Metadata Manager** node, right-click the **Active Server** node, and select **Properties**.
3. In the Active Server Properties dialog box, select **Send Test Message**.
4. In the Send Alert E-mail Test Message dialog box, enter text to be included in the e-mail. Then select **OK**.

The metadata server sends an e-mail message to the addresses that are specified in the OMA ALERTEMAIL option in the omaconfig.xml file. The generated e-mail has `Metadata Server Alert Email Test` in the subject line. The body includes the text that you entered in the dialog box, followed by the name of the metadata server host machine, the metadata server port, and the location of the metadata server log.

If the test is successful, the following message is displayed: `An alert e-mail message was sent to` *address*. If the test is not successful, an explanatory error message is displayed.

You can also use the OMA ALERTEMAILTEST option in the omaconfig.xml file to test alert e-mails. However, this method requires stopping the server.

Chapter 15

Managing Metadata Server Memory

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 “Options for the Metadata Server Invocation Command” on page 339.

Input and 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 and 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 188.

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 optimizeIMDB to Optimize and Report on Metadata Server Memory Usage

About the Metadata Server optimizeIMDB Command Option

To optimize the usage of memory on the metadata server, you should occasionally use the optimizeIMDB command option of the metadata server script. This option analyzes the handling of character variables, checking for items such as unnecessary indexes, long character variables that can be stored as variable-length strings, and duplicate strings. A report is produced showing the expected amount of memory that will be required for each repository after the optimizations are applied. The optimizations are then applied the next time you pause and resume (or restart) the metadata server.

The optimizeIMDB command option is new with SAS 9.3. It replaces the %OMARUNAN macro and the RUNANALYSIS option of %OMABAKUP, which provided similar functionality in previous releases.

When to Optimize Memory Usage

It is not necessary to perform memory optimization 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. Optimization 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 use the optimizeIMDB command option:

- The subprogram accesses the repository data sets in the backup destination. In each data set, it analyzes the handling of character variables. The analysis checks for such items as unnecessary indexes, long character variables that can be stored as variable-length strings, and duplicate strings.
- If the subprogram identifies optimizations for a repository, it stores the results of the analysis in a data set called DS_VCHAR. This data set is stored in the directory that contains the repository files (for example, **`MetadataRepositories/Foundation`**) and in the directory that contains the most recent backup of the repository files.
- The next time the metadata server is paused and resumed or restarted, it merges the information in each repository's DS_VCHAR data set with the repository's METAIMDB data set. The updated information in the METAIMDB data set is then used to optimize memory usage. The optimizations are not applied to repository data sets on disk.

  After the changes are applied, the DS_VCHAR data sets are deleted from the repository directories (but not from the backup directories).

  CAUTION:

  The metadata server recovery facility deletes the DS_VCHAR data sets from the backup directories before it restores the repositories to their registered

locations. Therefore, the optimizations will be lost if you run a metadata server recovery before pausing and resuming (or restarting) the server.

Running the optimizeIMDB Command Option

You should be aware of the following considerations before using the optimizeIMDB command option:

- Because optimizeIMDB analyzes backup copies of your metadata repositories, a backup copy of the repositories must have been created by using the metadata server backup facility.
- If your backup directory is on the same machine as the metadata server, then you might want to run optimizeIMDB during a period of low activity to minimize the impact on client applications.
- The backup facility creates the OMA_Varan.sas7bcat catalog that is used by optimizeIMDB command option. The information in OMA_Varan.sas7bcat reflects the last backup that was made. Therefore, optimizeIMDB analyzes all repositories that were registered at the time of the backup.
- Because optimizeIMDB does not interact directly with the metadata server, no special user IDs or server privileges are required. However, the user executing the command needs full operating system access to the SAS Metadata Server backup destination path and all subdirectories.

To run the optimizeIMDB command option:

1. Open a command prompt and navigate to the metadata server start-up directory, which is in the following path: ***`SAS-configuration-directory`*****`/Lev1/SASMeta/MetadataServer`**.
2. Type one of the following commands:
   - On Windows: **`MetadataServer -optimizeIMDB`**
   - On UNIX or z/OS: **`MetadataServer.sh -optimizeIMDB`**
3. When the command is finished, you can review the following reports that are written to the backup directory:
   - Repository_memory_usage_on_*date and time*.html, which shows the amount of memory that is expected to be required for each repository after the optimizations are applied.
   - reports called *repository_id*_metaimdb_changes.txt, which list the specific memory optimizations that were identified for each repository
4. Be sure to pause and resume (or restart) the metadata server to apply the results of the analysis.

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 67. Pause and Resume actions can also be issued by using PROC METAOPERATE. See *SAS Language Interfaces to Metadata*.

Part 6

Managing SAS Metadata

Chapter 16

Managing SAS Metadata Repositories

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 205.

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 Chapter 5 of *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:

- 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 224.
- In SAS 9.2 and SAS 9.3, 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 9.3 migration process. However, the existing associations between metadata objects still remain in effect

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 82. 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.

: For example, 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 example, 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.

: For example, 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 206.

Create a Custom Repository

To organize your metadata, the recommended practice is to create folders instead of creating separate repositories. See "Working with SAS Folders" on page 217. 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 36.
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).

     Note: If the last directory in the specified path does not exist, it is created. If the path includes any higher-level directories that do not exist, an error message is displayed.
   - 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 205.

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 "Authorization Model" in Chapter 5 of *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 36.
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. (For a description of the access modes, see "About SAS Metadata Repository Access Modes" on page 205.) Then click **OK**.
6. The **Registration** tab provides a field in which you can change the location of users' home folders. If you need to make a change to this field, it is recommended that you first contact SAS Technical Support for additional steps that might be required.

Copying, Promoting, Importing, Exporting, and Analyzing Your Metadata

Tools for Copying, Promoting, Importing, Exporting, and Analyzing Metadata

The SAS Intelligence Platform includes several tools to assist you in copying, promoting, importing, exporting, and analyzing 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:

Table 16.1 *Tools for Managing Metadata Repositories*

Tool Name	Purpose	Location in SAS Management Console	Documentation Reference
Metadata server backup and recovery facility	Copy all of your metadata repositories, the repository manager, and the contents of the metadata server's configuration directory to or from the configured backup location.	**Metadata Manager** ⇨ **Metadata Utilities** ⇨ **Server Backup**	"Backing Up and Recovering the SAS Metadata Server" on page 167

Tool Name	Purpose	Location in SAS Management Console	Documentation Reference
Import Metadata and Export Metadata*	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 208
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.	In the **Folders** tab, right-click a folder and select **Export SAS Package** or **Import SAS Package**.	"Promotion Tools Overview" on page 237 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 **Folders** tab, right-click a folder and select **Copy**. Then right-click another folder and select **Paste** or **Paste Special**.	SAS Management Console Help
SAS Migration Utility	Populate a SAS deployment with metadata, associated physical data, and configuration information from an existing deployment.	Not applicable. The utility is executed from a command line.	"Performing SAS 9.3 to SAS 9.3 Migration" on page 373
Metadata Analyze and Repair	Run selected tests on metadata to locate common problems, and (when possible) repair problems that are found.	**Metadata Manager** ⇨ **Active Server** ⇨ **Analyze/ Repair Metadata**	"Analyzing and Repairing Metadata" on page 212

* 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.

** This feature is also available in SAS Data Integration Studio and SAS OLAP Cube 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 SASMCErrorLog.txt, which is located in one of the following paths:

- On Windows Vista, Windows 7, and Windows Server 2008 systems: **`C:\Users\`***user***`\AppData\Roaming\SAS\SASManagementConsole\9.3`**
- On other Windows systems: **`C:\Documents and Settings\`***user***`\Application Data\SAS\SASManagementConsole\9.3`**
- On UNIX systems: *user's-home-directory***`/SAS/SASManagementConsole/9.3`**

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.

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

In most cases, it is neither necessary nor advisable to move metadata repositories to a new location, for the following reasons:

- Metadata server performance is unlikely to be improved by moving repositories to a faster file system. The metadata server uses an in-memory database, and file system updates are performed in a background thread in SAS 9.2 and later releases.
- Repositories are unlikely to require a larger amount of disk space than is available on the current file system. In SAS 9.2 and later releases, repositories are stored in compressed data sets. Except in the very largest deployments, storage requirements are unlikely to exceed a total of 1 gigabyte.

  Note: Metadata server logs can cause disk space to fill up if additional logging has been enabled for troubleshooting or other purposes. To avoid this situation, be sure to reset logging to the default level when troubleshooting has been completed.

Because of the complexities involved, metadata repositories should not be moved unless there is a strong reason to do so. Implementation of high-availability failover for the metadata server is one of the few situations in which it might be appropriate to consider moving metadata repositories to a network-available file system.

CAUTION:

Moving a metadata repository is a complex task and is not necessary in most cases. If you determine that you need to move a metadata repository (for example, to support failover), it is strongly recommended that you first contact SAS Technical Support for advice and assistance.

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 if you have moved the repository to a new locationon the same metadata server and you want to register its new location.

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 36.
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 and Re-registering 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 when you are moving the repository to a new location on the same metadata server.

CAUTION:

Unregistering the foundation repository makes the metadata server unusable. Therefore, you should unregister a foundation repository only in special situations such as moving the repository.

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 36.
- 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 206.

To unregister a repository:

1. Log on to SAS Management Console as a user with the appropriate credentials.
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 36.
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.

   Note: When a repository is created, the repository type (foundation, custom, or project) is recorded in the repository. When you re-register a repository, the repository type must match the type that was previously recorded.

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 from its previous location if you have moved the repository to a new location on the same metadata server.

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 36.
- 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 206.

To delete a repository:

1. Log on to SAS Management Console as a user with the appropriate credentials.
2. In the Metadata Manager, expand the **Active Server** node.
3. Right-click the repository that you want to delete, and select **Delete**.

Analyzing and Repairing Metadata

About the Metadata Analyze and Repair Tools

The Metadata Analyze and Repair tools, which can be accessed from the Metadata Manager node in SAS Management Console, enable you to run selected tests on metadata to locate common problems. When possible, the tools also repair problems that the analysis has identified.

The Analyze and Repair tools provide the following options:

- You can analyze one repository at a time, or you can analyze multiple repositories simultaneously.
- You can choose to run the repair phase immediately following the analysis, or you can examine the analysis results and then apply the repairs in a separate step. (The latter option is recommended.)
- Several metadata analysis and repair tools are available, as described in the following topic. You can run either a single tool or multiple tools in a single execution.

When you run the Metadata Analyze and Repair tools, the metadata server is automatically paused to ADMINISTRATION mode. Therefore, you might want to run the tools during times of low metadata server activity. Pausing the server prevents any changes to the repositories from occurring while the analysis process is running. The server is automatically resumed when the analysis and repair processes is completed.

Metadata Analyze and Repair Tool Descriptions

Verify Metadata Files

The **Verify Metadata Files** tool analyzes key metadata server files to determine whether they are corrupt and, when possible, recommends repairs that can be applied. This tool might be useful if a metadata server failure occurs or if a user experiences unexpected results when accessing the server. For example, it might be useful if a repository cannot be opened, a new repository cannot be registered, or metadata that was previously accessible cannot be accessed. It might also be useful if metadata associations have been broken.

When you run this tool, do not select the option to **Repair immediately**. When the analysis is finished, select **View log** to determine whether any repairs are recommended. The metadata server log provides further details. Then proceed as follows:

- If any errors are reported, do not proceed with the repair process. Instead, contact SAS Technical Support for assistance.
- If warnings are reported, but no errors, then you can proceed with the repair process.

You can then proceed with the repairs, if necessary.

Verify Associations

The **Verify Associations** tool checks the metadata repository for associations in which one or the other associated object does not exist. This tool might be useful if a user experiences unexpected results when accessing the server.

When you run this tool, do not select the option to **Repair immediately**. When the analysis is finished, select **View log**. The log provides the following information:

- The total number of associations in the repository.
- The number of broken associations in the repository. These are associations within the same repository that involve at least one metadata object that does not exist. Broken associations are candidates for removal by the repair tool.

  To see details about each broken association, access the metadata server log and search for the string `AnalyzeAssociations`. You can then remove the broken associations by proceeding with the repair step.
- The number of suspect associations in the repository. These are associations between objects in different repositories that involve at least one metadata object that cannot be located. The Verify Associations tool cannot determine whether the missing object is actually missing or whether it resides in a repository that is currently offline. Therefore, suspect associations cannot be repaired.

Verify Permissions

The **Verify Permissions** tool verifies that permission objects exist only in the Foundation repository and recommends repairs that can be applied.

Verify Authentication Domains

The **Verify Authentication Domains** tool checks authentication domain objects to ensure that the object names are valid and unique. It also checks to ensure the valid usage of these objects. When possible, the tool recommends repairs that can be applied.

Orphaned Objects

The orphaned objects tools locate metadata objects that are no longer being referenced. Metadata can become orphaned when an application deletes a metadata object but does not delete metadata that is associated with the deleted object. Orphaned objects are candidates for removal by the repair tool.

It can be helpful to run the orphaned metadata tools after migrating or promoting your metadata from a previous SAS release. In particular, deleting orphaned objects after upgrading from SAS 9.1.3 can free up a significant amount of disk space. After you run one of these tools with the repair option, be sure to back up the metadata server with the Reorganize Repositories option to reclaim the unused disk space.

You can choose from the following orphaned metadata tools:

- **Orphaned Access Controls**
- **Orphaned Directories**
- **Orphaned Job Components**
- **Orphaned Directories**
- **Orphaned Properties**
- **Orphaned Source Code**

Validate SAS Folders

The **Validate SAS Folders** tool analyzes the integrity of objects contained in the SAS Folders tree and, when possible, recommends repairs that can be applied. This tool identifies issues such as duplicate, missing, or invalid object names; objects that are present in more than one folder; and objects that are stored in a different repository than their parent folders.

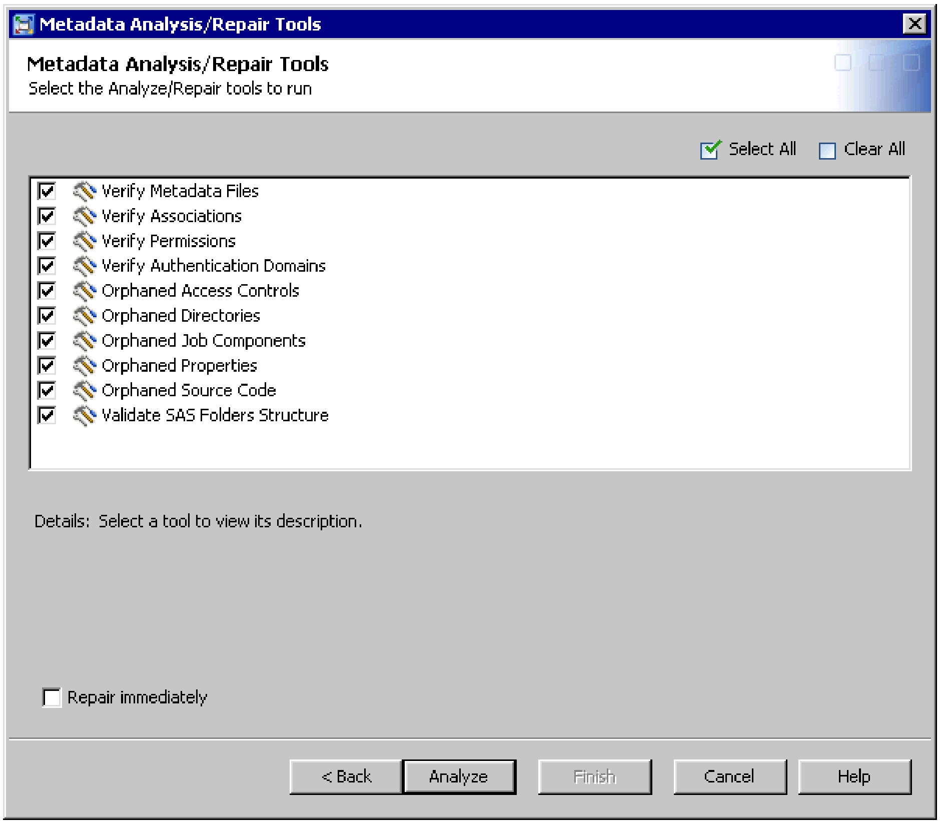
Running the Metadata Analyze and Repair Tools

CAUTION:

Be sure to back up the metadata server before you perform the repair phase of the Metadata Analyze and Repair Tool. See "Using SAS Management Console to Run an Ad Hoc Metadata Server Backup" on page 173.

To run the Metadata Analyze and Repair Tool:

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 36.
2. Expand the **Environment Management** and **Metadata Manager** nodes. Then right-click the **Active Server** node, and select **Analyze/Repair Metadata**. (You can also start the process by right-clicking one of the repositories that appears under the **Active Server** node.)
3. The following message is displayed: `The metadata server will be paused in administration mode to run the analyze and repair tools. Only administrative users will be able to access the metadata server while the process is running. Do you wish to proceed?` Click **Yes** if you are ready to begin the analysis. The server will be paused after you complete the next two wizard pages.
4. On the first wizard page, select one or more repositories to analyze and repair. Click **Next**.
5. The next wizard page lists the analysis tools that are available:



Metadata Analysis/Repair Tools

Metadata Analysis/Repair Tools
Select the Analyze/Repair tools to run

☑ Verify Metadata Files
☑ Verify Associations
☑ Verify Permissions
☑ Verify Authentication Domains
☑ Orphaned Access Controls
☑ Orphaned Directories
☑ Orphaned Job Components
☑ Orphaned Properties
☑ Orphaned Source Code
☑ Validate SAS Folders Structure

Details: Select a tool to view its description.

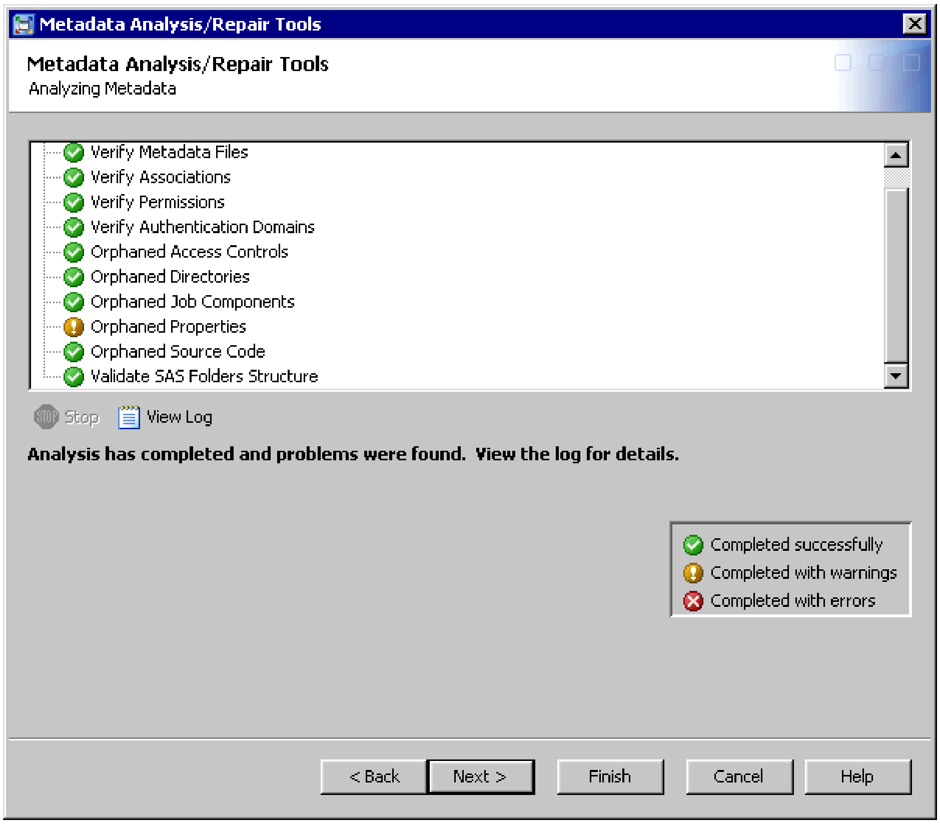
□ Repair immediately

Select or deselect the check boxes to specify the analyses to perform. For descriptions of each tool, see "Metadata Analyze and Repair Tool Descriptions" on page 212. You can also click the check box to **Repair immediately**, but it is usually recommended that you perform the repairs in a separate step.

Click **Analyze**.

Note: If you select **Repair immediately**, the **Analyze** button is replaced by the **Repair** button. You can then click this button and proceed as described in steps 9 and 10.

6. A message is displayed stating that the server is being paused to Administration mode. The analysis is then performed. When it is finished, the results are displayed as shown in the following example:



Metadata Analysis/Repair Tools
Metadata Analysis/Repair Tools
Analyzing Metadata

✓ Verify Metadata Files
✓ Verify Associations
✓ Verify Permissions
✓ Verify Authentication Domains
✓ Orphaned Access Controls
✓ Orphaned Directories
✓ Orphaned Job Components
! Orphaned Properties
✓ Orphaned Source Code
✓ Validate SAS Folders Structure

Analysis has completed and problems were found. View the log for details.

✓ Completed successfully
! Completed with warnings
✗ Completed with errors

7. If no problems are found, the following message is displayed: `Analysis has completed with no problems found.` You can then click **Finish** to exit the wizard.

   If problems are found, the following message is displayed: `Analysis has completed and problems were found. View the log for details.` Click **View Log** to see information about the errors. Additional details might also be available in the metadata server log.

8. The next wizard page displays a list of the analysis tools that found problem situations. Select one or more tools to run in repair mode, and click **Repair**.

   CAUTION:

   If the Verify Metadata Files tool reports errors (as compared to warnings), do not run the repair for that tool. Instead, contact SAS Technical Support for assistance.

9. A message reminds you to back up your metadata before running the repairs. Click **Yes** to continue; or if you have not taken a backup, click **No**, run the backup, and then restart the Analyze and Repair process from the beginning.

10. If you click **Yes**, the repairs are executed. A dialog box indicates whether each repair was completed successfully.

    You can click **View Log** to see details about the repairs. Then click **Finish** to exit the wizard.

Chapter 17
Working with SAS Folders

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.

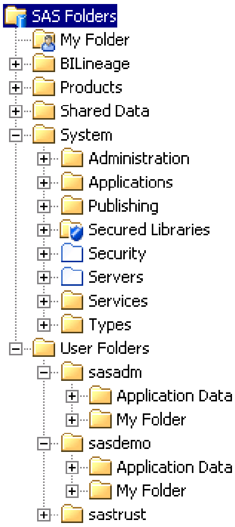
If you have installed Web applications such as SAS Web Report Studio, then some SAS folders (for example, folders that contain metadata for reports) have associated physical content that resides on the SAS Content Server. In these instances, a procedure called content mapping maps the metadata folder 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 SAS 9.3 system, effective with the second maintenance release, the initial folder structure is similar to the following:



```
SAS Folders
    My Folder
  + BILineage
  + Products
  + Shared Data
  - System
      + Administration
      + Applications
      + Publishing
      + Secured Libraries
      + Security
      + Servers
      + Services
      + Types
  - User Folders
      - sasadm
          + Application Data
          + My Folder
      - sasdemo
          + Application Data
          + My Folder
      + sastrust
```

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 222 and "Best Practices for Maintaining Associations among Objects in SAS Folders" on page 223.

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 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 224.

`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.

`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.

`Secured Libraries`

contains secured data folders, secured library objects, and secured table objects that have been created to support metadata-bound libraries. This folder is new with the second maintenance release for SAS 9.3. See the *SAS Guide to Metadata-Bound Libraries*.

`Security` and **`Servers`**

contain references to security objects (users, user groups, roles, access control templates, and authentication domains) and server objects. The white folder icons (🗀) indicate that these are virtual folders. The folders are displayed only in SAS Management Console to support operations such as promotion. See "Promoting Security Objects and Server Objects" on page 292.

`Services`

is used by SAS BI Web Services to store metadata for generated Web services.

`Types`

contains type definitions for public objects that exist on this metadata server.

`User Folders`

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: The following details apply to the creation of users’ home folders:

- 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 **`User Folders`** folder. Therefore, your site might not have a **`User Folders`** folder. If you do not have a **`User Folders`** folder, but you would like to create one, see “Change the Properties of a Metadata Repository” on page 206.
- In SAS 9.2, the folder that contains users’ home folders is called **`Users`**. If you use the SAS Migration Utility to upgrade from SAS 9.2 to SAS 9.3, then the folder name **`Users`** is retained in SAS 9.3. If you choose to use promotion instead of the SAS Migration Utility to move your content to SAS 9.3, see “Promoting Personal Folders from SAS 9.2 to SAS 9.3” on page 299.

Each home folder contains the following folders:

`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 an application needs to use it. The folder is to be accessed only by applications and is not accessed directly by the user.

`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.

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 205.

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 Systems That Are Migrated from SAS 9.1.3

If you migrate your system from SAS 9.1.3 to SAS 9.3, folders are created as follows:

- **`Products`**, **`User Folders`**, **`Shared Data`**, and **`System`** folders are created just as they are for newly installed system.
- The SAS 9.1.3 Shared Data folder structure is moved in its entirety to SAS 9.3. 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 9.3 "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 9.3 "User Folders" folder. Users' objects are moved from the SAS 9.1.3 personal folders to the SAS 9.3 personal folders.
- The SAS 9.1.3 BIP tree folder structure is moved in its entirety to SAS 9.3 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).

Using Search Folders and Favorites Folders

Search folders (new with SAS 9.3) are represented by the following icon: 🔍. The **Search** tab in SAS Management Console, which is also new with SAS 9.3, enables users to search for metadata objects based on object names, locations, descriptions, object types, creation or modification dates, keywords, and responsibilities. After performing a search, click the **Save** button on the **Search** tab to specify a folder and location in which to store the search criteria. Opening a search folder causes the search to be rerun and updated search results to appear.

Favorites folders (new with SAS 9.3) are represented by the following icon: ⭐. Favorites folders enable users to store references to often-used metadata objects. To create a Favorites folder, right-click a folder in the SAS Folders tree (for example, your personal folder), select **New** ⇨ **Favorites Folder**, and specify a folder name. Users can further organize favorites by selecting **New Favorites Group** when a Favorites folder is selected. To store a favorite, right-click the Favorites folder (or Favorites group), select **Add Favorite**, and navigate to the object that you want to reference.

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 230.

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 "Managing SAS Metadata Repositories" on page 203.

It is recommended that you not delete or rename the `User Folders` folder, even if you have permission to do so.

If you have a reason to delete or rename the `User Folders` folder, then you must change the metadata repository configuration to reflect the change. See "Change the Properties of a Metadata Repository" on page 206.

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 223.

When you create new folders, the security administrator should set permissions to ensure appropriate access.

For details, see "Access Management" in Chapter 3 of *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.

> To identify objects that are associated with information maps and reports, you can use the BI Lineage plug-in for SAS Management Console. For details, see "Using the BI Lineage Plug-in" on page 224.
>
> 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 “Promotion Tools Overview” on page 237.

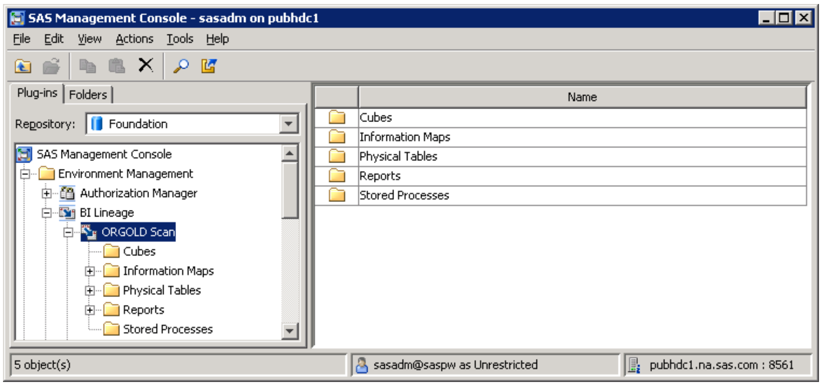
Using the BI Lineage Plug-in

About the BI Lineage Plug-in

The BI Lineage plug-in for SAS Management Console 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:


SAS Management Console - sasadm on pubhdc1
File Edit View Actions Tools Help
Plug-ins
Folders
Name

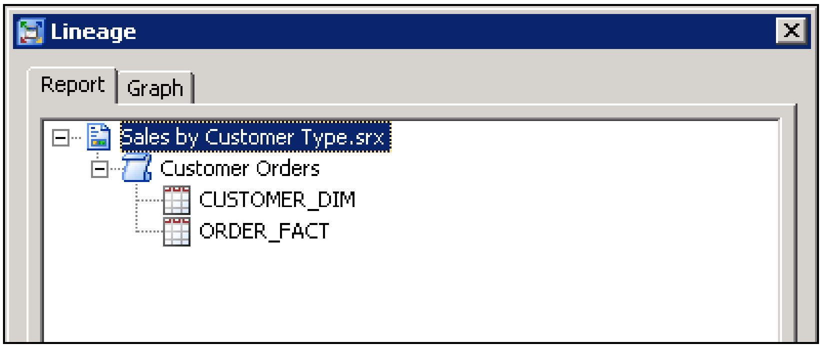
Repository: Foundation

- SAS Management Console
  - Environment Management
    - Authorization Manager
    - BI Lineage
      - ORGOLD Scan
        - Cubes
        - Information Maps
        - Physical Tables
        - Reports
        - Stored Processes

Name
Cubes
Information Maps
Physical Tables
Reports
Stored Processes

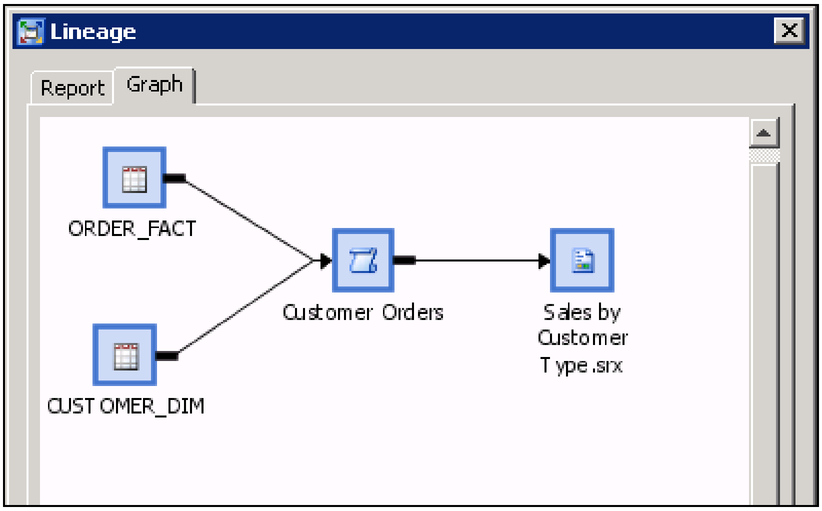
5 object(s) | sasadm@saspw as Unrestricted | pubhdc1.na.sas.com : 8561

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.



- Sales by Customer Type.srx
  - Customer Orders
    - CUSTOMER_DIM
    - ORDER_FACT

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 227.

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. 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:

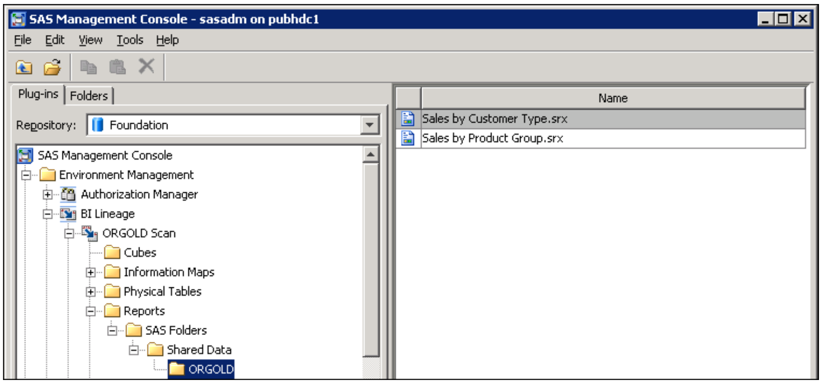
1. Log on to SAS Management Console as an unrestricted administrative user. On **Plug-ins** tab, select **BILineage** from the Repository drop-down list.
2. Navigate to **Environment Management** ⇨ **Access Control Templates** ⇨ **Default ACT**, and open the properties window for the Default ACT.
3. 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:



Repository: Foundation

SAS Management Console
Environment Management
Authorization Manager
BI Lineage
ORGOLD Scan
Cubes
Information Maps
Physical Tables
Reports
SAS Folders
Shared Data
ORGOLD

Sales by Customer Type.srx
Sales by Product Group.srx

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 “Manage the Key User Action Log File” in Chapter 4 of *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 233.

Do not set up folders based on SAS client applications.
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 that 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 “Baseline ACTs” in Chapter 6 of *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.

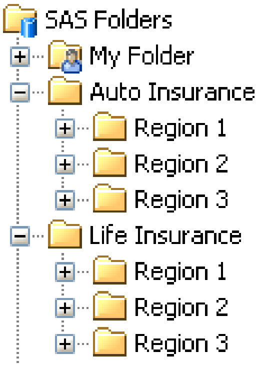
If you have separate environments for development, test, and production, then use the same folder structure across environments.

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.

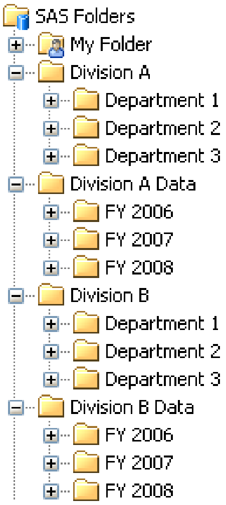


- SAS Folders
  - My Folder
  - Auto Insurance
    - Region 1
    - Region 2
    - Region 3
  - Life Insurance
    - Region 1
    - Region 2
    - Region 3

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.



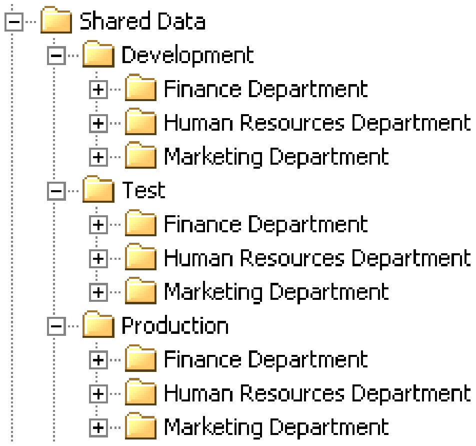
- SAS Folders
  - My Folder
  - Division A
    - Department 1
    - Department 2
    - Department 3
  - Division A Data
    - FY 2006
    - FY 2007
    - FY 2008
  - Division B
    - Department 1
    - Department 2
    - Department 3
  - Division B Data
    - FY 2006
    - FY 2007
    - FY 2008

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:



- Shared Data
  - Development
    - Finance Department
    - Human Resources Department
    - Marketing Department
  - Test
    - Finance Department
    - Human Resources Department
    - Marketing Department
  - Production
    - Finance Department
    - Human Resources Department
    - Marketing Department

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 "Promotion Tools Overview" on page 237. To facilitate promotion, use the same folder structure within each of the three folders.

Part 7

Using the Promotion Tools

Chapter 18
Promotion Tools Overview

Introduction to the Promotion Tools

About Promotion

Promotion is the process of copying selected metadata and associated physical content within or between planned deployments of SAS software. This process is repeatable for a particular deployment.

About the Promotion Tools

The SAS Intelligence Platform provides tools that enable you to promote individual metadata objects or groups of objects from one SAS deployment to another, or from one folder location to another within the same deployment. You can promote metadata as well as the physical files that are associated with 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 Packageand Import SAS Package wizards.

See Also

A different promotion tool is provided for portal content. See “Promoting Portal Content” in Chapter 12 of *SAS Intelligence Platform: Web Application Administration Guide*.

When to Use the Promotion Tools

The promotion tools (including the Export SAS Package and Import SAS Package wizards and the batch export and import tools) support the following types of scenarios:

- promotion of content 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) content from a user's home folder to a location under the `Shared Data` folder.
- promotion of content from a test environment to a production environment. For example, you might want to promote newly developed (or newly modified) content from a metadata server that is installed on a test machine to a metadata server that is installed on a production machine.
- importing content that was exported from a SAS 9.2 (or SAS 9.1.3) environment. This capability is useful in the following situations:
  - You are upgrading from SAS 9.1.3 or SAS 9.2 to SAS 9.3, but you chose not to use the SAS Migration Utility. For details, see *SAS Intelligence Platform: Migration Guide*.
  - You are running SAS 9.3 in parallel with SAS 9.1.3 or SAS 9.2, and you want to update your SAS 9.3 system with new content that you created in the earlier release.
- 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.

Types of Objects That Can Be Promoted

You can use the promotion tools to promote almost all types of content that are stored in the SAS Folders tree. The content types include libraries, tables, jobs, cubes, information maps, reports, and many other types depending on which applications you have installed.

In SAS 9.3, support was added for promoting additional object types including security objects (users, user groups, roles, and access control templates) and server objects (SAS Application Servers and their component servers, SAS object spawners, and

SAS/CONNECT spawners). In the second maintenance release of SAS 9.3, promotion support was added for the following server types: content servers, database servers, DataFlux Federation Servers, enterprise application servers, generic batch servers, grid monitoring servers, Hadoop servers, message queue polling servers, SAS/SHARE servers, SAS Scalable Performance Data (SPD) servers, and all types of scheduling servers.

Virtual folders called Servers and Security are displayed in the SAS Folders tree in SAS Management Console for use in promoting these objects.

Limitations of the Promotion Tools

The following limitations apply to the promotion tools that are described in this documentation:

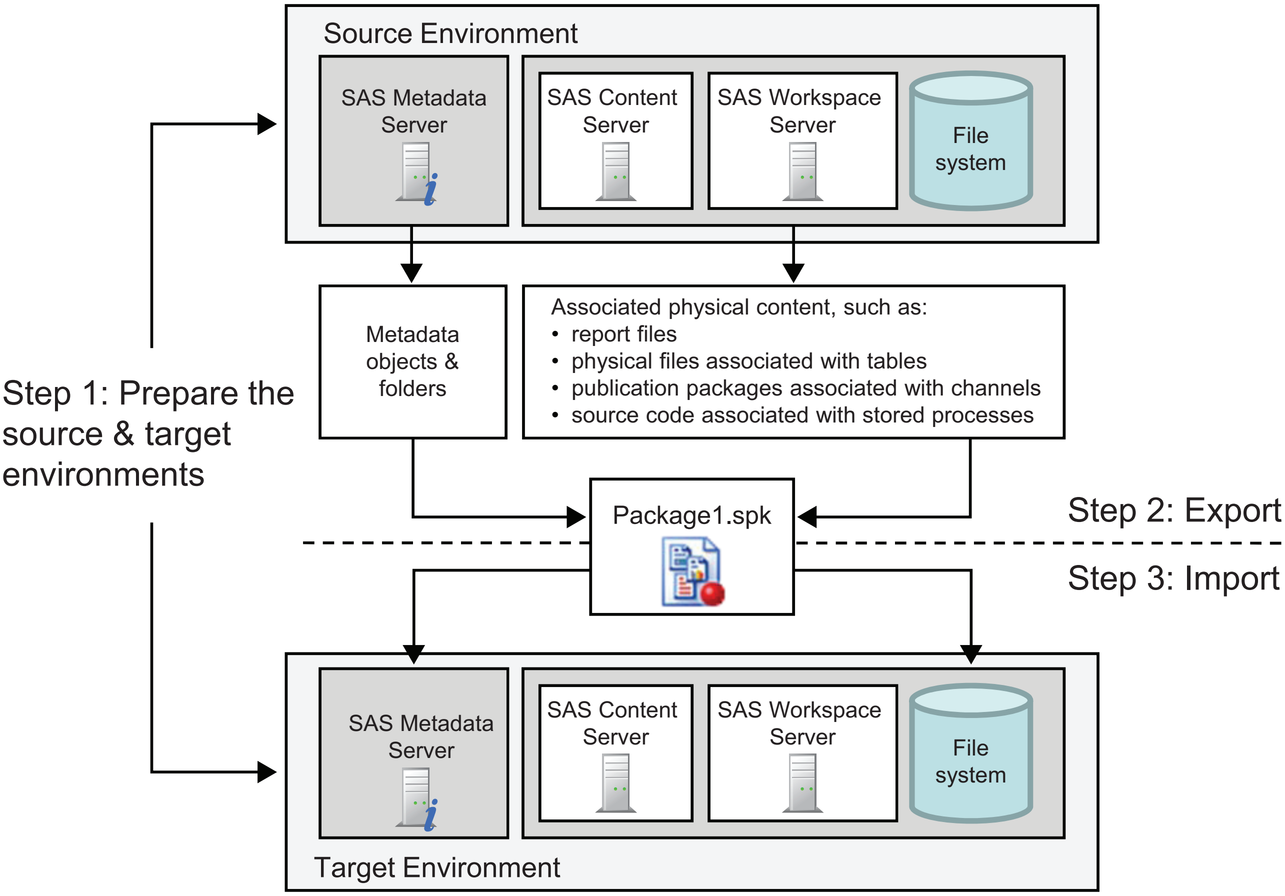
- When using the export and import wizards in SAS Data Integration Studio and SAS OLAP Cube Studio, you can promote only the objects that pertain to the application. For example, in SAS Data Integration Studio, you can promote objects such as libraries, tables, jobs, and cubes, but you cannot promote objects that are unrelated to data integration (such as reports and dashboard objects).
- You cannot use the export and import wizards or the batch export and import tools to promote portal pages, page templates, portlet instances, and the portal's application and user permissions trees. This content is not stored in the SAS Folders tree. To promote these items, you can use the content promotion tool for the SAS Information Delivery Portal. See “About the Portal Promotion Tools” in Chapter 12 of *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 directories for the physical files. For details, see “Preparing to Use the Promotion Tools” on page 243.
- 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 transport the package by e-mail or FTP 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 either of the following:
  - another metadata server, which can be located either on the same host machine or a different host machine
  - a different location in the SAS Folders tree on the source metadata server (except for server and security metadata, which can be imported only to the Server and Security folders, respectively)

Figure 18.1 Three-step Promotion Process


Source Environment
SAS Metadata Server
SAS Content Server
SAS Workspace Server
File system
Metadata objects & folders
Associated physical content, such as:
• report files
• physical files associated with tables
• publication packages associated with channels
• source code associated with stored processes
Step 1: Prepare the source & target environments
Package1.spk
Step 2: Export
Step 3: Import
SAS Metadata Server
SAS Content Server
SAS Workspace Server
File system
Target Environment

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 export and import only the objects that pertain to the application.
- 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.

Features of the Promotion Tools

Selecting Objects to Export

The Export SAS Package wizard and the batch export tool provide the following options for specifying which objects are to be exported:

- You can select multiple nested folders.
- You can select **Include empty folders** if you want to export folders that do not contain any subfolders or objects. (Otherwise, the empty folders are not exported.)
- You can include all of the objects that are contained in a folder or just selected objects.
- You can filter content, and you can identify and include dependent objects, as described in the next two topics.

Filtering Content

The export wizard includes a Filter feature that enables you to select objects based on the object name, object description, object type, or the time period during which the object was created or last modified. The Advanced Filter Options (new with SAS 9.3) enable you to identify objects for export based on their keywords, notes, extended attributes, and responsibilities fields. The same filtering capabilities are included in the batch export tool.

Identifying and Including Dependent Objects

CAUTION:

> **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, the information maps that the reports depend on must be present. If a report has stored processes or images associated with it, then those objects must be present in the target system.

The BI Lineage tool can assist you in identifying associations between objects. In addition, the promotion tools provide the following features to help you ensure that you include all of the necessary objects:

- You can use the **Include dependent objects when retrieving initial collection of objects** option of the export wizard (or the **-includeDep** option of the export batch tool) to specify that all of an object’s dependencies be identified recursively and included in the export. For example, if you are promoting a report, this feature identifies and automatically exports objects such as information maps, cubes, tables, and libraries that the report depends on.
- In the export wizard, the **Dependencies** tab lists the highlighted objects’s direct dependencies, and the **Used by** tab identifies objects that directly use the highlighted object. You can then specify which of the identified objects to include in the export. After these objects are added to the list of objects to export, you can view the new

objects' **Dependencies** and **Used by** tabs to identify additional associated objects to export.

For example, suppose the **Dependencies** tab for a report lists an information map. You can select the map to add it to the list of objects to export. You can then highlight the map and display its **Dependencies** tab, which lists any libraries, tables, and cubes that the map depends on. These objects can then be selected to be added to the export.

Note:

- The **Used by** tab does not display reports, because associations between reports and other objects are not stored in metadata. If you need to identify the reports that use a given table, cube, or information map, use the Reverse Lineage feature of the BI Lineage plug-in.
- **Used by** information cannot be displayed for dashboard objects.
- The **Dependencies** and **Used by** tabs do not display information for roles. In addition, the tabs do not list user groups that contain (or belong to) another user group. The **User Manager** plug-in provides dependency and "used by" information for these objects.

Promoting Physical Content along with Metadata

For most object types, the export and import wizards (and the batch export and import tools) promote both the physical content and the metadata. For some object types, the wizards enable you to select whether to include or exclude physical content. For example, when you export a stored process, you can choose to include or exclude the associated source code files. When you export a table or external file, you can choose to include or exclude the associated physical file.

When you use the batch tools, the promotion automatically includes all associated content except physical files for tables and external files. However, you can use a global option to prevent the batch import tool from importing source code files for stored processes.

For a detailed list of the types of physical content that can be promoted, see "Types of Physical Content That Are Promoted with Metadata Objects" on page 296.

Restoring Metadata Associations

Depending on the types of objects that you are importing, you might need to associate imported objects with objects or entities on the target system. The import wizard displays prompts that enable you to establish the needed associations. For example:

- Some objects, such as libraries and physical files, must be associated with a physical directory on the target system. Physical directories cannot be promoted. Therefore, the import wizard prompts you to specify an already existing physical directory on the target system.
- Some objects, such as libraries and stored processes, must be associated with a SAS Application Server on the target system. When you import these objects, the import wizard prompts you to select the appropriate application server on the target system. (Although it is possible to promote servers, you will generally want to use the servers that already exist on the target system.)

If you are using the batch tools, then you can specify the associations in the substitution properties file.

Reorganizing Content By Moving Objects to Different Folders

In SAS 9.3, the import wizard and the batch import tool enable you to reorganize content as part of the promotion process. You can do so by specifying new folder locations for individual objects that you are importing. The connections among objects are maintained. This feature is new with SAS 9.3.

In the batch import tool, you can change the location of an object by specifying a new property for the object in the substitution properties file. The new folder location must already exist in metadata when you use this method.

Resolve Column Differences When Importing Tables and External Files

When you import metadata for a table or external file, and a table or external file of the same name already exists in the target folder, you can use the Change Analysis feature of the import wizard to identify and resolve differences in column metadata.

In the batch import tool, you can use global options to globally control the handling of column differences.

Preparing to Use the Promotion Tools

Ensure That the Required Servers Are Present

Before using the export and import wizards (or the batch export and import tool), make sure that the appropriate servers exist and are 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 standard (non-pooled) 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) has the following credentials and permissions:

- ReadMetadata access to the objects that are being exported
- WriteMetadata access to the objects that are being imported
- WriteMemberMetadata access to the folders where objects are imported

- appropriate access to associated physical content (for example, tables and source code files) that is being exported or imported
- credentials to access the workspace server, if you are exporting or importing physical content that is associated with stored processes, tables, jobs, libraries, or external files (see "Requirements for Accessing a Standard Workspace Server" on page 39)

  Note: If your user account does not have workspace server credentials, the wizard prompts you to enter additional credentials. For example, you will be prompted to enter additional credentials if you are running the wizard using an internal account (a user ID with the suffix @saspw).

Ensure Adequate Memory

If you are planning to export or import a large number of sizable objects, then you might need to increase the amount of memory that is allocated to the import or export process. Information maps, SAS Data Integration Studio jobs, and tables are the object types that most frequently require additional memory. To increase allocated memory, follow these steps:

1. On the client machine where you are executing the export or import, 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\9.3`.
   - If you are using SAS Data Integration Studio, open distudio.ini, which is located in *SAS-installation-directory*`\SASDataIntegrationStudio\4.3`.
   - If you are using SAS OLAP Cube Studio, open olapa.ini, which is located in *SAS-installation-directory*`\SASOLAPCubeStudio\4.3`
   - 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\9.3`.
2. Locate the following line:

   ```
   JavaArgs_1=-Xmx512m
   ```
3. Replace `512` with a higher value (for example, `1024`), and save your changes.

   Note:

   - For SAS Management Console and SAS Data Integration Studio, this value is set to `1024` by default. For the batch tools and SAS OLAP Cube Studio, the default value is `512`.
   - 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. All versions of Citrix support a maximum Java heap size of 512 megabytes, and some versions support higher limits.

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 Security for Folders and Objects

In some cases, you might want to set up and test access controls for folders or objects in the source environment, and then include these access controls in the promotion. To use this approach, see "Promoting Access Controls" on page 288.

As an alternative, you might want to set up folders on the target metadata server and apply access controls to them before you import your metadata.

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 directories for the following:

- physical locations of libraries
- physical files and external format files (EFFs) that are associated with external files
- custom code that is associated with jobs, if the code is stored outside of the metadata
- source code repositories for stored processes
- **File or directory** prompts for which a default file or directory value is specified (see "Promoting Prompts" on page 291)

The user who will execute the import should have Write access to these directories.

About the Promotion Tools Documentation

To learn to use the promotion tools, access the following additional documentation:

- "Using the Export SAS Package and Import SAS Package Wizards" on page 247
- "Using the Batch Export and Import Tools" on page 264
- "Reference Information for Promotion Tools" on page 287, which provides important supplemental information that applies to the wizards as well as the batch export and import tools. The following information is included:
  - details about promoting specific object types
  - the types of physical content that are promoted along with metadata
  - considerations for promoting metadata and physical content from previous versions of SAS
  - troubleshooting information
- the product Help for SAS Management Console, SAS Data Integration Studio, or SAS OLAP Cube Studio

Chapter 19
Using the Export SAS Package and Import SAS Package Wizards

About the Export SAS Package and Import SAS Package Wizards

The Export SAS Package and Import SAS Package wizards provide a graphical user interface for performing promotions. The wizards are available in SAS Management Console, SAS Data Integration Studio, and SAS OLAP Cube Studio. You can use the wizards 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.

This chapter tells how to access the wizards. It then provides an example scenario that shows how to use the wizards to export and import a package containing several types of related business intelligence objects. The chapter also provides instructions for using the advanced features of the wizards.

As you use the wizards, be sure to refer to the "Promotion Tools Overview" on page 237 and the "Reference Information for Promotion Tools" on page 287. You can also refer to the product Help for SAS Management Console, SAS Data Integration Studio, or SAS OLAP Cube Studio.

Accessing the Export SAS Package and Import SAS Package Wizards

To access the Export SAS Package Wizard in SAS Management Console, SAS OLAP Cube Studio, or SAS Data Integration Studio:

1. Right-click any folder or object. (In SAS Management Console, you must first open the **Folders** tab.)
2. In SAS Management Console or SAS OLAP Cube Studio, select **Export SAS Package**. In SAS Data Integration Studio, select **Export** ⇨ **SAS Package**.

To access the Import SAS Package Wizard in SAS Management Console, SAS OLAP Cube Studio, or SAS Data Integration Studio:

1. Right-click the folder under which you want to import the folders and objects.
2. In SAS Management Console or SAS OLAP Cube Studio, select **Import SAS Package**. In SAS Data Integration Studio, select **Import** ⇨ **SAS Package**.

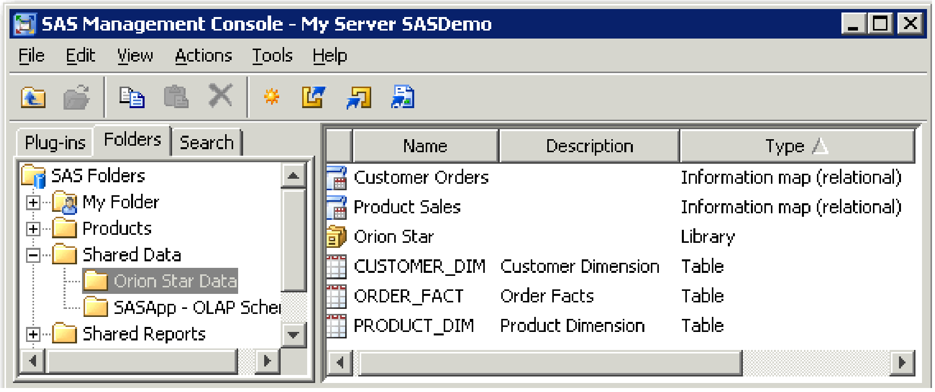
Example Usage Scenario for Export SAS Package and Import SAS Package Wizards

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:

- two reports: Sales by Customer Type and Sales by Product Type
- a stored process called Sales Analysis
- two information maps: Customer Orders and Product Sales
- a library called Orion Star
- three SAS tables: CUSTOMER_DIM, PRODUCT_DIM, and ORDERS_FACT

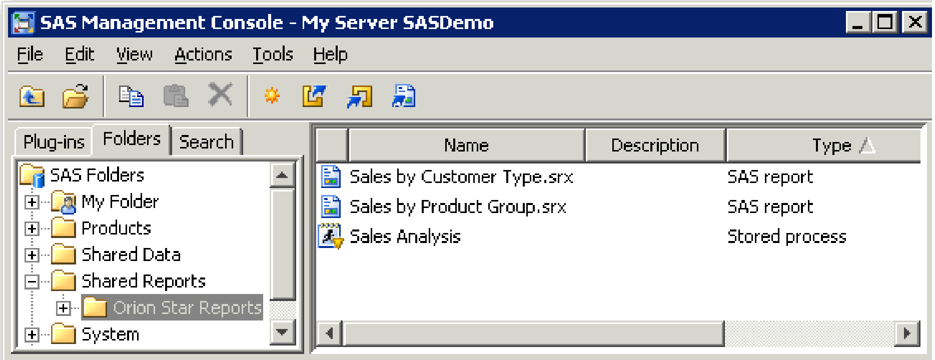
The metadata for the library, the tables, and the information maps is stored in **`/Shared Data/Orion Star Data`**, as shown here:



SAS Folders
- My Folder
- Products
- Shared Data
  - Orion Star Data
  - SASApp - OLAP Schei
- Shared Reports

Name	Description	Type
Customer Orders		Information map (relational)
Product Sales		Information map (relational)
Orion Star		Library
CUSTOMER_DIM	Customer Dimension	Table
ORDER_FACT	Order Facts	Table
PRODUCT_DIM	Product Dimension	Table

The metadata for the reports and the stored process is stored in `Shared Reports/Orion Star Reports`, as shown here:



SAS Folders
- My Folder
- Products
- Shared Data
- Shared Reports
  - Orion Star Reports
- System

Name	Description	Type
Sales by Customer Type.srx		SAS report
Sales by Product Group.srx		SAS report
Sales Analysis		Stored process

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 the associations that are involved in the example scenario, as well as the actions that must be taken to ensure that the objects will function properly in the target environment.

***Table 19.1** Example Scenario: Objects to Be Promoted and Their Associations*

Metadata Object or Folder Name	Associated Objects or Content in the Source Environment	Action Needed
Orion Star Reports and Orient Star Data folders	Reports_ A ACT (associated with the `/Shared Reports/Orion Star Reports` folder) and Data_A ACT (associated with the `/Shared Reports/Orion Star Reports` folder)	If you want to include access controls in the import: Promote the ACTs and the user groups to which they refer (or create them in the target environment) before you promote the affected objects. Make sure that the ACTs and groups have the same names in both environments.

Metadata Object or Folder Name	Associated Objects or Content in the Source Environment	Action Needed
Sales by Customer Type report	Sales by Customer Type.srx file (on SAS Content Server)	Before the promotion, ensure that a SAS Content Server exists on the target system.
	Customer Orders information map	Promote the information map and its associated tables with the report.
Sales by Product Group report	Sales by Product Group.srx file (on SAS Content Server)	Ensure that a SAS Content Server exists on the target system.
	Product Sales information map	Promote the information map and its associated tables with the report.
	Sales Analysis stored process	Promote the stored process with the report.
Sales Analysis stored process	Analysis.sas source file	Before the promotion, set up a source code directory in the file system of the target system and apply appropriate permissions.
	Source file repository definition	Before the promotion, ensure that a workspace server exists on the target system.
	Stored process server	Before the promotion, ensure that a stored process server exists on the target system.
Orion Star library	CUSTOMER_DIM, PRODUCT_DIM, and ORDER_FACT tables	Before the promotion, identify a physical location for the library on the target system.
CUSTOMER_DIM, PRODUCT_DIM, and ORDER_FACT tables	Physical tables	Import the physical tables with the table objects.

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:

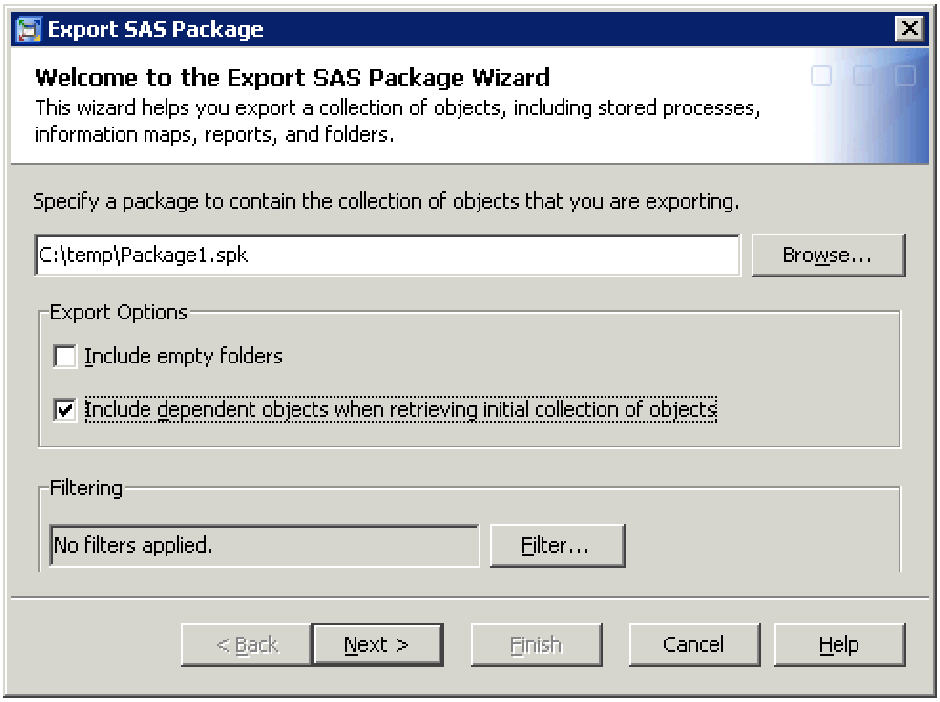
1. Make sure that you have prepared the source environment as described in “Preparing to Use the Promotion Tools” on page 243.
2. Open SAS Management Console, and connect to the source metadata server (that is, the metadata server from which you want to promote the folders and objects).
3. Select the **Folders** tab. (This step is not necessary in SAS Data Integration Studio or SAS OLAP Cube Studio.)
4. 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.)

5. 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`**).

      Or you can click **Browse**, navigate to the location on the file system where you want the SPK file to be saved, and specify a name for the file. For convenience, you might want to choose a location that will be accessible from the target environment.

   b. Select **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.



Export SAS Package

Welcome to the Export SAS Package Wizard

This wizard helps you export a collection of objects, including stored processes, information maps, reports, and folders.

Specify a package to contain the collection of objects that you are exporting.

C:\temp\Package1.spk

Export Options

□ Include empty folders

☑ Include dependent objects when retrieving initial collection of objects

Filtering

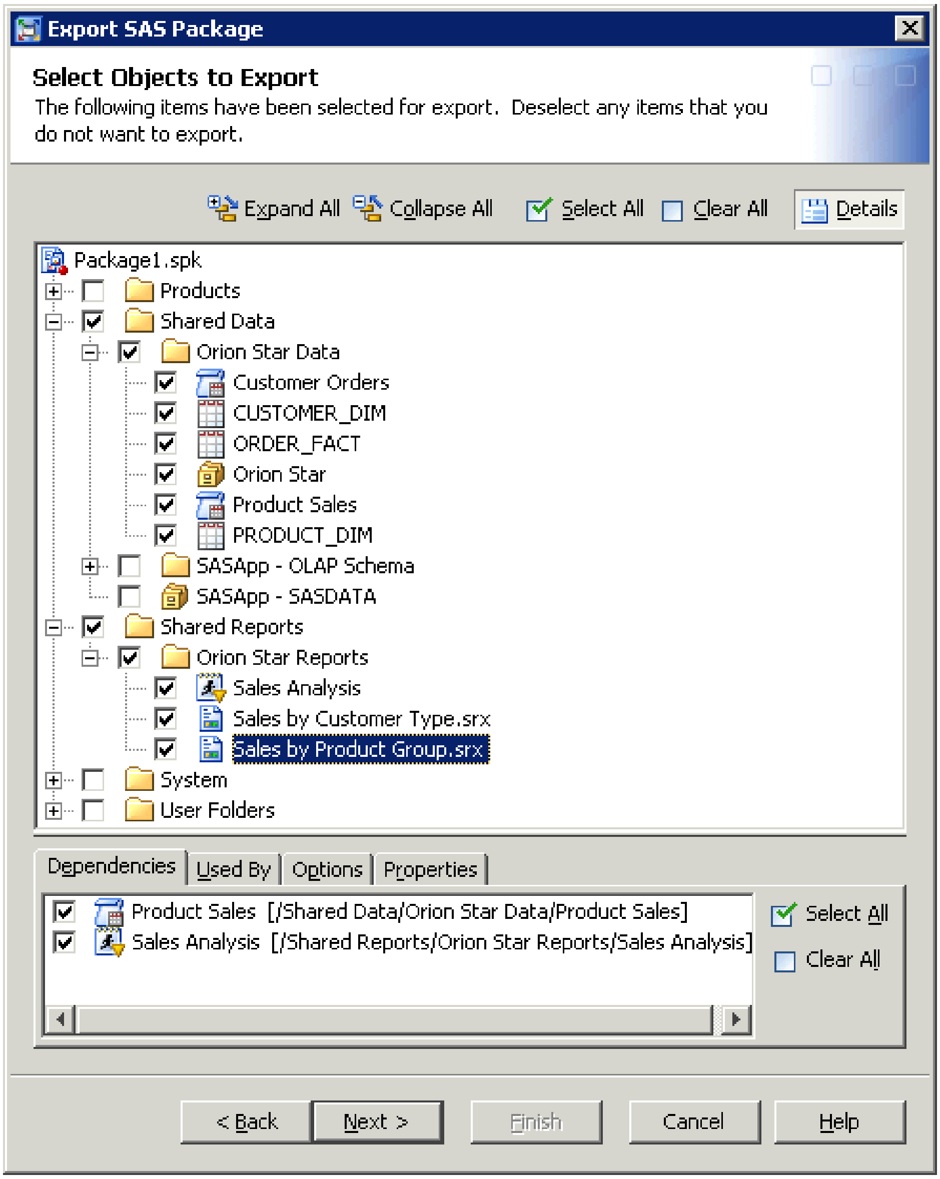
No filters applied.

   Note: On this page, you can 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. You can also use the **Advanced Filter Options** to select objects based on keywords, notes, extended attributes, and responsibilities fields. (This scenario does not use the filter features.)

   c. Click **Next**.

6. 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.



Select Objects to Export

The following items have been selected for export. Deselect any items that you do not want to export.

- Package1.spk
  - □ Products
  - ☑ Shared Data
    - ☑ Orion Star Data
      - ☑ Customer Orders
      - ☑ CUSTOMER_DIM
      - ☑ ORDER_FACT
      - ☑ Orion Star
      - ☑ Product Sales
      - ☑ PRODUCT_DIM
    - □ SASApp - OLAP Schema
    - □ SASApp - SASDATA
  - ☑ Shared Reports
    - ☑ Orion Star Reports
      - ☑ Sales Analysis
      - ☑ Sales by Customer Type.srx
      - ☑ Sales by Product Group.srx
  - □ System
  - □ User Folders

Dependencies:

☑ Product Sales [/Shared Data/Orion Star Data/Product Sales]

☑ Sales Analysis [/Shared Reports/Orion Star Reports/Sales Analysis]

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.

You can also display the **Used By** tab to identify objects that make use of the objects that you have selected.

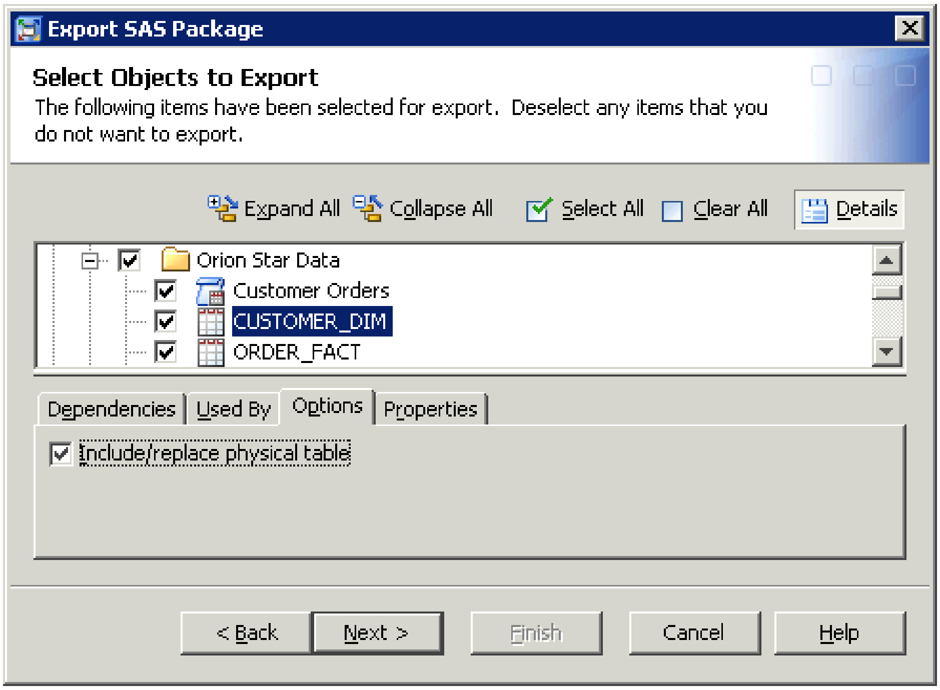
Note:

- The **Include dependent objects when retrieving initial collection of objects** option (selected on the first wizard page) identifies dependencies only for the objects that are initially selected on this wizard page. If you click **Next** to go to the next wizard page, and you then click **Back** to return to this page and select additional objects, you must use the **Dependencies** tab to identify dependencies for the additional objects.

- If you press the CTRL key and highlight multiple objects, the **Dependencies** tab displays dependencies for all of the highlighted objects.
- The **Dependencies** and **Used By** tabs are subject to the following limitations:
- The **Used By** tab does not identify reports. For example, it cannot identify reports that use the selected information map.
- The **Used By** tab does not display information for dashboard objects.
- The **Dependencies** and **Used By** tabs do not display information for roles or for associations between one user group and another.

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.



Export SAS Package

Select Objects to Export

The following items have been selected for export. Deselect any items that you do not want to export.

☑ Orion Star Data
☑ Customer Orders
☑ CUSTOMER_DIM
☑ ORDER_FACT

☑ Include/replace physical table

Note:

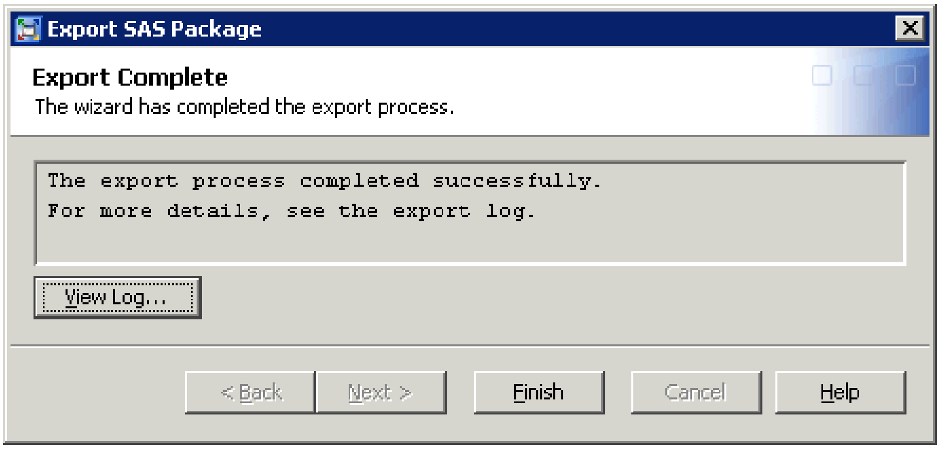
- If you choose the option to export physical tables, you should be aware that large data sets could require lengthy processing times (since data sets are packaged in the SPK file along with the metadata).
- If you press the CTRL key and highlight multiple objects of the same type, the **Options** tab 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**.

7. The next Export SAS Package page displays a summary of the objects and folders that you selected. If the list is correct, click **Next** to begin the export process.

8. 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 39 . 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.

9. 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:



Export Complete

The wizard has completed the export process.

```
The export process completed successfully.
For more details, see the export log.
```

If errors or warnings are present, you can click **View Log** to view explanatory messages. For assistance in resolving errors, see "Special Considerations for Promoting Content to SAS 9.3 from an Earlier SAS Version" on page 298.

Note: For complete step-by-step instructions for using the export 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 the export without errors.

1. Make sure that you have completed all of the necessary actions to prepare the target environment. See "Preparing to Use the Promotion Tools" on page 243 and the Action Needed column of Table 19.1 on page 249.

2. Locate the SPK file that was created by the export wizard. If necessary, copy it, move it, or e-mail it to a location that is accessible from the target environment.

3. 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).

4. Select the folder under which you want to import the folders and objects.

5. Right-click to display the menu, and click **Import SAS Package**.

6. On the first Import SAS Package page, complete the following steps:

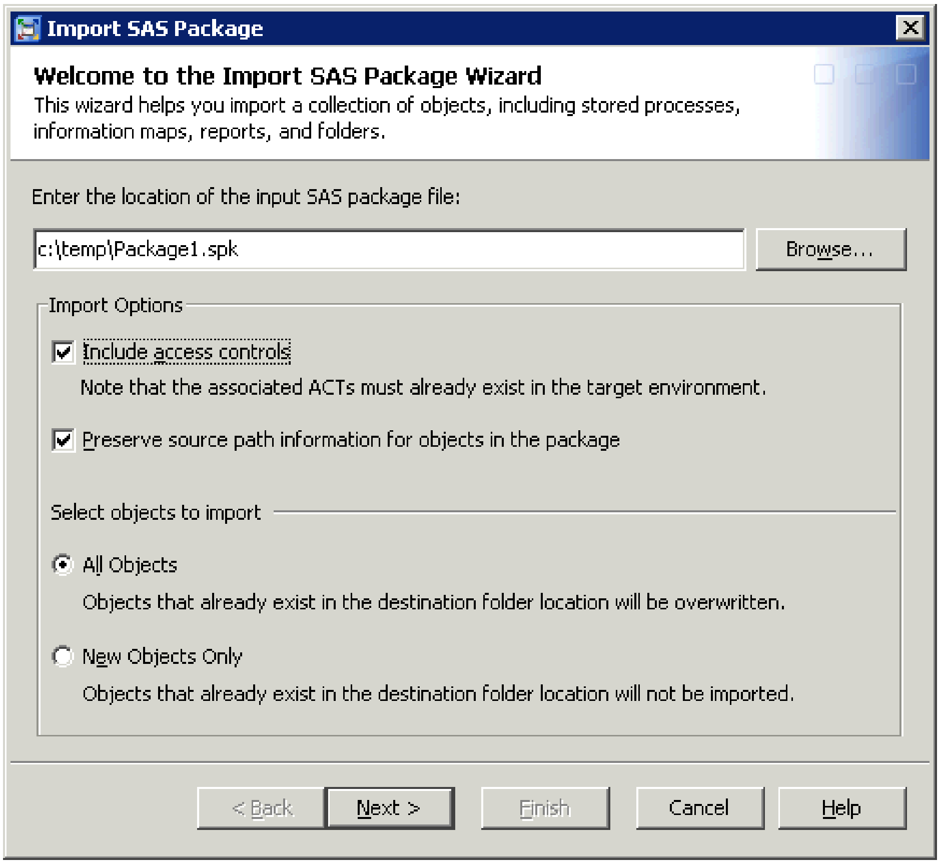
   a. Click **Browse**, and navigate to the SPK file that contains the folders and objects.

   b. Select **Include access controls**. This option will retain the associations that have been established between folders and ACTS.

c. Select **Preserve source path information for objects in the package**. This option places objects and folders 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. (You can select this option only if you are importing to the SAS Folders level.)

d. Select 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 select **New Objects Only**.



Import SAS Package

Welcome to the Import SAS Package Wizard

This wizard helps you import a collection of objects, including stored processes, information maps, reports, and folders.

Enter the location of the input SAS package file:

c:\temp\Package1.spk

Import Options

☑ Include access controls

Note that the associated ACTs must already exist in the target environment.

☑ Preserve source path information for objects in the package

Select objects to import

● All Objects

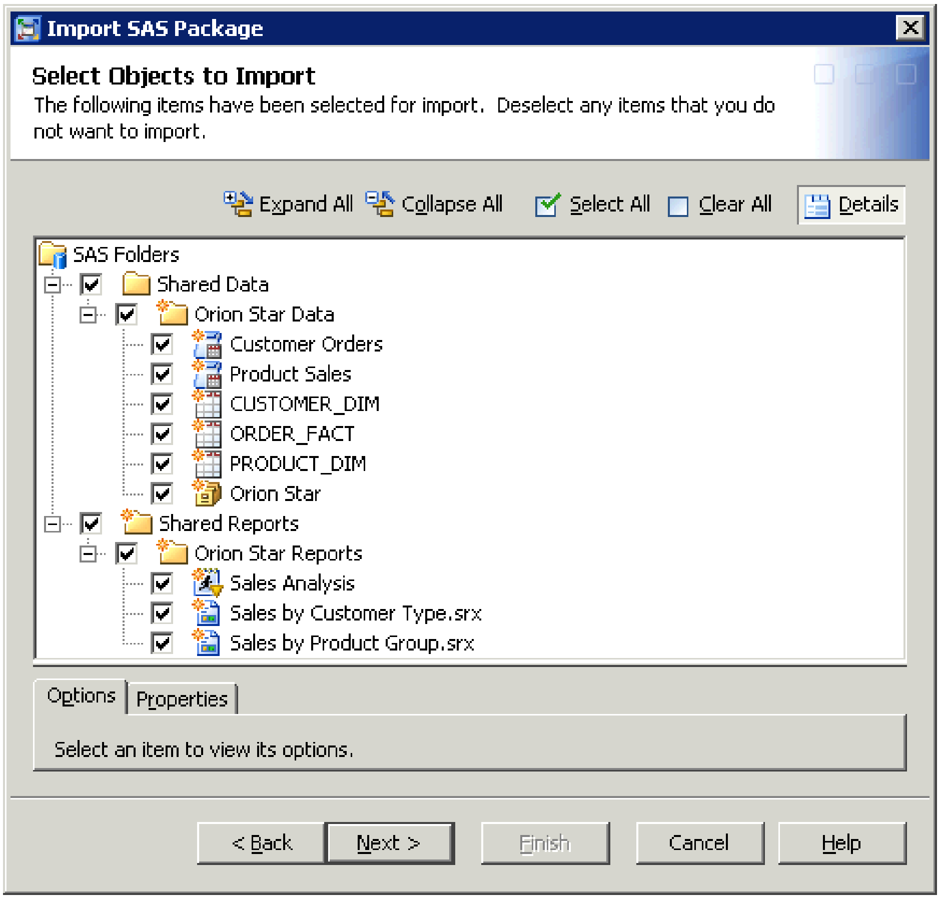
Objects that already exist in the destination folder location will be overwritten.

○ New Objects Only

Objects that already exist in the destination folder location will not be imported.

e. Click **Next**.

7. The next Import SAS Package page displays a tree that shows the objects that are included in the package.



Import SAS Package

Select Objects to Import

The following items have been selected for import. Deselect any items that you do not want to import.

- SAS Folders
  - ☑ Shared Data
    - ☑ Orion Star Data
      - ☑ Customer Orders
      - ☑ Product Sales
      - ☑ CUSTOMER_DIM
      - ☑ ORDER_FACT
      - ☑ PRODUCT_DIM
      - ☑ Orion Star
  - ☑ Shared Reports
    - ☑ Orion Star Reports
      - ☑ Sales Analysis
      - ☑ Sales by Customer Type.srx
      - ☑ Sales by Product Group.srx

Select an item to view its options.

The icons indicate whether each folder or object is new (✸) or already existing (❗) in the target environment.

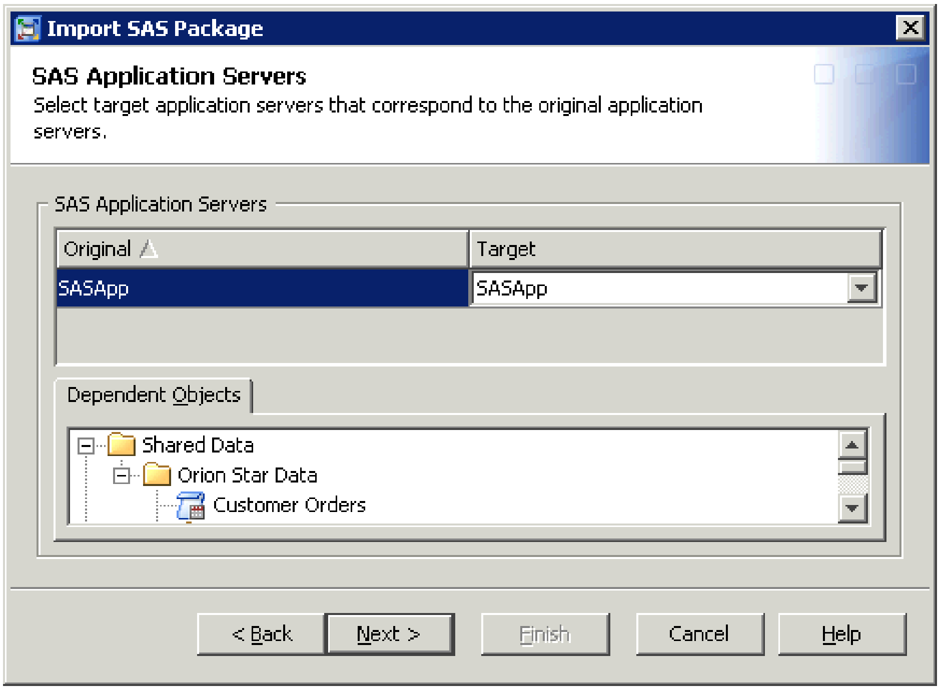
On this page, you can highlight one or more table objects, and then use the **Options** tab to indicate whether the physical tables are to be imported along with the metadata. You can also edit the table names. (These options are available only if you exported the physical tables.)

If the tables already exist on the target system, the **Options** tab also enables you to specify how differences in column metadata are to be handled. You can use the Change Analysis feature to resolve individual column differences.

Then click **Next**.

8. 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**.

9. In the **Target** column of the next page, select the application server that the imported objects will be associated with in the target environment.



Import SAS Package

SAS Application Servers

Select target application servers that correspond to the original application servers.

SAS Application Servers

Original	Target
SASApp	SASApp

Click **Next**.

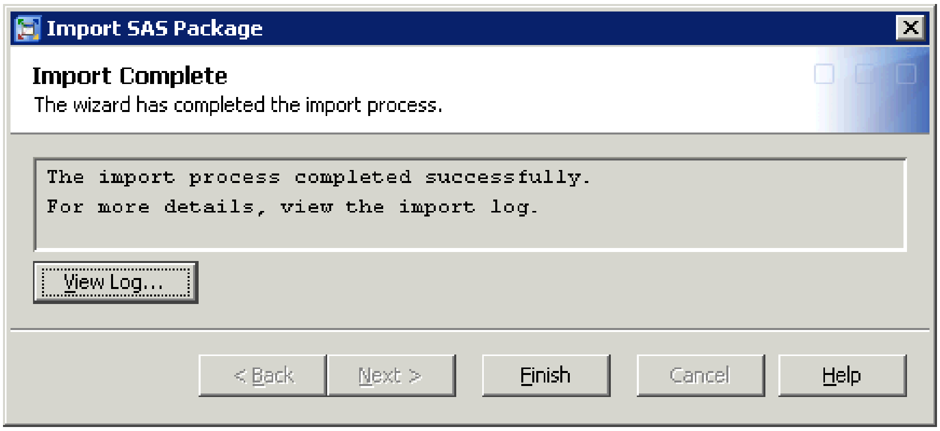
10. In the **Target** column of the next page, select the path where the source code file for the stored process should be stored in the target environment. (If the source code repository definition does not yet exist, then click **Add** to create it.) Then click **Next**.

11. 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**.

12. 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**.

13. As the import process begins, the import 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 39. 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.

14. 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:



Import SAS Package

Import Complete

The wizard has completed the import process.

```
The import process completed successfully.
For more details, view the import log.
```

If errors or warnings are present, you can click **View Log** to view explanatory messages. For assistance in resolving errors, see "Special Considerations for Promoting Content to SAS 9.3 from an Earlier SAS Version".

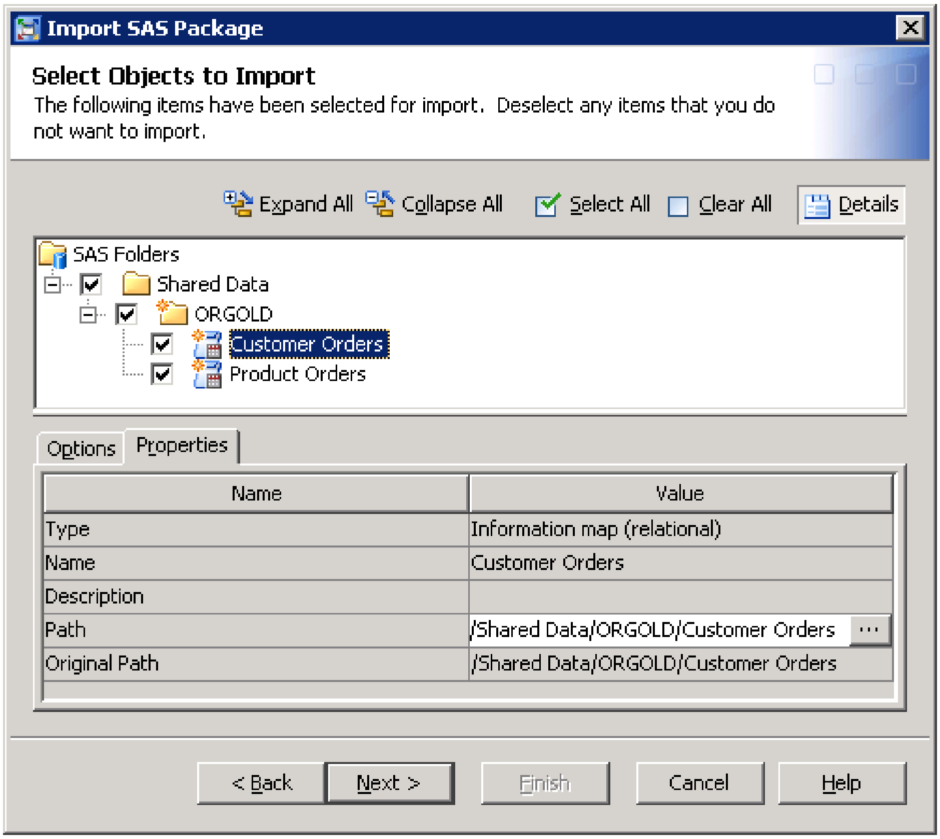
Note: The preceding steps are for the example scenario. Additional wizard pages might appear, depending on which types of objects you are promoting. For complete step-by-step instructions for using the import wizard, see the product Help for SAS Management Console, SAS Data Integration Studio, or SAS OLAP Cube Studio.

Using Advanced Features of the Import and Export Wizards

Move Objects to Different Folders When Using the Import Wizard

In SAS 9.3, the import wizard and the batch import tool enable you to reorganize content as part of the promotion process. You can do so by specifying new folder locations for individual objects that you are importing. The connections among objects are maintained.

To use this feature, select an object (for example, an information map) on the Select Objects to Import wizard page. Then select the **Properties** tab, as shown here:



Select Objects to Import

The following items have been selected for import. Deselect any items that you do not want to import.

SAS Folders
- ☑ Shared Data
  - ☑ ORGOLD
    - ☑ Customer Orders
    - ☑ Product Orders

Name	Value
Type	Information map (relational)
Name	Customer Orders
Description	
Path	/Shared Data/ORGOLD/Customer Orders …
Original Path	/Shared Data/ORGOLD/Customer Orders

In the Value column of the Path field, click the **Browse** button (…). When the Select a Location dialog box appears, select a target location in the SAS Folders tree. You can use the New Folder button (📁) in this dialog box to create a new folder.

Note:

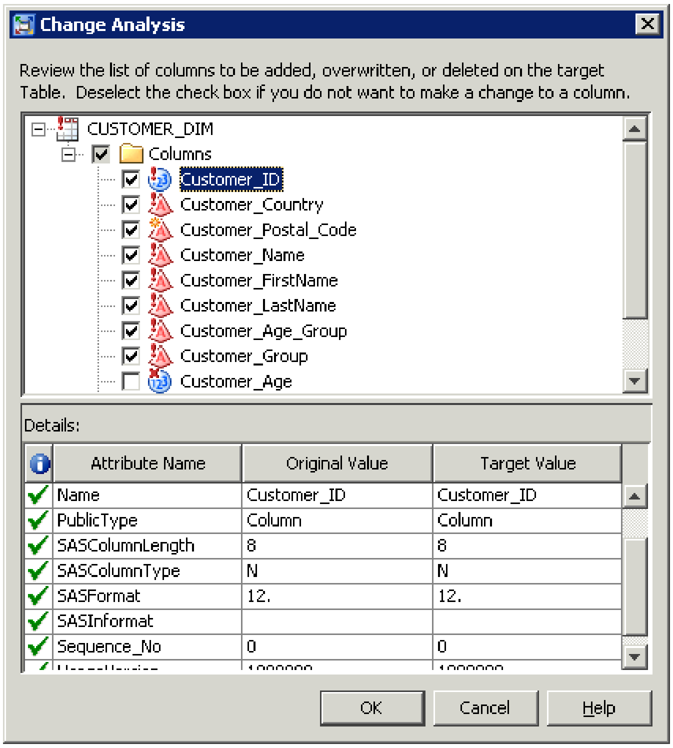
- You must repeat these steps for each object that you want to move.
- You can use this feature to move folders, except that you cannot move the highest-level folder that you are importing.
- You cannot move objects from the Servers and Security virtual folders.
- In the batch import tool, you can change the location of an object by specifying a new property for the object in the substitution properties file. For details, see "Use the Substitution Properties File to Specify New Folder Locations for Objects" on page 278.

Use the Import Wizard's Change Analysis Feature to Resolve Column Differences in Tables and External Files

When you import metadata for a table or external file, and a table or external file of the same name already exists in the target folder, you can use the Change Analysis feature of the import wizard to identify and resolve differences in column metadata. This feature is new with SAS 9.3. Use the feature to control column handling on either a global or a fine-grained basis, as follows:

Note: This feature applies only if you are overwriting existing objects as a result of selecting the **All Objects** option on the first wizard page. In addition, the feature is available only if you are importing content from SAS 9.2 or 9.3. It is not available when you import content from SAS 9.1.3.

1. On the Select Objects to Import page, highlight one or more tables or external files that you are importing.
2. On the **Options** tab, you can select or deselect the following check boxes to specify how column differences are to be handled by default for this import:
   - **Keep target columns not found in the source** is selected by default. This option retains columns in the target table if the columns are not present in the imported table. Deselect this check box if you want to delete columns from the target table if the columns are not also in the imported table. (In the next step, you will be able to override this action for individual columns.)
   - **Include source columns not found in the target** is selected by default. This option adds columns from the imported table if the columns are not present in the target table. Deselect this check box if you do not want to add any new columns to the target table. (In the next step, you will be able to override this action for individual columns.)
3. Click the **Change Analysis** button to view column information for each table that you are importing. In the Change Analysis dialog box, you can update the check boxes as needed to specify how each column is to be handled during the import. Select a column to view its change analysis information, as shown in this example:



Change Analysis

Review the list of columns to be added, overwritten, or deleted on the target Table. Deselect the check box if you do not want to make a change to a column.

- CUSTOMER_DIM
  - ☑ Columns
    - ☑ Customer_ID
    - ☑ Customer_Country
    - ☑ Customer_Postal_Code
    - ☑ Customer_Name
    - ☑ Customer_FirstName
    - ☑ Customer_LastName
    - ☑ Customer_Age_Group
    - ☑ Customer_Group
    - □ Customer_Age

Details:

	Attribute Name	Original Value	Target Value
✓	Name	Customer_ID	Customer_ID
✓	PublicType	Column	Column
✓	SASColumnLength	8	8
✓	SASColumnType	N	N
✓	SASFormat	12.	12.
✓	SASInformat		
✓	Sequence_No	0	0

The dialog box might also list items such as table indexes and keys, which cannot be deselected.

The following table explains the meaning of the icons and check boxes in the Change Analysis dialog box:

Icon Explanation	What the Import Will Do If the Check Box Is Selected	What the Import Will Do If the Check Box Is Not Selected
✘ The column metadata exists in the target table but not in the imported table.	☑ ✘ The column metadata will be deleted from the target table. If **Keep target columns not found in the source** is deselected on the **Options** tab, the check boxes for these columns are selected by default.	□ ✘ The column metadata will remain in the target table. If **Keep target columns not found in the source** is selected on the **Options** tab, the check boxes for these columns are deselected by default.
✸ The column metadata exists in the imported table but not in the target table.	☑ ✸ The column metadata will be added to the target table. If **Include source columns not found in the target** is selected on the **Options** tab, then the check boxes for these columns are selected by default.	□ ✸ The column metadata from the imported table will not be imported. If **Include source columns not found in the target** is deselected on the **Options** tab, then the check boxes for these columns are deselected by default.

Icon Explanation	What the Import Will Do If the Check Box Is Selected	What the Import Will Do If the Check Box Is Not Selected
❗ The column metadata exists in both the imported table and the target table.	☑ ❗ The column metadata from the imported table will overwrite the column metadata for the target table.	□ ❗ The column metadata will not be imported from the imported table.

For more information about a column, refer to the **Details** section of the dialog box, which displays source and target metadata values for the currently selected column.

See Also

- For information about controlling the handling of column differences in the batch import tool, see "Set Global Options for the Batch Export and Import Tools" on page 265.

Chapter 20

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 physical content such as table data and source code files. When you use the batch tools, the promotion automatically includes all associated content except physical files for tables and external files. For tables and external files, the batch tools do not provide the option of promoting physical files. For stored processes, you can use a global option to prevent the batch import tool from importing source code files.
- In the export 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

This chapter tells how to access the batch export and import tools and provides detailed syntax descriptions and examples. As you use the batch tools, be sure to refer to the "Promotion Tools Overview" on page 237 and the "Reference Information for Promotion Tools" on page 287.

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 285.

Preparing to Use the Batch Export Tool or the Batch Import Tool

Perform Standard Steps to Prepare for Promotion

The batch export tool and import tool require the same preparation steps as the Export SAS Package and Import SAS Package wizards. See "Preparing to Use the Promotion Tools" on page 243. Make sure that the user who is specified in the options has the appropriate credentials. See "Ensure Appropriate User Credentials" on page 243.

Set the DISPLAY Variable (UNIX Only)

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*

Set Global Options for the Batch Export and Import Tools

Editing the .ini Files to Set Global Options

In the .ini files for the batch export and import tools, you can set global options to override the default handling of certain object types. To set these options:

1. Go to the directory *SAS-installation-directory*`/`**`SASPlatformObjectFramework/9.3`** and locate the files ExportPackage.ini and ImportPackage.ini.
2. Make a backup copy of the .ini file that you want to modify.
3. Open the .ini file in a text editor, and add a new Java argument for each option that you want to specify. For example:

   ```
   JavaArgs_10=-Dsas.promotion.stp.includesourcecode=false
   JavaArgs_11=-Dsas.promotion.columns.excludenewcolumns=true
   JavaArgs_12=-Dsas.promotion.columns.deletetargetcolumns=true
   ```
4. Save the file.

The available options are described in the following topics.

Global Option to Include or Exclude Source Code Files for Stored Processes

The following global option applies to exporting or importing stored processes using the batch tools. You can set this option in both ExportPackage.ini and ImportPackage.ini.

`-Dsas.promotion.stp.includesourcecode`
specifies whether source code files are to be included when a stored process is exported or imported.

- Specify **`-Dsas.promotion.stp.includesourcecode=false`** if you want source code files to be excluded when stored processes are imported or exported.
- Specify **`–Dsas.promotion.stp.includesourcecode=true`** (the default setting) if you want source code files to be included when stored processes are imported or exported.

Global Options to Resolve Column Differences in Tables and External Files

The following global options apply to importing tables and external files using the batch import tool. These options specify how column differences are to be handled if you import metadata for a table or external file and a table or external file of the same name already exists in the target folder. You can set these options in ImportPackage.ini.

`–Dsas.promotion.columns.deletetargetcolumns`
specifies the action that is to be taken if the target table includes columns that are not present in the imported table.

- Specify **–Dsas.promotion.columns.deletetargetcolumns=true** if you want columns to be deleted from the target table if the columns are not also present in the imported table.
- Specify **–Dsas.promotion.columns.deletetargetcolumns=false** (the default setting) if you want columns in the target table to be retained if the columns are not also present in the imported table.

–Dsas.promotion.columns.excludenewcolumns
specifies the action that is to be taken if the imported table includes columns that are not present in the target table.

- Specify **–Dsas.promotion.columns.excludenewcolumns=true** if you do not want to add any new columns to the target table.
- Specify **–Dsas.promotion.columns.excludenewcolumns=false** (the default setting) if you want columns from the imported table to be added if the columns are not present in the target table.

Accessing the Batch Export and Import Tools

The batch import tool and export tools are called **ImportPackage** and **ExportPackage** are located in the following path:

SAS-installation-directory/**SASPlatformObjectFramework/9.3**

You can execute these tools on any machine where the SAS Platform Object Framework is installed.

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*>
<-excludeSubTypes>
<-includeDep>
<-name *character-string*>
<-nameMatchType CONTAINS | EQUALS | STARTSWITH>
<-includeDesc>
<-keywords *keyword*>
<-notes *character-string*>
<-extName *extended-attribute-name*>

<-extValue *extended-attribute-value*>
<-respName *name-of-responsible-user*>
<-respRole OWNER | ADMINISTRATOR>
<-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.

Requirement: Required if –profile is not specified.

-port *source-port-number*
specifies the port number of the metadata server from which the package is being exported.

Requirement: Required 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 243.

Requirement: Required 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.

The password can be encrypted using SAS proprietary 32-bit encryption. To obtain the encrypted password, use PROC PWENCODE. For example, to encrypt the password "mypassword," you would submit this code in the SAS Program Editor:

```
proc pwencode in="mypassword" method=sas002;
run;
```

Requirement: Required 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.

Requirement: Not required. 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 client applications such as SAS Management Console, SAS Data Integration Studio, or SAS OLAP Cube 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.

Requirement: Not required. You can omit this option if –host and –port are provided.

Interaction: If you specify this parameter, then –host, –port, and –domain 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.

If the path or filename includes spaces, then use quotation marks.

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.

Default: If you do not specify a path, then the package is placed in the directory where the command was launched.

Requirement: You must specify a filename, but the path is optional.

-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 enclose the location in 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 (**/User Folders/***user-name***/My Folder** or **/Users/***user-name***/My Folder**) rather than the shortcut (**/My Folder**). Note that the name of the parent folder for user folders varies depending on the SAS release in which the folders were created.
- 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 274. Object types are case-sensitive.

The following are examples of locations:

- –objects/

  This example exports the entire SAS Folders hierarchy and all of its objects.

- –objects*"/User Folders/sasdemo/My Folder"* or –objects*"/Users/sasdemo/My Folder"*

  This example 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)"*.

  This 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)"*

  This example 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`**.

Requirement: Required

Tips:

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 274. Object types are case-sensitive.

By default, if you specify an object type that has one or more subtypes, then the subtypes are included in the export. For example, if you specify **`Report`**, then **`Report.Component`**, **`Report.Image`**, and **`Report.StoredProcess`** are automatically included.

Requirement: Not required.

Interaction: If you want to exclude subtypes of these object types from the export, then use the –excludeSubTypes option with this option.

Note: **`Folder`** is not a valid value for the –types option. When an object of the specified type is exported, the folder structure that contains it is also exported.

-excludeSubTypes

specifies that subtypes of the specified object types are to be excluded from the export. In the list of object types, include one or more object types that are specified in the –types option.

For example, suppose you specify the following:

```
-types Report,Job -excludeSubtypes
```

Objects with the type **`Report`** and **`Job`** are included in the export. Objects with the type **`Job.Cube`**, **`Report.Component`**, **`Report.Image`**, and **`Report.StoredProcess`** are excluded.

Default: If you do not specify –excludeSubTypes with the –types option, then the export automatically includes all subtypes.

Requirement: Not required.

-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.

Requirement: Not required

Interaction: The identified dependent objects will include all types of objects, regardless of the values that are specified in the –types option.

-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 only identifies objects. It does not apply to folders.

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.

Requirement: Not required

Interactions:

If you specify –includeDesc along with –name, then the object is exported if the text string is present in either the object’s name or its description.

You can use the –nameMatchType option to further define how the –name option is to be applied.

-nameMatchType CONTAINS | EQUALS | STARTSWITH

specifies how the –name option is to be applied. The following values are valid:

CONTAINS	specifies that the object’s name must contain the character string that is specified in –name in order for the object to be exported.
EQUALS	specifies that object’s name must exactly match the character string that is specified in –name in order for the object to be exported.
STARTSWITH	specifies that the object’s name must begin with the character string that is specified in –name in order for the object to be exported.

The matching of character strings is not case sensitive.

Default: CONTAINS

Requirement: This option is not required.

Interactions:

This option is ignored if –name is not specified.

This option applies only to the object name. If you specify –includeDesc, this option does not apply to the description.

-includeDesc

specifies that the character string that is specified in –name will be searched for in both the name and the description of each object.

Requirement: Not required

Interactions:

This option is ignored if –name is not specified.

The –nameMatchType option does not apply to the object description. It applies only to the object name.

-keywords *keyword*

specifies one or more keywords to be used as filtering criteria. If an object has a keyword that exactly matches one or more of the specified keywords, then the object will be included in the export. If a keyword contains spaces, then put it in quotation marks.

-notes *character-string*

specifies one or more character strings to be used as filtering criteria for public and private Notes. If an object's public or private Notes field contains one or more of the specified strings, then the object will be included in the export. If a string contains spaces, then use quotation marks.

Only certain types of objects (for example, objects used by SAS Data Integration Studio) support the Notes field.

Requirement: Not required

-extName *extended-attribute-name*

specifies the name of an extended attribute. Objects that have an extended attribute that exactly matches the specified attribute name will be included in the export.

Only certain types of objects (for example, objects used by SAS Data Integration Studio) support extended attributes.

Requirement: Not required.

Interaction: You can use –extValue to specify a particular value for the attribute.

-extValue *extended-attribute-value*

specifies a value for the extended attribute that is specified in –extName. Objects that have a value for this attribute that exactly matches the specified value will be included in the export.

Requirement: Not required

Interaction: This option is ignored if –extName is not specified.

-respName *name-of-responsible-user*

specifies a user name that must be present in an object's Responsibilities metadata in order for the object to be exported. The specified name must exactly match the user's Name field in the metadata.

Requirement: Not required.

Interaction: You can use –respRole to specify whether the user is to be an owner or an administrator for the object.

-respRole OWNER | ADMINISTRATOR

specifies the role of the user that is specified in –respName. The valid values are OWNER and ADMINISTRATOR. Objects will be exported only if the object's Responsibilities metadata specifies this role for the user that you specify in –respName.

Requirement: Not required

Interaction: This option is ignored if –respName is not specified.

-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.
from-relative-date	specifies a date relative to the current date.
relative-time-period	specifies a time period relative to the current date.

Requirement: Not required

Interaction: You can specify either –created or –modified with this option. The default setting is –modified.

See: "Specifying an Absolute Date (or an Absolute Date and Time)" on page 275, "Specifying a Relative Date" on page 275, and "Specifying a Relative Time Period" on page 276

-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.
from-relative-date	specifies a date relative to the current date.

Requirement: No

Interaction: You can specify either –created or –modified with this option. The default setting is –modified.

Tip: To specify a date range, specify –since along with this option.

See: "Specifying an Absolute Date (or an Absolute Date and Time)" on page 275 and "Specifying a Relative Date" on page 275

-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.

Default: If –since or –before is specified but –created is not specified, then –modified is assumed.

Requirement: Not required.

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.

Default: If –since or –before is specified, and if you specify neither –created nor –modified, then –modified is assumed.

Requirement: Not required.

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.

Default: If this option is not specified, then empty folders are not exported.

Requirement: Not required.

-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. You can also use this file to reorganize content by specifying new folder locations for individual objects that you are importing. For detailed instructions, see "Using the Substitution Properties File" on page 277.

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. See "Using the Import SAS Package Wizard to Import a Package That Was Created by the Batch Export Tool" on page 285.

Requirement: Not required.

-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.

Default: If you do not specify this option, then the log is directed to the default log path and filename as described in "Viewing the Batch Export or Import Log" on page 284.

Requirement: Not required.

-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.

Default: If you do not specify this option, then the package is exported to the specified package.

Requirement: Not required.

[-? | --help]

specifies to display help for the batch export tool.

Requirement: Not required.

Special Instructions for z/OS Systems

When entering batch export and import commands on z/OS systems, you might need to precede special characters with a backslash character (\). Characters other than letters (A-Z and a-z), numbers (0-9), underscores (_), hyphens (-), and periods (.) might require a preceding backslash.

About Case Sensitivity

In the batch export and import commands, you must use the correct case for option names (for example, –includeDep and –newOnly) and object types (for example, **`InformationMap`**).

All other elements of the commands are case insensitive.

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. This list is valid as of the second maintenance release for SAS 9.3.

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.

ACT
Action
Application
ApplicationServer
BurstDefinition
Channel
Condition
ConditionActionSet
ContentSubscriber
Cube
DataExploration
DeployedFlow

InformationMap
InformationMap.OLAP
InformationMap.Relational
JMSDestination (Java Messaging System message queue)
Job
Job.Cube
Library
MessageQueue
MiningResults
MQM.JMS (queue manager for Java Messaging Service)
MQM.MSMQ (queue manager for MSMQ)
MQM.Websphere (queue manager for WebSphere MQ)

PromptGroup
Report
Report.Component
Report.Image
Report.StoredProcess
Role
SearchFolder
SecuredLibrary
Server
Service.SoapGenerated
SharedDimension
Spawner.Connect

DeployedJob
Document
EventSubscriber
ExternalFile
FavoritesFolder
Folder
Folder.SecuredData
GeneratedTransform
Note
OLAPSchema
Project
Project.EG
Project.AMOExcel
Project.AMOPowerPoint
Project.AMOWord
Prompt
Spawner.IOM (object spawner)
StoredProcess
SubscriberGroup.Content
SubscriberGroup.Event
Table
User
UserGroup

Note: By default, if you specify an object type that has one or more subtypes, then the subtypes are included in the export. For example, if you specify Report, then Report.Component, Report.Image, and Report.StoredProcess are automatically included.

Note: For server objects, the preceding list does not include all of the various subtypes. If you want to specify a server subtype, you can use SAS Management Console to look up the subtype name. On the **Folders** tab, navigate to **`System/Types`**. On the right pane, right-click an object type and select **Properties**. The Type Name is displayed on the **Advanced** tab of the Properties dialog box.

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, based on the date and time on the host machine for your client software (for example, SAS Management Console)

`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, 2011, then **`Current day of last year`** is October 12, 2010. February 29 is replaced with February 28.

`"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, 2011, then **`"Current day of last month"`** is September 12, 2011. If the previous month has fewer days, the date is adjusted downward as necessary. For example, if the current date is October 31, 2011, then **`"Current day of last month"`** is September 30, 2011.

`"Current day of last week"`
is seven days previous to the current date. For example, if the current date is October 12, 2011, then **`Current day of last week`** is October 5, 2011.

"*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.

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.

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.

Note: When you use the batch tools, the promotion automatically includes all associated content except physical files for tables and external files. (The batch tools do not provide the option of promoting physical files for tables and external files.)

- 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 establish associations with objects and entities on the target system by specifying appropriate server names, source code repository paths, database schemas, archive paths, and other values. In addition, you can specify a different location in the SAS Folders tree for a particular object or objects.

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 285.

Use the Substitution Properties File to Update Metadata Associations

Before you import a package, you must open the substitution properties file in a text editor and make any necessary updates to associations with objects and entities on the target system.

For example, when you import a library, you must establish an association between the library and a SAS Application Server on the target system; and when you import a stored process, you must associate the stored process with a directory path for the associated source code on the target system. These are just examples of the many types of associations that might be necessary.

Follow these guidelines when updating the substitution properties file:

- In the **`Connections:`** sections of the file (for example, **`Connections: Application Server`**), do not modify properties that have the word **`Source`** in the suffix. For example, do not modify **`ApplicationServer[1].SourceName=`**. Instead, modify the corresponding target property (for example, **`ApplicationServer[1].TargetName=`**).
- You can modify any of the other properties as needed, unless the comments in the file indicate that the properties are not to be modified.

- When updating properties, use 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 metadata definition for the source code repository will be created if it does not exist on the target system. (The physical location for the source code must already exist.)

```
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
```

Use the Substitution Properties File to Specify New Folder Locations for Objects

In SAS 9.3, the batch import tool enables you to reorganize content as part of the promotion process. You can do so by specifying new folder locations for individual objects that you are importing, as follows:

1. Determine whether the substitution properties file contains a section for the object that you want to move. For example, the following section title applies to the Orion Star library that is located in **`Shared Data/Orion Star Data`**:

   ```
   [/Shared Data/Orion Star Data/Orion Star(Library)]
   ```

   If you do not find a section title for the object that you want to move, then add a section title to the file using the following syntax. Be sure to include the entire path, relative to **`SASFolders`**. (You can obtain the path information from the export log, if necessary.)

   ```
   [path-relative-to-SASFolders/object-name(object-type)]
   ```

2. Immediately following the object's section title, add an entry that specifies the folder to which you want to move the object. The specified folder must already exist on the target metadata server. Be sure to include the entire path, relative to **`SASFolders`**. Use the following syntax:

   ```
   Framework.FolderPath=path-relative-to-SASFolders(Folder)
   ```

Add these entries for each object or folder that you want to move. For example:

- Suppose **`/Shared Data/Orion Star Data`** on the source metadata server contains three tables (CUSTOMER_DIM, ORDER_FACT, and PRODUCT_DIM) and two information maps (Customer Orders and Product Sales). To move these objects to an already-existing folder called **`/Shared Data/Imported Data`** on the target metadata server, you would update the substitution properties file with the following entries:

  ```
  [/Shared Data/Orion Star Data/CUSTOMER_DIM(Table)]
  Framework.FolderPath=/Shared Data/Imported Data(Folder)

  [/Shared Data/Orion Star Data/ORDER_FACT(Table)]
  Framework.FolderPath=/Shared Data/Imported Data(Folder)

  [/Shared Data/Orion Star Data/PRODUCT_DIM(Table)]
  Framework.FolderPath=/Shared Data/Imported Data(Folder)

  [/Shared Data/Orion Star Data/Customer Orders(InformationMap)]
  Framework.FolderPath=/Shared Data/Imported Data(Folder)

  [/Shared Data/Orion Star Data/Product Sales(InformationMap)]
  Framework.FolderPath=/Shared Data/Imported Data(Folder)
  ```

  On the target system, the three tables will appear directly under the **`Imported Data`** folder.

- If you want to move the entire **`/Shared Data/Orion Star Data`** folder and its contents to an already-existing folder called **`/Shared Data/Imported Data`** on the target metadata server, you would update the substitution properties file with the following entries:

  ```
  [/Shared Data/Orion Star Data(Folder)]
  Framework.FolderPath=/Shared Data/Imported Data(Folder)
  ```

  The imported objects will appear in the path **`/Shared Data/Imported Data/Orion Star Data`** on the target system.

Note: If you used the –includeDep option to export an object's dependent objects, the dependent objects are not moved automatically. You must move the dependent objects separately.

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-metadata-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.

Requirement: Required if –profile is not specified.

-port *target-port-number*
specifies the port number of the metadata server to which the package is being imported.

Requirement: Required 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 243.

Requirement: Required 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.

Requirement: Required 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.

Requirement: Not required. 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 client applications such as SAS Management Console, SAS Data Integration Studio, and SAS OLAP Cube 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).

Requirement: You can omit this option if –host and –port are provided.

Interaction: If you specify this parameter, then –host, –port, and –domain 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. The path is optional, but if you do specify a path, it must be fully qualified.

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 284 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 285 before you import this type of package)

Default: If you do not specify a path, then the package is assumed to be located in the directory where the command was launched.

Requirement: You must specify a filename, but the path is optional.

-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 (**`/User Folders/`***user-name***`/My Folder`** or **`/Users/`***user-name***`/My Folder`**) rather than the shortcut (**`/My Folder`**).

Requirement: Required

-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 274. Object types are case sensitive.

Requirement: Not required.

Note: `Folder` is not a valid value for the –types option. When an object of the specified type is imported, the folder structure that contains it is also imported.

-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.

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.

Requirement: Not required.

-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 (including access controls that are inherited) will be overwritten by the imported access controls.

Requirement: Not required.

See: "Promoting Access Controls" on page 288.

-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 "Using the Substitution Properties File" on page 277.

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.

Requirement: Not required.

-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.

Requirement: 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.

Default: If you do not specify this option, then the log is directed to the default log path and filename as described in “Viewing the Batch Export or Import Log.”

Requirement: Not required.

-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.

Default: If you do not specify this option, then the package is imported to the target metadata server.

Requirement: Not required

[-? | --help]
specifies to display help for the batch import tool.

Requirement: Not required

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

  Note: When you use the batch tools, the promotion automatically includes all associated content except physical files for tables and external files. (The batch tools do not provide the option of promoting physical files for tables and external files.)

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 and Windows 7 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 302.

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 wizard. When you run the export wizard, it 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.

   CAUTION:

   Do not save the package file after you perform the extract. Packages cannot be imported after they have been saved using a process other than the promotion tools.

2. Modify the target values in the substitution properties file as needed. See "Using the Substitution Properties File" on page 277.
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 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 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 "Ensuring the Correct Software Version When Exporting from SAS 9.1.3" on page 301.
- 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 and SAS 9.3, a larger number of object types can be imported and exported.
- 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 SAS Management Console, SAS Data Integration Studio, or SAS OLAP Cube Studio.

Chapter 21

Reference Information for Promotion Tools

About the Reference Information for Promotion Tools

The reference information in this chapter applies to the following promotion tools:

- the Export SAS Package and Import SAS Package wizards in SAS Management Console, SAS Data Integration Studio, and SAS OLAP Cube Studio
- the batch export and import tools

The reference information includes:

- details about promoting specific object types
- the types of physical content that are promoted along with metadata
- considerations for promoting metadata from previous versions of SAS
- troubleshooting information.

Promotion Details for Specific Object Types

Promoting Access Controls

About Promoting Access Controls

When you promote objects, you can choose to also promote the objects' access controls, subject to the following limitations:

- In order for an object's ACT association to be promoted, an ACT of the same name must exist in the target metadata server. For best results, you should always promote security objects (including users, user groups, roles, and ACTs) first, in a separate package, before you promote the objects with which they are associated.
- An object's access control entries are not promoted if they refer to users or groups that do not 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.

Using the Include Access Controls Option

If you want to include direct assignments of access control entries (ACEs) and access control templates (ACTs) in the promotion process, the following steps are recommended:

- Ensure that you have done the following in the source environment:
  - Set up user groups that mirror the groups that will exist in the target environment, and define a small number of users in each group for testing purposes.
  - Create ACTs that define the permission levels for each user group, apply the ACTs to objects and folders, and test the ACTs to ensure that the appropriate level of security is applied.
  - Instead of (or in addition to) ACTs, apply ACEs to folders and objects to define the permission levels for each user group. Test the ACEs to ensure that the appropriate level of security is applied.
- Promote the ACTs and user groups to the target environment (or create them in the target environment) before you import the affected objects. Make sure that the ACTs and groups have the same names in both environments.
- When you import the affected objects to the target system, select the **Include access controls** option in the import wizard (or specify **`-includeACL`** in the batch import command). Each object's access control template (ACT) associations and access control entries (ACEs) will then 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.
- If the imported objects already exist on the target metadata server, then any existing access controls for these objects (including access controls that are inherited) will be overwritten by the imported ACTs and ACEs.
- Access controls for folders are promoted only if the imported folders do not already exist on the target metadata server. If a folder already exists, its access controls are not overwritten by the imported ACTs and ACEs.

Applying Access Controls Directly in the Target Environment

Instead of promoting access controls, you can choose to 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. This option involves fewer steps, but it does not provide the opportunity to test your security settings before promotion.

For this option, do not select the **Include access controls** option in the import wizard. If you are using the batch import command, then omit the `-includeACL` option. Access controls for the imported objects will be 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.

Promoting Dashboard Objects

Dashboard objects (including dashboards, indicators, indicator data, ranges, indicator configurations, and dashboard components) can be promoted only within the same release. For example, you can promote dashboard objects from one SAS 9.3 deployment to another, but you cannot promote them from a SAS 9.2 deployment to a SAS 9.3 deployment.

For additional details about promoting dashboards, see “Promoting SAS BI Dashboard Content” in Chapter 14 of *SAS Intelligence Platform: Web Application Administration Guide*.

Promoting Libraries and Tables

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 import tools check to determine whether the directory that is defined in metadata exists in the target environment.
- If you choose to export physical tables along with metadata, you should be aware that large data sets could require lengthy processing times (since data sets are packaged in the SPK file along with the metadata).

  Note: When you use export table objects with the batch export tool, the physical tables cannot be 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 by default. 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`**.

  To override the default behavior for resolving column differences, you can use the Change Analysis feature of the import wizard or set global options for the batch import tool.

See Also

"Promoting Tables and Libraries from SAS 9.1.3 to SAS 9.3" on page 301

Promoting OLAP Content

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 wizard prompts you to either deselect the schema or select a different application server.

The following special considerations apply to promoting OLAP cubes:

- Before you promote cubes, 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).
- OLAP cubes must be promoted along with their associated jobs. The export wizard will not allow you to continue if you do not select both objects. You can use the **Dependencies** tab to identify the jobs.
- After you promote the metadata and jobs for an OLAP cube, you must rebuild the cube.

For important details about these and other tasks that are associated with promoting cubes, see the *SAS OLAP Server: User's Guide*.

See Also

"Promoting OLAP Cubes from SAS 9.1.3 to SAS 9.3" on page 302

Promoting Portal Content

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 "About the Portal Promotion Tools" in Chapter 12 of *SAS Intelligence Platform: Web Application Administration Guide*.

Promoting 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.

See Also

"Promoting Prompts from SAS 9.1.3 to SAS 9.3" on page 301

Promoting Queue Managers

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.

Promoting SAS Enterprise Guide Projects and SAS Add-In for Microsoft Office Documents

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.

Promoting Secured Data Folders, Secured Library Objects, and Secured Table Objects

The following special considerations apply to promoting secured data folders, secured library objects, and secured table objects that have been created to support metadata-bound libraries:

- The target metadata folder path and folder name must be the same as the source metadata folder path and folder name. Import and export of secured data folders, secured library objects, and secured table objects is primarily for transfer of these metadata objects from one metadata server to another. For example, you might promote these objects from a development server to a test server and then to a production server.

Note: It is also possible to import these objects into the same metadata server, folder path, and folder from which you exported them. You might use this method to back up and restore these objects.

- You cannot explicitly choose which secured table objects to import or export. All secured table objects within each secured library object are ported with that secured library object.
- After the initial import of a secured library object into a new target metadata server, you must use SAS code to apply passwords to the new target library. (For security reasons, passwords are not ported.) See documentation for the MODIFY statement in the *SAS Guide to Metadata-Bound Libraries*.

  TIP When you submit the code to add password information to the secured library object in the new target server, remember to specify options (or adjust configuration) in your SAS session so that it connects to the new target metadata server (where the passwords do not yet exist), instead of the source metadata server (where the passwords already exist).

  This requirement applies to only the first import of each secured library object into a new server (and to the first such import after the passwords for the source library object are changed). Other imports of the same secured library object into the same target server include a step that retrieves the already-present passwords from the pre-import target objects and updates the post-import target objects with those passwords.
- As a best practice, you should maintain a separate copy of your physical data for each environment (for example, development, test, and production). The alternative, pointing multiple metadata servers at the same physical data, is supported but introduces significantly more complexity.

Note: Like promotion of a regular folder, promotion of a secured data folder can include direct access controls (both explicit and ACT) only if the corresponding secured data folder does not already exist in the target server. If the corresponding secured data folder already exists in the target server, that folder's direct access controls are not modified, updated, or overwritten by the import process.

For information about metadata-bound libraries, see the *SAS Guide to Metadata-Bound Libraries*.

Promoting Security Objects and Server Objects

About Promoting Security Objects and Server Objects

In SAS 9.3, the object types that you can promote include security objects (users, user groups, roles, and access control templates) and server objects (SAS Application Servers and their component servers, SAS object spawners, and SAS/CONNECT spawners). In the second maintenance release for SAS 9.3, promotion support was added for additional server types.

The promotion of security and server objects can be useful when you are creating or refreshing a SAS environment that mirrors an existing environment. For example, you can use this feature to help create a test environment that mirrors your production environment and to refresh the test environment as needed. Or you might choose to create server or security objects in a test environment and then promote the objects to your production environment after the objects have been tested.

To support the promotion of server and security objects, the SAS Folders tree in SAS Management Console contains virtual folders called `Servers` and `Security`. These folders are referred to as virtual folders because they do not actually store metadata objects. The virtual folders are represented by white folder icons (□) and are located in the path `/System` in the Foundation repository and in each custom repository. These folders appear only in SAS Management Console. Therefore, you can promote server and security objects only with SAS Management Console and the batch export and import tools. You cannot use SAS Data Integration Studio or SAS OLAP Cube Studio to promote these objects.

Note: Server and security objects can reside only in the appropriate virtual folders. They cannot be moved or promoted to other folders. In addition, the `Server` and `Security` virtual folders can be used only for server and security objects. You cannot move or promote other types of objects into these folders.

Follow these guidelines when importing security and server objects:

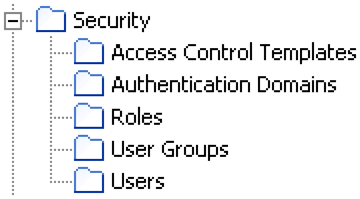
- If a package includes security objects and server objects, then you must import the package to `SASFolders/System` or higher. Be sure to select the option **Preserve source path information for objects in the package** so that the wizard will place each object in the appropriate virtual folder.
- If the package contains additional objects that are not security or server objects (for example, if the package contains stored processes), then you must import the package at the `SASFolders` level. Be sure to select the option **Preserve source path information for objects in the package** so that the wizard will place each security and server object in the appropriate virtual folder.

CAUTION:

For best results, you should always promote security or server objects first, in a separate package, before you promote the objects with which they are associated.

Promoting Security Objects (Users, User Groups, Roles, and Access Control Templates)

You can promote security objects from the `Security` virtual folder, which is located in the path `/System`. The folder has subfolders for access control templates, authentication domains, roles, user groups, and users, as shown here:



Security
- Access Control Templates
- Authentication Domains
- Roles
- User Groups
- Users

Note:

- Authentication domains cannot be promoted. If an imported user or server depends on an authentication domain that does not exist in the target environment, the domain is automatically created during the import process.
- The `Authentication Domains` and `Users` subfolders are present only in the Foundation repository.

The following special considerations apply to the promotion of security objects:

- Associations among objects are retained as follows:

  - When you import objects (for example, information maps) whose security is being controlled, their associations with security objects (users, user groups, and ACTs) are retained if the security objects already exist on the target system or are promoted in the same package. As a best practice, you should import security objects first in a separate package.
  - When users are imported, their associations with ACTs, user groups, and roles are retained if these objects either exist in the target system or are promoted in the same package as the users.
  - When user groups are imported, their associations with ACTs, users, and roles are retained if these objects exist in the target system or are promoted in the same package as the groups.
  - When roles are imported, their associations with users and user groups are retained if the users and user groups exist on the target system or are promoted in the same package as the roles.
  - When ACTs are imported, their associations with users and user groups are retained if the users and user groups exist on the target system or are promoted in the same package as the ACTs.

  If the associated objects do not exist in the target system (or are not included in the package) as described in the preceding list, the associations are removed.

  On the Select Objects to Export page, you can use the **Dependencies** and **Used by** tabs as needed to identify and select the groups that a user belongs to, the users that make up a group, and the users and groups that an ACT depends on. (Dependency and "used by" information is not available for roles.)

- If a package includes multiple types of security objects, then you must import the package to **`SASFolders/System/Security`** or higher. Be sure to select the option **Preserve source path information for objects in the package**.
- When users are imported, the following special considerations apply to the promotion of user login (account) information:
  - If a user already exists on the target system, the import operation does not overwrite the user's existing logins. However, if an imported user object contains a login for an authentication domain for which the user has no existing logins, the login is added.

    For example, suppose the target system contains one login for a user named Joel, and that the login specifies the DefaultAuth domain. If the imported package contains a login for Joel that specifies the DefaultAuth authentication domain, then that login is not imported. If the imported package contains a login for Joel that specifies a different domain (for example, Auth2), then that login is imported.
  - Passwords that are included in the logins of external accounts are not included in the import. These passwords must be added to logins manually after the import is complete.

    Note: Passwords for internal accounts are included in the import.
  - Machine names in stored user IDs are not updated during the import process. You must update them manually after the import is complete.
- Users cannot be promoted to a custom repository.
- When a user is promoted, application preferences that are associated with the user are not included in the promotion. (However, preference information is included in

migrations that are performed using the SAS Migration Utility. See the *SAS Intelligence Platform: Migration Guide*.)

- When a user object is overwritten by an imported user object, any application preferences that are associated with the user on the target system are erased.

Promoting Server Objects

You can promote server objects from the `Server` virtual folder, which is located in the path `/System`. The initial release of SAS 9.3 supports the promotion of SAS Application Servers and their component servers, SAS object spawners, and SAS/CONNECT spawners.

The second maintenance release for SAS 9.3 supports the promotion of additional server types including the following: content servers, database servers, DataFlux Federation Servers, enterprise application servers, generic batch servers, grid monitoring servers, Hadoop servers, message queue polling servers, SAS/SHARE servers, SAS Scalable Performance Data (SPD) servers, and all types of scheduling servers. If you need to promote one of these servers, but the server does not appear in the `System/Servers` folder in SAS Management Console, then follow these steps to adjust the server metadata:

1. Make sure your software, including SAS Management Console, is current as of the second maintenance release for SAS 9.3.

2. Back up your metadata server. See "Backing Up and Recovering the SAS Metadata Server" on page 167.

3. Run the Upgrade Metadata utility. To do so, go to the **Plug-ins** tab of SAS Management Console and select **Environment Management** ⇨ **Metadata Manager** ⇨ **Active Server**. Then right-click the **Active Server** node, and select **Upgrade Metadata**.

   After the utility executes, the servers should appear in the `System/Servers` folder.

The following special considerations apply to the promotion of servers:

- Server names must be unique within a metadata server. If the target metadata server already has a server with the same name as an imported object, then the promotion tool renames the imported object to `Copy of` *object-name*.

- The import wizard displays prompts that enable you to provide new values for host machine names, server ports, and paths and script names for server start-up commands. For each of these attributes, provide values that are valid for the target environment. The Summary page that appears at the end of the wizard process displays the old and new values for each attribute. Before completing the import operation, review this page to ensure that your entries are correct.

  If you are using the batch import tool, then use the substitution properties file to specify these values.

  Note: When you promote a SAS/SHARE server, the servers's port number cannot be updated.

  Note: When you promote an HTTP server, the base path cannot be updated. However, you can edit the path in the server properties after the promotion is completed.

- The promotion tools do not copy the physical files that are associated with servers, such as start-up scripts and server configuration files. These files must be copied to the target environment manually.

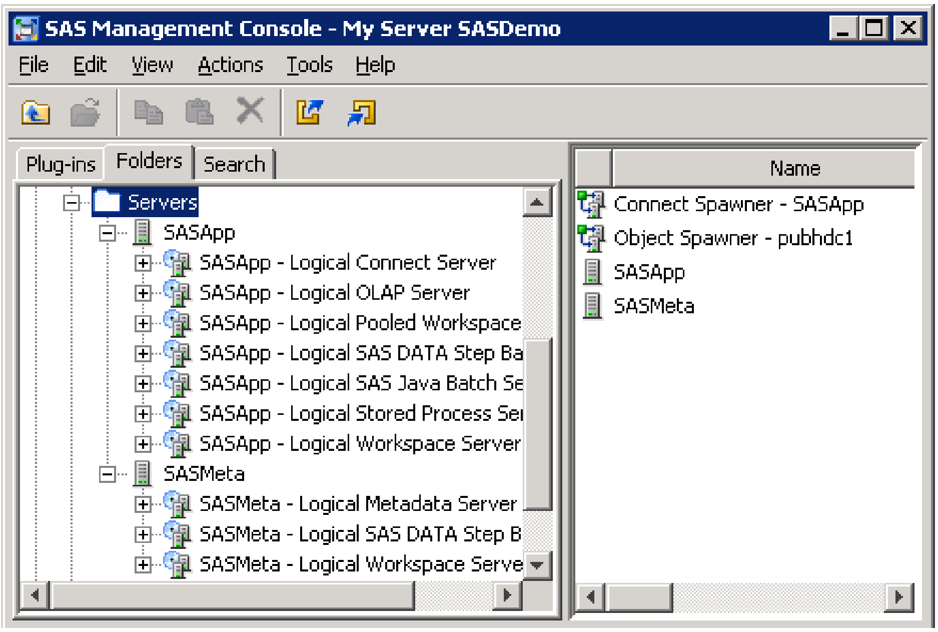
- SAS Framework Data Servers cannot be promoted.

If you need to promote a SAS Application Server and its components, see the additional considerations in "Promoting SAS Application Servers and Associated Spawners" on page 296.

Promoting SAS Application Servers and Associated Spawners

Generally, when you promote objects that depend on SAS Application Servers, it is not necessary to promote the application server objects. Instead, you can use the import wizard (or the substitution properties file) to associate imported objects with servers that already exist on the target system.

For situations in which you do need to promote metadata for application server objects, the `Servers` virtual folder contains a node for each SAS Application Server and each application server component that is configured on your system, as shown here:


SAS Management Console - My Server SASDemo
File
Edit
View
Actions
Tools
Help
Plug-ins
Folders
Search
Servers
SASApp
SASApp - Logical Connect Server
SASApp - Logical OLAP Server
SASApp - Logical Pooled Workspace
SASApp - Logical SAS DATA Step Ba
SASApp - Logical SAS Java Batch Se
SASApp - Logical Stored Process Se
SASApp - Logical Workspace Server
SASMeta
SASMeta - Logical Metadata Server
SASMeta - Logical SAS DATA Step B
SASMeta - Logical Workspace Serve
Name
Connect Spawner - SASApp
Object Spawner - pubhdc1
SASApp
SASMeta

The spawners that are associated with the servers are displayed in the right panel when the `Servers` folder is selected. The spawners do not appear on the **Folders** tab.

The following special considerations apply to the promotion of SAS Application Servers:

- When you promote an application server, all of its components are included in the promotion. You cannot select individual server components (for example, specific workspace servers) to export or import.
- On the Select Objects to Export page, you can use the **Dependencies** tab as needed to identify and select the spawners that are associated with each application server.

Types of Physical Content That Are Promoted with Metadata Objects

For most object types, the export and import wizards (and the batch export and import tools) promote both the physical content and the metadata. When you use the batch tools, the promotion automatically includes all associated physical content except tables and external files. When you use the wizards, you can choose whether to include or exclude some of the content types by using the **Options** tab.

The following table lists the types of physical content that are exported and imported with metadata objects.

Table 21.1 *Promotion of Physical Content with Metadata Objects*

Object	Associated Physical Content	Promotion Details
Channel	Packages published to the channel's archive or WebDAV location	Associated physical 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 physical content is not promoted. For details, see the *SAS OLAP Server: User's Guide*.
External file	Physical file	Associated physical content is not promoted with batch tools and is optionally promoted with wizards.
	External format file (EFF)	Associated physical content is not promoted with batch tools and is optionally promoted with wizards.
Job	Custom code, preprocesses, and postprocesses	Associated physical content is always promoted.
Library	XML files and XML maps that are contained in the library	Associated physical content is always promoted with batch tools and is optionally promoted with wizards.
Prompt	Physical files that are associated with `File or directory` prompts	Associated physical content is not promoted. For details, see "Promoting Prompts" on page 291.
Report	XML (.srx) file that defines the report	Associated physical content is always promoted. The target is the path on the SAS Content Server that is mapped to the target metadata folder via content mapping.

Object	Associated Physical Content	Promotion Details
Stored process	Source code (.sas) file	Associated physical 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 wizard enables you to add it. 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.
Table	Physical table	Associated physical content is not promoted with the batch tools and is optionally promoted with the wizards. 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.

See Also

"Promoting Physical Content along with Metadata" on page 242

Special Considerations for Promoting Content to SAS 9.3 from an Earlier SAS Version

About Promoting Content to SAS 9.3 from an Earlier SAS Version

You can export a package from a SAS 9.1.3 or SAS 9.2 metadata server and import it to a SAS 9.3 metadata server. This capability is useful in the following situations:

- You are upgrading from SAS 9.1.3 or SAS 9.2 to SAS 9.3, but you chose not to use the SAS Migration Utility. For details, see *SAS Intelligence Platform: Migration Guide*.
- You are running SAS 9.3 in parallel with SAS 9.1.3 or SAS 9.2, and you want to update your SAS 9.3 system with new content that you created in the earlier release.

The following information is important when you promote content to SAS 9.3 from an earlier version.

- the need to correctly promote personal folders from SAS 9.2 to SAS 9.2
- special considerations for promoting folders, dashboard objects, portal content, and data explorations from SAS 9.1.3 or SAS 9.2 to SAS 9.3
- additional considerations for promoting content from SAS 9.1.3 to SAS 9.3

Promoting Personal Folders from SAS 9.2 to SAS 9.3

Options for Correctly Promoting Personal Folders from SAS 9.2

In SAS 9.2, users' personal folders are located by default under a parent folder called `Users`. The name of this folder in your SAS 9.3 deployment depends on which approach you use to upgrade:

- If you upgrade using the SAS Migration Utility, then the name of the folder (`Users`) is retained in SAS 9.3. You can promote users' personal folders along with your other content without the need for additional steps.
- If you choose not to use the SAS Migration Utility, then your SAS 9.3 deployment contains the new default name for the parent folder, which is `User Folders`. In this situation, you must use one of the approaches described in this section to correctly promote the content of your personal folders.

CAUTION:

If you do not use the SAS Migration Utility to upgrade from SAS 9.2 to SAS 9.3, be sure to follow the instructions in this section to ensure that your users' personal folders will work properly after promotion.

You can use either of the following approaches to correctly promote your users' personal folders:

- "Option 1: Promote Personal Folders to the New Location in SAS 9.3"
- "Option 2: Configure SAS 9.3 to Recognize the "Users" Folder Name, and Promote the Existing Folder"

Option 1: Promote Personal Folders to the New Location in SAS 9.3

You can promote users' personal folders from SAS .2 to the new location (`User Folders`) in SAS 9.3. To do so, you must promote the personal folders separately from other metadata. In addition, you must promote the personal folders without promoting the `Users` folder itself. Follow these steps:

1. Use SAS Management Console to connect to the SAS 9.2 metadata server, and select the **Users** node in the SAS Folders tree.
2. In the right pane, select all of the personal folders that you want to export. (Press CTRL to enable the selection of multiple items.) Then right-click the mouse, select **Export SAS Package**, and complete the export.
3. Use SAS Management Console to connect to the SAS 9.3 metadata server. Select the **User Folders** node in the SAS Folders tree, and import the package.

Option 2: Configure SAS 9.3 to Recognize the "Users" Folder Name, and Promote the Existing Folder

If you do not want to switch to the new folder name (`User Folders`) for personal folders, you can reconfigure SAS to use the previous folder name (`Users`). For instructions, contact SAS Technical Support.

Restriction on Specifying Locations for SAS 9.1.3 or SAS 9.2 Folders in a SAS 9.3 Custom Repository

In SAS 9.3, the import wizard and the batch import tool enable you to reorganize content as part of the promotion process. You can do so by specifying new folder locations for individual objects or folders that you are importing.

When you are importing SAS 9.1.3 or SAS 9.2 content to a custom repository, you cannot use this feature to change the location of a folder. However, you can use the feature to change the location of individual objects in the custom repository.

Promoting Dashboard Objects from SAS 9.1.3 or SAS 9.2 to SAS 9.3

Dashboard objects (including dashboards, indicators, indicator data, ranges, indicator configurations, and dashboard components) can be promoted only within the same release. To move these objects from a SAS 9.1.3 or SAS 9.2 deployment to a SAS 9.3 deployment, you must use the SAS Migration Utility.

Promoting Portal Content from SAS 9.1.3 or SAS 9.2 to SAS 9.3

To promote portal pages, page templates, portlet instances, and the portal's application and user permissions trees from SAS 9.1.3 or SAS 9.2, use the content promotion tool for the SAS Information Delivery Portal 4.3. See “About the Portal Promotion Tools” in Chapter 12 of *SAS Intelligence Platform: Web Application Administration Guide*.

Converting Data Explorations to Reports after Promotion from SAS 9.1.3 or SAS 9.2 to SAS 9.3

In SAS 9.3, SAS Web OLAP Viewer is discontinued. After you have promoted your data explorations to SAS 9.3 from SAS 9.1.3 or 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 or SAS 9.2 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, be 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 Chapter 5 of *SAS Intelligence Platform: Migration Guide*.

Additional Considerations for Promoting Content from SAS 9.1.3 to SAS 9.3

Types of Objects That Can Be Promoted from 9.1.3 to SAS 9.3

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 and 9.3, 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.

Promoting Tables and Libraries from SAS 9.1.3 to SAS 9.3

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.

When you use the Import SAS Package Wizard to import tables from SAS 9.1.3, the following options are disabled: **Keep target columns not found in the source**, **Include source columns not found in the target**, and **Change Analysis**. For more information, see “Use the Import Wizard’s Change Analysis Feature to Resolve Column Differences in Tables and External Files” on page 259.

Promoting Prompts from SAS 9.1.3 to SAS 9.3

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 9.3 (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 SAS 9.3

To promote OLAP cubes from SAS 9.1.3 to SAS 9.3, use the ExportCubes and ImportCubes batch tools. For detailed instructions, see "Importing and Exporting SAS OLAP Cubes" in the *SAS OLAP Server: User's Guide*.

Promoting Metadata from SAS 9.1.3 to SAS 9.3 By Using the Batch Import Tool

You can use the batch import and export tools only with a SAS 9.2 or SAS 9.3 metadata server. You can use the batch import tool to import a package that was created by the export wizard 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 285.

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 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:

Solutions for Error Messages

`Out of memory error occurred during the import (or export) process.`

This message indicates that an out of memory error occurred on the client machine. The 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 244.

`ERROR - Failure exporting metadata from source repository. Reason: org.omg.CORBA.NO_MEMORY: Insufficient memory`

This message, or a similar message, appears if an out-of-memory error occurs on the metadata server machine. Solutions might include the following:

- Check the memory settings on the server machine and adjust them as necessary.

- Use the Analyze and Repair Metadata utility in SAS Management Console to clean up orphaned metadata. Then use the Reorganize Repositories option of the metadata server backup facility to reclaim unused disk space left from deleted metadata objects.
- Instead of exporting all of your objects at the same time, perform multiple exports that include a smaller number of objects.

`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.`

Content mapping, which is set up during the installation and configuration process, establishes the location of the XML (.srx) files that define reports. The location must be a path on the SAS Content Server. If this message appears when you try to run the import wizard, click **OK** to close the message box and click **Cancel**. Then follow these steps to check content mapping on the target metadata server:

1. Right-click the appropriate folder in the SAS Folders tree, and click **Properties**. Generally, content mapping is specified in the root folder (**SAS Folders**).
2. Open the **Content Mapping** tab, and make any necessary changes. When you specify the location on the SAS Content Server, you must enter the user ID and password for the SAS Trusted User (sastrust) in the appropriate fields. Then 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.

Part 8

Appendixes

Chapter 22

Default SAS Ports

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. Middle-tier SAS applications communicate with each other using a combination of multicast IP addresses and multicast UDP ports. 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. For some servers, the SAS Deployment Wizard assigns the next available port number in the range if the default port number is already in use. In addition, the SAS Deployment Wizard allows different port numbers to be specified for any server. Therefore, your site might use different port numbers than the ones that are shown here.

Table 22.1 *Default Port Numbers and Descriptions for SAS Servers and Spawners*

Default Port Value	Description
5091	SAS Remote Services application port
5451	SAS OLAP Server port

Default Port Value	Description
7551	SAS/CONNECT server port
8451	Operating System Services scheduler port
8551	SAS/SHARE server port
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)
8621	SAS Stored Process Server: load balancing connection 2 (MultiBridge)
8631	SAS Stored Process Server: load balancing connection 3 (MultiBridge)
10021	SAS Deployment Tester server port
22031	SAS Framework Data Server

* In SAS 9.2 and 9.3, two or more workspace servers can share the same port even if they are running at the same time.

Default Port Numbers for SAS Middle-Tier Applications

The following table shows the default port numbers for SAS middle-tier applications. In some cases, the SAS Deployment Wizard assigns the next available port number in the range if the default port number is already in use. In addition, 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 22.2 Default Port Numbers and Descriptions for SAS Middle-Tier Applications

Default Port Value	Description
5091	SAS Remote Services application port
7450 - 7459	IP Multicast UDP Scheduler Main Channel port 1
7460 - 7469	IP Multicast UDP Scheduler Main Channel port 2
7570	SAS Web Report Studio IP Scheduling UDP port 1
7571	SAS Web Report Studio IP Scheduling UDP port 2
7572	SAS Web Report Studio IP Scheduling UDP port 3
8561	Multicast UDP port

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. For some servers, the SAS Deployment Wizard assigns the next available port number in the range if the default port number is already in use. In addition, the SAS Deployment Wizard allows different port numbers to be specified for any server. Therefore, your site might use different port numbers than the ones that are shown here.

Table 22.3 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
	RMI ports for application servers	9811, 9812 (additional administrative consoles increment by 1)
JBoss Application Server	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)
	Secure listen port for managed servers	8443, 8543 (additional managed servers increment by 100)
Platform Suite for SAS	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.

For some servers, the SAS Deployment Wizard assigns the next available port number in the range if the default port number is already in use. In addition, the SAS Deployment

Wizard allows different port numbers to be specified for any server. Therefore, your site might use different port numbers than the ones that are shown here.

Table 22.4 *Examples of Incremented Port Numbers in Level 2 of a Multiple-Level Environment*

Default Port Value	Description
5452	SAS OLAP Server port
7552	SAS/CONNECT server port
8552	SAS/SHARE server port
8562	SAS Metadata Server
8582	SAS object spawner: operator port
8592	SAS Workspace Server port
8592	Metadata utilities SAS Workspace Server port
8602	SAS Stored Process Server: bridge connection
8612, 8622, 8632	SAS Stored Process Server: load balancing connections
8702	SAS Pooled Workspace Server port
8802, 8812, 8822	SAS object spawner: pooled workspace server port banks

Chapter 23

Reference: Configuration Directories

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 20.

Contents of the Levn Subdirectory

The `Lev`*n* 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 `Lev`*n* directory can contain the subdirectories and files that are listed in the following table:

Table 23.1 *Contents of the Levn Subdirectory*

Subdirectory or File	Description
`AppData`	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).
`ConnectSpawner`	Contains the management script, configuration files, and logs for the SAS/CONNECT spawner.
`Data`	Can be used to store user data.
`DeploymentTesterServer`	Contains files that are used by the Deployment Tester plug-in to SAS Management Console.
`Documents`	Contains Instructions.html, which contains post-installation configuration instructions; DeploymentSummary.html; ConfigurationErrors.html; and other application-specific documents.
`FrameworkServer`	Contains the management script, configuration information, log files, and database files for the SAS Framework Data Server.
`Logs`	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 "Default Locations for Server Logs" on page 319.
`Logs/Configure`	Contains logs that are created by the SAS Deployment Wizard.
`ObjectSpawner`	Contains a management script, configuration files, and logs for the object spawner.
`SASApp`	Contains management scripts, configuration files, and logs for SAS Application Server components. See "Contents of the Application Server Subdirectory (SASApp or SASMain)" on page 317.
`SASMeta`	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 315.
`SASMain`	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 315 and "Contents of the Application Server Subdirectory (SASApp or SASMain)" on page 317.
`ShareServer`	Contains the management script, configuration information, and log files for the SAS/SHARE server.

Subdirectory or File	Description
Utilities	Contains XML files that are used as input to the SAS Deployment Wizard and utilities that are associated with this configuration instance.
Web	See "Contents of the Web Subdirectory" on page 318.
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 61.
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 and SAS/SHARE server.

Contents of the Metadata Server Subdirectory (SASMeta)

On the machines where the metadata server is installed, the **Lev***n* 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:

- **SASEnvironment** and **Data** subdirectories that can be used by the workspace server.
- 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 reserved for use by metadata utilities.
- several files that control the configuration of the SASMeta application server. For details, see "Reference: Configuration Files for SAS Servers" on page 321.
- 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 322.

Table 23.2 *Contents of the MetadataServer Subdirectory of SASMeta*

Subdirectory or File	Description
Journal	Contains the data set for the metadata server journal, if JOURNALTYPE="SINGLE" has been specified in omaconfig.xml. For details, see "Configuring Metadata Server Journaling" on page 190.

Subdirectory or File	Description
Logs	Is the default location for metadata server logs.
MetadataRepositories	Contains subdirectories for each metadata repository. These subdirectories contain the SAS data sets that contain your metadata.
rposmgr	Contains data sets for the repository manager.
sasuser	Is a SAS library that can be used to store SAS catalogs that enable you to customize certain features of SAS to meet specific needs. SAS assigns the SASUSER library at invocation.
adminUsers.txt, trustedUsers.txt, and trustedPeers.xml	Contain security-related configuration information. See "About the Security-Related Configuration Files" on page 24.
backupServer.sas	Is invoked by MetadataServer.bat when the backup option is used. This program uses the metadata server backup facility to back up the metadata server repositories, repository manager, and server configuration information. For details, see "Backing Up and Recovering the SAS Metadata Server" on page 167.
logconfig.xml	Specifies the logging configuration for the metadata server. For details, see "Administering Logging for SAS Servers" on page 119.
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.
MetadataServerBackupConfiguration.xml	Contains the configuration options that are in effect for the metadata server backup facility.
MetadataServerBackupHistory.xml	Contains summary information about each metadata server backup and recovery that has been performed.
MetadataServerBackupManifest.xml	Contains detailed information about the most recent metadata server backup. *Note:* Do not delete or move this file from the metadata server directory. Doing so will prevent the server from starting successfully.
MetadataServerRecoveryManifest.xml	Contains detailed information about the most recent metadata server recovery. The file exists only if a recovery operation has been performed.
MetadataServerSSCU.ini	Configures the metadata server service in the Windows Services Manager.
MetadataServer_shortcuts.ini	Configures shortcuts for the Windows Start menu.

Subdirectory or File	Description
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.
optimizeIMDB.sas	Is invoked by MetadataServer.bat or MetadataServer.sh when the optimizeIMDB option is used. See "Using optimizeIMDB to Optimize and Report on Metadata Server Memory Usage" on page 197.
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 **`Lev`***n* 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 the SAS Deployment Wizard creates additional directories 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 reserved for specialized use.

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 321.
- 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 321. For details about start-up scripts, see "Operating Your Servers" on page 54.

- 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: Middle-Tier Administration Guide* and the *SAS Intelligence Platform: Web Application Administration Guide*.

Chapter 24
Default Locations for Server Logs

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 24.1 *Location of Server Logs*

Server	Log Location within the *SAS-configuration-directory*
SAS Framework Data Server	Windows: **Lev1\FrameworkServer\Logs** UNIX or z/OS: **Lev1/FrameworkServer/Logs**
SAS Metadata Server	Windows: **Lev1\SASMeta\MetadataServer\Logs** UNIX or z/OS: **Lev1/SASMeta/MetadataServer/Logs**
SAS object spawner	Windows: **Lev1\ObjectSpawner\Logs** UNIX or z/OS: **Lev1/ObjectSpawner/Logs**
SAS OLAP Server	Windows: **Lev1\SASApp\OLAPServer\Logs** UNIX or z/OS: **Lev1/SASApp/OLAPServer/Logs**

Server	Log Location within the *SAS-configuration-directory*
SAS Pooled Workspace Server	Windows: **Lev1\SASApp\PooledWorkspaceServer\Logs** UNIX or z/OS: **Lev1/SASApp/PooledWorkspaceServer/Logs**
SAS Stored Process Server	Windows: **Lev1\SASApp\StoredProcessServer\Logs** UNIX or z/OS: **Lev1/SASApp/StoredProcessServer/Logs**
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: **Lev1\ShareServer\Logs** UNIX or z/OS: **Lev1/ShareServer/Logs**
SAS/CONNECT spawner	Windows: **Lev1\ConnectSpawner\Logs** UNIX or z/OS: **Lev1/ConnectSpawner/Logs**
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: **Lev1\SASApp\BatchServer\Logs** UNIX or z/OS: **Lev1/SASApp/BatchServer/Logs**

For more information about server logging, see “Administering Logging for SAS Servers” on page 119.

Chapter 25
Configuration Files

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 20. 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 321.

Table 25.1 Configuration Files for the SAS Metadata Server

Order of Precedence	Path and Filename	Purpose
1	Windows: **\Lev1\SASMeta\MetadataServer\sasv9.cfg** UNIX and z/OS: **/Lev1/SASMeta/MetadataServer/sasv9.cfg**	Specifies server start-up 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*****\SASFoundation\9.3\sasv9.cfg** UNIX: ***SAS-install-directory*****/SAS-install-directory/SASFoundation/9.3/sasv9.cfg** z/OS: Not applicable	Specifies system options that apply to all SAS sessions on the host machine. On UNIX systems, do not modify this file. If you need to make changes, modify the sasv9_local.cfg file that is in the same directory.
3	UNIX only: ***SAS-install-directory*****/SASFoundation/9.3/sasv9_local.cfg**	On UNIX systems, contains site-specific modifications to the sasv9.cfg file that is in the same directory.
4	Windows: ***SAS-install-directory*****\SASFoundation\9.3\*****locale*****\sasv9.cfg** UNIX: ***SAS-install-directory*****/SASFoundation/9.3/*****locale*****/sasv9.cfg** z/OS: **<HLQ>.V93D8*****ddd*****.CNTLDSN(BAT*****locale*****W0)**	Specifies system options that apply to all SAS sessions on the host machine in the specified locale.
5	Windows: **\Lev1\SASMeta\MetadataServer\sasv9_usermods.cfg** UNIX and z/OS: **/Lev1/SASMeta/MetadataServer/sasv9_usermods.cfg**	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 "Options for the Metadata Server Invocation Command" on page 339.
Not applicable	Windows: **\Lev1\SASMeta\MetadataServer\omaconfig.xml** UNIX and z/OS: **/Lev1/SASMeta/MetadataServer/omaconfig.xml**	Specifies options for the metadata server and metadata repositories. See "Reference Information for omaconfig.xml" on page 329.

Order of Precedence	Path and Filename	Purpose
Not applicable	Windows: `\Lev1\SASMeta\MetadataServer\logconfig.xml` UNIX and z/OS: `/Lev1/SASMeta/MetadataServer/logconfig.xml`	Specifies the logging configuration for the server. See “Administering Logging for SAS Servers” on page 119.
Not applicable	Windows: `\Lev1\SASMeta\MetadataServer\adminUsers.txt` UNIX and z/OS: `/Lev1/SASMeta/MetadataServer/adminUsers.txt`	Defines your system's initial unrestricted user. See “Metadata Server: Unrestricted Role” on page 26.
Not applicable	Windows: `\Lev1\SASMeta\MetadataServer\trustedUsers.txt` UNIX and z/OS: `/Lev1/SASMeta/MetadataServer/trustedUsers.txt`	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 28.
Not applicable	Windows: `\Lev1\SASMeta\MetadataServer\trustedPeers.xml` UNIX and z/OS: `/Lev1/SASMeta/MetadataServer/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. See the *SAS Intelligence Platform: Security Administration Guide*.
Not applicable	Windows: Not applicable UNIX and z/OS: `/Lev1/SASMeta/MetadataServer/MetadataServer.srv`	On UNIX and z/OS, contains information that is used by generate_boot_scripts.sh to regenerate the sas.servers script.

In z/OS environments, the following additional configuration files affect the metadata server:

Table 25.2 Additional Configuration Files for the SAS Metadata Server 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/SASMeta/MetadataServer/tkmvsenv.cfg`	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 tkmvsenv_usermods.cfg file that is in the same directory.

3	z/OS:
	`/Lev1/SASMeta/MetadataServer/tkmvsenv`
	`usermods.cfg`
		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 tkmvsenv.cfg file in the same directory. See the *Configuration Guide: SAS 9.3 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 321.

Table 25.3 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: `\Lev1\`***`server-context`***`\`***`server-name`***`\sasv9.cfg` UNIX and z/OS: `/Lev1/`***`server-context`***`/`***`server-name`***`/sasv9.cfg`	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: `\Lev1\`***`server-context`***`\sasv9.cfg` UNIX and z/OS: `/Lev1/`***`server-context`***`/sasv9.cfg`	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 sasv9_usermods.cfg file that is in the same directory. See "Overview of System Options for SAS Application Server Components" in Chapter 14 of *SAS Intelligence Platform: Application Server Administration Guide*.
3	Windows: ***`SAS-install-directory`***`\SASFoundation\9.3\sasv9.cfg` UNIX: ***`SAS-install-directory`***`/SASFoundation/9.3/sasv9.cfg` z/OS: Not applicable	Specifies system options that apply to all SAS sessions on the host machine. On UNIX systems, do not modify this file. If you need to make changes, modify the sasv9_local.cfg file that is in the same directory.
4	UNIX only: ***`SAS-install-directory`***`/SASFoundation/9.3/sasv9_local.cfg`	On UNIX systems, contains site-specific modifications to the sasv9.cfg file that is in the same directory.
5	Windows: ***`SAS-install-directory`***`\SASFoundation\9.3\`***`locale`***`\sasv9.cfg` UNIX: ***`SAS-install-directory`***`/SASFoundation/9.3/`***`locale`***`/sasv9.cfg` z/OS: `<HLQ>.V93D8`***`ddd`***`.CNTLDSN(BAT`***`locale`***`W0)`	Specifies system options that apply to all SAS sessions on the host machine in the specified locale.

Order of Precedence	Path and Filename	Purpose
6	Windows: **\Lev1\*server-context*\sasv9_usermods.cfg** UNIX and z/OS: **/Lev1/*server-context*/sasv9_usermods.cfg**	Contains site-specific modifications to the server start-up 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 "Overview of System Options for SAS Application Server Components" in Chapter 14 of *SAS Intelligence Platform: Application Server Administration Guide*.
7	Windows: **\Lev1\*server-context*\appserver_autoexec.sas** UNIX and z/OS: **/Lev1/*server-context*/appserver_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 appserver_autoexec_usermods.sas file that is in the same directory.
8	Windows: **\Lev1\*server-context*\appserver_autoexec_usermods.sas** UNIX and z/OS: **/Lev1/*server-context*/appserver_autoexec_usermods.sas**	Contains site-specific modifications to the appserver_autoexec.sas file that is located in the same directory.
9	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 start-up options. Use this file to change the options that are specified in the sasv9.cfg file in the same directory. See "Overview of System Options for SAS Application Server Components" in Chapter 14 of *SAS Intelligence Platform: Application Server Administration Guide*.
10	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.
11	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: UNIX and z/OS: **/Lev1/*server-context*/*server-name*/logconfig.xml**	Specifies the logging configuration for the server. See "Administering Logging for SAS Servers" on page 119.
Not applicable	Windows: **\Lev1\sasv9_meta.cfg** UNIX and z/OS: **/Lev1/sasv9_meta.cfg**	Specifies metadata server connection information for the SAS OLAP Server and SAS/SHARE server.
Not applicable	Windows: Not applicable UNIX and z/OS: **/Lev1/*server-context*/*server-name*/*server-name*.srv**	On UNIX and z/OS, contains information that is used by generate_boot_scripts.sh to regenerate the sas.servers script.

In z/OS environments, the following additional configuration files affect components of a SAS Application Server:

Table 25.4 *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: `<HLQ>.TKMVSENV(TKMVSENV)`	Contains name/value pairs (environment variables) that apply to all SAS sessions on the machine.
2	z/OS: **/Lev1/*server-context*/*server-name*/tkmvsenv.cfg**	Contains name/value pairs (environment variables) that apply to this 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-context*/tkmvsenv.cfg**	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 tkmvsenv_usermods.cfg file that is in the same directory.
4	z/OS: **/Lev1/*server-context*/tkmvsenv_usermods.cfg**	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 tkmvsenv.cfg file in the same directory. See the *Configuration Guide: SAS 9.3 Foundation for z/OS*.
5	z/OS: **/Lev1/*server-context*/*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: SAS 9.3 Foundation for z/OS*.

Configuration Files for SAS/SHARE Servers

The following table shows the configuration files that affect SAS/SHARE servers. For an explanation of each columns' contents, see "Introduction to the Configuration File Reference" on page 321.

Table 25.5 *Configuration Files for SAS/SHARE Servers*

Order of Precedence	Path and Filename	Purpose
1	Windows: `\Lev1\ShareServer\sasv9.cfg` UNIX and z/OS: `/Lev1/ShareServer/sasv9.cfg`	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.

Order of Precedence	Path and Filename	Purpose
2	Windows: `SAS-install-directory\SASFoundation\9.3\sasv9.cfg` UNIX: `SAS-install-directory/SASFoundation/9.3/sasv9.cfg` z/OS: Not applicable	Specifies system options that apply to all SAS sessions on the host machine. On UNIX systems, do not modify this file. If you need to make changes, modify the sasv9_local.cfg file that is in the same directory.
3	UNIX only: `SAS-install-directory/SASFoundation/9.3/sasv9_local.cfg`	On UNIX systems, contains site-specific modifications to the sasv9.cfg file that is in the same directory.
4	Windows: `SAS-install-directory\SASFoundation\9.3\locale\sasv9.cfg` UNIX: /SAS-install-directory/SASFoundation/9.3/*locale*/sasv9.cfg z/OS: `<HLQ>.V93D8ddd.CNTLDSN(BATlocaleW0)`	Specifies system options that apply to all SAS sessions on the host machine in the specified locale.
5	Windows: `\Lev1\ShareServer\sasv9_usermods.cfg` UNIX and z/OS: `/Lev1/ShareServer/sasv9_usermods.cfg`	Contains site-specific modifications to the server's start-up options. Use this file to change the options that are specified in the sasv9.cfg file in the same directory. See the *SAS/SHARE User's Guide*.
6	Windows: `\Lev1\ShareServer\autoexec.sas` UNIX and z/OS: `/Lev1/ShareServer/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.
7	Windows: `\Lev1\ShareServer\autoexec_usermods.sas` UNIX and z/OS: `/Lev1/ShareServer/autoexec_usermods.sas`	Contains site-specific modifications to the autoexec.sas file that is located in the same directory.
Not applicable	Windows: \Lev1\sasv9_meta.cfg UNIX and z/OS: `/Lev1/sasv9_meta.cfg`	Specifies metadata server connection information for the SAS OLAP Server and SAS/SHARE server.
Not applicable	Windows: `\Lev1\ShareServer\logconfig.xml` UNIX and z/OS: `/Lev1//logconfig.xml`	Specifies the logging configuration for the server. See "Administering Logging for SAS Servers" on page 119.
Not applicable	Windows: Not applicable UNIX and z/OS: `/Lev1/ShareServer.srv`	On UNIX and z/OS, contains information that is used by generate_boot_scripts.sh to regenerate the sas.servers script.

In z/OS environments, the following additional configuration files affect SAS/SHARE servers:

Table 25.6 Additional Configuration Files for SAS/SHARE Servers

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/ShareServer/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/ShareServer/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: SAS 9.3 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 321.

Table 25.7 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 *Communications Access Methods for SAS/CONNECT and SAS/SHARE*.
1	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 Chapter 12 of *SAS Intelligence Platform: Application Server Administration Guide*.

Order of Precedence	Path and Filename	Purpose
2	Windows: `\Lev1\ObjectSpawner` \ObjectSpawner_usermods.bat UNIX: `/Lev1/ObjectSpawner/ObjectSpawner_usermods.sh` z/OS: Not applicable	Can be used to specify site-specific modifications to the spawner's start-up options. Sites that need to set JREOPTIONS on the spawner can uncomment parameters that are included in this file.
Not applicable	Windows: `\Lev1\`***`spawner-name`***`\logconfig.xml` UNIX and z/OS: `/Lev1/`***`spawner-name`***`/logconfig.xml`	Specifies the logging configuration for the spawner. See "Administering Logging for SAS Servers" on page 119.
Not applicable	Windows: Not applicable UNIX and z/OS: `/Lev1/`***`server-or-spawner-name`***`.srv`	On UNIX and z/OS, contains information that is used by generate_boot_scripts.sh to regenerate the sas.servers 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)
- the e-mail addresses to which alert e-mails are to be sent
- 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 "Operating Your Servers" on page 54.

CAUTION:

Stopping the metadata server stops other components.

The file can contain the following XML elements:

<OMA>
: specifies options for the metadata server. See "<OMA> Options" on page 330.

<RPOSMGR>
: specifies options for the repository manager.

<InternalAuthenticationPolicy>
: specifies options for the assignment and management of passwords for internal users. Internal accounts are accounts that exist only in metadata. See "SAS Internal Authentication" in Chapter 10 of *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 in the event of a metadata server backup error, a metadata server recovery error, or an error that prevents the repository data sets from being updated from the journal.

To specify more than one address, enclose each address in single quotation marks, place a blank space between each address, and enclose the list in parentheses. For example: **"('Bill@mycompany.com' 'Susan@mycompany.com')"**

Interaction: For the ALERTEMAIL 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. If your e-mail server requires authentication, the appropriate information must be provided in EMAILAUTHPROTOCOL, EMAILID and EMAILPASSWORD. See "Options for the Metadata Server Invocation Command" on page 339.

Tip: If you modify any of the e-mail options, be sure to test metadata server e-mail alerts. See "Testing E-mail Alerts for the SAS Metadata Server" on page 192.

See: "Managing Alert E-mail Options for the SAS Metadata Server" on page 191 for more information.

ALERTEMAILTEST="*text*"
: enables you to verify that the ALERTEMAIL option is set correctly. ALERTEMAILTEST is a temporary omaconfig.xml option that causes a test message to be sent at server start up in order to verify that the e-mail addresses specified in the ALERTEMAIL option 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 text string

with this option. Using the option without a text string can also prevent the e-mail from being sent.

Note: In SAS 9.3 and later releases, the **Send Test Message** button in SAS Management Console is the preferred method for testing alert e-mails. This feature enables you to test alert e-mails without stopping the metadata server. The button is located on the Active Server Properties dialog box in the Metadata Manager plug-in. See "Testing E-mail Alerts for the SAS Metadata Server" on page 192.

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.

JOURNALPATH="*path-and-filename*"
specifies a location to create the journal file if JOURNALTYPE="SINGLE" has been specified. (If JOURNALTYPE is set to NONE or ROLL_FORWARD, then the JOURNALPATH option is ignored.)

Note: The settings JOURNALTYPE="SINGLE" and JOURNALTYPE="NONE" are not recommended.

When specifying a location, keep these things in mind:

- For consistency, we recommend that you specify the filename **`MetadataServerJournal.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.
- 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.

Default: If you specify JOURNALTYPE="SINGLE" but you do not specify a JOURNALPATH, then the journal is created in ***SAS-configuration-directory*/Lev1/SASMeta/MetadataServer/Journal/MetadataServerJournal.dat**.

Interaction: This option is ignored if OMA JOURNALTYPE is set to NONE or ROLL_FORWARD.

JOURNALSIZE="*number-of-bytes*"

specifies a size, in bytes, for the journal file if JOURNALTYPE="SINGLE" has been specified. (If JOURNALTYPE is set to NONE or ROLL_FORWARD, then the JOURNALSIZE option is ignored.)

Note: The settings JOURNALTYPE="SINGLE" and JOURNALTYPE="NONE" are not recommended.

The space that you assign is reserved in the file system and is unavailable to other processes, whether the journal file is full or not. Journal entries are appended to the file until the file grows to the size that is specified. When the file reaches the specified size, subsequent journal entries overwrite previously written entries starting at the beginning of the file.

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.`**

Default: If you specify JOURNALTYPE="SINGLE" but you do not specify a JOURNALSIZE, then the default size is used, as follows: 200MB on 32–bit machines and z/OS machines, and 500MB on 64–bit machines.

Interaction: This option is ignored if OMA JOURNALTYPE is set to NONE or ROLL_FORWARD.

JOURNALTYPE="NONE | SINGLE | ROLL_FORWARD"

specifies whether metadata server journaling is enabled and, if so, which type of journal file is created. 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 applied all of the updates from the journal file, the server recovers the updates from the journal file when it is restarted.

You can specify one of the following journal types:

NONE

(Not recommended) disables the journaling feature so that no journal file is created. The JOURNALPATH and JOURNALSIZE options are ignored. The roll-forward option is not available when the metadata server is recovered from a backup. In addition, the metadata server cannot be backed up while the server is in an ONLINE state. The server will pause itself to a READONLY state each time a backup is executed, and any update transactions that are issued during the backup will fail.

SINGLE

(Not recommended) enables the journaling feature by creating a single, circular journal file of a fixed size at the location that is specified in JOURNALPATH. Journal entries are appended to the file until the file grows to the size that is specified in JOURNALSIZE. When the file reaches the JOURNALSIZE limit, subsequent journal entries overwrite previously written entries starting at the beginning of the file. The roll-forward recovery option is not supported because there is no guarantee that all transactions that occurred since the last backup will still be available when a recovery is executed.

ROLL_FORWARD
: enables the journaling feature by creating a linear journal file that permanently stores all transactions that have occurred since the most recent backup. The journal file is written to the same location as the associated backup files. Each time a new backup is executed, journaling stops and a new journal file is started in the new backup location. The JOURNALPATH and JOURNALSIZE options are ignored. If ROLL_FORWARD is specified, the roll-forward option is available when the metadata server is recovered from a backup. If you change the option setting to ROLL_FORWARD from NONE or SINGLE, a metadata server backup is automatically performed when the server is restarted.

Default: The SAS Deployment Wizard sets this option to ROLL_FORWARD during installation or during migration from SAS 9.1.3 or 9.2.

Interaction: If ROLL_FORWARD or NONE is specified, then JOURNALPATH and JOURNALSIZE are ignored.

Tip: 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.

See: "Configuring Metadata Server Journaling" on page 190

CAUTION: If you want to be able to use the roll-forward option in the event of a metadata server recovery, then make sure that a JOURNALTYPE of ROLL_FORWARD is specified.

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 188.

Default: 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 113.

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 113.

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**.

The initial setting for this parameter is 1.

Notes:

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.

For a few solutions desktop clients (for example, SAS Model Manager, SAS Enterprise Miner, and SAS Forecast Studio), the ability to store credentials in client-side connection profiles is instead controlled by the Policy.AllowClientPasswordStorage property. To access this property, open the **Plug-ins** tab of SAS Management Console and navigate to **Application Management** ⇨ **Configuration Manager** ⇨ **SAS Application Infrastructure** ⇨ **Settings** ⇨ **Policies** ⇨ **Allow client password storage**.

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*.

The initial setting for this parameter is SAS002.

Default: If no value is specified for this parameter and SAS/SECURE is not installed, the SAS002 encoding method is used. If no value is specified for this parameter and SAS/SECURE is installed, the SAS003 encoding method is used.

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*.

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.

The initial setting for this parameter is SAS002.

Default: If no value is specified for this parameter and SAS/SECURE is not installed, the SAS002 encoding method is used. If no value is specified for this parameter and SAS/SECURE is installed, the SAS003 encoding method is used.

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.

Default: 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 start-up 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 "SAS Internal Authentication" in Chapter 10 of *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.

Default: 0

Range: 0–1440

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.

Default: 0

Range: 0–32767

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`**.

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.

Default: T

Interaction: This option is valid only if Expiration Days is greater than 0.

MinLength="*number-of-bytes*"
specifies the minimum allowed length for the passwords of internal users.

Default: 6

Range: 1–32

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.

Default: 5

Range: 0–5

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.

Default: 0

Range: 0–32767

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="number-of-failures"” on page 338.)

Default: 60

Range: 1–2^{31}

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.

Default: 3

Range: 0–100

Chapter 26
Options for the Metadata Server Invocation Command

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 340
- "Optional Parameters" on page 343

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 following parameters which are used for generating e-mail alerts for the metadata server:

  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 SMPT server that you specified for the EMAILHOST option.

  EMAILID "*server-email-address* | *server-user-id*"
  specifies the e-mail address that is to appear in the **From** field for alert e-mail messages. By default, the SAS Deployment Wizard places the following value in this parameter: **`SASMeta - Metadata Server (Config - Lev1) <noreply@mailhost.fyi.sas.com>`**.

  If you want to specify a different e-mail address, use one of the following formats:

  - *"server-name<user-account@domain>"*
  - "*<user-account@domain>*"

  Note:

  - Some electronic mail systems do not allow semicolons or colons in the **From** field.

  - If your e-mail server requires user authentication, then this option specifies the user ID that is to be used for authentication when an e-mail alert is generated. See "Optional Parameters" on page 343.

  TIP If you modify any of the e-mail options, be sure to test metadata server e-mail alerts. See "Testing E-mail Alerts for the SAS Metadata Server" on page 192.

  Note:

  - If your mail server requires authentication, then additional parameters are needed. See "Optional Parameters" on page 343.
  - For more information about metadata server alerts, see "Managing Alert E-mail Options for the SAS Metadata Server" on page 191.

- 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 Chapter 14, “System Options for SAS Application Server Components,” in *SAS Intelligence Platform: Application Server Administration Guide*.
- If you need to make any changes to the OBJECTSERVERPARMS option, make sure that the sasv9_usermods.cfg file contains the entire list of parameters that you want to specify for this option. (The list of object server parameters that you specify in sasv9_usermods.cfg overrides the list that is specified in sasv9.cfg.
- For details about the other options, see the *SAS System Options: Reference*.

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 340:

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 340:

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 following parameters, which might be needed to generate e-mail alerts for the metadata server:

EMAILID "*server-email-address* | *server-user-id*"
specifies the user ID that is to be used for authentication when an e-mail alert is generated. Use this option for this purpose only if your e-mail server requires user authentication.

EMAILPW "*server-email-password*"
specifies the password that is associated with the user ID that is specified in EMAILID. Specify this option only if your e-mail server requires user authentication. The password should be encrypted using SAS proprietary 32-bit encryption. To obtain the encrypted password, use PROC PWENCODE.

Note: If you provide this password in the SAS Deployment Wizard, then the wizard automatically encrypts it. To update this password, you can use the Update Passwords feature of the SAS Deployment Manager. See "Update a Managed Password" in Chapter 4 of *SAS Intelligence Platform: Security Administration Guide*.

EMAILAUTHPROTOCOL LOGIN | NONE
specifies the protocol that is to be used to authenticate the user to the e-mail server. If your e-mail server requires user authentication, specify a value of LOGIN. Otherwise, omit this option or specify the default value (NONE).

TIP If you modify any of the e-mail options, be sure to test metadata server e-mail alerts. See "Testing E-mail Alerts for the SAS Metadata Server" on page 192.

Note: For more information about metadata server alerts, see "Managing Alert E-mail Options for the SAS Metadata Server" on page 191.

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 Chapter 10 of *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.

By default, the server uses this formula to compute the minimum number of threads:

```
MAX(5,((number-of-processors * 2) + 1))
```

This means that the value is set to either 5 or to ((*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 AUTHPROVIDERDOMAIN option, see the *SAS System Options: Reference*.

Chapter 27
Using the SAS Deployment Manager to Update Host Name References

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 351. 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 “Automating the SAS Installation on Multiple Machines” in Chapter 5 of *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 361.

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:

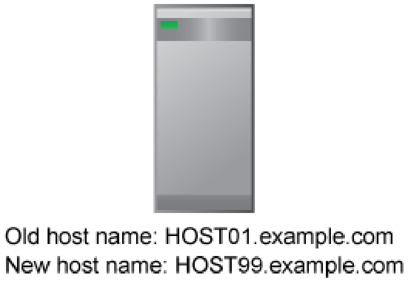
The Update Host Name References tool is not supported by certain SAS products and solutions. Refer to SAS Installation Note 39344 to determine whether your installed products and solutions are supported.

When to Use the Update Host Name References Tool

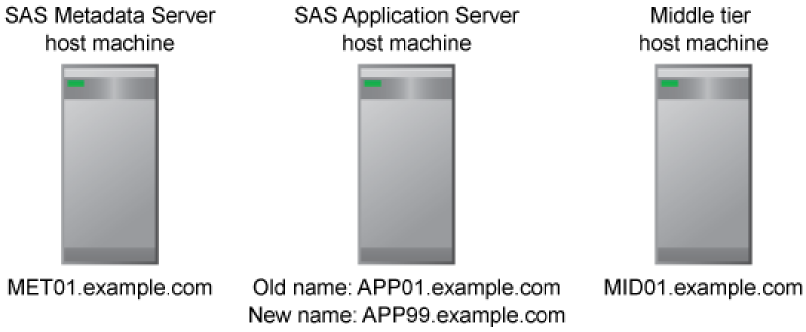
Use Case 1: An Existing Host Machine Name or Network Domain Name Has Changed

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:

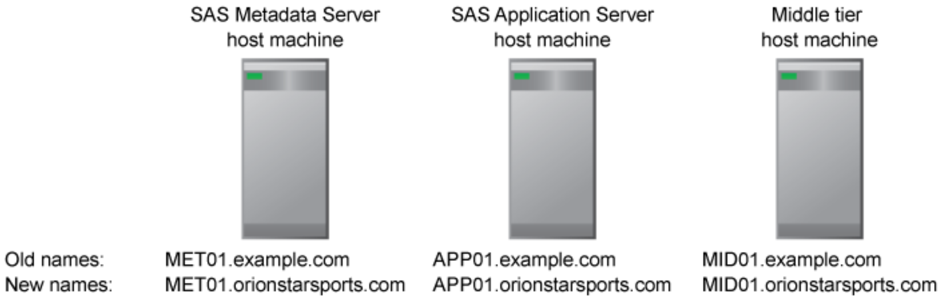
- The name of the machine on which a single-machine deployment is installed has changed, as shown in the following example. You can use the Update Host Name References tool to change all occurrences of the machine name in the deployment to the new name.


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
SAS Application Server host machine
Middle tier host machine
MET01.example.com
Old name: APP01.example.com
New name: APP99.example.com
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.


SAS Metadata Server host machine
SAS Application Server host machine
Middle tier host machine
Old names:
New names:
MET01.example.com
MET01.orionstarsports.com
APP01.example.com
APP01.orionstarsports.com
MID01.example.com
MID01.orionstarsports.com

Use Case 2: One or More Servers Have Been Cloned to New Machines

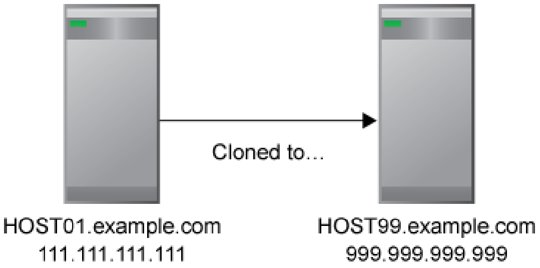
The Update Host Name References tool can be used when you have created an exact clone of one or more SAS servers or middle-tier machines on another machine or machines with different host names. The following examples illustrate this use case:

- 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.

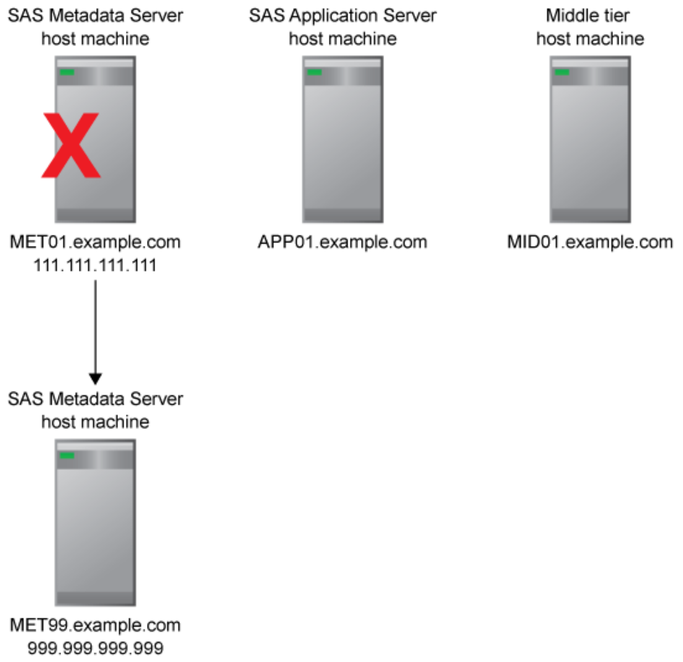
To create the clone, you must use cloning, ghosting, or virtualization software to transfer full-system images. On Windows, use a utility such as Norton Ghost from Symantec. On UNIX, use the cloning tool that is appropriate for your flavor of UNIX. After cloning the machine, you can use the Update Host 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:

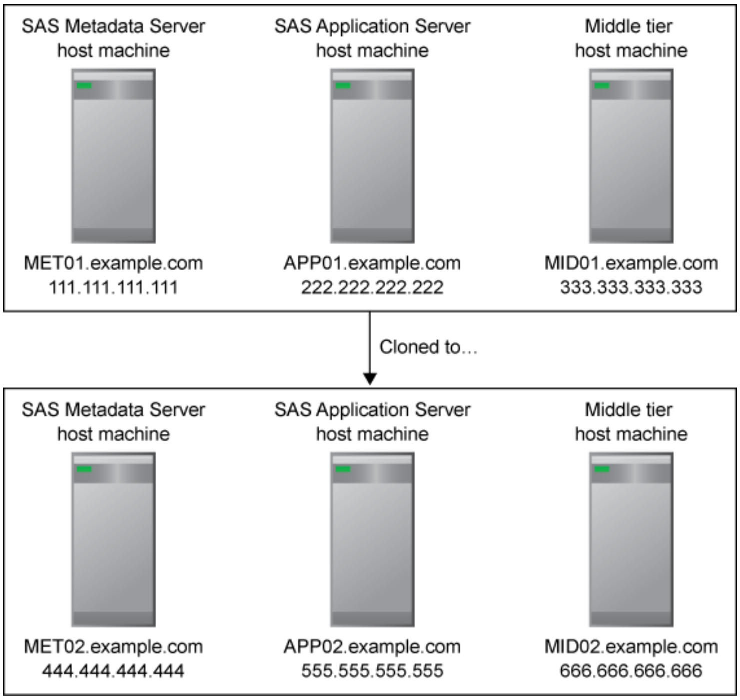
- A single-machine deployment has been cloned to another machine, as shown in the following example. You can use the Update Host Name References tool in the new deployment to update all occurrences of the host name to the new host name.



- 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.


SAS Metadata Server host machine
SAS Application Server host machine
Middle tier host machine
X
MET01.example.com 111.111.111.111
APP01.example.com
MID01.example.com
SAS Metadata Server host machine
MET99.example.com 999.999.999.999

- 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.


SAS Metadata Server host machine
SAS Application Server host machine
Middle tier host machine
MET01.example.com 111.111.111.111
APP01.example.com 222.222.222.222
MID01.example.com 333.333.333.333
Cloned to...
SAS Metadata Server host machine
SAS Application Server host machine
Middle tier host machine
MET02.example.com 444.444.444.444
APP02.example.com 555.555.555.555
MID02.example.com 666.666.666.666

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 347. 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.

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.

Table 27.1 *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.	"Using the SAS Deployment Manager to Update Host Name References" on page 346
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 Chapter 10 of *SAS Intelligence Platform: Application Server Administration Guide*
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.	"Promotion Tools Overview" on page 237 and SAS Management Console Help
SAS Migration Utility	Populate a SAS deployment with metadata, associated physical data, and configuration information from an existing deployment.	"Performing SAS 9.3 to SAS 9.3 Migration" on page 373

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 361.

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 start-up 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.

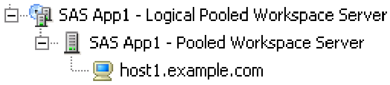
    These updates affect some of the previously listed files, as well as the following additional files:

    - adminUsers.txt and trustedUsers.txt, which are used by the metadata server.
    - sasv9_meta.cfg, which is used by the SAS OLAP Server 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.

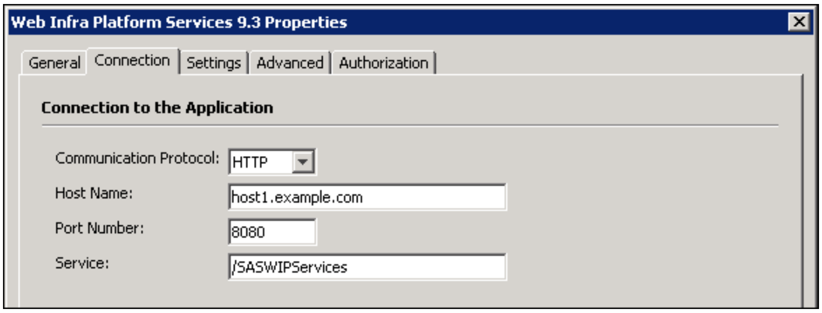
    - configuration files for the Stored Process Web application, if your configuration includes the middle tier. The WAR and EAR files for the application are then rebuilt. The Stored Process Web application is redeployed automatically if you use JBoss as your Web application server and if you selected 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:



SAS App1 - Logical Pooled Workspace Server
SAS App1 - Pooled Workspace Server
host1.example.com

  - 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:

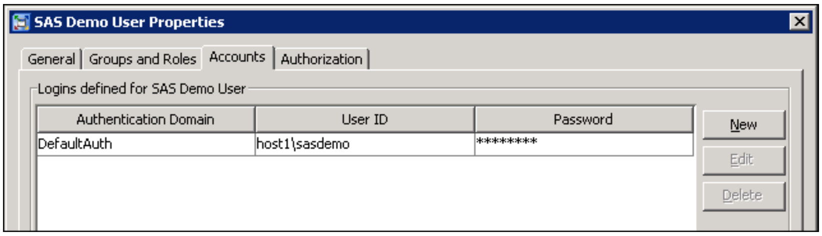


Web Infra Platform Services 9.3 Properties

Connection to the Application

Communication Protocol: HTTP
Host Name: host1.example.com
Port Number: 8080
Service: /SASWIPServices

  - 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:



Logins defined for SAS Demo User

Authentication Domain	User ID	Password
DefaultAuth	host1\sasdemo	********

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 re-creates 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 re-created.

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 customized 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 351). 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 361 .

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. Update host name references in Platform Suite for SAS, if applicable.
9. Start your servers and perform validation tasks.
10. 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, verify 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 “Best Practices for Backing Up and Restoring Your SAS Content” on page 155.

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 351. 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 365.

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.

   CAUTION:

   > **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. As a best practice, you should either stop or pause the old metadata server.** You will be able to start the metadata server 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 361.

3. Log on to the machine with the same user account that was used to deploy SAS 9.3. (On Windows, the user account must have Administrator rights.) Start the SAS Deployment Manager on the machine by navigating to *SAS-installation-directory*`/SASDeploymentManager/9.3` and launching **`sasdm.exe`** (Windows), **`sasdm.sh`** (UNIX) or **`sasdm.rexx`** (z/OS).

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 354.

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:

Table 27.2 *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 **Perform DNS lookup**.
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 **Perform DNS lookup**.
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 **Perform DNS lookup** (which is strongly recommended), then you do not need to enter the machines' IP addresses or short names.	As a best practice, select **Perform DNS lookup**. Otherwise, you must enter all forms of the machine names.

For example entries, see "Example Scenarios for Updating Host Name References" on page 365.

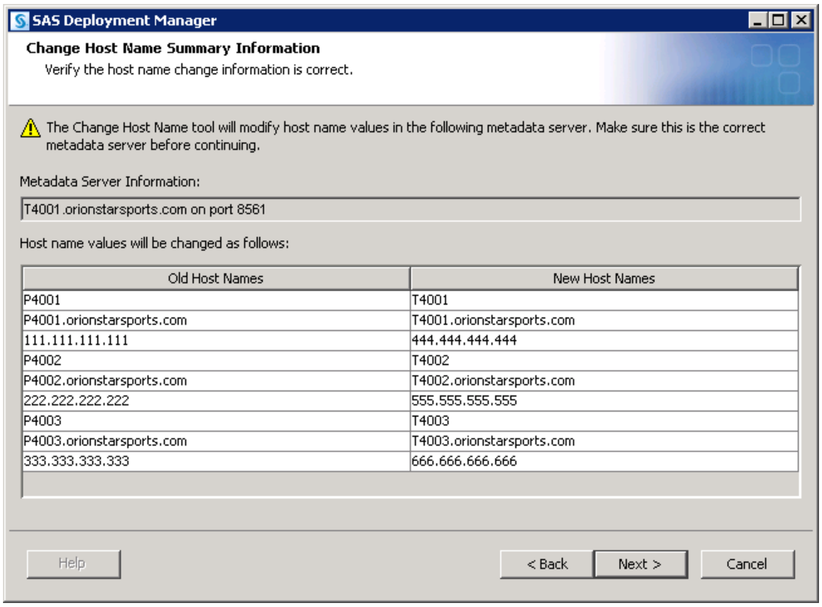
If you select **Perform DNS lookup**, then the lookup is performed when you click **Next**. The results of the lookup appear on the next page (Update Host Name References Summary).

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 354. 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:



Change Host Name Summary Information

Verify the host name change information is correct.

The Change Host Name tool will modify host name values in the following metadata server. Make sure this is the correct metadata server before continuing.

Metadata Server Information:

T4001.orionstarsports.com on port 8561

Host name values will be changed as follows:

Old Host Names	New Host Names
P4001	T4001
P4001.orionstarsports.com	T4001.orionstarsports.com
111.111.111.111	444.444.444.444
P4002	T4002
P4002.orionstarsports.com	T4002.orionstarsports.com
222.222.222.222	555.555.555.555
P4003	T4003
P4003.orionstarsports.com	T4003.orionstarsports.com
333.333.333.333	666.666.666.666

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”, 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 364.

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 customized 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 351). 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 Chapter 2 of *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 Chapter 13, "Administering Multicast Options," in *SAS Intelligence Platform: Middle-Tier 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 Appendix 1, "Configuring the SAS Environment File," in *SAS Intelligence Platform: Middle-Tier 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.
- If you use Platform Suite for SAS for scheduling, update the host name references in this software as described in "Step 8: Update Host Names in Platform Suite for SAS".

Step 8: Update Host Names in Platform Suite for SAS

About Updating Host Names in Platform Suite for SAS

If you use Platform Suite for SAS for scheduling, the Update Host Name References tool automatically updates host names that exist in metadata for scheduling servers and grid resources. However, you must manually update the host name references in Platform Suite for SAS. The following topics provide instructions:

- "Update Host Names in Platform Suite for SAS: Windows Instructions"
- "Update Host Names in Platform Suite for SAS: UNIX Instructions"
- "Update Host Name References for Grid Resources"

Update Host Names in Platform Suite for SAS: Windows Instructions

On Windows systems, follow these steps to update host name references in Platform Suite for SAS:

1. Make sure that the Process Manager service and the LSF Cluster services have been stopped.
2. Update the host name references in the following files:
   - ***JS_TOP*\conf\js.conf**
   - ***JS_TOP*\conf\lsf.conf**
   - ***JS_TOP*\conf\lsf.cluster.*cluster_name***
   - ***JS_TOP*\conf\ego\cluster_name\kernel\ego.conf**
3. Update the host name references in the following environment variables:
   - LSF_ENVDIR
   - EGO_CONFDIR
4. In the Windows registry, update the host name references in the values of the following keys:
   - **HKEY_LOCAL_MACHINE\SOFTWARE\Platform Computing Corporation\EGO\EGO_CONFDIR**
   - **HKEY_LOCAL_MACHINE\SOFTWARE\Platform Computing Corporation\EGO\EGO_SHAREDTOP**
   - **HKEY_LOCAL_MACHINE\SOFTWARE\Platform Computing Corporation\LSF\LSF_ENVDIR**
5. Start the Process Manager service and the LSF services.

Update Host Names in Platform Suite for SAS: UNIX Instructions

On UNIX systems, follow these steps to update host name references in Platform Suite for SAS:

1. Make sure that the Process Manager service and the LSF Cluster services have been stopped.

   To stop the services, log on to the Process Manager server host either as Root or as the primary Process Manager administrator, and use the following commands:

```
jadmin stop
badmin hshutdown all
lsadmin resshutdown all
lsadmin limshutdown all
```

2. Update the host name references in the following files:
   - ***JS_TOP*/conf/js.conf**
   - ***JS_TOP*/conf/lsf.conf**
   - ***JS_TOP*/conf/lsf.cluster.*cluster_name***
   - ***JS_TOP*/conf/ego/*cluster_name*/kernel/ego.conf**
3. Update the host name references in the process manager and LSF environments, as follows:
   - If you are using the C shell (csh or tcsh), use the following scripts:
     - **source *JS_TOP*/conf/cshrc.js**
     - **source *JS_TOP*/conf/cshrc.lsf**
   - If you are using the Bourne shell (sh), the bash shell, or the Korn shell (ksh), use the following scripts:
     - **.*JS_TOP*/conf/profile.js**
     - **.*JS_TOP*/conf/profile.lsf**
4. Use the following commands to reconfigure the LSF cluster and restart the services:

```
jadmin start
lsadmin reconfig
badmin reconfig
badmin mbdrestart
```

Update Host Name References for Grid Resources

To update host names in grid resource specifications, follow these steps:

1. Log on to the grid control machine as the LSF administrator.
2. Issue the following command:

   ```
   addresource -r resource_name -m machine-name
   ```

   In *machine-name*, specify the new host name. If the name contains spaces, you must change the spaces to underscores.
3. Run the LSF commands to reconfigure the grid to recognize the new resource.

See Also

"Defining and Specifying Resources" in *Grid Computing in SAS*

Step 9: 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:

- "Operating Your Servers" on page 54
- "Validate the SAS 9.3 Servers" in Chapter 5 of *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 10: 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 "Best Practices for Backing Up and Restoring Your SAS Content" on page 155.

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*`/Lev`*n*`/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, Windows 7, and Windows Server 2008 systems: `C:\Users\`*user*`\AppData\Local\SAS\SASDeploymentWizard\9.3`
- On other Windows systems: `C:\Documents and Settings\`*user*`\Local Settings\Application Data\SAS\SASDeploymentWizard\9.3`
- On UNIX systems: *user's-home-directory*`/.SASAppData/SASDeploymentWizard/9.3`

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:

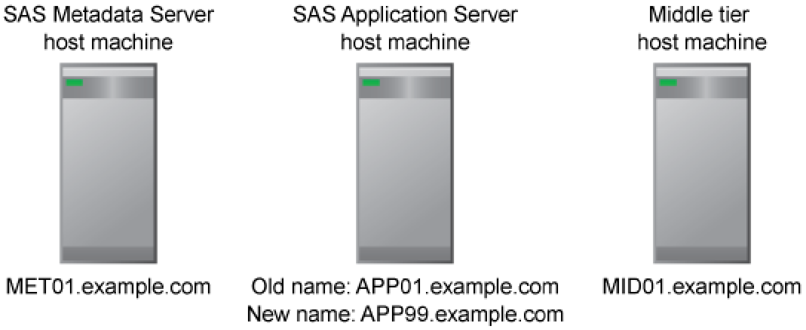
1. Create a complete system backup of the machine, including a correct backup of the metadata server.
2. Start the metadata server and stop all of the other servers.
3. Run the Update Host Name References tool on the machine.

   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.



In this scenario, you would use the Update Host Name References tool as follows:

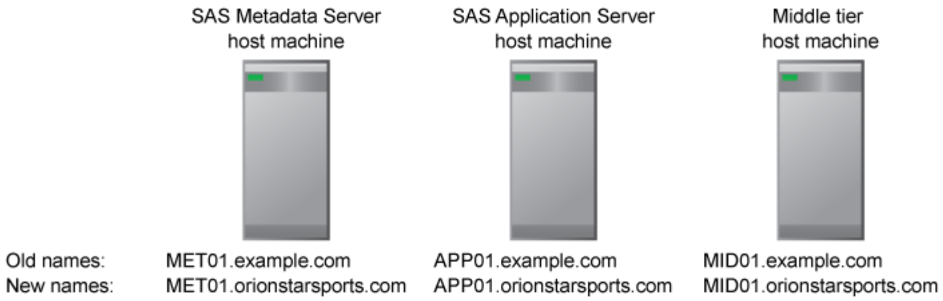
1. Create a complete system backup of each machine in the deployment, including a correct backup of the metadata server.
2. Start the metadata server, and stop all of the other servers.
3. Run the Update Host Name References tool on MET01.example.com.

   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.


SAS Metadata Server host machine
SAS Application Server host machine
Middle tier host machine
Old names:
New names:
MET01.example.com
MET01.orionstarsports.com
APP01.example.com
APP01.orionstarsports.com
MID01.example.com
MID01.orionstarsports.com

In this scenario, you would use the Update Host Name References tool as follows:

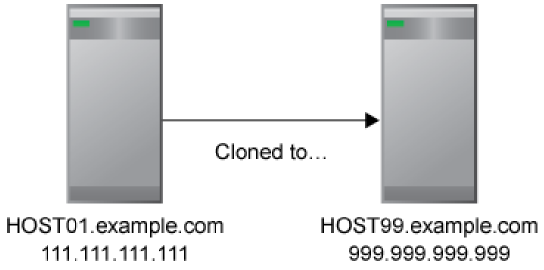
1. Create a complete system backup of each machine in the deployment, including a correct backup of the metadata server.
2. Start the metadata server, and stop all of the other servers.
3. Run the Update Host Name References tool on MET01.orionstarsports.com.

   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:

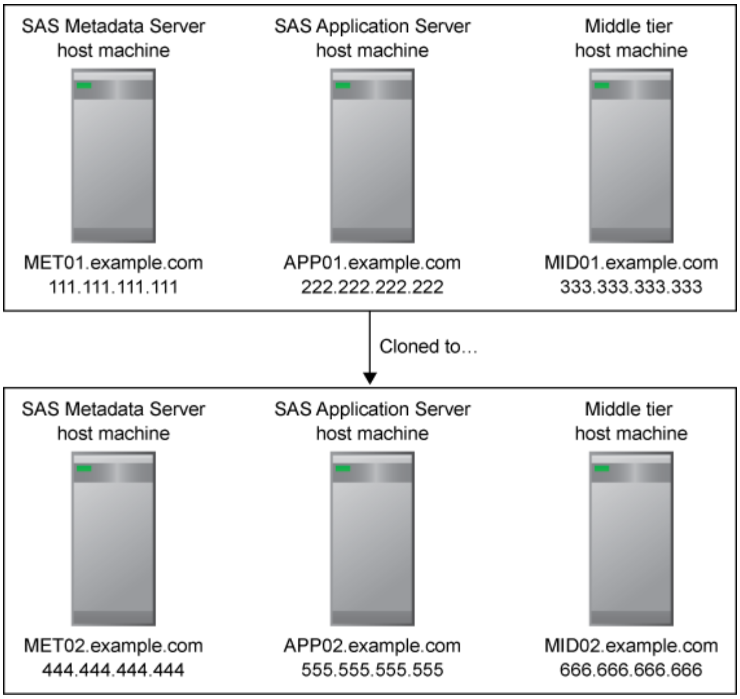
1. Create a complete system backup of HOST01.example.com, including a correct backup of the metadata server.
2. Set up the new machine (HOST99.example.com) by using cloning, ghosting, or virtualization software. For details, see "Step 3: Change Existing Names at the Operating System Level, or Set Up New Machines to Be Identical to the Current Machines" on page 355.
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 355.
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.

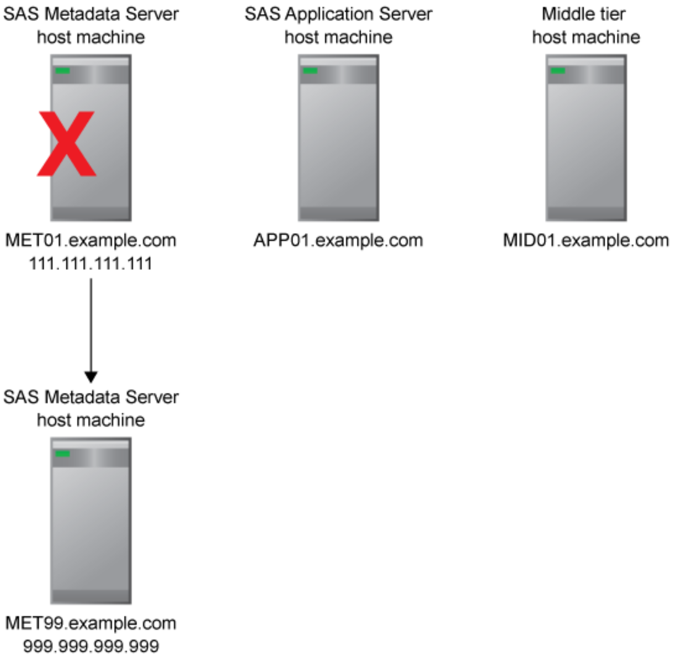
   In the Specify Host Names dialog box, you need to enter only the fully qualified host names. The DNS lookup identifies all other forms of the old and new host names so that they can be updated correctly. Specify the following:

   - **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 355.
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.

Chapter 28

Performing SAS 9.3 to SAS 9.3 Migration

About Using the SAS Migration Utility to Copy an Existing Deployment

You can use the SAS Migration Utility along with the SAS Deployment Wizard to create a second SAS 9.3 deployment that contains the same SAS content and the same configuration as an existing SAS 9.3 deployment. This capability is useful in the following situations:

- You have created and verified a SAS deployment in a test environment, and you want to replicate that deployment in a production environment.
- You want to use the SAS deployment in your production environment as the basis for creating a deployment in a test environment or a development environment.
- You need to create a mirror of your SAS deployment for disaster recovery purposes.
- You want to move your SAS deployment to upgraded hardware.

The following limitations apply to the use of the SAS Migration Utility:

- The two deployments must have the same topology. That is, the SAS architecture must be deployed across physical machines in the same fashion.
- The SAS components must run on the same operating system family in both deployments. You can migrate within the Windows family (for example, Windows 32-bit to Windows 64-bit) or migrate within the UNIX family (for example, Solaris to HP-UX).

- You can use the SAS Migration Utility only for the initial replication of a deployment. The utility cannot be used to refresh or update an existing deployment. You can refresh the deployment as needed by using the promotion tools to copy specific content items from one deployment to another.

Be sure to use the correct version of the SAS Migration Utility. When you migrate from a SAS 9.3 deployment, use smu93_32.exe, smu93_x64.exe, smu93.sh, or smu93.zos.

About the SAS Migration Utility

The SAS Migration Utility is an Ant-based framework and set of SAS product extensions whose primary purpose is to create a package of your current SAS content—data and configuration—that the SAS Deployment Wizard will use when it installs and configures SAS 9.3. The migration utility framework consists of a two-phase process: an analyze phase, and a package phase.

In the analyze phase, the migration utility locates the SAS content required for migration, performs a validation to ensure the content can be reliably migrated, and saves the required information for the migration utility package phase. During the analyze phase, the migration utility also generates a migration analysis report that you can use to inventory and enumerate those SAS products found on your current SAS system to be eligible for automatic migration with the deployment wizard.

In the package phase, the migration utility searches the system and copies SAS content into a migration package that will be used by the deployment wizard as it installs and configures SAS 9.3. For a description of the content that is migrated, see “What Can Be Migrated?” in Chapter 1 of *SAS Intelligence Platform: Migration Guide*.

Overview of the SAS 9.3 to SAS 9.3 Migration Process

The SAS 9.3 to SAS 9.3 migration process includes the following steps:

- “Step 1: Design Your Migration”
- “Step 2: Perform Pre-migration Tasks”
- “Step 3: Run the SAS Deployment Wizard and Migrate Your SAS Content”
- “Step 4: Complete Manual Migration Steps”
- “Step 5: Validate the New Environment”
- “Step 6: Maintain and Update the New Environment”

Step 1: Design Your Migration

In this step, you determine how you are going to perform the migration. The following options are available:

- Migrate to the same machine as the source environment. You might want to use this option if you are creating a new environment for development or testing purposes.

If you choose this option, it is recommended that you use the same instance of installed software (SASHome) for both environments.

When you perform the migration, select a configuration level that is different from the level of the source environment. For example, if the source environment uses Lev1, select **Lev2** for the new environment. Under the configuration directory path, a new subdirectory will be created with the level number that you specify.

The new environment will have its own configuration files, metadata servers, and other SAS servers. Based on the level number, the SAS Deployment Wizard will automatically increment the port numbers for your SAS servers. In addition, each environment will have its own set of physical files (including files for tables, deployed jobs, and stored process source code) on the machine. The two environments can share the same Web application server, or each environment can have its own server.

- Migrate to a different machine.

  If you choose this option, a new instance of installed software is created on the new machine, along with a new configuration directory, new server instances, and copies of physical files. If you choose this option, it is not necessary to specify a different level number or to use different port numbers for servers.

Note: The procedures in this chapter are for a single-machine configuration. However, same-version migration is also supported for multiple-machine configurations. For guidance, see "Designing Your Migration" in Chapter 2 of *SAS Intelligence Platform: Migration Guide*.

See Also

"Designing Your Migration" in Chapter 2 of *SAS Intelligence Platform: Migration Guide*

Step 2: Perform Pre-migration Tasks

Pre-migration tasks include backing up the source environment, preparing the new machine for the migration (when applicable), and creating the migration package. Follow these steps to prepare for the migration:

1. Make a complete backup of the SAS 9.3 deployment from which you want to migrate (referred to as the source environment). See "Best Practices for Backing Up and Restoring Your SAS Content" on page 155.

2. If you are migrating to a different machine, use the appropriate pre-installation checklist to prepare the machine. Preparation tasks include installing required third-party software, creating required user accounts, and reviewing port numbers. For details, see the following topics:

   - "Completing the Pre-migration Checklists" in Chapter 3 of *SAS Intelligence Platform: Migration Guide*
   - "Installing Third-Party Software" in Chapter 3 of *SAS Intelligence Platform: Migration Guide*

3. Create the migration package. In this step, you use the SAS Migration Utility to analyze the source environment. After correcting any detected issues, you run the utility again to extract the source environment's content and configuration and create a migration package. Follow these steps:

a. Log on to the metadata server host machine in the source environment, and navigate to the directory that contains the SAS Migration Utility executables. By default, the executables are located in the SAS Software Depot in the path **`utilities/smu`**.

b. Open the utility template file (smu.properties.template) in a text editor. Edit the file to specify properties including the following:

   - fully qualified paths for the source environment's configuration directory (SMU.config.dir), the SAS executable (SMU.SASROOT), the source environment's SASHome (SMU.SASHOME), and an output directory for the migration package (SMU.Output.Dir).
   - the metadata server host name (SMU.host.metadata), administrative user ID (SMU.user), and administrative user password (SMU.password) to access the source environment's metadata server; or the name of a connection profile (SMU.profile) that contains this information. You must specify an unrestricted user (for example, sasadm@saspw).
   - the user ID (SMU.webinfpltfm.dbms.userid) and password (SMU.webinfpltfm.dbms.password) that can be used to open a JBDC connection to the SAS Web Infrastructure Platform database in the source environment. For JBoss Application Server, this information is stored in ***`JBoss-Home`*****`/SASServer1/deploy/SharedServices-ds.xml`**. For WebLogic Server and WebSphere Application Server, see the server's administrative console.

   Additional properties might also be needed. See "Develop a Common SAS Migration Utility Properties File" in Chapter 2 of *SAS Intelligence Platform: Migration Guide*.

c. Run the migration utility in analyze mode, using the following command from an operating system prompt or command line:

   - Windows 32-bit:

```
smu93_32 -analyze -properties absolute-pathname-to-properties-file -replace
```

   - Windows 64-bit:

```
smu93_64 -analyze -properties absolute-pathname-to-properties-file -replace
```

   - UNIX:

```
./smu93 -- -analyze -properties absolute-pathname-to-properties-file -replace
```

   - z/OS:

```
./smu93.zos -- -analyze -properties absolute-pathname-to-properties-file -replace
```

d. Check the migration utility log file (migrate.log), which is located in your migration package output directory. If the utility completed successfully, the following messages should appear at the end of the log:

```
[INFO] SMU Product analyses completed
[INFO] SMU No packaging was performed due to the mode setting
```

e. Check the Migration Analysis report (AnalysisReport.xml), which is located in the path ***`host-machine-name`*****`/Analysis Report`** within your migration package output directory. Red Xs indicate error conditions that must be addressed before the migration can be completed successfully. For an explanation of the messages in this report, go to **`http://support.sas.com/migration/utilitynotes`**.

f. Correct any errors that are reported in the migrate.log file or in AnalysisReport.xml, and then rerun the migration utility in analyze mode. Repeat this process until all error situations have been resolved.

g. Run the migration utility using the same command that you used in step 4, but without the **`-analyze`** option. The command creates the migration package. Check the migration utility log file (migrate.log) and the Migration Analysis report (AnalysisReport.xml) to verify that the migration package was created successfully.

See Also

"Performing Pre-migration Tasks" in Chapter 3 of *SAS Intelligence Platform: Migration Guide*

Step 3: Run the SAS Deployment Wizard and Migrate Your SAS Content

In this step, you run the SAS Deployment Wizard with the migration package as input. The wizard configures the target environment based on the source environment and loads your content to the target environment. Follow these steps:

1. Log on to the machine with the appropriate user ID. As a best practice, we recommend logging on with the same account that was used to install the source environment. On Windows, the account must be a member of the Administrators group. On UNIX, do not use **`root`**.
2. Go to the highest-level directory in the SAS Software Depot. From that directory, use one of the following commands to start the SAS Deployment Wizard:
   - Windows: **`setup.exe -record -deploy`**
   - UNIX: **`setup.sh -record -deploy`**
   - z/OS: **`setup.rexx -record -deploy`**
3. As you respond to the SAS Deployment Wizard prompts, follow the guidelines that are provided in the following tables. (These tables list only the pages that pertain to migrations.)

   For detailed instructions for using SAS Deployment Wizard for a migration, see "Install and Migrate SAS Interactively" in Chapter 4 of *SAS Intelligence Platform: Migration Guide*.

   If you are migrating to the same machine as the source environment, respond to the prompts as follows:

Wizard Page	How to Respond When Migrating to the Same Machine (Shared SASHome, Different Level Number)
Select Deployment Task	Select **Install SAS software**.
Select Deployment Type	Select **Perform a Planned Deployment** and **Configure SAS Software**. When migrating to the same machine, you do not need to select **Install SAS Software**, because the software is already installed on the machine.

Wizard Page	How to Respond When Migrating to the Same Machine (Shared SASHome, Different Level Number)
Specify Deployment Plan	Specify the same deployment plan that was used to create the source environment.
Specify Configuration Prompting Level	Select the same prompting level that was used to create the source environment (for example, **Typical**).
Specify Configuration Information	**Configuration Directory:** You can accept the default directory or specify a different directory. **Configuration Level:** Select a level that is different from the level of the source environment. For example, if the source environment is using Lev1, you might want to select **Lev2**. Under the configuration directory path, a new subdirectory will be created with the level number that you specify. Based on the level number, the SAS Deployment Wizard automatically increments the port numbers for your SAS servers. (See "Port Numbering in a Multiple-Level Environment" on page 310.)
Local Machine Name	Enter the fully qualified host name and short host name for the machine.
Migration Information	Select **Perform migration**, and then browse to the highest-level directory of the migration package that you created in "Step 2: Perform Pre-migration Tasks" on page 375.
Remap Ports	If you want to specify any port numbers that are different from the defaults, click the Port for Target cell in the appropriate row, and enter the new port number. Port numbers that appear in boldface indicate a potential port conflict. (In some cases, servers such as the SASApp and SASMeta logical workspace servers can share the same port.)
External Account: Installer	As a best practice, specify the same installer account that was used to create the source environment.
External Account: SAS Spawned Servers Account	If you want, you can specify the same SAS Spawned Servers account that is used by the source environment.
SAS Framework Data Server: Transactional Database Files	Edit the path so that it specifies the correct configuration directory (for example, change **Lev1** to **Lev2**).
Web Application Server: Managed Server	Enter a managed server name that corresponds to the new environment.
Web Application Server: Managed Server Ports	Add 100 or 200 to each port number so that the ports are unique to the new environment. For example, change 8080 to 8180, change 8443 to 8543, and change 1099 to 1199.
SAS Web Infrastructure Platform Scheduling Services Configuration Dependencies	Change the SAS Application Server Context from **SASMeta** to **SASApp**.
IP Multicast	Be sure to modify the multicast settings if you are migrating to the same machine as the source environment. The combination of multicast IPD address and multicast UDP port should be different for each SAS deployment and also different from those used by other multicast applications at your site. See "Designating Ports and Multicast Addresses" in Chapter 2 of *SAS Intelligence Platform: Installation and Configuration Guide*.
SAS Distributed In-Process Scheduling Server	Increment each port number so that the ports are unique to the new environment. For example, change 7541 to 7542, and change 7641 to 7642.

Wizard Page	How to Respond When Migrating to the Same Machine (Shared SASHome, Different Level Number)
SAS Content Server: Repository Directory	**Repository Directory:** Edit the path so that it specifies the correct configuration directory (for example, if you are migrating to Level 2, change **Lev1** to **Lev2**).
DataFlux Data Management Server	**DataFlux Data Management Server Name:** Specify a unique name for this server. For example, if you are creating a development environment, you could add `DEV` to the end of the server name.

If you are migrating to a different machine, respond to the prompts as follows:

Wizard Page	How to Respond When Migrating to a Different Machine (Same or Different Level Number)
Select Deployment Task	Select **Install SAS software**.
Select Deployment Type	Select **Perform a Planned Deployment**, **Install SAS Software**, and **Configure SAS Software**.
Specify Deployment Plan	Specify the same deployment plan that was used to create the source environment.
Specify Configuration Prompting Level	Select the same prompting level that was used to create the source environment (for example, **Typical**).
Specify Configuration Information	**Configuration Directory:** You can accept the default directory or specify a different directory. **Configuration Level:** You can use the same level number that is used in the source environment. This option is appropriate if you are creating a mirror of your SAS deployment for disaster recovery purposes, or if you are migrating your deployment to new hardware. If you are creating a separate test or development environment, you might want to specify a different level number. If you do so, the SAS Deployment Wizard automatically increments the port numbers for your SAS servers. See "Port Numbering in a Multiple-Level Environment" on page 310.
Local Machine Name	Enter the fully qualified host name and short host name for the machine.
Migration Information	Select **Perform migration**, and then browse to the highest-level directory of the migration package that you created in "Step 2: Perform Pre-migration Tasks" on page 375.
Remap Ports	If you want to specify any port numbers that are different from the defaults, click the Port for Target cell in the appropriate row, and enter the new port number. Port numbers that appear in boldface indicate a potential port conflict. (In some cases, servers such as the SASApp and SASMeta logical workspace servers can share the same port.)
External Account: Installer	As a best practice, specify the same installer account that was used to create the source environment.
External Account: SAS Spawned Servers Account	Specify the SAS Spawned Servers account that you created on this machine.

Wizard Page	How to Respond When Migrating to a Different Machine (Same or Different Level Number)
SAS Framework Data Server: Transactional Database Files	If you specified a different level number, then edit the path so that it specifies the correct configuration directory (for example, change **Lev1** to **Lev2**).
Web Application Server: Managed Server	You can accept the default server name.
Web Application Server: Managed Server Ports	You can accept the default ports.
SAS Web Infrastructure Platform Scheduling Services Configuration Dependencies	Change the SAS Application Server Context from **SASMeta** to **SASApp**.
IP Multicast	When migrating to a different machine, it is usually not necessary to modify the multicast settings. See "Designating Ports and Multicast Addresses" in Chapter 2 of *SAS Intelligence Platform: Installation and Configuration Guide*.
SAS Distributed In-Process Scheduling Server	You can accept the default port numbers.
SAS Content Server: Repository Directory	**Repository Directory:** If you specified a different level number, then edit the path so that it specifies the correct configuration directory (for example, change **Lev1** to **Lev2**).
DataFlux Data Management Server	**DataFlux Data Management Server Name:** You can accept the default server name.

4. When the wizard is finished, read the warnings and notices that appear in the instructions.html document.
5. On Windows systems, check to ensure that a unique set of Windows services has been created for the new environment. A separate set of services should appear for each level (for example, Lev1 and Lev2).

See Also

"Installing SAS 9.3 and Migrating Your SAS Content" in Chapter 4 of *SAS Intelligence Platform: Migration Guide*

Step 4: Complete Manual Migration Steps

In this step, you manually perform certain migration tasks that are not performed by the SAS Deployment Wizard. Refer to the following sources for information about manual steps that might be required to complete the migration:

- the instructions.html document for your new deployment
- "Overview of Performing Post-migration Tasks" in Chapter 5 of *SAS Intelligence Platform: Migration Guide*

The manual steps include tasks such as the following:

- Create new connection profiles for the new environment. For same-machine migrations, profile names should include the level number so that users will select the correct profile when logging on.
- Copy physical files such as data tables, deployed jobs, and stored process source files to the new environment. (These files are copied automatically only if they are stored in the configuration directory.)

  Then modify the path information for these files. Use SAS Management Console to modify path locations for libraries, stored process repositories, and deployment directories. Use SAS OLAP Cube Studio to modify physical paths for cubes. Use SAS Data Integration Studio to modify file references that are stored in jobs or job flows.
- Rebuild cubes if necessary.
- Manually apply any needed customizations to autoexec and configuration files for SAS servers.

Note: In a SAS 9.3 to SAS 9.3 migration, it is generally not necessary to run the Authorization Differences Reports.

For details about these and other manual tasks, see the following topics:

- "Server Tier Post-migration Tasks" in Chapter 5 of *SAS Intelligence Platform: Migration Guide*
- "Middle-Tier Post-migration Tasks" in Chapter 5 of *SAS Intelligence Platform: Migration Guide*

Step 5: Validate the New Environment

The final step is to validate the new environment to ensure that it is configured properly and is operational. For instructions, see "Validating Your SAS Migrated Deployment" in Chapter 6 of *SAS Intelligence Platform: Migration Guide*.

Step 6: Maintain and Update the New Environment

You can refresh the new environment as needed by using the promotion tools to copy specific content items from one deployment to another. See "Introduction to the Promotion Tools" on page 237.

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.

ARM

See Application Response Measurement.

authentication

the process of verifying the identity of a person or process for security purposes.

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 noninteractive method of running SAS programs by which a file (containing SAS statements along with any necessary operating system commands) is submitted to the batch queue of the operating environment for execution.

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 to enable quick analysis of data. 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, or the end of the job. The DATA step enables you to read raw data or other SAS data sets and to create SAS data sets.

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 or process of converting data to a form that is unintelligible except to the intended recipients.

external file

a file that is created and maintained by a host operating system or by another vendor's software application. An external file can read both data and stored SAS statements.

format

See SAS format.

foundation repository
the metadata repository that is used to specify metadata for global resources that can be shared by other repositories. For example, a foundation repository is used to store metadata that defines users and groups on the metadata server.

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.

level

See diagnostic level.

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 setting that reflects the language, local conventions, and culture for a geographic region. Local conventions can include specific formatting rules for paper sizes, dates, times, and numbers, and a currency symbol for the country or region. Some examples of locale values are French_Canada,Portuguese_Brazil, and Chinese_Singapore.

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

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

descriptive data about data that is stored and managed in a database, in order to facilitate access to captured and archived data for further use.

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.

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 container for OLAP 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 tables (cubes).

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 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.

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. In general, each user who participates in change management has an individual project repository. (Project repositories are available for SAS Data Integration Studio only.)

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 files 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.

role

a set of capabilities. In some applications, certain actions are available only to users or groups that have a particular role.

root logger

the highest-level logger in the logger hierarchy. In a logging configuration, all other loggers inherit the root logger's attributes.

SAS Application Server
a logical entity that represents the SAS server tier, which in turn comprises servers that execute code for particular tasks and metadata objects.

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 prompts the customer for other necessary input at the start of the session, so that there is no need to monitor the 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 Framework Data Server
a database server that is the default location for middle-tier data such as alerts, comments, and workflows, as well as data for the SAS Content Server and SAS Service Parts Optimization. The server is provided as an alternative to using a third-party DBMS. The server cannot be used as a general-purpose data store.

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

one or more files that are defined, recognized, and accessible 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

a container for metadata that is managed by the SAS Metadata Server.

SAS Metadata Server

a multi-user server that enables users to read metadata from or write metadata to one or more SAS Metadata Repositories.

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 table
another term for SAS data set.

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 SAS session that delivers services, data, or other resources to a requesting client. The server can run on the same computer as the client, or on a networked computer.

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 code repository
the location on the application server that contains stored process source code.

source metadata server
the metadata server from which metadata is promoted or replicated.

spawner
See object spawner.

stored process
See SAS Stored Process.

subscriber
a recipient of information that is published to a SAS publication channel.

target metadata server
the metadata server to which the metadata is promoted or replicated.

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 identity
a user or group that has all capabilities and permissions in the metadata environment due to membership in the META: Unrestricted Users Role (or listing in the adminUsers.txt file with a preceding 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 access them.

workspace

in the IOM object hierarchy for a SAS Workspace Server, an object that represents a single session in SAS.

Index