



SAS Publishing



SAS[®] 9.1.2 Integration Technologies

Server Administrator's Guide

The correct bibliographic citation for this manual is as follows: SAS Institute Inc. 2004. *SAS® 9.1.2 Integration Technologies: Server Administrator's Guide*. Cary, NC: SAS Institute Inc.

SAS 9.1.2 Integration Technologies: Server Administrator's Guide

Copyright © 2002-2004, SAS Institute Inc., Cary, NC, USA

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

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

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

May 2004

SAS Publishing provides a complete selection of books and electronic products to help customers use SAS software to its fullest potential. For more information about our e-books, e-learning products, CDs, and hard-copy books, visit the SAS Publishing Web site at support.sas.com/pubs or call 1-800-727-3228.

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

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

Table of Contents

SAS® 9.1.2 Integration Technologies: Server Administrator's Guide.....	1
Getting Started.....	2
Overview of Administration.....	3
Getting Started without the SAS Configuration Wizard.....	4
Choosing a Server Configuration.....	7
Planning for Metadata Definitions.....	9
Getting Started With the SAS Configuration Wizard.....	11
Initial Directory Structure.....	12
Initial Security Configuration.....	16
Initial Servers and Services Setup.....	20
Initial Load–Balancing Stored Process Server Configuration and Security.....	23
Additional Planning.....	25
Setting Up Libraries.....	27
Initial Access Control.....	28
Pooling and Load Balancing.....	29
Choosing Pooling or Load Balancing.....	31
Overview of Pooling.....	33
Locations for Specifying Pooling Parameters.....	34
Overview of Load Balancing.....	35
Setting Up a COM/DCOM Connection.....	39
Server and Client Requirements.....	40
Summary of Setup Steps (COM/DCOM).....	41
Planning Your Server Configuration Metadata.....	43

Table of Contents

Standard Server Metadata.....	44
Pooling Metadata.....	45
Creating Metadata Using SAS Management Console.....	47
Using SAS Management Console to Define Servers.....	48
Using SAS Management Console to Modify Servers.....	51
Using SAS Management Console to Define Custom Parameters for a Workspace Server (COM/DCOM).....	52
Using SAS Management Console to Define an OLAP Server (COM/DCOM).....	53
Using SAS Management Console to Define a Pooled Logical Server (COM).....	54
Enabling DCOM on the Server and the Client.....	56
Configuring SAS for DCOM.....	58
Setting SAS Permissions on the Server (COM/DCOM).....	59
Setting Default COM Security on Windows NT/2000.....	60
Setting Permissions per Application on Windows NT/2000.....	63
Setting Default COM Security on Windows XP.....	68
Setting Permissions per Application on Windows XP.....	71
Configuring COM/DCOM for Active Server Page Access.....	76
Accessing a Local COM IOM Server from an Active Server Page.....	78
Accessing a Remote DCOM IOM Server from an Active Server Page.....	80
Creating a Metadata Configuration File in SAS.....	85
Using the SAS Integration Technologies Configuration Utility (ITConfig).....	87
Using ITConfig to Create Metadata Configuration Files.....	88
Using ITConfig to Test Connections.....	90
Troubleshooting a COM/DCOM Connection.....	92

Table of Contents

AppIDs for Configuring DCOM.....	94
Object Server Parameters.....	95
Fields for the Server Definition.....	100
Fields for the Pooled Logical Server and Puddle Definitions.....	106
Setting Up an IOM Bridge Connection.....	108
Best Practices: Server and Spawner Setup.....	110
Quick Start: Standard Workspace Server and Spawner.....	112
Quick Start: Load–Balancing Stored Process Server and Spawner.....	115
Summary of Setup Steps (IOM Bridge).....	119
Spawner Overview.....	121
Spawner Requirements.....	123
Planning the Configuration Metadata.....	124
Security Metadata.....	125
Standard Workspace or Stored Process Server Metadata.....	129
Standard OLAP Server Metadata.....	132
Pooling Metadata.....	134
Load Balancing Metadata.....	137
Creating the Metadata Using SAS Management Console.....	143
Using SAS Management Console to Define Servers.....	144
Using SAS Management Console to Modify Servers.....	147
Using SAS Management Console to Define Custom Parameters for Workspace or Stored Process Servers (IOM Bridge).....	149
Using SAS Management Console to Define an OLAP Server (IOM Bridge).....	152
Using SAS Management Console to Define or Modify a Spawner (IOM Bridge).....	153

Table of Contents

Using SAS Management Console to Modify SAS Workspace or Stored Process Servers for Pooling or Load Balancing.....	156
Using SAS Management Console to Define a Pooled Logical Server (IOM Bridge).....	158
Using SAS Management Console to Define a Load–Balancing Logical Server (IOM Bridge).....	160
Configuring a UUID Generator.....	162
Configuring and Starting the Object Spawner on z/OS.....	163
Creating a Metadata Configuration File in SAS.....	168
Using the SAS Integration Technologies Configuration Utility (ITConfig).....	170
Using ITConfig to Create Metadata Configuration Files.....	171
Using ITConfig to Test Connections.....	175
Using SAS Management Console to Test Server Connections.....	177
Using Telnet to Administer the Spawner.....	178
Spawner Error Messages.....	180
Fields for the Server Definition.....	197
Object Server Parameters.....	203
Fields for the Spawner Definition.....	208
Fields for the Pooled Logical Server and Puddle Definitions.....	212
Fields for the Load–Balancing Logical Server Definition.....	214
Administering HTTP Servers and WebDAV.....	216
Using SAS Management Console to Define an HTTP Server.....	217
Starting Servers.....	219
Server Startup Command.....	222
Specifying Metadata Connection Information (required if METAAUTOINIT is specified).....	227
Customizing the Workspace Server Startup Command for COM/DCOM Connections.....	231

Table of Contents

Initializing UNIX Environment Variables for SAS Workspace Servers.....	232
Invoking (Starting) the Spawner.....	234
Starting the Spawner on Windows.....	236
Starting the Spawner on UNIX.....	238
Starting a Spawner on Alpha/VMS.....	240
Server and Services Startup Order.....	241
Object Server Parameters.....	242
Spawner Invocation Options.....	247
Moving Servers.....	250
Moving the SAS Stored Process Server.....	251
Moving the SAS Workspace Server.....	254
Moving the SAS OLAP Server.....	256
Moving Both the SAS Stored Process Server and SAS Workspace Server to the Same New Machine.....	258
Moving the SAS Stored Process Server and SAS Workspace Server to Separate Machines.....	261
Security.....	265
Overview of Domains.....	266
Implementing Authentication.....	267
Defining Users for Host Authentication.....	269
Setting System Access Permissions on Windows NT.....	271
Setting System Access Permissions on Windows 2000.....	272
Setting System Access Permissions on Windows XP.....	274
Setting System Access Permissions on UNIX.....	276
Specifying Default Host Domains When Starting Servers That Only Use Host Authentication.....	277

Table of Contents

How Hosts Handle Domains.....	279
Implementing Trusted Authentication Mechanisms.....	280
Implementing Alternative Authentication Providers.....	284
Specifying Authentication Provider and Default Domains When Starting Servers.....	288
How Servers Determine the Authentication Provider.....	290
Scenario: Alternate Authentication Provider.....	292
Defining Users, Groups, and Logins on the SAS Metadata Server.....	295
Implementing Authentication and Authorization for the Xythos WFS WebDAV Server.....	301
Scenario: Using the Xythos Administration GUI and SAS User Management Customization with the Portal.....	302
Implementing Encryption with Integration Technologies.....	308
Setting Up Additional Server Security.....	311
Planning Security on Workspace and Stored Process Servers (IOM Bridge Connection Only).....	313
Planning the Spawner Security.....	316
Spawner Security Scenario.....	320
Planning the Pooling Security (IOM Bridge only).....	322
Planning the Load Balancing Security (IOM Bridge only).....	328
Implementing Security in Client Applications.....	334

SAS® 9.1.2 Integration Technologies: Server Administrator's Guide

This is the Server Administrator's Guide for SAS Integration Technologies. This guide is provided for SAS Integration Technologies customers who use the SAS Open Metadata Architecture.

Here you will find detailed instructions for all of the server administration tasks that are required for SAS Workspace servers and spawners, and SAS Stored Process servers and spawners. It also provides general information for administering all IOM servers. Many of these tasks can be performed using the SAS Management Console application. SAS Management Console is a graphical user interface that enables you to easily enter and modify metadata on your SAS Metadata Server.



Before you begin performing SAS Integration Technologies server administration tasks, refer to the Getting Started chapter for important introductory information for administrators, including these topics:

- a high-level summary of the administrative steps involved in a SAS Integration Technologies implementation, both with and without the use of the SAS Configuration Wizard (including determining whether you should use a COM/DCOM connection or an IOM Bridge connection).
- information about setting up libraries.

Then refer to the other chapters in the Administrator's Guide for detailed documentation of the following administrative tasks:

- determining whether you should choose pooling or load balancing
- setting up an IOM server using COM/DCOM
- setting up an IOM Bridge server and spawner
- configuring HTTP servers and WebDAV
- initializing spawners and starting servers
- implementing authentication and server configuration security for your installation.
- moving servers for your installation

Use this Server Administrator's Guide in conjunction with the following titles:

- *SAS Integration Technologies Administrator's Guide*, which provides details about using administering SAS Foundation Services, SAS Stored Processes, and the SAS Publishing Framework
- *SAS Integration Technologies Developer's Guide*, which provides details about using SAS Integration Technologies to develop and integrate applications.
-  *SAS OLAP Server Administrator's Guide*, which provides information about how to perform administrative tasks such as installing, configuring, securing, and monitoring a SAS OLAP Server. It also includes information about building and using data cubes.
-  *SAS Metadata Server: Setup Guide*, which contains step-by-step instructions for setting up and administering a SAS Metadata Server.

Note: If you are implementing SAS Integration Technologies using the Lightweight Directory Access Protocol (LDAP) instead of the Open Metadata Architecture, refer to the Administrator's Guide (LDAP Version).

Getting Started

Getting Started

This chapter describes the processes for configuring and administering a SAS Integration Technologies implementation. In general, SAS Integration Technologies configuration and administration involves configuring the application resources for your site and deploying them for access by applications or users. The specifics of the process depend on the requirements for your implementation.

If you have used the new SAS Configuration Wizard to plan, install, and define your implementation, you already have an initial configuration of resources. If you have not used the new SAS Configuration Wizard, you must plan, install, and define your configuration. Depending on whether you used the SAS Configuration Wizard, the configuration and administration steps are as follows:

1. **Understand the general administration tasks and software.** For details, see [Overview of Administration](#).
2. **Plan and set up your server resources.** Depending on how you configure your initial setup, see the appropriate section in order to plan and set up your implementation:
 - ◆ Configuration Without the SAS Configuration Wizard. To understand how to plan and set up your implementation without using the SAS Configuration Wizard, see [Getting Started Without the SAS Configuration Wizard](#).
 - ◆ Configuration With the SAS Configuration Wizard. To understand the initial configuration that was planned, installed, and set up by the SAS Configuration Wizard, and how to plan for additional features and modify the initial configuration, see [Getting Started With the SAS Configuration Wizard](#).
3. **Set up libraries (optional).** For details about setting up libraries, see [Setting up Libraries](#).
4. **Set up other SAS Integration Technologies features.** To set up other features, you must define the resources on the SAS Metadata Server. For details about setting up SAS Stored Processes, SAS Publication Channels, and SAS Foundation Services on the SAS Metadata Server, see [Getting Started](#) in the *SAS Integration Technologies: Administrator's Guide*.
5. **Start your servers and services.** To understand how to start servers, and the order in which servers and services must be started in order to run your implementation, see the [Starting Servers](#) chapter.
6. **Update your configuration.** After you set up your implementation and roll it out to the user community, you might occasionally need to change your SAS Integration Technologies configuration. Therefore, you should establish a maintenance procedure for making changes to the configuration. If you need to move servers, refer to the detailed instructions in the [Moving Servers](#) chapter.

For details about coding client applications to access the server, user, and other resource metadata on the SAS Metadata Server, see the [SAS Integration Technologies Developer's Guide](#).

Getting Started

Overview of Administration

Administration of the IOM servers includes the following tasks:

- planning and setting up the SAS Metadata Server, for storing metadata definitions in a SAS Metadata Repository
- planning and setting up the server configurations
- planning and setting up users who will access the servers
- planning and setting up additional resources, such as libraries
- planning and setting up which users and group have what access to the servers and other resources
- starting the servers
- administering the servers

To set up the servers, you might use the following software:

- **SAS Configuration Wizard:** an application that automates configuration of the IOM servers within a planned (project) SAS installation.
- **SAS Management Console:** a Java application that provides a single point of control for performing the administrative tasks that are required to create and maintain an integrated SAS environment.

Getting Started

Getting Started without the SAS Configuration Wizard

If you are not using the SAS Configuration Wizard, follow these steps:

1. Plan your implementation:

- ◆ Determine how your organization intends to use these features of SAS Integration Technologies:

- ◇ distributed applications
- ◇ SAS Stored Processes
- ◇ publish/subscribe
- ◇ SAS Foundation Services

In addition, you might want to set up tables and libraries.

- ◆ Determine the hardware and software elements that will be involved in your SAS Integration Technologies implementation. For example, if you are administering a distributed application implementation, you will need to know the communication requirements for connecting your client and server platforms. The IOM servers are as follows:

- ◇ SAS Workspace Server
- ◇ SAS Stored Process Server
- ◇ SAS OLAP Server
- ◇ SAS Metadata Server

For some features, a WebDAV server might be required. The following table details the servers that are required to implement each SAS Integration Technologies feature:

Metadata on the SAS Metadata Server	
Feature	Required Server Definition
Metadata Storage (provides central, shared location)	SAS Metadata Server
Submit and Generate SAS Code	SAS Workspace Server and Spawner
Tables and Libraries	SAS Workspace Server and Spawner
Packages	<ul style="list-style-type: none">◇ if published to an archive on a SAS Workspace Server, a SAS Workspace Server and Spawner◇ if published to WebDAV, a WebDAV server◇ if published to a file, no server definition is needed for the package
Publication Channels	<ul style="list-style-type: none">◇ if publishing to an archive on a SAS Workspace Server, a SAS Workspace Server and Spawner◇ if publishing to an archive on a WebDAV server, a WebDAV server

	◇ if publishing to an archive in the file system, no server definition is needed for the publication channel
SAS Stored Processes – Package Results	if stored in an archive on a SAS Workspace Server, a SAS Workspace Server and Spawner if outputting a package to DAV, a WebDAV server
SAS Stored Processes – Streaming Results	SAS Stored Process Server and Login

In addition, you might also require certain servers for Business Intelligence content. The following table details the servers that are required to implement SAS Information Maps and SAS Reports:

Metadata on the SAS Metadata Server	
Feature	Required Server Definition
SAS Information Maps (preproduction feature)	◇ for relational data, a SAS Workspace Server and Spawner ◇ for multi-dimensional data, a SAS OLAP Server
SAS Reports (preproduction feature)	◇ for relational data, SAS Workspace Server and Spawner ◇ for multi-dimensional data, a SAS OLAP Server if storing reports in DAV, a WebDAV server

Finally, certain SAS clients rely on the servers. For details about how different SAS clients interact with the IOM servers, see "How SAS Clients Interact with SAS Application Servers" in the [SAS Intelligence Architecture: Planning and Administration Guide](#).

2. **Set up the SAS Metadata Server.** SAS Integration Technologies uses the SAS Metadata Server which provides a central, shared location for storing metadata. The metadata server provides a common repository from which user, resource, and security-policy information can be centrally managed. Because all of the SAS Integration Technologies administration tasks involve working with metadata information, the first administration task is to install and set up a metadata server. For reference material, see the [SAS Open Metadata Architecture](#) in the *SAS Integration Technologies Technical Overview*. For details about setting up the SAS Metadata Server, see [SAS Metadata Server: Setup Guide](#).
3. **Understand, plan, and implement security features.** For details about security, refer to "Understanding the Security Concepts in the SAS Intelligence Architecture," "Developing your Security Plan," and "Implementing Security" in the [SAS Intelligence Architecture: Planning and Administration Guide](#).
4. **Set up SAS Stored Process, SAS Workspace, and SAS OLAP Servers.**

To set up, start, and administer an IOM server, you must complete each of the following tasks:

- ◆ Choose the appropriate connection(s) for your server configuration:
 - ◇ A COM/DCOM connection, which enables client access using COM/DCOM
 - ◇ An IOM Bridge connection, which enables client access using the SAS Integration Technologies IOM Bridge for COM or IOM Bridge for Java.For details, see [Choosing a Server Configuration](#).
 - ◆ Plan your configuration metadata, which you will define on the SAS Metadata Server using SAS Management Console. For details, see [Planning for Metadata Definitions](#).
 - ◆ For IOM Bridge connections, determine the SAS users, groups, and login definitions that are specified in the server connection configuration:
 - ◇ login definitions for spawner security. For details, see [Planning the Spawner Security](#) in the Security chapter.
 - ◇ login definitions for pooling security. For details, see [Planning the Pooling Security](#) in the Security chapter.
 - ◇ login definitions for load balancing security. For details, see [Planning the Load-Balancing Security](#) in the Security chapter.
 - ◆ Set up a server using either a COM/DCOM connection or an IOM Bridge connection. For details about setting up a server that uses a COM/DCOM connection, see the [COM section](#). For details about setting up a server that uses an IOM Bridge connection, see the [IOM Bridge section](#). These sections provide detailed instructions for creating the metadata to define your server configuration, and for performing server administration and troubleshooting tasks.
- Note:** Servers that use a COM/DCOM connection do not use a spawner to launch the server. For SAS Workspace and SAS Stored Process Servers that use an IOM Bridge connection, you must set up a spawner to launch the server.
- In addition, for SAS Workspace Servers, you can choose to set up pooling or load balancing. For SAS Stored Process Servers, you *must* set up load balancing. For details, see the [Pooling and Load Balancing](#) chapter.
- ◆ Configure the startup commands for the spawners and servers as specified in the [Starting Servers](#) chapter. Instructions are provided for enabling and launching the spawners and servers on the host machine.
5. **Set up WebDAV Servers.** To set up WebDAV server definitions, see [HTTP Servers and WebDAV](#). If you are using the Xythos WFS WebDAV server, you can implement authentication using the SAS Metadata Server's authentication provider, and implement authorization using the Xythos WFS WebDAV Administration GUI to specify access controls for the SAS users and groups that are defined on the SAS Metadata Server. For details, see [Implementing Authentication and Authorization for the Xythos WFS WebDAV Server](#)
 6. **Set up metadata definitions for additional features.** For details about setting up libraries, see [Setting Up Libraries](#). For details about setting up SAS Foundation Services deployments, SAS Publishing Framework, and SAS Stored Processes, see [SAS Integration Technologies Administrator's Guide](#).
 7. **Implement security for resource authorization.** Set up access controls for your resources as specified by your security plan.
 8. **Start your servers and services in the appropriate order.** For details, see [Server and Service Startup Order](#)

Getting Started

Choosing a Server Configuration

SAS Integration Technologies supports two types of connections for server configurations:

- **COM/DCOM Connection.** A COM/DCOM connection enables client access using the native Windows Component Object Model (COM) or Distributed Component Object Model (DCOM). In a client–server environment, DCOM must be enabled on both the client machine and the machine where the server runs.
- **IOM Bridge Connection.** An IOM Bridge connection enables client access using the SAS Integration Technologies IOM Bridge for COM or IOM Bridge for Java. The IOM Bridge for COM allows you to develop native COM/DCOM applications that access server data on UNIX, VMS, or z/OS. The IOM Bridge for Java allows you to develop applications using Java that access server data on Windows, UNIX, VMS, or z/OS platforms.

To understand how clients connect to servers using COM/DCOM and IOM Bridge connections, see [Connecting Clients to IOM Servers](#) in the *SAS Integration Technologies Technical Overview*.

The following table shows the supported connection types for each of the four types of IOM servers.

Supported Connection Types			
IOM Server Type	COM Connection	IOM Bridge Connection	Both COM and IOM Bridge Connection
SAS Workspace Server	x	x	
SAS Stored Process Server		x	
SAS OLAP Server	x	x	x
SAS Metadata Server	x (experimental only)	x	x (experimental only)

When to Use a COM/DCOM Connection

For SAS Workspace Servers, SAS OLAP Servers, and SAS Metadata Servers (experimental only), you can use a COM/DCOM server configuration in either of the following situations:

- the server will run on a Windows machine and will be accessed by Windows client applications running on remote machines. In this situation, the connection uses DCOM.
- the server will run on a Windows machine and will be accessed by Windows client applications running on the same machine. In this situation, the connection uses COM.

If the object server will be accessed by a Java client, you must use an IOM Bridge connection instead.

For more information about COM/DCOM distributed clients, refer to [Connecting Clients to IOM Servers](#) in the *SAS Integration Technologies Technical Overview*.

When to Use an IOM Bridge Connection

For any IOM server (SAS Workspace Servers, SAS Stored Process Servers, SAS OLAP Servers, and SAS Metadata Servers), you must use an IOM Bridge connection if any of the following are true:

- the server will run on a UNIX, VMS, or z/OS machine
- the server will be accessed by Java client applications
- you want to use load balancing to balance work across server processes on the same or separate machines.

You can also use an IOM Bridge connection if the server will run on a Windows machine and will be accessed by Windows clients. In this situation, clients will connect to the server using the IOM Bridge connection instead of a COM/DCOM connection.

For more information about the IOM Bridge for COM and the IOM Bridge for Java, refer to [Connecting Clients to IOM Servers](#) in the *SAS Integration Technologies Technical Overview*.

When to Use Both IOM Bridge and COM/DCOM Connections

For SAS OLAP Servers and SAS Metadata Servers (where COM server connections are experimental only), you can set up a multi-user server to run IOM Bridge and COM/DCOM connections simultaneously if the server runs on Windows.

When you run the IOM Bridge and COM/DCOM connections simultaneously, you enable both of the following:

- single-signon capabilities of the Windows network environment for COM clients and Windows SAS clients
- support for Java and for UNIX, VMS, and z/OS SAS clients.

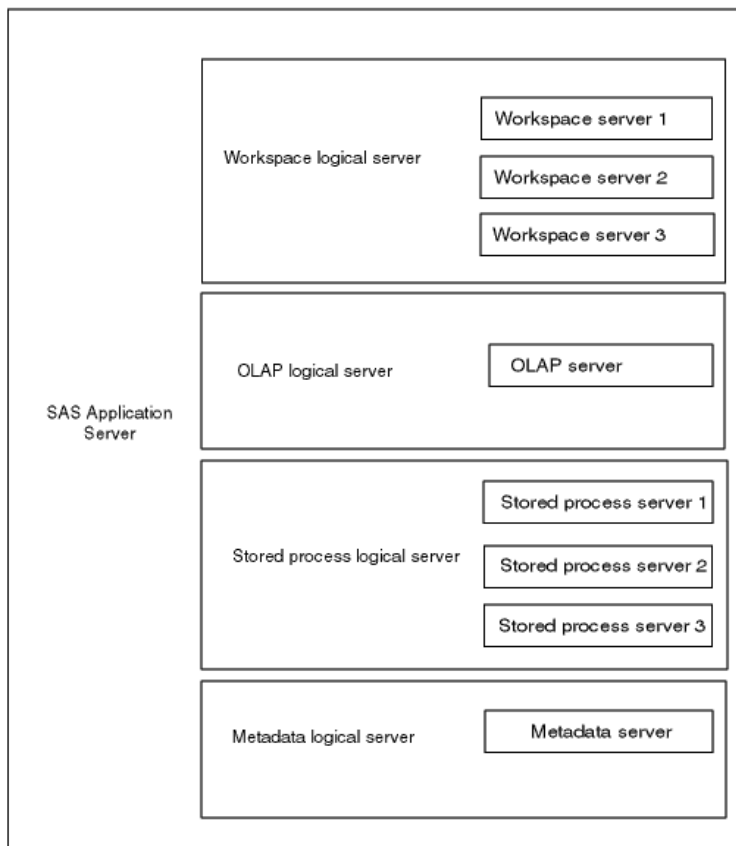
Getting Started

Planning for Metadata Definitions

To plan your server metadata, you must first plan for your SAS application server and logical server definitions. This page describes the SAS application server and logical server concepts.

Understanding SAS Application Servers and Logical Server Definitions

At a minimum, the metadata for a SAS server consists of three definitions: a SAS application server, a logical server, and a server. The following diagram shows how the SAS application server (server context), logical server, and servers are related.



- **SAS application server.** A SAS application server is a server context for your metadata. A server context allows you to specify metadata that applies to all of the logical servers and servers that it contains. A server context is a container to which you can assign libraries, schemas, directories and other resources that are available to SAS servers, regardless of the type of server. For example, when you define a SAS library in the Data Library Manager, you can assign the library to the server context and all the servers within the context will have access to the library.

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

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

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

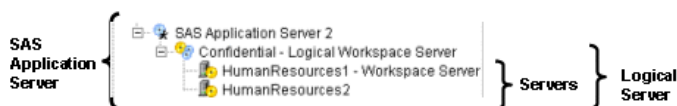
A SAS application server can contain one or more logical server definitions. However, each SAS application server can contain only one of each type of logical server. The types of logical servers correspond to the four types of IOM servers: SAS Metadata Server (COM connection is experimental), SAS Workspace Server, SAS Stored Process Server (IOM Bridge connection only), and SAS OLAP Server. For example, in one SAS application server definition, you can have up to 4 logical servers: one logical metadata server (COM connection is experimental), one logical workspace server, one logical stored process server (IOM Bridge connection only), and one logical OLAP server.

- **Logical servers.** For SAS Stored Process Servers (IOM Bridge connection only) and SAS Workspace Servers, a logical server definition contains one or more server definitions. For SAS OLAP Servers and SAS Metadata Servers (COM connection is experimental), a logical server contains one server definition. However, each logical server can contain only one type of IOM server. For example, a logical workspace server can only contain workspace servers.

After you have created a particular type of logical server within the SAS application server definition, you can then add servers to this logical server and do one of the following:

- ◆ leave the servers as standard servers that do not pool or load balance within the logical server group. In this case, the multiple standard servers are used as redundant servers to provide fail-over capability. Clients that cannot connect to the first server will try subsequent servers within the logical server.
- ◆ for workspace servers (IOM Bridge or COM connection), convert the logical server to a pooled logical server and set pooling parameters on the servers that are contained within the logical server group.
- ◆ for workspace or stored process servers (IOM Bridge connection only), convert the logical server to a load-balancing logical server and set load-balancing parameters on the servers that are contained within the logical server group.
- **Servers.** A server definition contains the actual server metadata required to connect to a SAS server on a particular machine.

The following diagram shows the SAS application server, logical server, and servers defined in SAS Management Console.



After you have planned for your metadata, you can set up a server with a COM or IOM Bridge connection:

- For information about using SAS Integration Technologies to set up an IOM Bridge connection, see [Setting Up an IOM Bridge Connection](#).
- For information about using SAS Integration Technologies to set up a COM/DCOM connection, see [Setting Up a COM/DCOM Connection](#).

Getting Started

Getting Started With the SAS Configuration Wizard

When you use the SAS Configuration Wizard, you plan for and set up your implementation within a project installation. The project installation:

- provides a pre–installation checklist that you use to create the required operating system user accounts
- uses a planning file to install and configure your licensed software, such as Base SAS, SAS Management Console, and SAS Integration Technologies
- installs the required version of the Java Runtime Environment (JRE) (if needed)
- prompts you to register a foundation repository on the SAS Metadata Server
- walks you through using SAS Management Console to write metadata definitions for users, user groups, and installed servers, such as the SAS Workspace Server, SAS Stored Process, and SAS OLAP Server
- instructs you to define authorization metadata for the initial users and user groups
- optionally starts servers as services on Windows (creates start–up scripts for other operating systems)
- creates a physical directory structure on each host in your environment. The directory structure stores, for example, log files, server start–up scripts, and format catalogs (see [Initial Directory Structure](#)).

For details about the project installation using SAS Configuration Wizard, see the [SAS Intelligence Architecture: Planning and Administration Guide](#).

After you have performed the project installation, you have an initial security and server configuration. The following sections provide details about your initial setup and links to configuration sections that provide details about how to modify your setup:

- [Initial Directory Structure](#)
- [Initial SAS Metadata Server Configuration](#)
- [Initial IOM Server Configuration](#)
- [Initial WebDAV Server Configuration](#)

In addition, you might need to configure additional servers and resources for your implementation, as outlined in [Additional Planning](#). When you modify your setup, you must implement the appropriate authorization (access controls) for the new resources.

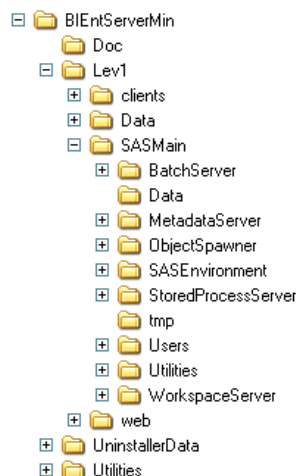
Getting Started

Initial Directory Structure

After initial configuration, you will have a server environment that is represented by a physical directory structure and by metadata registered in a SAS Metadata Repository.

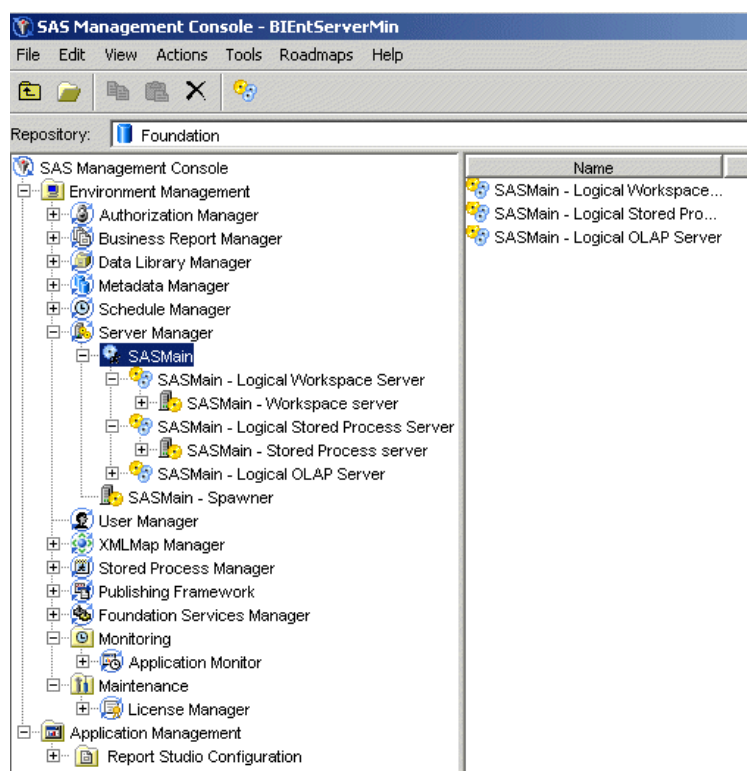
The physical directory structure is created automatically for you by the SAS Configuration Wizard based on your input into each wizard window. The SAS Configuration Wizard creates a physical directory structure on each host in your environment. The following illustration is a typical directory structure on a Windows machine that is hosting four servers and one spawner: a SAS Metadata Server, a SAS OLAP Server, a SAS Stored Process Server, a SAS Workspace Server, and an object spawner.

A Typical Directory Structure for a Windows Machine that Is Hosting Four Servers



The metadata environment is created by you by following instructions that are contained in a generated HTML file. The following illustration shows the corresponding Server Manager definitions in SAS Management Console for an implementation that has three servers and an object spawner defined on the SAS Metadata Server: a SAS OLAP Server, a SAS Stored Process Server, a SAS Workspace Server, and an object spawner.

SASMain Server Definitions in SAS Management Console



Here are descriptions of the contents of the object spawner, SAS Workspace Server, SAS Stored Process Server, SAS OLAP Server directory, and its parent directories:

Initial Directories

BIArchitecture

the configuration directory, which is the root of the environment. The environment identifies an entire set of related information such as production levels, SAS application servers, scripts, utilities, and documentation. The name of this directory is site-specific.

Lev1

identifies the production status of the information contains within the level directory. **Lev1** is the production level.

SASMain

the name of the **SAS application server** on the machine that hosts the SAS Metadata Server. The SAS application server is a logical framework under which SAS applications execute. A SAS application server provides a place to attach libraries, schemas, directories, and other resources that are available to SAS servers, regardless of the type of server. Providing this framework separate from the launching mechanism enables the administrator to deploy applications in several modes while ensuring that the applications will execute properly in that mode.

ObjectSpawner

the physical directory that corresponds to a server definition in a SAS Metadata Repository. This directory contains the start-up scripts needed to run a process-spawning service that instantiates SAS Workspace Servers and load-balanced SAS Stored Process Servers. Typically, the spawner is started as a service on Windows.

Note: During the deployment process, you are given the option to install and configure the spawner. However, if you are installing SAS Workspace Servers or SAS Stored Process Servers, then the spawner is automatically installed because those servers must be started by a spawner.

OLAPServer

(shown selected and expanded) the physical directory that corresponds to the SAS OLAP Server definition in the SAS Metadata Repository. It contains these items:

- the **OLAPServer.bat** file, which is a script for maintaining the SAS OLAP Server. You should not modify this file.
- the **sasv9_OLAPServer.cfg** file, which includes standard configuration information required for the SAS OLAP Server plus additional options needed by the SAS Metadata Server.
- the **startOLAPServer.bat** file, which is a script to start up the server manually. Typically, the server is started as a service on Windows.
- the **OLAPServerSSCU.ini** file, which is the SAS Services Control Manager Initialization File for the SAS OLAP Server.
- the **sasuser** subdirectory, which is a SAS library that contains SAS catalogs that enable you to tailor features of SAS for your needs. SAS assigns the SASUSER library at invocation.
- the **logs** subdirectory that stores any log files generated by the SAS OLAP Server invocation.
- the **work** subdirectory that stores temporary work files.

StoredProcess Server

the physical directory that corresponds to a server definition in a SAS Metadata Repository. This server executes stored processes. A stored process is a SAS program that is stored on a server and can be executed as required by requesting applications. This directory contains these items:

- a sample SAS application named **LoadPlannedStoredProcessSamples.sas** that can be used to load stored process samples into the metadata repository.
- the **logs** subdirectory that stores any log files generated by this server invocation.
- any other utility files that are used to manage the server.

WorkspaceServer

the physical directory that corresponds to a server definition in a SAS Metadata Repository. This server fulfills client requests for specific SAS sessions. This directory contains these items:

- the **logs** subdirectory that stores any log files generated by this server invocation.
- any other utility files that are used to manage the server.

In addition to the SASMain directory, the **Lev1** directory contains the **web** subdirectory.



Depending on your operating system and your installed products, the **web** directory might contain the following subdirectories:

Initial Directories For Web Content

Deployments	contains information related to deploying Web applications, including policy files, service configuration files, and service deployment files.
webapps	contains the Web applications archive (WAR) files for your Web applications such as SAS Information Delivery Portal. These WAR files are actually JAR files that contain all of the files that make up the Web application, including servlets, JavaServer Pages, and HTML documents.
<p>Note: If you are using the Tomcat servlet container to execute your Web applications, the SAS Configuration wizard has already copied these WAR files to Tomcat's webapps directory. If you are using a J2EE application server for this purpose, you must manually deploy these files to your server's webapps directory. Follow the instructions in your vendor's documentation for deploying an application.</p>	

For detailed information about the configuration environment, refer to the [SAS Intelligence Architecture: Planning and Administration Guide](#).

Getting Started

Initial Security Configuration

After you perform your pre-installation tasks, run the SAS Configuration Wizard, and perform the post-installation manual setup, your initial security setup includes the following user and group definitions on the SAS Metadata Server:

- **SAS Administrator (for example, sasadm).** This user's ID will be written to a special file called `adminUsers.txt`, which gives the user unrestricted access to the metadata server. For information about administrative users, see [Server Administrative Privileges](#) in the *SAS Metadata Server: Setup Guide*. You can use the SAS Administrator to log on to SAS Management Console and create metadata on the SAS Metadata Server.
- **SAS Trusted User (for example, sastrust).** This user's ID is written to the file `trustedUsers.txt` and has trusted access to the metadata server. It is used for the following tasks:
 - ◆ If you have installed a SAS OLAP Server, this user is used for a trusted connection from the SAS OLAP server to the SAS Metadata Server.
 - ◆ The object spawner that starts your workspace and stored process servers uses this account to connect to the metadata server in order to read the appropriate server and spawner definition.
 - ◆ If you configure Web server authentication, this user enables mid-tier (Web-tier) users to be viewed as already authenticated by the Web server and connect to the SAS Metadata Server for authorization purposes.

For information about trusted users, see [Trusted Users](#) in the *SAS Metadata Server: Setup Guide*.

Important Note: This is a highly privileged account and should be protected accordingly.

- **SAS Guest (for example, sasguest).** This user is a guest user. If you have installed the Web Infrastructure Kit or the SAS Information Delivery Portal, this user configures the Public Kiosk for the portal Web application.
- **SAS General Servers group.** This group contains a group login that is used by the spawner to start the load-balancing SAS Stored Process Servers. This group login is also used by servers to connect back to the SAS Metadata Server. The SAS Trusted User is a member of the SAS General Servers group.
- **SAS System Services group.** The group is for users that make server-to-server connections. The group initially contains the SAS Web Administrator (for example, saswbadm) and the SAS Trusted User (for example, sastrust).

Mid-tier Credentials

If you have set up mid-tier (Web-tier) software, the initial security setup also includes the following users and groups:

- **SAS Web Administrator (e.g. saswbadm).** This user has permission to administer the portal Web application. The portal Web application shell uses the SAS Web administrator to perform specific tasks, such as deploying portlets and creating SAS group permission trees. The SAS Web administrator has administrative privileges for all of the portal Web application content. The SAS Web administrator can access any portal user's pages and share content with any SAS group.
- **SAS Demo User (for example, sasdmo).** This user is the general demo user for the portal Web application.
- **Portal Admins group.** The group `Portal Admins` is for the users that are SAS Web administrators. The group initially contains the `saswbadm` user. Each member of the `Portal Admins` group is a SAS Web administrator and has administrative permissions to view any user's content and share that content with any SAS group.

- **Portal Demos** group. The group `Portal Demos` is for the portal's demo users. The group initially contains the `sasdemo` user.

Unix and z/OS Systems Credentials

If you installed the portal Web application with the project installation on Unix or z/OS, you created one additional user and one additional group on the operating system:

- **SAS user:** The default SAS user is `sas`. The SAS user should be used to start the following servers (if they are not started as a service) and spawners:
 - ♦ Start the spawner that starts the SAS Workspace Server(s) and SAS Stored Process Server(s).
 - ♦ If you are not starting the SAS Metadata Server as a service, start the SAS Metadata Server.
 - ♦ If you have installed a SAS OLAP Server and are not starting the OLAP server as a service, start the OLAP server.
- **SAS group:** The default SAS group is `sas` on Unix and `sasgrp` on z/OS. This group is used to control access to some directories and files.

For additional details about the SAS user and group, see "Pre-Installation Checklist for Unix" and "Pre-Installation Checklist for z/OS" in the [SAS Intelligence Architecture: Planning and Administration Guide](#).

Initial User Accounts

If you deploy a distributed server configuration, or authenticate some users against an alternative authentication provider, the following table shows the required locations of the user accounts that you create before beginning your installation:

Summary of Required Accounts for Authentication of Initial Credentials				
User Name (User ID)	SAS Metadata Server's authentication provider	SAS Workspace Server's host authentication provider	SAS Stored Process Server's host authentication provider	SAS OLAP Server's authentication provider
SAS Administrator (e.g., <code>sasadm</code>)	Yes	No	No	Yes
SAS Trusted User (e.g., <code>sastrust</code>)	Yes	No	No	No
SAS Guest (e.g., <code>sasguest</code>)	Yes	Yes*	Yes	Yes
SAS Demo User (e.g., <code>sasdemo</code>)	Yes	Yes*	Yes	Yes
SAS General Server (e.g., <code>sassrv</code>)	Yes	Yes	Yes	No

Note: If you set up the SAS Workspace Server in a pooled configuration, you are not required to have an account for these user credentials on the host for the SAS Workspace Server.

User and Group Metadata Identities

The following table summarizes the User and Group objects that you have defined in the metadata in order for your servers and applications to work correctly. You can use the User Manager plug-in in SAS Management Console to verify that these objects have been created properly.

Metadata Identities	Logins			Group Membership Information
	User ID*	Password**	Authentication Domain	
User: SAS Administrator	sasadm			
User: SAS Trusted User	sastrust			member of: SAS System Services group member of: SAS General Servers group
User: SAS Guest User	sasguest	*****	DefaultAuth	
User: SAS Demo User	sasdemo	*****	DefaultAuth	member of: Portal Demos
User: SAS Web Administrator***	saswbadm	*****	DefaultAuth	member of: SAS System Services group member of: Portal Admins
Group: SAS System Services				members: SAS Trusted User, SAS Web Administrator
Group: SAS General Servers	sassrv	*****	DefaultAuth	members: SAS Trusted User
Group: Portal Admins***				members: SAS Web Administrator
Group: Portal Demos***				members: SAS Demo User
<p>* These are the recommended IDs. They should correspond to accounts in your authentication provider. On Windows, the user ID in the login should be fully qualified with a host or domain name, for example, <i>host-name\sasadm</i>.</p> <p>** If you are logged in to SAS Management Console as an unrestricted user, you will always see ***** in the password column, even if no password was specified.</p> <p>*** You only need this metadata identity if you have a mid-tier.</p>				

For information about the SAS General Servers group set up, and about the problems you will see if it is not set up correctly, see [Initial Load Balancing Stored Process Server Configuration and Security](#).

To add new SAS users and groups, refer to "Understanding the Security Concepts in the SAS Intelligence Architecture," "Developing your Security Plan," and "Implementing Security" in the [SAS Intelligence Architecture: Planning and Administration Guide](#).

To implement authentication against an alternate authentication provider, see [Implementing Authentication](#) in the Security chapter of this guide.

Getting Started

Initial Servers and Services Setup

After you run the SAS Configuration Wizard, you have an initial server configuration. To understand the locations of your configuration information, see "Overview of Understanding the SAS Configuration Environment" in the [SAS Intelligence Architecture: Planning and Administration Guide](#). Your initial server setup includes a SAS Metadata Server and might include one or more additional server configurations.

Initial SAS Metadata Server Configuration

After you run the SAS Configuration Wizard, the initial SAS Metadata Server configuration includes the following:

- **SAS Metadata Repository on the SAS Metadata Server's host machine**, located in the `MetadataServer\MetadataRepositories\<RepositoryName>` directory of the installation. To modify the SAS Metadata Server and SAS Metadata Repository setup, see the [SAS Metadata Server: Setup Guide](#).
- **SAS Metadata Server startup script or service configuration**, located in the `MetadataServer` directory of the installation. To modify the startup script or service configuration for the SAS Metadata Server, see [Starting the SAS Metadata Server](#) in the *SAS Metadata Server: Setup Guide*.

Initial IOM Server Configuration

After you run the SAS Configuration Wizard, you will have one or more servers configured. The following table details the server configurations that might be set up.

Initial IOM Server Configurations					
Server	Location and Type of Server Startup	Credentials Used for Server Startup and SAS Metadata Server Connection	Metadata	Credentials Used in Metadata Configuration	Modifying the Configuration
SAS Workspace Server	Spawner startup script , located in the <code>ObjectSpawner</code> subdirectory of your install. See Spawner Overview and Invoking the Spawner .	To connect to the SAS Metadata Server , the <code>ObjectSpawner</code> subdirectory also contains the spawner's metadata configuration file, <code>OMRConfig.xml</code> , which specifies the SAS Trusted User's user ID to connect to the SAS Metadata Server*.	Standard logical server definition named Main–Logical Workspace Server.	N/A	Pooling. See Pooling Overview and Pooling Metadata .
			Server definition named Main – Workspace Server	N/A	Adding a new server and spawner. See Standard Server Metadata .
			Spawner definition	N/A	Adding a new server with a COM connection. See

		To start the server, the spawner uses the client's credentials to launch the SAS Workspace Server.			Standard Server Metadata.
SAS Stored Process Server	Spawner startup script , located in the ObjectSpawner subdirectory of your installation. Modifying the startup script. See Spawner Overview and Invoking the Spawner .	To connect to the SAS Metadata Server , the ObjectSpawner subdirectory also contains the spawner's metadata configuration file, OMRConfig.xml, which specifies the SAS Trusted User's user ID to connect to the SAS Metadata Server*. To start the server, the spawner uses the login credentials owned by the SAS General Servers group to launch the SAS Stored Process Server.**	Load-balancing logical server definition named Main– Logical Stored Process Server	Logical server credentials , which are specified as the group login of the SAS General Servers group.	Load balancing. See Load Balancing Overview and Load Balancing Metadata .
			Load-balancing server definition named Main – Stored Process Server	Multi-user login, which is specified as the group login of the SAS General Servers group. This login is the user ID under which the SAS Stored Process Server runs.	Adding a new server and spawner. See Standard Server Metadata .
			Spawner definition named Main– Logical OLAP Server	N/A	
SAS OLAP Server	Startup script or service startup , located in the OLAPServer subdirectory of your installation. Modifying the startup script or service configuration. See Creating and Modifying the SAS OLAP Server Script and Starting the SAS OLAP Server as a Service in the <i>SAS OLAP Server Administrator's Guide</i> .	To connect to the SAS Metadata Server , the SAS OLAP server uses the SAS Trusted User's user ID.* To start the server (if not started as a service) , the SAS user credentials are used.	Spawner definition named Main– Logical OLAP Server	N/A	Adding a COM connection to the server. See Adding a COM Connection .
			Server definition named Main – OLAP Server	N/A	Adding a New Server Definition. See Standard OLAP Server Metadata (IOM Bridge) and Standard Server Metadata (COM) . Also see the Server Manager Help in SAS Management Console.

*The SAS Trusted User is a member of the SAS System Services group, which has permission to read the metadata (ReadMetadata permission) for the server definitions.

**The SAS General Servers group owns the login that the spawner uses to start the SAS Stored Process Server. The spawner (which connected to the SAS Metadata Server as the SAS Trusted User) can see the SAS General Servers group login because SAS Trusted User is a member of the SAS General Servers group.

To move servers to a different machine, depending on whether you have both the SAS Workspace Server and SAS Stored Process Server installed, or installed on the same or separate machines, see the following:

- [Moving the SAS Workspace Server](#)
- [Moving the SAS Stored Process Server](#)
- [Moving both the SAS Workspace Server and SAS Stored Process Server to the Same Machine](#)
- [Moving the SAS Stored Process Server and SAS Workspace Server to Separate Machines](#)
- [Moving the SAS OLAP Server](#)

Initial WebDAV Server Configuration

If you use the SAS Configuration Wizard to configure the Apache HTTP server as a WebDAV server, it creates a server definition, "HTTP DAV Server." To modify the WebDAV server definition, see the Server Manager Help in SAS Management Console. To add a new WebDAV server definition, see [HTTP Servers and WebDAV](#).

If you are using the Xythos WFS WebDAV server, you can implement authentication using the SAS Metadata Server's authentication provider, and implement authorization using the Xythos WFS WebDAV Administration GUI to specify access controls for the SAS users and groups that are defined on the SAS Metadata Server. For details, see [Implementing Authentication and Authorization for the Xythos WFS WebDAV Server](#).

Initial SAS Foundation Services Configuration

The project installation places an XML service deployment file into the Deployments directory of the project installation. For details, see the topic "Web Contents" in the [SAS Intelligence Architecture: Planning and Administration Guide](#). The service deployment file contains the local and remote service deployment configurations. For details about working with service deployments, see the [Foundation Services](#) chapter in the *SAS Integration Technologies Administrator's Guide* and the [Deployment](#) chapter in the *Web Infrastructure Kit Administrator's Guide*.

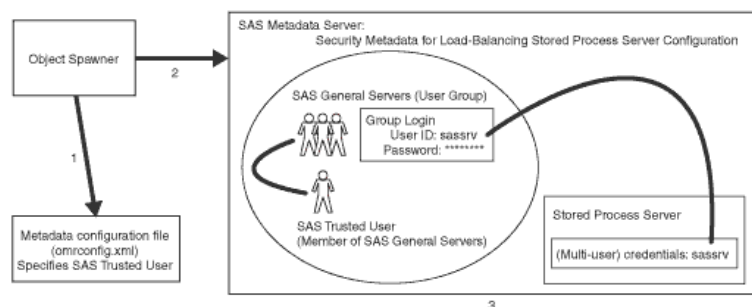
Getting Started

Initial Load–Balancing Stored Process Server Configuration and Security

After you run the SAS Configuration Wizard to setup a stored process server, the initial load–balancing SAS Stored Process Server configuration is set up with three MultiBridge connections so that the object spawner can start up to three stored process server processes. The object spawner will balance the workload across these processes. The object spawner runs on the server host, listens for client requests, and connects clients to the appropriate server process.

The SAS Metadata Server contains the spawner, server, and security metadata for the load–balancing stored process server configuration. The object spawner must connect to the SAS Metadata Server and the metadata must be appropriately configured to enable the spawner to start the load–balancing stored process server processes. The following diagram shows the initial security setup and process flow for the load–balancing stored process server and spawner configuration:

Note: On Windows, all user IDs would be machine or domain qualified. For example, europe\sastrust.

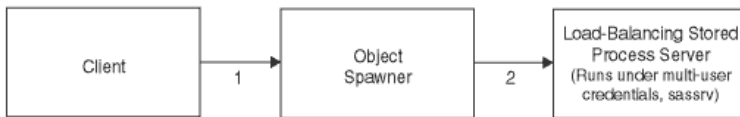


In the previous diagram, the Object Spawner obtains the metadata information to start a load–balancing stored process server as follows:

1. When the spawner is started, it reads a metadata configuration file named `omrconfig.xml` that contains information to access the SAS Metadata Server. This metadata configuration file specifies the following information:
 - ◆ the location of the SAS Metadata Server
 - ◆ the user ID that the spawner will use to connect to the metadata server.By default, the `omrconfig.xml` file contains the user ID `sastrust`, which is owned by the SAS Trusted User.
2. The object spawner connects to the SAS Metadata Server using the user ID specified in `omrconfig.xml`. (By default, this is SAS Trusted User (e.g. `sastrust`)). The SAS Trusted User's credentials are authenticated against the SAS Metadata Server's authentication provider.
3. On the SAS Metadata Server, the connection from the object spawner is associated with the user that owns the `sastrust` user ID, SAS Trusted User. The spawner (as the SAS Trusted User) reads the metadata information for the server and spawner configuration.

Note: The SAS Trusted User's login credentials can view the server's multi–user login credentials (`sassrv`) because the SAS Trusted User is a member of the SAS General Server group and the SAS General Servers group owns the server's multi–user login credentials (`sassrv`).

The object spawner then has the necessary metadata to launch a server. The following diagram shows the flow for a client request and server launch.



1. When a client requests a server, the client is authenticated against the host authentication provider for the server.
2. If the object spawner needs to launch a new stored process server, the object spawner uses the server's multi-user login credentials (sassrv) to launch the load-balancing stored process server.

Note: Because the stored process server runs under the credentials for the multi-user stored process server, each client can only access information for which the multi-user credentials are authorized.

To summarize, in your initial load-balancing stored process server configuration, you must ensure that the following security is configured properly:

- On the SAS Metadata Server, ensure that the SAS Trusted User is a member of the SAS General Servers group
- In the metadata configuration file, omrconfig.xml, ensure that the SAS Trusted User's credentials are specified.
- On the SAS Metadata Server, ensure that the group login owned by the SAS General Servers group is specified in the stored process server definition (on the Credentials tab).
- Ensure that the user ID and password of the group login for the SAS General Servers group matches the account on the host authentication provider for the stored process server.

To improve performance, you can add a second load-balancing stored process server machine. For details, see [Load Balancing Metadata](#).

Getting Started

Additional Planning

In addition to the initial SAS Configuration Wizard implementation, you might need to plan for resources for SAS Integration Technologies features as follows:

1. Determine how your organization intends to use the features of SAS Integration Technologies. These features include the following:

- ◆ distributed applications
- ◆ SAS Foundation Services
- ◆ SAS Stored Processes
- ◆ publish and subscribe

In addition, you might want to set up tables and libraries.

2. Determine the hardware and software elements that will be involved in your SAS Integration Technologies implementation. For example, if you are administering a distributed application implementation, you will need to know the communication requirements for connecting your client and server platforms.

For certain features, you might also require a WebDAV server. The following table details the servers that are required to implement each SAS Integration Technologies feature:

Metadata on the SAS Metadata Server	
Feature	Required Server Definition
SAS Code Submit and Generate	SAS Workspace Server and Spawner
Tables and Libraries	SAS Workspace Server and Spawner
Packages	<ul style="list-style-type: none">◆ if published to an archive on a SAS Workspace Server, a SAS Workspace Server and Spawner◆ if published to WebDAV, a WebDAV server◆ if published to a file, no server definition is needed for the package
Publication Channels	<ul style="list-style-type: none">◆ if publishing to an archive on a SAS Workspace Server, a SAS Workspace Server and Spawner◆ if publishing to an archive on a WebDAV server, a WebDAV server◆ if publishing to an archive in the file system, no server definition is needed for the publication channel
SAS Stored Processes – Package Results	<ul style="list-style-type: none">◆ if stored in an archive on a SAS Workspace Server, a SAS Workspace Server and Spawner◆ if outputting a package to DAV, a WebDAV server

SAS Stored Processes – Streaming Results	SAS Stored Process Server and Login
--	-------------------------------------

3. Determine the additional users that will access the servers, data, and other resources.
4. Determine security roles and authorization policies for data and other resources.

After you have planned for any additional resources for SAS Integration Technologies, refer to the appropriate section for links that contain instructions about how to modify the initial setup:

- [Initial Security Configuration](#)
- [Initial SAS Metadata Server Configuration](#)
- [Initial IOM Server Configuration](#)
- [Initial WebDAV Server Configuration](#)
- [Initial SAS Foundation Services Configuration](#)

To administer SAS Stored Process or SAS Publishing Framework resources, refer to the [*SAS Integration Technologies Administrator's Guide*](#).

Getting Started

Setting Up Libraries

To use libraries, you must define the appropriate servers and libraries on the SAS Metadata Server or in a configuration file. You can then access the library information as required for your implementation. (To access the library definition on the SAS Metadata Server, the user ID that reads the definition must have the ReadMetadata permission. You should have already determined the appropriate authorization (access controls) for the libraries that you will define and access on the SAS Metadata Server).

To set up libraries, define (pre-assign) and access the library definitions in one of the following ways:

- **Pre-assign libraries in the metadata (SAS Workspace, SAS Stored Process, and SAS OLAP Servers only).** To pre-assign and access library definitions, follow these steps.

1. Use SAS Management Console to create a library definition, and then pre-assign the library. When you use the Data Library Manager plug-in of SAS Management Console to register the library in the SAS Metadata Server, you can identify the library as preassigned on the Options tab under **Advanced Options**. For details, see [Managing Libraries](#) in the *SAS Management Console: User's Guide*.
2. Set the SERVER= and, if required, set the METAUTOINIT parameters, then specify information to connect to the SAS Metadata Server. For details, see [Specifying Metadata Connection Information](#). If the library has not already been preassigned by using one of the SAS 8.2 methods, the metadata LIBNAME engine will use the library's metadata definition to assign the libref automatically when access to the data is requested. For details about the metadata LIBNAME engine, see the [SAS Metadata LIBNAME Engine User's Guide](#).

- **Pre-assign the library definition on the server startup command or in the SAS Config file.**

To access a library that has been pre-assigned to an environment variable, use the SET system option (on the server startup command or in the SAS Config file) to define an environment variable that is valid within the SAS session. (The server startup command is supplied either on the command line or in the metadata's server definition, on the Options tab under Launch Commands). For example:

```
* in the config file
-set GRAPHDATA "c:\sasv9\samples\graph\data"
```

When you refer to GRAPHDATA as a library name during your SAS session, SAS automatically assigns the library with the path that is listed in the SET command. For example:

```
/* SAS Language submitted by the client */
proc datasets library=graphdata; run;
```

- **Pre-assign a library definition by using a SAS Autoexec file.** To pre-assign and access a library definition using a SAS Autoexec file, see [Specifying a SAS Autoexec File](#).

Getting Started

Initial Access Control

When you initialize a metadata repository, a repository access control template is created and applied to the repository. A repository access control template controls access to a particular metadata repository and to resources that are registered in that repository for which definitive access controls are not specified. You can designate one repository ACT for each metadata repository at your site.

The repository ACT is also called the **Default ACT**, as shown in the following illustration.

Default ACT Available Under Authorization Manager



By default, all users who can access the metadata server are granted ReadMetadata and WriteMetadata permission to the repositories on the SAS Metadata Server. If your environment is not under change management, then ReadMetadata and WriteMetadata are the only two metadata permissions that are required to

- define metadata about servers, spawners, libraries, and other resources in a metadata repository.
- connect and disconnect from a SAS Workspace Server, SAS Stored Process Server, or SAS OLAP Server.

Note: For more information about granting or denying access to metadata resources, see the help for the Authorization Manager.

Pooling and Load Balancing

Pooling and Load Balancing

The following table shows the pooling and load balancing configurations that are supported for each type of IOM server:

Supported Pooling/Load Balancing Configurations		
IOM Server Type	Pooling	Load Balancing
SAS Workspace Server with COM connection	yes	no
SAS Workspace Server with IOM Bridge connection	yes	yes
SAS Stored Process Server with IOM Bridge Connection	no	yes (required), by either or both of these methods: <ul style="list-style-type: none">• Adding multiple stored process servers to a load-balancing logical stored process server• Adding multiple connections to each stored process server within a load-balancing logical stored process server
SAS OLAP Server	no	no
SAS Metadata Server	no	no

Project Installation Note: For the project installation (using SAS Configuration Wizard), the servers are configured as follows:

- SAS Workspace Servers are set up as standard servers
- SAS Stored Process Servers are set up as load-balancing stored process servers that each have three multibrige connections. The load-balancing stored process servers are all configured within the same load-balancing logical server (cluster).

Pooling and Load Balancing SAS Workspace Servers

For SAS Workspace Servers, in addition to the standard configuration, you can also choose to set up pooling or load balancing.

- For COM connections, you can set up pooling to improve the efficiency of connections between clients and servers. For details, see [Overview of Pooling](#).
- For IOM Bridge connections, you can set up either of these configurations:
 - ♦ Pooling, to improve the efficiency of connections between clients and servers. See [Overview of Pooling](#).
 - ♦ Load balancing, to distribute the server workload between processes or machines. See [Overview of Load Balancing](#).

For IOM Bridge connections, you must also set up additional [Pooling Security](#) or [Load Balancing Security](#).

To choose between pooling and load balancing, see [Choosing Pooling or Load Balancing](#).

Load Balancing SAS Stored Process Servers

For SAS Stored Process Servers, you *must* use load balancing. Pooling and standard configurations are not supported. For more information, see [Overview of Load Balancing](#).

Pooling and Load Balancing

Choosing Pooling or Load Balancing

For SAS Workspace Servers, you can choose between standard, pooling, and load balancing configurations.

Note: For SAS Workspace Servers, it is recommended that you use pooling.

- Pooling and load balancing are most useful for applications that require the use of a server for a short period of time. For example, pooling and load balancing are useful for Web applications, such as JavaServer Pages (JSPs).
- Pooling and load balancing are least useful for applications that acquire a server and use the server for a long period of time.

Note: When you use pooling or load balancing, applications might not connect to the same server that they used for a previous connection. Therefore, a particular state is not automatically maintained between connections. If state must be maintained between connections, applications must save the state via a mechanism that is accessible to all servers in the pool or load-balancing cluster.

To choose whether to use pooling or load balancing, the administrator must determine whether the client-side applications are (or will be) coded for pooled servers or for standard servers.

- Application code that is written for pooled servers requires the administrator to set up pooling.
- Application code that is written for standard non-pooled servers can also be used for load balancing servers.

Additionally, use the information in the following table to determine whether you should choose to pool or load balance your servers.

Pooling or Load Balancing Decision Table		
Implementation Features	Pooling	Load Balancing
Implementation Location:	Client-side implementation. Pool can only be shared within a process on a single machine.	Server-side implementation. Load-balancing cluster can be shared from multiple client processes and machines.
Fault Recovery:	Single point of failure on the one machine where the pool is defined.	If one machine in a load-balancing cluster fails, the cluster continues to function.
Connection Establishment:	Client machine establishes a connection and keeps that connection.	New connection is established for every request.
Security:	Security supplied by mapping users to puddles.	SAS server authenticates the caller's credentials.
Client-Side Coding Version:	Requires SAS Integration Technologies Release 8.2 or later client application components.	Requires SAS Integration Technologies 9.1 or later client application components.
Process Distribution:	The spawner starts the first server process for the first client that requests a connection to the pool. When a client subsequently connects, if the process is available,	The spawner starts a new process for each client that requests a connection to the

	a new workspace is created in the same process instead of creating a new process.	load-balancing cluster.
--	---	-------------------------

For more information about pooling, see [Overview of Pooling](#). For more information about load balancing, see [Overview of Load Balancing](#).

Pooling and Load Balancing

Overview of Pooling

Note: For SAS 9.1, you can only use pooling with SAS Workspace Servers.

Pooling occurs at the client side and allows a client to use an existing pooled connection (server process) instead of establishing a connection and starting a SAS server every time a client wants to use SAS.

Pooling enables you to create a pool of connections to IOM servers. Without pooling, a connection (server process) must be created for each client connection request and must remain available for that client, regardless of the client's level of activity. These dedicated connections consume resources that are then unavailable for future client connections.

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

When to Use Pooling

Pooling is most useful for applications that require the use of an IOM server for a short period of time. Because pooling reduces the wait that an application incurs when establishing a connection to SAS, pooling can reduce connection times in environments where one or more client applications make frequent but brief requests for IOM services. For example, pooling is useful for Web applications, such as JavaServer Pages (JSPs).

Pooling is least useful for applications that acquire an IOM server and use the server for a long period of time. A pooled connection does not offer any advantage in applications that use connections for an extended period of time.

Note: For Windows clients, you can choose between SAS Integration Technologies pooling or COM+ pooling. For details, see [Choosing SAS Integration Technologies or COM+ Windows Client Pooling](#) in the *SAS Integration Technologies Developer's Guide*.

Setting Up Pooling

For IOM Bridge connections, you can set up additional SAS Integration Technologies security for pooling. For details, see [Planning the Pooling Security \(IOM Bridge only\)](#). For information about where to specify the parameters that are needed to set up SAS Integration Technologies pooling, see [Locations for Specifying Pooling Parameters](#).

Pooling and Load Balancing

Locations for Specifying Pooling Parameters

You can specify parameters for SAS Integration Technologies pooling in different locations depending on which type of server connection and client coding you are using.

Java Clients

For Java clients using an IOM Bridge connection, you can specify pool parameters in either of the following locations:

- **SAS Metadata Server.** Use SAS Management Console to specify pool parameters on the SAS Metadata Server. For information about how to plan and set up pooling using SAS Management Console, see [Pooling Metadata](#).
- **Source code.** For information about providing pooling information in the Java client source code, see [Using Connection Pooling with Java](#) in the *SAS Integration Technologies Developer's Guide*.

Windows Clients

For Windows clients, you can specify pool parameters in either of the following locations:

- **SAS Metadata Server.** Use SAS Management Console to specify pool parameters on the SAS Metadata Server. For information about how to plan and set up pooling using SAS Management Console, see [Pooling Metadata](#) (for IOM Bridge connections) or [Pooling Metadata](#) (for COM/DCOM connections).
- **Source code.** For information about Windows programming calls to the pooled object, see [Using Connection Pooling with Windows](#) in the *SAS Integration Technologies Developer's Guide*.

Pooling and Load Balancing

Overview of Load Balancing

Load balancing occurs at the server side and helps to balance work across server processes on the same or separate machines. Object spawners manage load balancing across servers.

You can set up load balancing for IOM Bridge connections. You cannot set up load balancing for COM connections.

Note: Release 8.2 (and previous) clients cannot connect to a Version 9 or later load balancing server.

When to Use Load Balancing

You can use load balancing with SAS Workspace Servers and SAS Stored Process Servers. For SAS Stored Process Servers, you *must* use load balancing. Load balancing is most useful when you need to do one of the following:

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

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

For example, use load balancing with

- **Web applications.** When you configure load balancing for SAS Stored Process Servers that are accessed in Web applications, load balancing gives you scalability and reliability.
- **SAS Enterprise Guide.** When you configure SAS Enterprise Guide to connect to a load-balancing logical server (e.g., a single point of connection), load balancing distributes client connections across multiple machines.

How Load Balancing Works

You can use load balancing to balance work across server processes that are used to handle client requests. These server processes can exist on the same machine or across multiple machines. Load balancing works with single-user (SAS Workspace Servers) and multi-user servers (SAS Stored Process Servers).

Load balancing occurs within a set of machines called a cluster. Each machine in the cluster runs an object spawner that handles client requests for connections. A load balancer routine runs in the object spawner and directs client requests to the SAS process that is least loaded at the time the client request is made. Subsequent calls are then direct calls between the client and SAS. For an example of this process, see the Load Balancing Scenario.

Note: Each client's credentials must be able to authenticate against any server in the load-balancing cluster. Therefore, when you define servers within a load-balancing cluster, you must use the same authentication domain for each server.

When launching a load-balancing spawner, you must specify a metadata configuration file that contains information for accessing a SAS Metadata Server. The spawner processes information as follows:

1. The spawner uses the metadata configuration file to connect to and read metadata from the metadata server.
2. The metadata from the metadata server is then used to determine which machines or ports are in the cluster.
3. The spawner then attempts to establish an IOM connection to each spawner in the cluster.

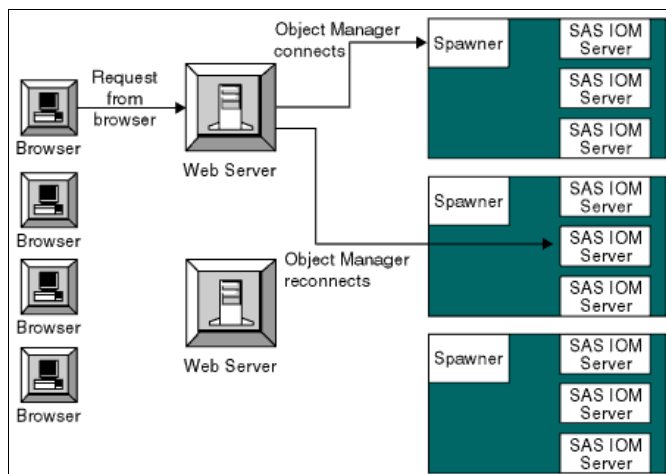
If one machine in the cluster becomes unavailable, then the other machines in the cluster detect that the machine is unavailable and continue to run and process any client requests.

Note: You can launch load-balancing spawners at any time:

- If the spawner is already configured in the metadata as being associated with a server in the load-balancing cluster, the spawner is immediately included as part of the load-balancing cluster.
- If you configure a new spawner (associated with a server in the load-balancing cluster) in the metadata after the other spawners have started, you must restart the currently running spawners in order to access the new spawner's metadata and allow it to participate in the load-balancing cluster.

Note: Because a client can connect to any machine, client programs should not depend on being able to reconnect to the same server.

Scenario



The following is a typical load balancing scenario, in which a Web application connects to SAS using the Windows Object Manager (or the Java Connection Factory):

1. The Web Server receives a request from the browser.
2. The Object Manager (running in the Web server) calls to the spawner.
3. The spawner tells the Object Manager to connect back to a given machine and port.
4. The Object Manager connects directly to SAS, giving the Web application a direct connection to SAS.

When the Web application disconnects, SAS notifies the spawner that the load on that server has changed.

MultiBridge Connections (SAS Stored Process Servers only)

MultiBridge connections allow multiple processes to run under the same server definition. When you set up load balancing for SAS Stored Process Servers, you *must* set up MultiBridge connections.

Each server in a cluster should have its own set of MultiBridge connections. Load balancing uses MultiBridge connections to redirect clients to different processes defined for the same or other servers within the load-balancing cluster. When a client connects to an IOM Bridge port that is defined on a server, if MultiBridge connections are

available, load balancing selects one of the MultiBridge connections to redirect the client to for a server connection.

When you define a MultiBridge connection, you define a unique port for the connection. Each MultiBridge connection then defines a different process for your server definition. For example, if you define a server definition with three MultiBridge connection definitions, you have three processes.

Note: MultiBridge connections use the server's credentials. When using MultiBridge connections, you must specify multi-user login credentials on the server definition.

Administration (SAS Stored Process Servers only)

For SAS Stored Process Servers, you can use the administrator command `cluster reset` to shut down load-balancing servers in a cluster. For details, see [Using Telnet to Administer the Spawner](#).

Note: The `cluster reset` command only affects servers that were launched from the spawner to which you are currently connected.

Security

With load balancing, every connection to the server is authenticated with the credentials of the client; depending on the type of server, the process then runs under the following credentials:

- for SAS Workspace Servers, the credentials of the client
- for SAS Stored Process Servers, the multi-user login credentials that are specified in the stored process server definition (**Advanced Options ▶ Credential**) in SAS Management Console.

Note: Because the load-balancing stored process server runs under the multi-user login credentials, the operating system account for these credentials must have access to any operating system resources used by stored processes that are hosted on the stored process server.

You can use normal server security mechanisms to protect sensitive data. For more information about server security, see the [Security](#) chapter.

Algorithms

Load balancing supports two different types of load-balancing algorithms:

Cost (SAS Workspace Servers and SAS Stored Process Servers)

The cost algorithm assigns a cost value (determined by the administrator) to each client that connects to the server. The algorithm can also assign cost values to servers that have not started yet. When a new client requests a connection, load balancing redirects the client to whichever server is determined to have the lowest cost.

Response Time (SAS Stored Process Server only)

Each spawner's load balancer maintains an ordered list of machines and their response times. Load balancing updates this list periodically, at an interval determined by the user. When a new client requests a connection, load balancing redirects the client request to the machine at the top of the list (i.e., a round-robin approach to the list).

For more details about load-balancing algorithms, see [Determining the Load-Balancing Parameters](#).

Setting up Load Balancing

To plan and set up load balancing, see [Planning the Configuration Metadata](#) and [Load Balancing Metadata](#).

COM/DCOM

Setting Up a COM/DCOM Connection

Introduction

SAS can be configured to enable client access through Component Object Model (COM) interfaces. A COM connection can be established either locally (on the same machine) or remotely (on a different machine). For remote connections, the Distributed Component Object Model (DCOM) interface is used.

Because COM launches the SAS server, spawners are not used in the COM/DCOM server environment. However, you can (and should) use the SAS 9.1 Object Manager (or SAS System Version 8 Workspace Manager) to obtain a DCOM server. The server definitions can be administered through a metadata server.

DCOM must be enabled on both the client machine and on the machine where the IOM server runs. The server machine requires additional configuration for DCOM object access and launch permissions.

When to Use a Server with a COM/DCOM Connection

For SAS Workspace Servers, SAS OLAP Servers, and SAS Metadata Servers, you can configure a server with a COM/DCOM connection in either of the following situations:

- the server will be installed on a Windows machine and will be accessed by Windows client applications running on remote machines. In this situation, the connection uses DCOM.
- the server will be installed on a Windows machine and will be accessed by Windows client applications running on the same machine. In this situation, the connection uses COM.

Caution: Use of the COM protocol to launch the SAS Metadata Server is experimental in SAS 9.1. Do not use this combination as a production environment.

If you use a Java client, you must use an IOM Bridge server configuration (or both configurations for OLAP and metadata servers, which can support both COM and IOM Bridge connections simultaneously).

COM/DCOM

Server and Client Requirements

SAS supports Windows NT 4 (Server and Workstation), Windows 2000, Windows XP, and Windows 2003 as either client or server machines. Windows 98 is not supported.

Server Requirements

Install the following software on the server machine:

- SAS 9.1 (or later)
- SAS Integration Technologies
- any other SAS products that your application will use

COM/DCOM

Summary of Setup Steps (COM/DCOM)

Standalone Windows Development Machine

To set up a standalone Windows development machine, simply install SAS 9.1 (including SAS Integration Technologies) on the machine. On Windows, the SAS Integration Technologies client is installed with Base SAS software.

You can then develop your Windows client application as described in [Developing Windows Clients](#) in the *SAS Integration Technologies Developer's Guide*. To use the server in a Visual Basic environment, for example, you would reference the IOM type libraries from within your Visual Basic project (refer to [Programming with Visual Basic](#) in the *SAS Integration Technologies Developer's Guide* for details).

For more information about developing Windows client applications, see [Windows Clients](#) in the *SAS Integration Technologies Developer's Guide*.

Separate Client and Server Machine

To set up the server machine:

1. Install SAS 9.1 (including SAS Integration Technologies) on the server machine. Refer to the SAS documentation for the details of this procedure.

Note: If you are using the SAS Integration Technologies client with 64-bit SAS, extra setup steps are required for IOM servers with a COM connection on 64-bit Windows. For details, see the SAS installation documentation.

2. Enable DCOM on the server machine. For details, see [Enabling DCOM on the Server and the Client](#).
3. Edit your SAS CONFIG file (SASV9.CFG) for use with DCOM. For details, see [Configuring SAS for DCOM](#).
4. Set SAS launch policies on the server. You can set global policies that affect all COM-enabled applications, or set application policies for individuals to grant permissions to users and groups specifically for accessing and launching the server. For details, see [Setting SAS Permissions on the Server](#).
5. Before attempting to run a COM/DCOM application, test the client/server connection by using the tips that are provided in [Troubleshooting a COM/DCOM Connection](#).
6. If your applications need to access metadata that describes your COM/DCOM server configuration, you must create the necessary metadata, including definitions for servers. You can use SAS Management Console to create the necessary metadata on the SAS Metadata Server.

Note: COM-based workspace servers do not necessarily require a definition within the SAS Metadata Server. The most common reason for defining a COM-based workspace server in metadata would be to enable clients to connect by logical server name, without needing to know the actual network location. Pooling parameters also require a metadata definition. Another possible reason to define a workspace server in metadata is so that you can use the METAUTOINIT option to enable metadata-based LIBNAME preassignments or OBJECTSERVERPARMS.

OLAP servers always require a definition within the SAS Metadata Server.

For planning details, see [Planning Your Server Configuration Metadata](#).

SAS® 9.1.2 Integration Technologies: Server Administrator's Guide

For details about using SAS Management Console to create the metadata, see [Creating the Metadata Using SAS Management Console](#).

7. Start the server. For details, see [Starting Servers](#).

To set up the client machines:

1. On each client machine, enable DCOM. For details, see [Enabling DCOM on the Server and the Client](#).
2. On each client machine, install the client software, either as part of the installation of a pre-written application or as a separate installation of a custom application. For details about installing custom applications, refer to [Developing Windows Clients](#) in the *SAS Integration Technologies Developer's Guide*.

This completes the basic configuration steps that are necessary to do client development on a Windows platform. For information about developing applications that access COM/DCOM servers, refer to [Developing Windows Clients](#) in the *SAS Integration Technologies Developer's Guide*.

COM/DCOM

Planning Your Server Configuration Metadata

To plan your SAS Application Server and logical server configuration metadata, determine the following:

- the number of SAS Application Servers
- the number and type of logical servers within each SAS Application Server.

To plan your other server configuration metadata, see the following topics:

- Standard Server Metadata
- Pooling Metadata (SAS Workspace Servers only)

COM/DCOM

Standard Server Metadata

For a server with a COM connection, you must decide if the server needs to be defined in SAS metadata. An OLAP server with a COM connection is required to be defined in SAS metadata; otherwise, a definition in SAS metadata is not necessary except in cases such as the following:

- You want to enable clients to connect to the server by logical server name without knowing the actual network location.
- You are setting up pooled logical servers (the pooling parameters are stored in metadata).
- You want to use the METAAUTOINIT option to enable metadata–based LIBNAME preassignments or OBJECTSERVERPARMs.

If you decide that your server needs to be defined in metadata, then you need to create the appropriate set of metadata definitions to describe your server configuration.

For information about the SAS Application Server and logical server definitions that contain the server definitions, see [Planning for Metadata Definitions](#).

To plan a standard server with a COM connection, you must determine the following:

- **How many servers do you need?** Decide how many servers you need for your implementation.
- **How many logical servers and SAS Application Server contexts do you need?** Decide which logical servers and SAS Application Server contexts will contain your server definitions.

To set up a server with a COM connection, you must plan and set up metadata for the servers. You must plan and define the servers that you will use to process client requests.

Planning for Servers

To plan each server, you must determine the following information:

- the server name
- the host name
- object server parameters, as required. For details, see [Object Server Parameters](#).
- SAS startup command and options, as required. For details, see [Server Startup Command](#).

For detailed information about the fields included in the metadata for a server, see the [Fields for the Server Definitions](#).

Defining the Servers

Use SAS Management Console to define the servers within the appropriate SAS Application Server and logical server. A server definition with a COM connection will specify that clients can connect to the server using COM. For detailed information about using SAS Management Console to add a new server definition, see [Using SAS Management Console to Define Servers](#).

COM/DCOM

Pooling Metadata

For SAS Workspace Servers, you can set up pooling for a group of servers within a logical server. Before you can plan and set up pooling, you must understand pooling (see [Overview of Pooling](#) for more information).

To plan a server pool, you must determine the following:

- The number of servers (one or more) in the pool.
- The standard server metadata. For each server in the pool, plan and set up the standard server and login definitions that are specified in [Standard Server Metadata](#).

Note: With COM, you can specify only one puddle for a pool, and you cannot specify a login for the puddle.

To set up a pool, you must plan and set up additional metadata for the following:

1. [Plan Pooling Security](#). To set up pooling security, you must plan the SAS group that can access the puddle.
2. [Plan Pooled Logical Server and Puddle](#). To set up a pool, you must plan converting a standard logical server to a pooled logical server, puddle, pooling parameters, and a SAS group that is granted access to the puddle.
3. [Plan for Server\(s\)](#). To set up each server for pooling, you must plan pooling parameters for each server.
4. [Set up Pooling Security](#). To set up pooling security, you must define the SAS groups and SAS group membership.
5. [Set up Pooled Logical Server and Puddle](#). To set up a pool, you must convert a standard logical server to a pooled logical server, create the puddle, specify pooling parameters, and associate a SAS group that can access the puddle.
6. [Set up Servers](#). To set up each server for pooling, on each server definition, you must specify pooling parameters for the server.

Step 1: Plan Security for Server Pooling

To plan security for server pooling, you must determine the SAS groups that can access the puddle in the pool and the person who will be authorized to update that SAS group.

Note: The ID that the SAS Workspace Server is started with is determined by the process ID of the process that calls `CreatePoolByServer` or `CreatePoolByLogicalName`. This ID is used to connect to the puddle.

Step 2: Plan the Pooled Logical Server and Puddle

When you convert the logical server to a pooled logical server, you can then define a puddle that associates the appropriate SAS group to use for access to the pool. The SAS users in the SAS group that is granted access to the puddle are also granted access to the servers that are in the puddle. To plan and set up the pooled logical server and puddle, do the following:

Determine the following parameters for each puddle associated with the pooled logical server definition.

- [Name](#) of the puddle
- [Minimum Available Servers](#)
- [Minimum Number of Servers](#)
- [SAS Group](#) that is granted access to the puddle

Step 3: Plan Pooled Servers

To plan server pooling, you must determine pooling parameters for the servers that are contained in the pooled logical server.

For each server in the pooled logical server, determine the appropriate pooling parameters.

- [Maximum Clients](#)
- [Recycle Activation Limit](#)
- [Inactivity Timeout](#)

Step 4: Set up Pooling Security

To set up pooling security, set up your SAS group definition and the person who is authorized to update that SAS group. To understand SAS group definition structure, and how to set up a group of SAS users, see [Defining SAS Users, Groups, and Logins](#).

Control access to the SAS group that is granted access to the puddle. You must control access for who is authorized to update the SAS group that is granted access to the puddle. To control who can update the SAS group that is granted access to the puddle, in SAS Management Console, after you set up the SAS group, use the Authorization tab for the SAS group to do both of the following:

- Deny "Write" permission to the Public group
- Grant "Write" permission to your metadata administrator

Control access to the pooled logical server. You must control access for who is authorized to update the pooled logical server. To control who can update the pooled logical server, in SAS Management Console, you must use the Authorization tab for the pooled logical server to do both of the following:

- Deny "Write" permission to the Public group
- Grant "Write" permission to your metadata administrator

Step 5: Set up Pooled Logical Servers

Use SAS Management Console to define a pooled logical server and puddle. To convert a logical server to a pooled logical server and define puddles, see [Using SAS Management Console to Define a Pooled Logical Server \(COM\)](#).

Step 6: Set up Pooled Servers

For each server in the pool, use SAS Management Console to specify pooling parameters on the server. To modify a server and set up the server pooling parameters, see [Adding Pooling Parameters to an Existing Server](#). To define a server and set up the server pooling parameters, see [Using SAS Management Console to Define Servers.COM/DCOM](#)

Creating Metadata Using SAS Management Console

If your applications need to access metadata from the SAS Metadata Server, you must create metadata that describes your server configuration.

If you are using the SAS Metadata Server, you can use the SAS Management Console graphical user interface to create and modify the metadata for your server configuration. For information about SAS Management Console, from the SAS Management Console menu bar, select **Help ➔ Help on SAS Management Console**. For Help on the fields in a particular window, click **Help** in that window.

Before you can create definitions on your SAS Metadata Server, you must set up a SAS Metadata Server. You must also use SAS Management Console to create a repository. For details, see [🌐 SAS Metadata Server: Setup Guide](#).

For instructions about how to add new servers, see [Using SAS Management Console to Define Servers](#).

For instructions about how to add custom parameters, see the following:

- [Using SAS Management Console to Define Custom Parameters for Workspace Servers \(COM/DCOM\)](#)
- [Using SAS Management Console to Define an OLAP Server \(COM/DCOM\)](#)

COM/DCOM

Using SAS Management Console to Define Servers

The SAS Management Console Server Manager provides a graphical user interface that allows you to create or modify a definition for a server with a COM connection. For details about using SAS Management Console, from the SAS Management Console menu bar, select **Help ► Help on SAS Management Console**. For more information about the fields in the New Server Wizard, click **Help** from within the wizard.

For an overview of SAS Application Server contexts and logical server groupings, see [Planning Your Server Configuration Metadata](#).

Before you begin defining servers, you must have a metadata profile for connecting to a metadata repository. For details about setting up this profile, see [SAS Metadata Server: Setup Guide](#).

To define a server with a COM connection using SAS Management Console:

1. Start SAS Management Console and connect to a metadata repository.
2. Choose the appropriate method for defining your server:
 - ◆ **New SAS Application Server, logical server, and server.** To define a server and logical server in a new SAS Application Server, see [Defining a Server and Logical Server in a New SAS Application Server Context](#).
 - ◆ **New logical server and server.** To define a server in an existing SAS Application Server but within a new logical server, see [Defining a Server and Logical Server in an Existing SAS Application Server](#).
 - ◆ **New server.** To define a server in an existing SAS Application Server and existing logical server, see [Defining a Server in an Existing Logical Server](#).

Defining a Server and Logical Server in a New SAS Application Server

To define a new server, logical server, and SAS Application Server context:

1. From the navigation tree, select the Server Manager, then select **Actions ► New Server** from the menu bar.
The New Server Wizard appears. A list of resource templates is displayed.
2. Select **SAS Application Server**. Click **Next**.
3. Enter the Name and Description. The name that you specify will be the name of the SAS Application Server context. Click **Next**.
4. Verify that the Minor Version Number, Major Version Number, Software Version, and the Vendor are correct. Click **Next**. A list of defined server resource templates is displayed.
5. Select the type of server you want to define. The type that you choose will be the type of the first logical server and server in the SAS Application Server context. For example, if you select workplace server as the server type, then the SAS Application Server context will contain a logical workspace server which in turn contains a workspace server. Click **Next**.
6. Select **Custom** and click **Next**.
7. Verify that the Minor Version Number, Major Version Number, Software Version, and the Vendor are correct.

The **Command** field is not used with COM; leave this field unchanged. The **Object Server Parameters** field

is optional. It can be used to supplement the OBJECTSERVERPARMS values that are specified on the server invocation command line. See [Server Startup Command](#) for details.

8. To continue configuring the server, see the appropriate topic for your type of server:

◆ [SAS Workspace Server](#)

◆ [SAS OLAP Server](#)

Note: Because a SAS Stored Process Server should be run by a load–balancing spawner, it should be defined with an IOM Bridge connection only. Running a SAS Stored Process Server with a COM connection is not recommended.

Defining a Server and Logical Server in an Existing SAS Application Server

To define a new server and new logical server in an existing SAS Application Server context:

1. From the navigation tree, expand the Server Manager and locate the SAS Application Server context under which you want to add the new server. The SAS Application Servers are located one folder level below the Server Manager. Select the appropriate SAS Application Server context, and then select **Actions ➤ Add Application Server Component** from the menu bar. The New Server Component Wizard appears. A list of server resource templates is displayed.

Note: A SAS Application Server context can contain only one logical server of each of the following types: SAS Workspace Server, SAS Metadata Server, SAS Stored Process Server, and SAS OLAP Server.

2. Select the type of server you want to define. Click **Next**.
3. Select **Custom**. Click **Next**.
4. Enter the Name and Description. The name that you specify will be the name of the server. Click **Next**.
5. Verify that the Minor Version Number, Major Version Number, Software Version, and the Vendor are correct.

The **Command** field is not used with COM; leave this field unchanged. The **Object Server Parameters** field is optional. It can be used to supplement the OBJECTSERVERPARMS values that are specified on the server invocation command line. See [Server Startup Command](#) for details.

6. To continue configuring the server, see the appropriate topic for your type of server:

◆ [SAS Workspace Server](#)

◆ [SAS OLAP Server](#)

Note: Because a SAS Stored Process Server should be run by a load–balancing spawner, it should be defined with an IOM Bridge connection only. Running a SAS Stored Process Server with a COM connection is not recommended.

Defining a Server in an Existing Logical Server

To define a new server in an existing logical server and SAS Application Server:

1. From the navigation tree, expand the **Server Manager**, then select and expand the SAS Application Server context that contains the logical server under which you want to add the new server. Select the appropriate logical server, and then select **Actions ➤ Add Server** from the menu bar. The New Server Wizard appears.

SAS® 9.1.2 Integration Technologies: Server Administrator's Guide

2. Enter the Name and Description. The name that you specify will be the name of the server. Click **Next**.
3. Verify that the Minor Version Number, Major Version Number, Software Version, and the Vendor are correct.

The **Command** field is not used with COM; leave this field unchanged. The **Object Server Parameters** field is optional. It can be used to supplement the OBJECTSERVERPARMS values that are specified on the server invocation command line. See Server Startup Command for details.

4. To continue configuring the server, see the appropriate topic for your type of server:

- ◆ SAS Workspace Server
- ◆ SAS OLAP Server

Note: Because a SAS Stored Process Server should be run by a load–balancing spawner, it should be defined with an IOM Bridge connection only. Running a SAS Stored Process Server with a COM connection is not recommended.

COM/DCOM

Using SAS Management Console to Modify Servers

SAS Management Console provides a graphical user interface that allows you to modify a definition for a server with a COM connection. For details about using SAS Management Console, from the SAS Management Console menu bar, select **Help ➤ Help on SAS Management Console**. For more information about the fields in a particular window, click **Help** in that window.

Modifying an Existing Server's Properties

To modify a server definition with a COM connection using SAS Management Console:

1. Start SAS Management Console and connect to a metadata repository.
2. In the navigation tree, expand the Server Manager to find the server object that you want to modify.
3. Select the server object, and then select **File ➤ Properties** from the menu bar.
4. Select the appropriate tabs, and enter the necessary changes. For a description and location of the fields, refer to the [Fields for the Server Definition](#). When you are finished, click **OK** to return to the SAS Management Console main window.

Adding a COM Connection

To add a connection using SAS Management Console:

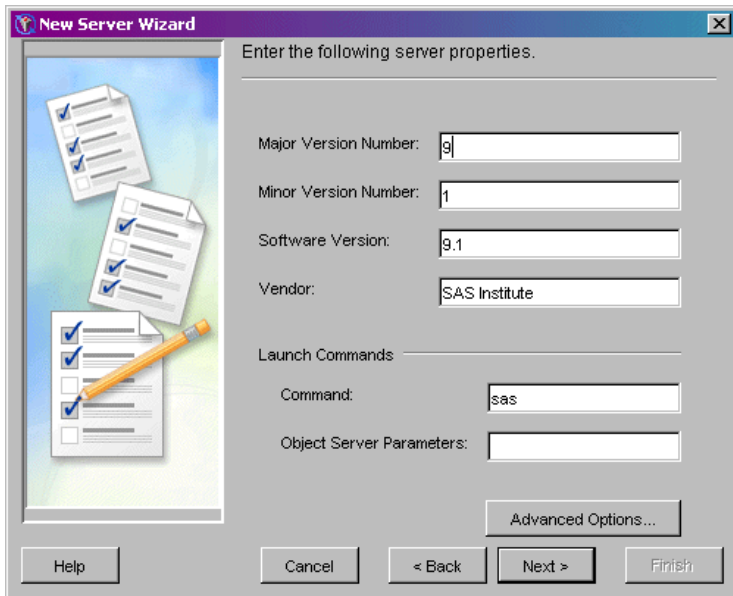
1. Start SAS Management Console and connect to a metadata repository.
2. In the navigation tree, expand the Server Manager to find the server object that you want to modify.
3. Select the server object, and then select **Actions ➤ Add Connection** from the menu bar. The New Connection Wizard appears.
4. Select **COM Connection**. (If a COM connection is already defined for this server, then the **COM Connection** selection is not available; click **Cancel**.)
5. Click **Next**.
6. Enter a **Name** and optionally, a **Description** for the connection. Click **Next**.
7. Enter the machine name ([HostName](#)) for the machine on which the server will run. Click **Next**.
8. Click **Finish** to add the connection and return to the SAS Management Console main window.

For a description and location of the fields, refer to the [Fields for the Server Definition](#).

COM/DCOM

Using SAS Management Console to Define Custom Parameters for a Workspace Server (COM/DCOM)

In order to define custom workspace server parameters, you must already have begun to add a server according to the instructions in [Using SAS Management Console to Define Servers](#). The New Server Wizard's Server Options window will be displayed as follows:

The screenshot shows the 'New Server Wizard' dialog box, specifically the 'Server Options' window. The title bar reads 'New Server Wizard'. The main area is titled 'Enter the following server properties.' and contains several input fields: 'Major Version Number' with the value '9', 'Minor Version Number' with '1', 'Software Version' with '9.1', and 'Vendor' with 'SAS Institute'. Below these is a section for 'Launch Commands' with a 'Command' field containing 'sas' and an empty 'Object Server Parameters' field. An 'Advanced Options...' button is located below the launch commands section. At the bottom of the dialog are buttons for 'Help', 'Cancel', '< Back', 'Next >', and 'Finish'.

To finish defining a server with a COM connection using SAS Management Console:

1. If you want to specify pooling properties, then click **Advanced Options**.

In the Advanced Options dialog box, select the Pooling Properties tab.

Specify the maximum number of clients in the pool ([Maximum Clients](#)) and the recycle activation limit ([RecycleActivationLimit](#)). If you want to shut down inactive servers, then select the **Inactivity Timeout** check box and specify the inactivity timeout ([InactivityTimeout](#))

When you are finished entering information in the Advanced Options dialog box, click **OK**.

2. Click **Next**.
3. Select the **COM** connection. Click **Next**.
4. Fill in the machine name ([HostName](#)) for the machine on which the server will run. Doing so enables clients to ask for this logical server by name and then be connected to the machine where the logical server is running.
5. Click **Next**.
6. Click **Finish** to create the SAS Workspace Server definition.

COM/DCOM

Using SAS Management Console to Define an OLAP Server (COM/DCOM)

An OLAP server is a high-capacity, multi-user data manipulation engine specifically designed to support and operate multi-dimensional data structures.

Use SAS Management Console to create the OLAP server definition. For details about SAS Management Console, from the SAS Management Console menu bar, select **Help ▶ Help on SAS Management Console**. For help about the fields in a particular window, click **Help** in that window.

The following documents provide additional information and Help for SAS OLAP Server:

- SAS OLAP Server Administrator's Guide
- SAS OLAP Server Help
- SAS OLAP Administrator Online Help

COM/DCOM

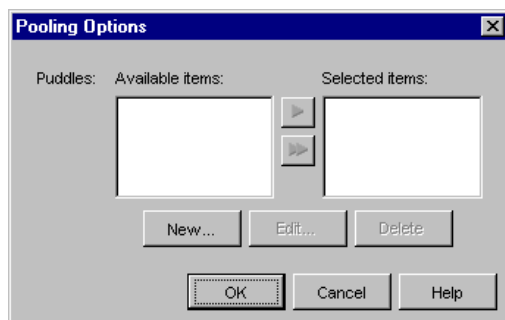
Using SAS Management Console to Define a Pooled Logical Server (COM)

SAS Management Console provides a graphical user interface that enables you to convert a logical server to a pooled logical server. For details about using SAS Management Console, from the SAS Management Console menu bar, select **Help ► Help on SAS Management Console**. For more information about the fields in a particular window, click **Help** in that window.

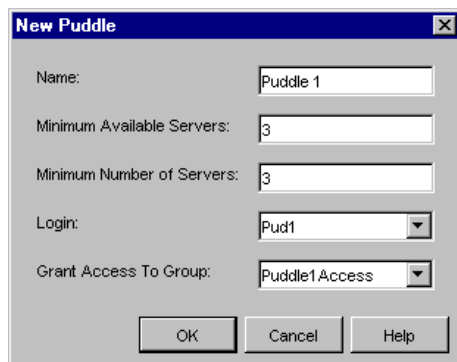
You can only convert workspace logical servers to pooled logical servers.

To convert a logical server to a pooled logical server using SAS Management Console:

1. Start SAS Management Console and connect to a metadata repository.
2. From the navigation tree, expand the Server Manager and select the logical server that you want to convert to a pooled logical server.
3. Select **Actions ► Convert to ► Pooling** from the menu bar. The Information dialog box appears.
4. Click **Yes** to continue. The Pooling Options window appears.



5. Click **New** to create a new puddle. The New Puddle window appears.



For the new puddle, enter the following:

- ◆ the name of the puddle (Name)
- ◆ the minimum number of connections that need to be available (Minimum Available Servers)
- ◆ the minimum number of connections to create when the pool is created (Minimum Number of Servers)

- ◆ the group that can have puddle access (Group).

Note: For COM, you cannot specify a **Login** for a puddle.

When you are finished entering the puddle parameters, click **OK**. The Pooling Options window appears and contains the new puddle.

6. From the Pooling Options window, you can do one of the following:

- ◆ select a puddle and click **Edit** to edit the puddle.
- ◆ select a puddle and click **Delete** to delete the puddle.

Note: For COM, you can create only one puddle on a pooled logical server.

Note: If you want to delete a puddle, then you must select the puddle and click **Delete**. Do not select a puddle and move it from the Selected Items list to the Available Items list. If you move a puddle to the Available Items list, then when you click **OK** the puddle will no longer appear as available to the pool; however, the puddle will still be stored in a SAS Metadata Repository and will consume memory on the repository's machine.

When you are finished defining, editing, or deleting the puddle, click **OK**.

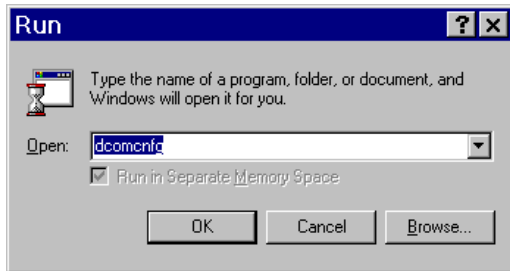
For a description and location of the fields, refer to the [Fields for the Pooled Logical Server and Puddle Definitions](#).

COM/DCOM

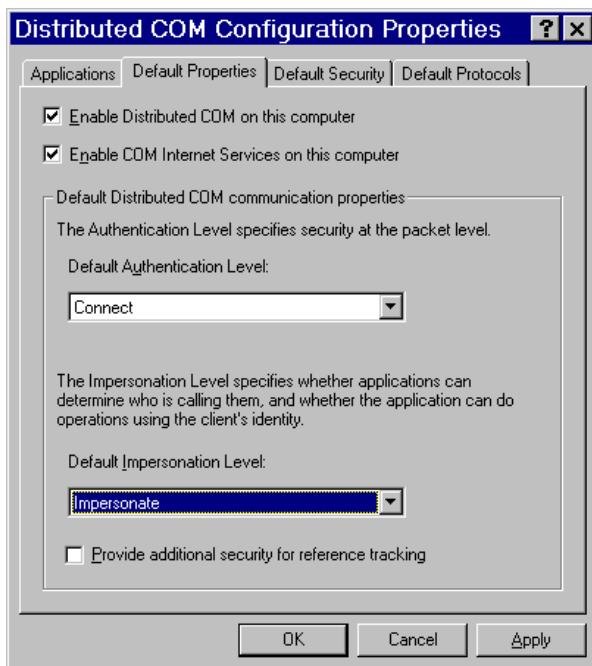
Enabling DCOM on the Server and the Client

To establish a DCOM session, you must ensure that DCOM is enabled on the server machine and on each client machine. Perform the following steps on each machine:

1. From the Windows Taskbar, click **Start ➔ Run**.
2. Type `dcomcnfg`, as shown in the illustration.



3. Click **OK**. The dialog box that appears depends on the Windows operating system you are using:
 - ◆ If you are using Windows NT/2000, the Distributed COM Configuration Properties dialog box appears.
 - ◆ If you are using Windows XP, the Component Services dialog box appears. Expand the Component Services folder, expand the Computers folder, then right-click on My Computer and select Properties.
4. Select the **Default Properties** tab.



Note: The dialog box might look slightly different than the illustration, depending on the version of Windows you are running and which Service Pack you have applied.

5. Select **Enable Distributed COM on this computer**.
6. COM uses the Default Authentication Level when a client or server does not provide a specific value, either programmatically or on the Applications tab (which creates an AppID-based setting in the Windows registry). For Default Authentication Level, choose the value that is most appropriate for applications that do not have a

specific setting of their own. This value will not be used by an IOM server if you set its authentication level individually using the Application tab (see Setting Permissions per Application on Windows NT/2000 and Windows XP).

Select an Authentication level of **Connect** to provide a good balance between security and system performance. More restrictive security levels can be required based on the needs of your site and your users. For a description of additional levels, consult the Windows NT Help.

Note: Currently, event output from the SAS server sent to client applications cannot be encrypted due to Microsoft COM restrictions.

7. It is recommended that you select an Impersonation Level of **Impersonate**.

This completes the steps necessary to enable DCOM on the clients and servers.

COM/DCOM

Configuring SAS for DCOM

The COM Service Control Manager (SCM), which launches single user servers such as the IOM Workspace, does not load a user profile or environment. As a result, SAS sessions launched via DCOM are not initialized with the user's home directory (typically `C:\Documents and Settings\<User Name>\My Documents`), environment variables or other profile settings.

The default SAS CONFIG file on Windows (`!SASROOT\nls\<Language Code>\SASV9.CFG`) contains a definition for SASUSER that contains the Windows shell enumeration `%CSIDL_PERSONAL`. For local SAS sessions, this enumeration refers to the user's home directory. However, when SAS is invoked by DCOM, `%CSIDL_PERSONAL` resolves to a system folder that can usually only be accessed if the client has administrator privileges at the server.

To correct this issue, you must edit the `-SET MYSASFILES` and `-SASUSER` commands in `SASV9.CFG` to refer to a location that all users can access. Additionally, because the `-SASUSER` setting will be shared, you should specify the `-RSASUSER` option to ensure that none of the users update the user settings.

Default Lines from SASV9.CFG:

```
/* Setup the MYSASFILES system variable */
-SET MYSASFILES "%CSIDL_PERSONAL\My SAS Files\9.1"

/* Setup the default SAS System user profile folder */
-SASUSER "%CSIDL_PERSONAL\My SAS Files\9.1"
```

Recommended Change:

```
/* Setup the MYSASFILES system variable */
-SET MYSASFILES "%CSIDL_COMMON_DOCUMENTS\My SAS Files\9.1"

/* Setup the default SAS System user profile folder */
-SASUSER "%CSIDL_COMMON_DOCUMENTS\My SAS Files\9.1"
-RSASUSER
```

On Windows XP, this change would typically place SASUSER at `C:\Documents and Settings\All Users\Documents\My SAS Files\9.1`.

On most systems, this path would be accessible to everyone. If you choose another path, you must make sure that all of your potential users have read permissions in that directory.

If you use SAS without IOM on the same system, you might want to create a separate default `SASV9.CFG` file. See [Customizing the Startup Command for Workspace Servers](#) for details on how to update the COM startup command to specify a different file in the `-CONFIG` option.

COM/DCOM

Setting SAS Permissions on the Server (COM/DCOM)

On the machine where the server runs, you must identify who can access and launch the server. A client that needs services from a multi-user server, such as an OLAP server running as a Windows service, must have access permissions for that server. A client that needs a single user server, such as a workspace server, must have both access and launch permissions on the server application. These permissions are defined in terms of one or more Windows users or groups.

There are two ways to identify users and groups that have launch or access permission. One way is to define permissions that are specific to a server application. The other way is to specify them in the default permissions. The default permissions are used for server applications that do not have their own application-specific permissions. Because an arbitrary COM server could potentially have significant capabilities over the system, it is usually best to keep the default launch and access permission well restricted, for example, to Administrators and the System account. Granting access permissions to users and groups on a per-application basis allows those users to access a particular application without permitting them to use other COM servers that might be installed on the server machine.

Each particular server application has a name that is listed in DCOMCNFG. When executing as a COM server, the application identifies itself with an AppID, which is a UUID that identifies the application in the Windows registry. DCOMCNFG enables you to select the server application and update the Windows registry settings to control the security policy for that particular application. In SAS System 9, each type of IOM server has its own name, permission policy settings, and AppID. [AppIDs for Configuring DCOM](#) lists each of these.

These methods are discussed in the following sections:

- [Setting Default COM Security on Windows NT/2000](#)
- [Setting Permissions per Application on Windows NT/2000](#)
- [Setting Default COM Security on Windows XP](#)
- [Setting Permissions per Application on Windows XP](#)

COM/DCOM

Setting Default COM Security on Windows NT/2000

Default COM security affects all COM applications that do not have launch permissions of their own.

- If authentication is used, client machines that receive events from a DCOM server must include "Everyone" in the default access permissions.
- If authentication is not used, client machines must specify "None" in the default access permissions.

To set default COM security on Windows NT/2000:

1. From the Windows taskbar, click **Start ▶ Run**.
2. Type `dcomcnfg` and click **OK**. The Distributed COM Configuration Properties window appears.
3. Select the Default Security tab.

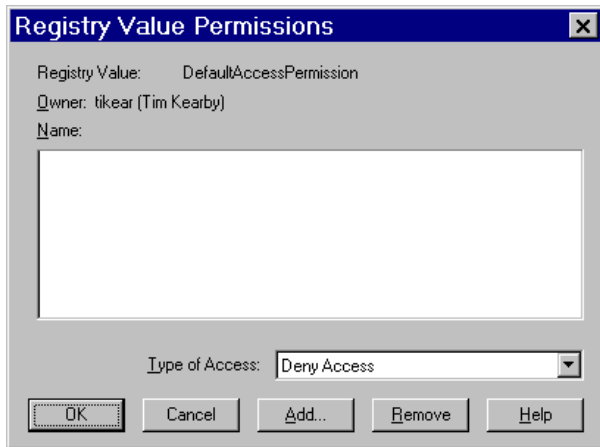


4. From the Default Security tab, you must edit the Default Access Permissions and the Default Launch Permissions. (The Default Configuration Permissions are adequate for a development environment). For details, see the following topics:
 - ◆ [Global Access Permissions](#)
 - ◆ [Global Launch Permissions](#)
 - ◆ [Global Configuration Permissions](#)

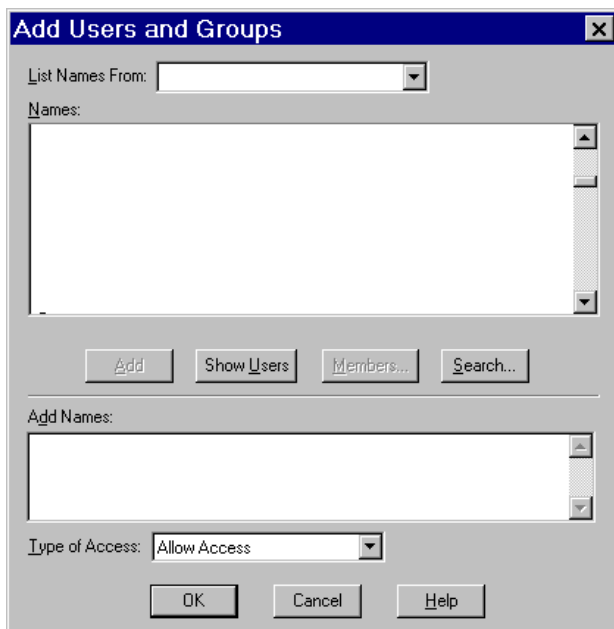
Global Access Permissions

To set global access policies for selected users and groups from the Default Security tab of `dcomcnfg`:

1. In the Default Access Permissions box, click **Edit Default**. The following dialog box appears, showing current registry settings for the Default Access Permissions:



2. To add users and groups to the list, click **Add**. The Add Users and Groups dialog box appears.

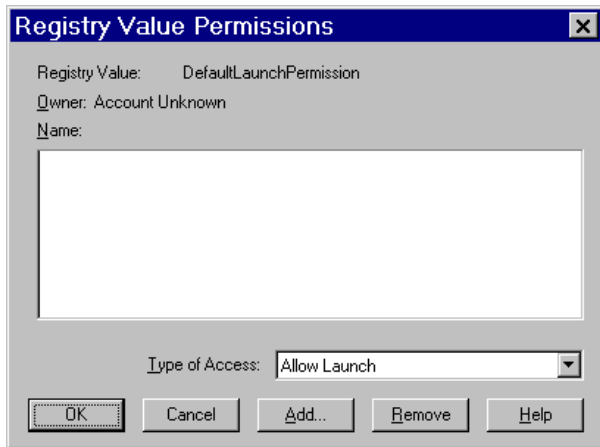


3. Use the Add Users and Groups dialog box to identify the users and groups at your site and the type of access (allow or deny access). You should also give access permission to System. For field descriptions, refer to the Windows NT or Windows 2000 Help. When you are finished, click **OK** and then **OK** again to return to the Default Security tab.

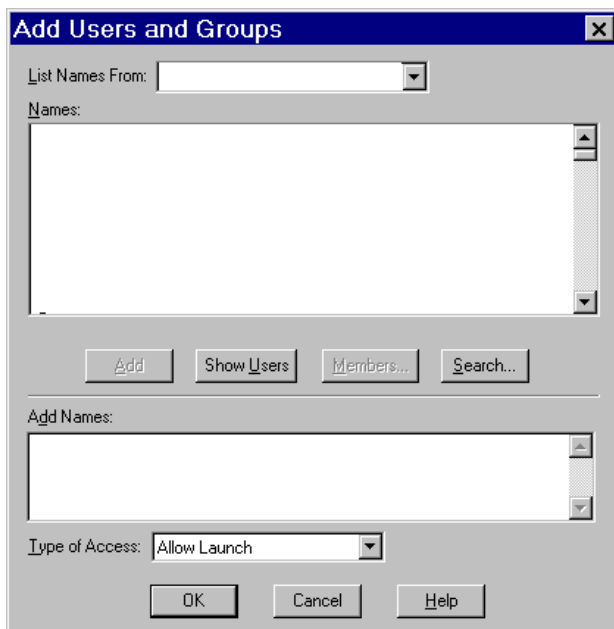
Global Launch Permissions

To set global launch permissions for selected users and groups from the Default Security tab of dcomcnfg:

1. In the Default Launch Permissions box, click **Edit Default**. The following dialog box appears, showing the current registry settings for Default Launch Permissions.



2. Click **Add** to add users and groups to the list. The following dialog box appears.



3. Use the Add Users and Groups dialog box to identify users and groups at your site and the type of access (allow or deny launch access). It is recommended that you enter the same values that you entered for the Default Access Permissions. You should also give launch permission to System. When you are finished, click **OK** and then **OK** again to return to the Default Security tab.

Global Configuration Permissions

To set global configuration permissions for selected users and groups from the Default Security tab of dcomcnfg:

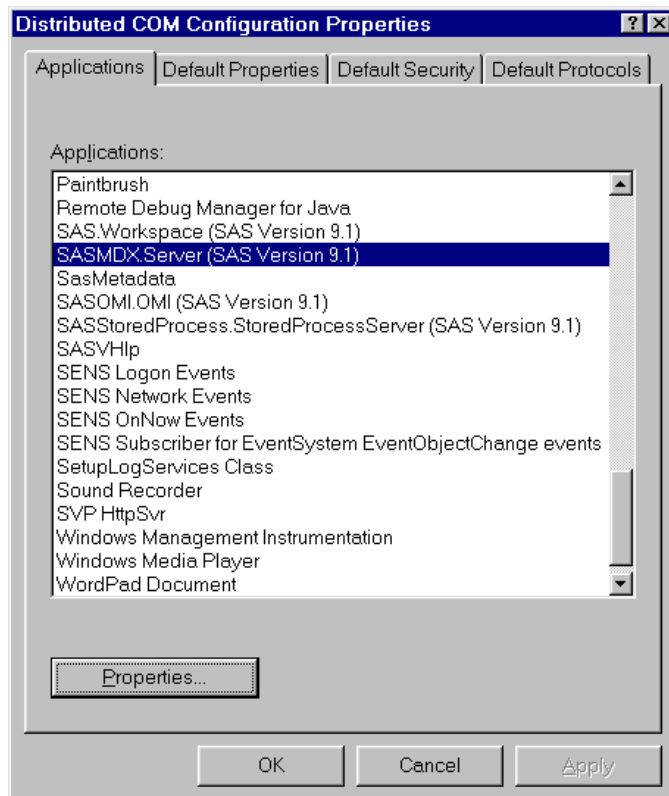
1. If you need to specify more restrictive configuration permissions, from the Default Security tab of dcomcnfg, click **Edit Default** in the Default Configuration Permissions box. Consult the Windows NT or Windows 2000 Help for further information.
2. When you are finished, click **OK** to save the new settings and exit from the dcomcnfg utility.

COM/DCOM

Setting Permissions per Application on Windows NT/2000

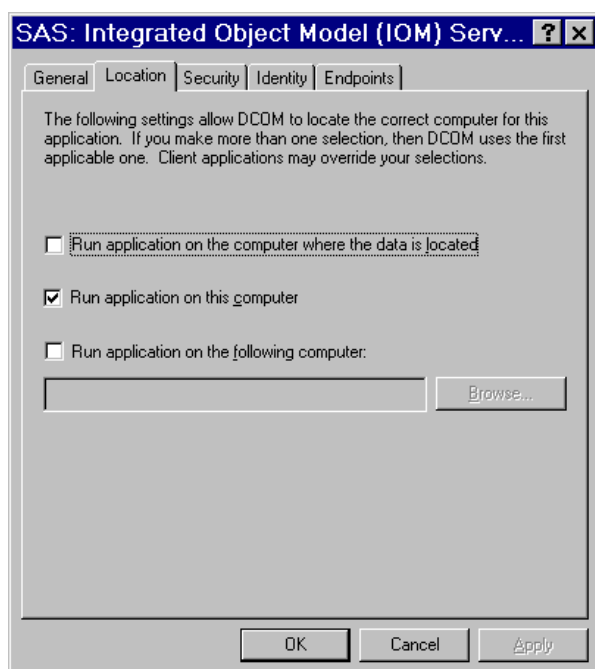
To grant permissions to users and groups specifically for accessing and launching the SAS server (instead of defining global permissions as shown in the previous section):

1. From the Windows taskbar, click **Start ▶ Run**.
2. Type `dcomcnfg` and click **OK**. The Distributed COM Configuration Properties window appears.
3. Select the Applications tab:

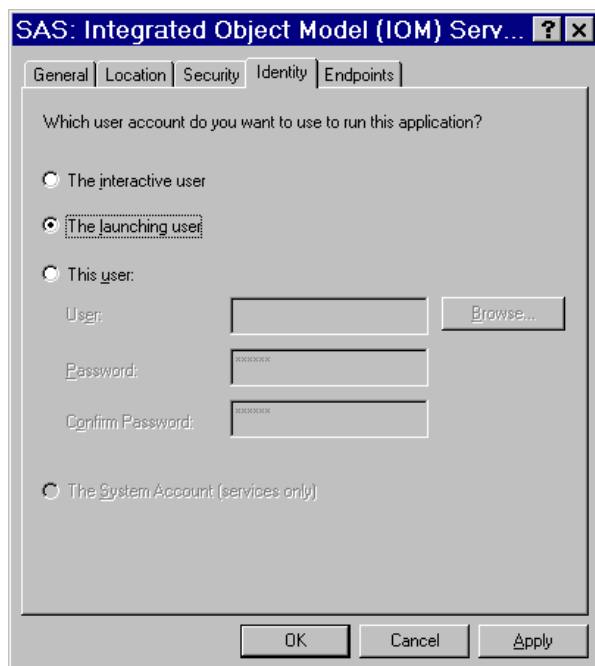


This tab shows the AppID description for each DCOM server that can be run on your machine. (The AppID GUID is shown for servers that register without a description.)

4. Locate the IOM server that you are configuring and select it. For example, if you want to set policies for the Workspace, select **SAS.Workspace (SAS Version 9.1)**. The application listing differs depending on which version of SAS is installed. See [AppIDs for Configuring DCOM](#) to determine which AppID to look for.
5. After you highlight the selection, click on the **Properties** button. The Properties dialog box for the server object appears.
6. Select the Location tab.



7. Check the default location setting. By default, the only option enabled is **Run application on this computer**, as shown in the illustration. No other options are required for SAS applications.
8. Select the Identity tab.



9. Select the identity based on the type of server:

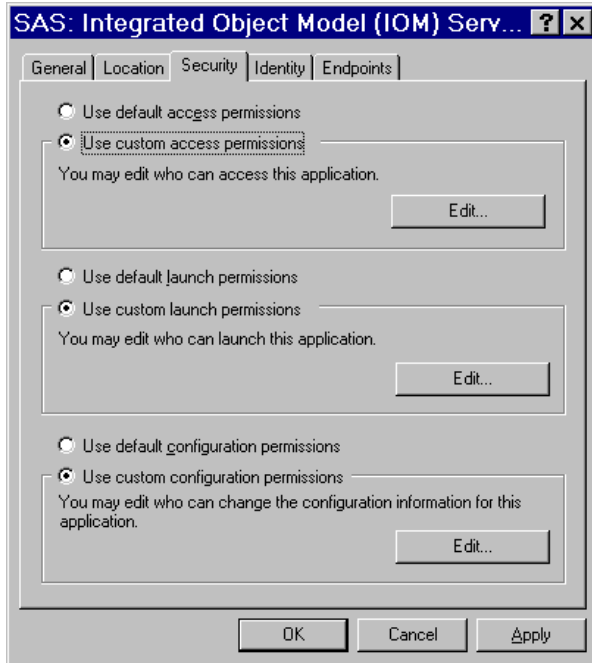
- ◆ For multi-user servers (SAS Metadata Server, and the SAS System 9 OLAP server), select **This user** and specify the **User**, **Password**, and **Confirm Password** information.

CAUTION: Support for the use of COM in the SAS Metadata Server is experimental in SAS 9.1. Do not use COM in the SAS Metadata Server in production jobs.

- ◆ For SAS Workspace Servers, check the desired default identity setting. For maximum security, select the option **The launching user**.

Note that some versions of Windows prevent servers with COM connections that are configured with the "This user" identity choice to be run from a command prompt. The recommended approach for multiuser servers is to install them as a Windows service, generally with "protocol=(com,bridge)" in order to support the maximum possible range of clients. See [Choosing a Server Configuration](#) for details.

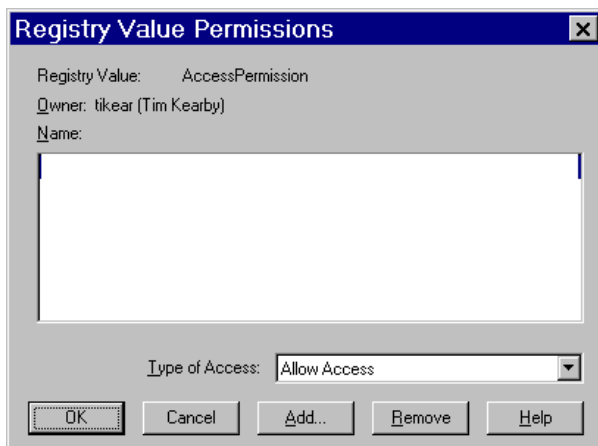
10. Select the Security tab.



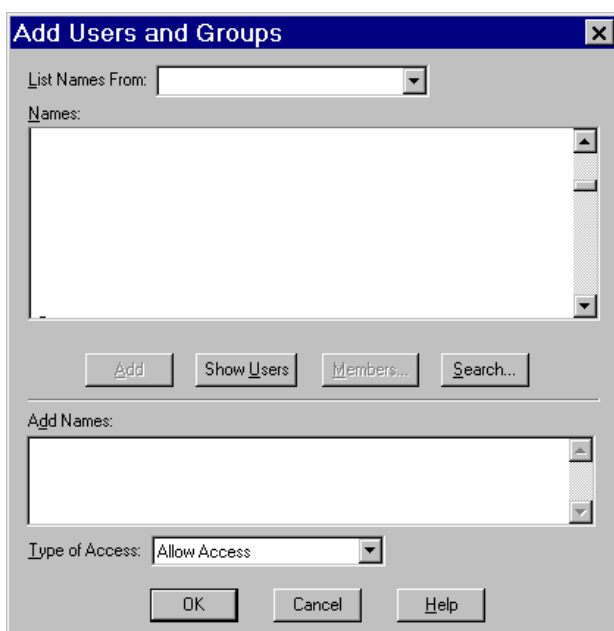
11. To use the system-wide default access permissions, select **Use default access permissions**, click **Apply**, and then continue with Step 12.

To give IOM server application its own set of access permissions:

- a. Select **Use custom access permissions** and click the adjacent **Edit** button. The Registry Value Permission dialog box appears:



- b. Select **Add**. The Add Users and Groups dialog box appears.

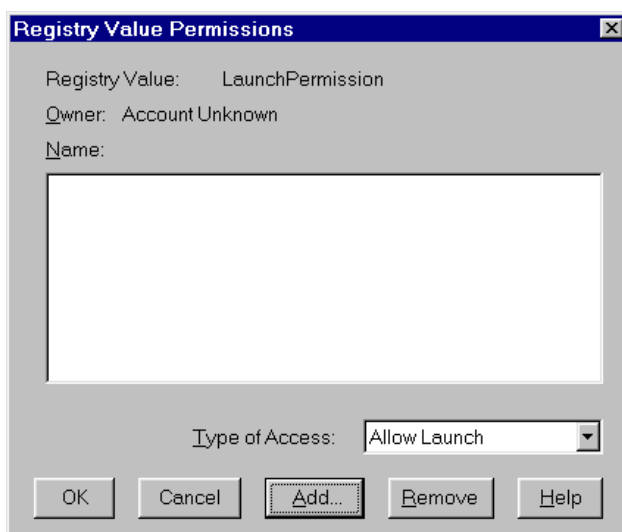


- c. Use this dialog box to grant users and groups access to SAS through DCOM. You should also give access permission to System. (For field descriptions, refer to the Windows NT Help.) You can also identify users and groups that are denied access to SAS by changing the selection in Type of Access.
 - d. When you are finished, click **OK** in the Add Users and Groups dialog box, and then click **OK** in the Registry Value Permissions dialog box.
12. If you are configuring a Workspace server, which is launched by COM, you will also need to choose your launch permissions. It is recommended that they be the same as the access permissions; additionally, ensure that the **System** account has launch permissions.

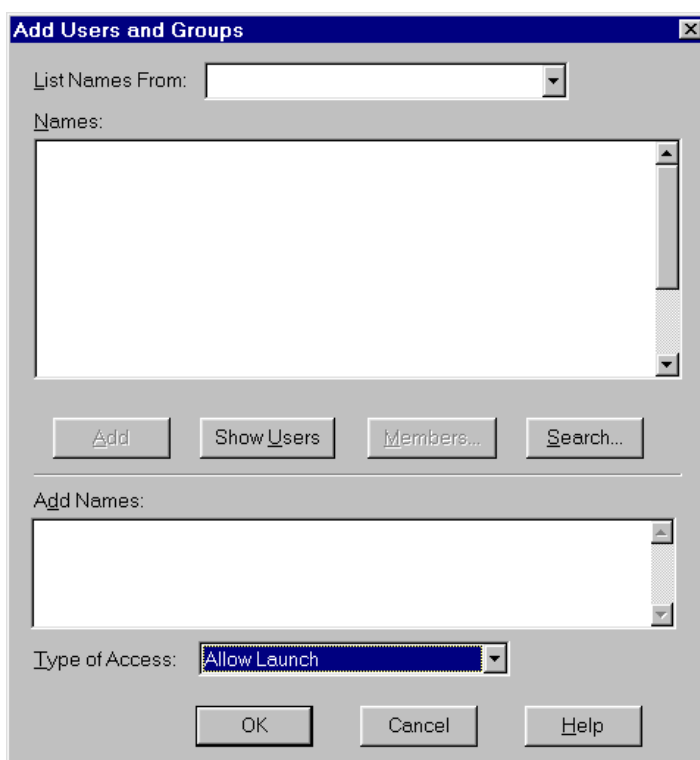
To use system-wide default launch permissions, select **Use default launch permissions**, click **Apply**, and then continue with Step 13.

To give the IOM server application its own set of launch permissions:

- ◆ On the Security tab, select **Use custom launch permissions** and click the adjacent **Edit** button. The Registry Value Permissions dialog box appears.



- ◆ Select **Add**. The Add Users and Groups dialog box appears.



- ◆ Use this dialog box to grant users and groups access to SAS through DCOM. It is recommended that you enter the same values that you entered for the Custom Access Permissions. You should also give launch permission to System. (For field descriptions, refer to the Windows NT or Windows 2000 Help.) When you are finished, click **OK**.

Note: If you grant launch permissions for an application to specific users and groups, then you might affect those users who previously had permission to the application through default permissions.

13. Click **OK** in each of the open dialog boxes to save your selections and exit the dcomcnfg utility.

COM/DCOM

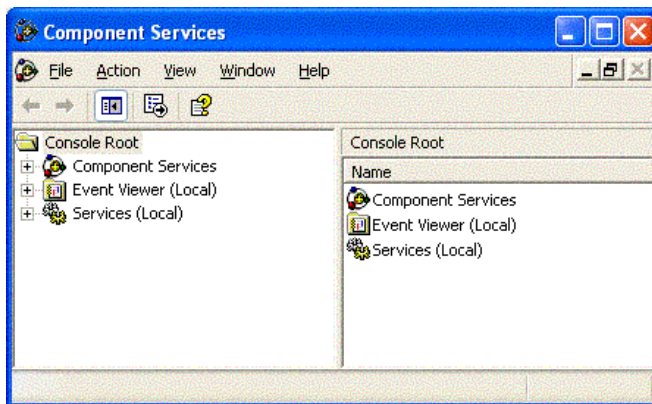
Setting Default COM Security on Windows XP

Default COM security affects all COM applications that do not have launch permissions of their own.

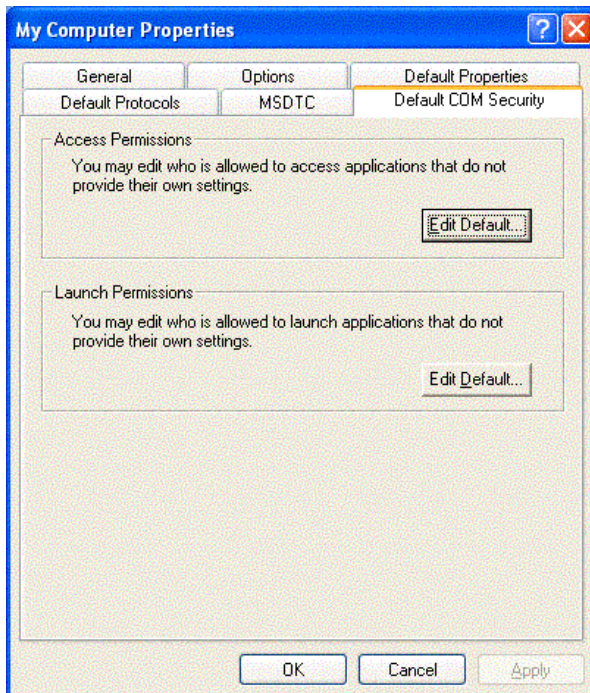
- If authentication is used, client machines that receive events from a DCOM server must include "Everyone" in the default access permissions.
- If authentication is not used, client machines must specify "None" in the default access permissions.

To set default COM security on Windows XP:

1. From the Windows taskbar, click **Start ▶ Run**.
2. Type `dcomcnfg` and click **OK**. The Component Services window appears.



3. Expand the Component Services folder and expand the Computers folder. Right-click the My Computer folder and select **Properties**.
4. Select the Default COM Security tab.



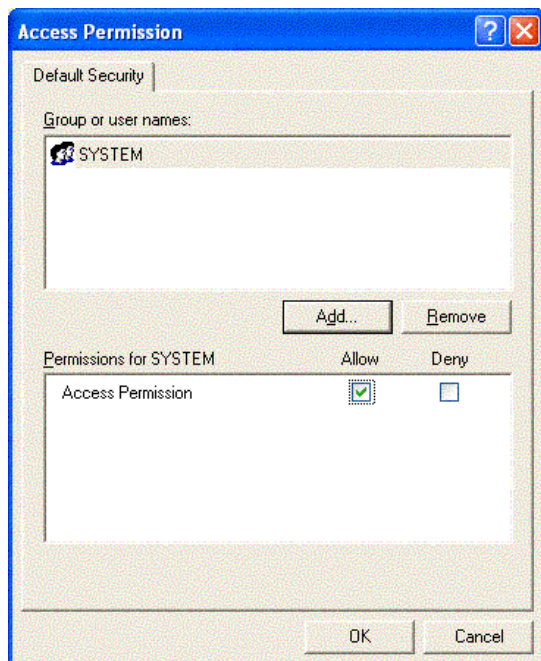
5. From the Default COM Security tab, you must edit the Access Permissions and the Launch Permissions. For details, see the following topics:

- ◆ Global Access Permissions
- ◆ Global Launch Permissions

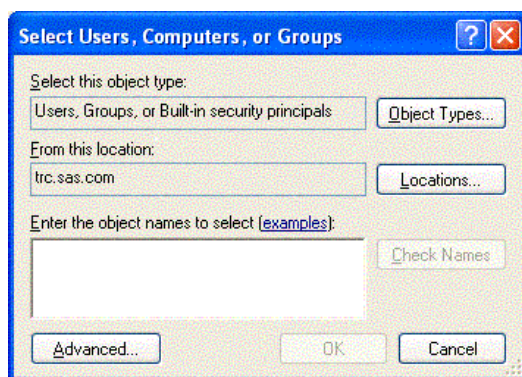
Global Access Permissions

To set global access policies for selected users and groups from the Default COM Security tab of dcomcnfg:

1. In the Access Permissions box, click **Edit Default**. The following dialog box appears, showing current registry settings for the Access Permissions:



2. To add users and groups to the list, click **Add**. The Select Users, Computers, or Groups dialog box appears.

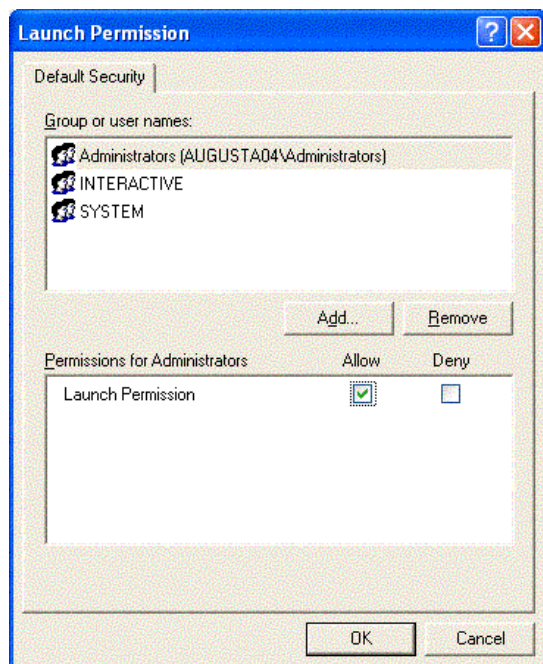


3. Use the Select Users, Computers, or Groups dialog box to identify the users and groups at your site and the type of access (allow or deny access). You should also give access permission to System. For field descriptions, refer to the Windows XP Help. When you are finished, click **OK** and then **OK** again to return to the Default COM Security tab.

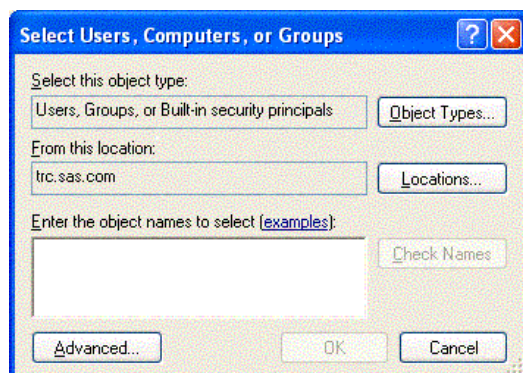
Global Launch Permissions

To set global launch permissions for selected users and groups from the Default COM Security tab of dcomcnfg:

1. In the Launch Permissions box, click **Edit Default**. The following dialog box appears, showing the current registry settings for Launch Permissions.



2. Click **Add** to add users and groups to the list. The following dialog box appears.



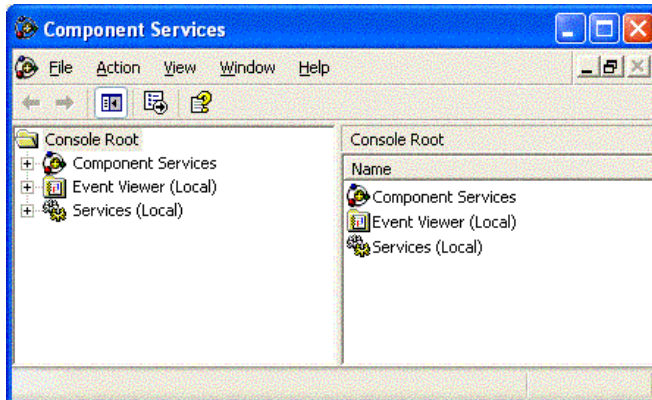
3. Use the Select Users, Computers, or Groups dialog box to identify users and groups at your site and the type of access (allow or deny launch access). It is recommended that you enter the same values that you entered for the Access Permissions. You should also give launch permission to System. When you are finished, click **OK** and then **OK** again to return to the Default COM Security tab.

COM/DCOM

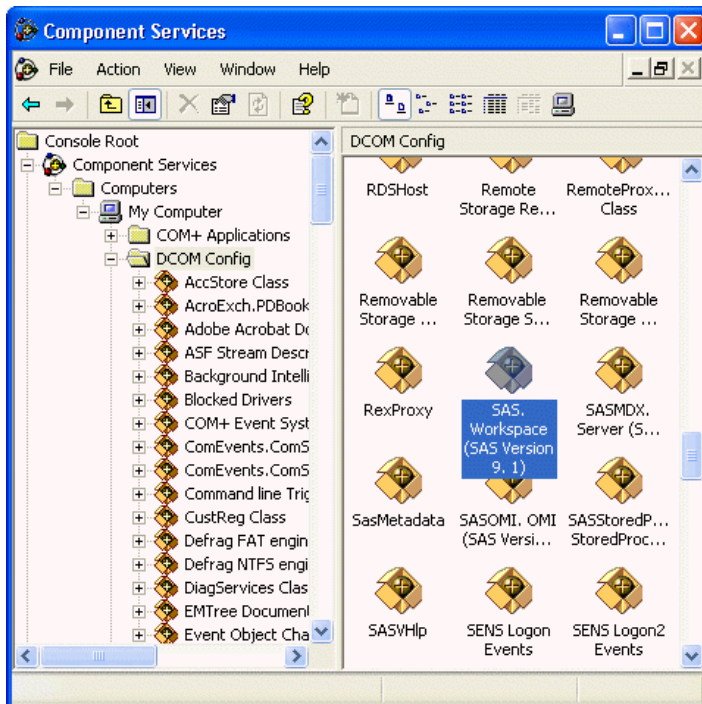
Setting Permissions per Application on Windows XP

To grant permissions to users and groups specifically for accessing and launching the SAS server (instead of defining global permissions as shown in the previous section):

1. From the Windows taskbar, click **Start ▶ Run**.
2. Type `dcomcnfg` and click **OK**. The Component Services window appears.



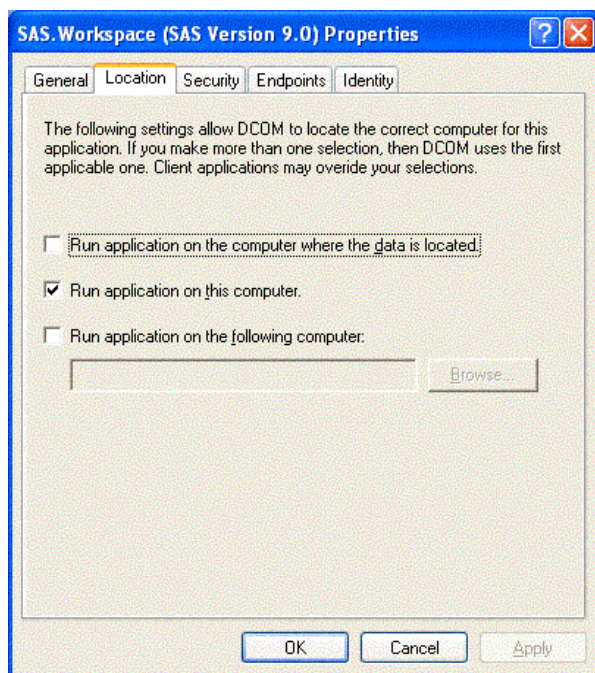
3. Expand the Component Services folder, expand the Computers folder, expand the My Computer folder, and then expand the DCOM Config folder.



This view shows the AppID description for each DCOM server that can be launched on your machine. (The AppID GUID is shown for servers that register without a description.)

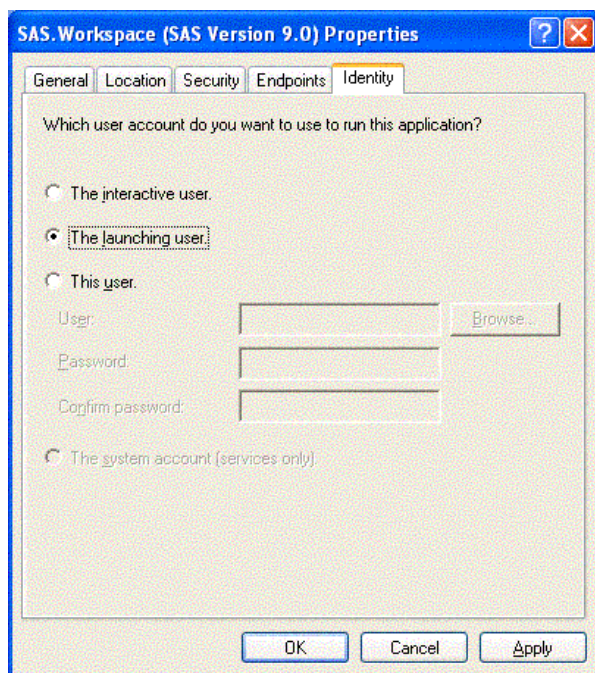
4. Select the AppID for the SAS Integrated Object Model (IOM) Server. The AppID differs depending on which version of SAS is installed. See [AppIDs for Configuring DCOM](#) to determine which AppID to look for.
5. After you highlight the selection, right-click and select **Properties**. The Properties dialog box for the server object appears.

6. Select the Location tab.



7. Check the default location setting. By default, the only option enabled is **Run application on this computer**, as shown in the illustration. No other options are required for SAS applications.

8. Select the Identity tab.



9. Select the identity based on the type of server:

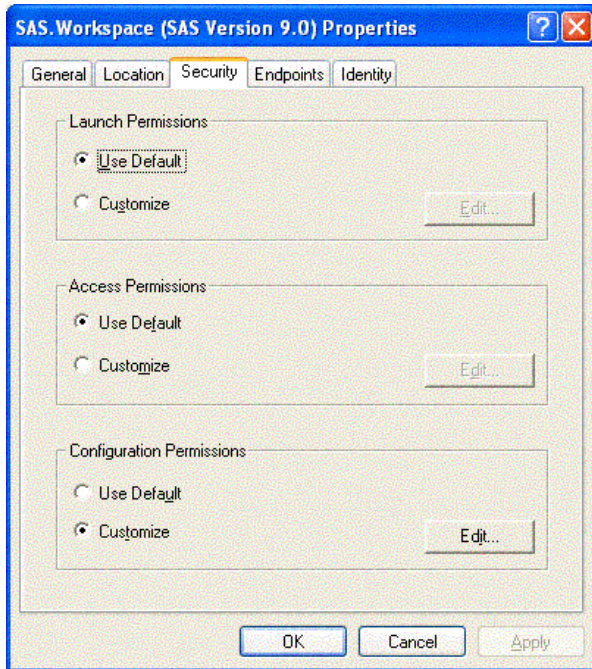
- ◆ For multi-user servers (SAS Metadata Server, SAS System 9 OLAP server), select **This user** and specify the **User**, **Password**, and **Confirm Password** information.

CAUTION: Support for the use of COM in the SAS Metadata Server is experimental in SAS 9.1. Do not use COM in the SAS Metadata Server in production jobs.

- ◆ For SAS Workspace Servers, check the desired default identity setting. For maximum security, select the option **The launching user**.

Note that some versions of Windows prevent servers with COM connections that are configured with the "This user" identity choice to be run from a command prompt. The recommended approach for multiuser servers is to install them as a Windows service, generally with "protocol=(com,bridge)" in order to support the maximum possible range of clients. See [LINK](#) for details.

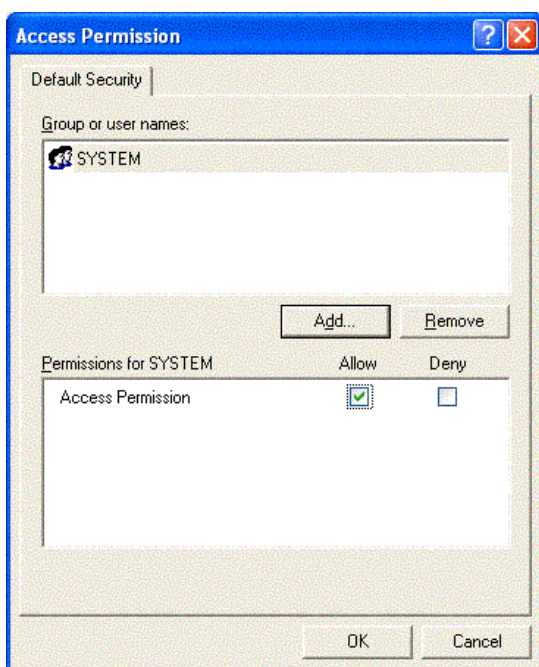
10. Select the Security tab.



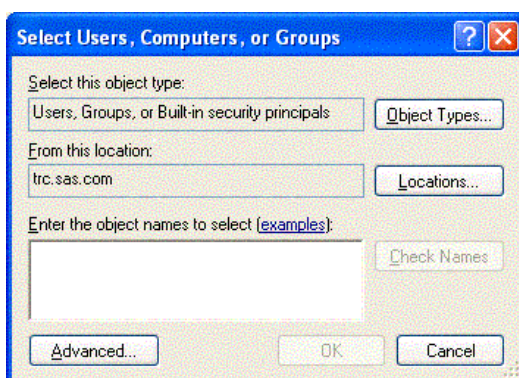
11. If you want to use default access permissions, select **Use Default**, click **OK**, and then continue with Step 12.

If you want to grant access to users who are not in the list of default access permissions:

- a. Select **Customize** and click the adjacent **Edit** button. The Access Permissions dialog box appears:



b. Select **Add**. The Select Users, Computers, or Groups dialog box appears:



c. Use this dialog box to grant users and groups (who are not listed in the Access Permissions) access to SAS through DCOM. You should also give access permission to System. (For field descriptions, refer to the Windows XP Help.) You can also identify users and groups that are denied access to SAS by changing the selection in Type of Access.

d. When you are finished, click **OK** in the Select Users, Computers, or Groups dialog box, and then click **OK** in the Access Permissions dialog box.

12. On the Security tab, in the Launch Permissions box, select **Customize** and click the adjacent **Edit** button. The Launch Permissions dialog box appears.
13. Click **Add**. The Select Users, Computers, or Groups dialog box appears.
14. Use this dialog box to identify users and groups at your site and the type of access (allow or deny launch). It is recommended that you enter the same values that you entered for the Custom Access Permissions. You should also give launch permission to System. (For field descriptions, refer to the Windows XP Help.) When you are finished, click **OK**.

Note: If you grant launch permissions for an application to specific users and groups, you might affect those users who previously had permission to the application through default permissions.

15. Click **OK** in each of the open dialog boxes to save your selections and exit the dcomcnfg utility.

Note: On Windows XP, if you have used the dcomcnfg utility to edit an application's security settings and you have

left the Authentication Level on the General tab as Default, then DCOMCNFG will store the "AuthenticationLevel" value under the HKEY_CLASSES_ROOT\AppID\{hexadecimal-appid} key in the Windows registry with a value of "0". This value is not defined as a supported value by the COM library (which reads these values at runtime to determine your application's security settings). When this occurs, the symptom is "0x80070005 – Access is denied" on the first call from the client to the IOM server.

The easiest workaround is to set the Authentication Level on the General tab to some specific value other than "Default".

For more information about this problem, see Microsoft Knowledge Base Article 814430.

COM/DCOM

Configuring COM/DCOM for Active Server Page Access

Note: You can also configure your Active Server Page (ASP) application to access SAS using the IOM Bridge for COM. You might want to use IOM Bridge for COM if either of the following are true:

- SAS is running on z/OS or a Unix machine
- you want to share a configured SAS server with Java applications.

If you are using the IOM Bridge for COM, the configuration in this section is not required. See [Choosing a Server Configuration](#) for details.

COM/DCOM Configuration

To configure a Windows Active Server Page (ASP) client running in Microsoft Internet Information Services (IIS) for access to a Windows server using DCOM, you must perform two different types of configuration:

1. A basic configuration that is similar to a standard Windows client that accesses a Windows server using DCOM.

To perform this basic configuration, follow the instructions in [Enabling DCOM on the Server and the Client](#).

2. Additional configuration steps that will enable a Web client to access an IOM server. There are two ways that you can access a Windows Server using COM/DCOM:

- ◆ To configure access to a local COM IOM server, see [Accessing a Local COM IOM Server from an Active Server Page](#).
- ◆ To configure access to a remote DCOM IOM server, see [Accessing a Remote DCOM IOM Server from an Active Server Page](#).

Permissions

Use **dcomcnfg** to configure the SAS.Workspace (SAS Version 9.1) application. To configure the DCOM or COM when using ASP, you must change access and launch permissions for the SAS.Workspace (SAS Version 9.1) application. Therefore, you should also familiarize yourself with [Setting Permissions per Application on Windows NT/2000](#) or [Setting Permissions per Application on Windows XP](#).

If you are experienced with using IIS and DCOM and only need to know the permissions required for your setup, see the following table for details about these permissions.

IOM Server	Web Server	DCOM Access Permission (on IOM Server)	DCOM Launch Permission (on IOM Server)	Other Notes
Local COM	IIS 4 All Authentication Methods	<ul style="list-style-type: none"> • System 	<ul style="list-style-type: none"> • System 	
	IIS 5 using Anonymous Access	<ul style="list-style-type: none"> • IUSR_<machine_name> • IWAM_<machine_name> 	<ul style="list-style-type: none"> • IUSR_<machine_name> • IWAM_<machine_name> 	The COM+ application can be configured to be launched as a different user; however, this is not necessary. Refer to Configure your IIS Application to use High (isolated) Application Protection for details.
	IIS 5 using Basic Authentication and Integrated	<ul style="list-style-type: none"> • IWAM_<machine_name> 	<ul style="list-style-type: none"> • IWAM_<machine_name> 	

	<u>Windows Authentication</u>	<ul style="list-style-type: none"> Any valid NT user account that will be accessing the ASP 		
<u>Remote DCOM</u>	<u>IIS 4 All Authentication Methods</u>	<ul style="list-style-type: none"> System Network 	<ul style="list-style-type: none"> System Network 	If you are setting up the remote DCOM IOM server on a Windows 2000 or XP computer, you must configure the DCOM server to run as a different user than the launching user.
	<u>IIS 5 All Authentication Methods</u>	<ul style="list-style-type: none"> User account launching IIS COM+ application Network 	<ul style="list-style-type: none"> User account launching IIS COM+ application 	The COM+ application must be configured so it is launched as a user that exists on both the Web server and DCOM IOM server. Refer to <u>Configure your IIS Application to use High (isolated) Application Protection</u> for details.

COM/DCOM

Accessing a Local COM IOM Server from an Active Server Page

When you access a local COM IOM server from an Active Server Page (ASP), SAS and Internet Information Services (IIS) are both installed on the same machine.

Note: This configuration is not recommended. If you have SAS and a Web server running on the same machine, they might compete for resources.

To configure local COM IOM in an ASP, you must ensure that the user who is launching the process has the proper permissions. Follow the configuration instructions to configure permissions either for Windows NT 4, or for Windows 2000 and XP.

Configuring Windows NT4 with IIS to Access a Local COM IOM Server

In IIS 4, the System account owns the IIS process and all of its child processes. When the local COM IOM server launches through an active server page (ASP), the launching user is identified as the System account. Use **dcomcnfg** to verify that the System account has launch and access permissions for the SAS.Workspace (SAS Version 9.1) application.

Note: This configuration will work for all of the supported authentication methods in IIS 4.

1. Start **dcomcnfg**.
2. Select **SAS.Workspace (SAS Version 9.1)** and then select **Properties**.
3. Select the Security tab. If the System account does not have access and launch permissions, add the access and launch permissions.

Configuring Windows 2000 or XP with IIS to Access a Local COM IOM Server

In IIS 5, all processes, both pooled and isolated, are now *COM+ Applications*. For this reason, you must configure an additional level of security and add different users to the access and launch permissions for the SAS.Workspace (SAS Version 9.1) application. For more details, refer to [Configuring Windows 2000 or XP with IIS 5 Remote DCOM and COM+ Settings](#).

There are two different types of authentication, [Anonymous Access](#) and [Basic Authentication](#).

Note: If you are using Windows XP as your Web server platform, it is recommended that you use Basic Authentication instead of Anonymous Access.

Anonymous Access

Enabling anonymous access allows all inbound Web clients to use the identity of the IUSR_<machine name> user. The IWAM_<machine name> user launches the IIS process. Therefore, you must configure the following security permissions:

- access permissions for both the IUSR_<machine name> and the IWAM_<machine name> users to access the SAS.Workspace (SAS Version 9.1) application
- launch permissions for the IWAM_<machine name> user

where *<machine name>* is the name of your machine or a slight variation. These users are part of the *\\<machine name>** domain and will appear if you click **Show Users**.

By default, the IUSR_*<machine name>* and IWAM_*<machine name>* users have launch permissions for all DCOM applications. However, use **dcomcnfg** to verify that the launch permissions are properly configured.

1. Start **dcomcnfg** and modify the properties for **SAS.Workspace (SAS Version 9.1)**.
2. Add access and launch permissions for the following:

- ◆ IUSR_*<machine name>* (Internet Guest Account)
- ◆ IWAM_*<machine name>* (Launch IIS Process Account)

Basic Authentication

Note: This configuration also works for **Integrated Windows authentication**.

For basic authentication, all inbound Web clients must authenticate as a specific user in order to gain access to the Web page. The following security options must be configured:

- access permissions for any user that will be accessing the Web page. Configure access permissions to the SAS.Workspace (SAS Version 9.1) application, as well as the IWAM_*<machine name>* user.
- launch permissions for the IWAM_*<machine name>* user. The IIS process is still launched by the IWAM_*<machine name>* user.

By default, the IWAM_*<machine name>* has launch permissions for all DCOM applications. However, use **dcomcnfg** to verify that the launch permissions are properly configured.

1. Start **dcomcnfg** and modify the properties for **SAS.Workspace (SAS Version 9.1)**.
2. Add launch and access permissions (Launch IIS Process Account) for the IWAM_*<machine name>* user.
3. Add access permissions for any user that will be accessing the ASP through the Web. To add access permissions for users, use **dcomcnfg** to do either of the following:
 - ◆ add each user individually
 - ◆ create a group of users and then add that group.

COM/DCOM

Accessing a Remote DCOM IOM Server from an Active Server Page

When you access a remote DCOM IOM server from an Active Server Page (ASP), your IOM server is on a different machine than your Web server and you access DCOM objects through the network.

Follow the configuration instructions for configuring permissions on either for Windows NT 4, or for Windows 2000 and XP.

Configuring Windows NT 4 with IIS to Access a Remote DCOM IOM Server

To enable the NT Anonymous Logon user with permissions to launch and access the DCOM server:

1. On your remote IOM server, start **dcomcnfg**.
2. Select **SAS.Workspace (SAS Version 9.1)**, and then select **Properties**.
3. Select the Security tab, and add launch and access permissions for the following users:

- ◆ System (the operating system)
- ◆ Network (users accessing this object remotely)

4. If your DCOM IOM server is on Windows NT 4, this configuration is sufficient.

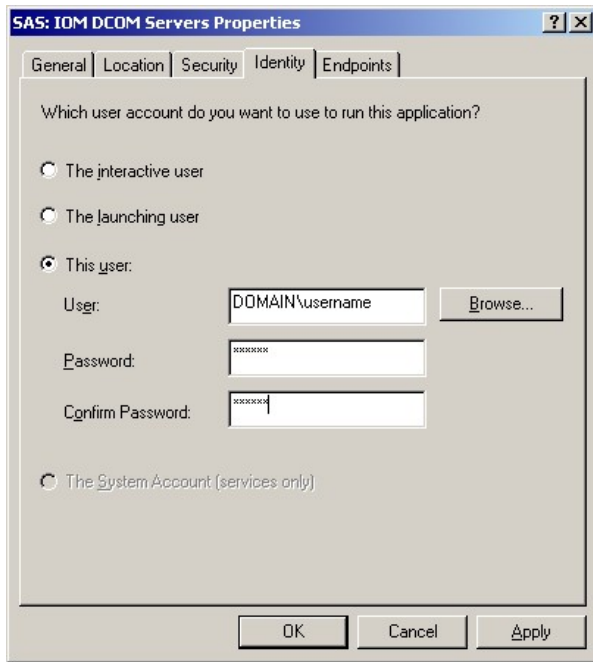
If your DCOM IOM server is on Windows 2000 or XP, you must change the identity of the user that will run the DCOM server process. The NT Anonymous Logon user account on Windows NT 4 does not have sufficient permission to run SAS on a Windows 2000 or XP server.

For Windows 2000 or XP, to change the user that will run the DCOM server process:

1. Select the Identity tab.
2. Select either **The interactive user** or **This user**.

Note: If you have selected **The interactive user** and no users are logged onto the computer, the application will fail. It is recommended that you select **This user** and indicate a specific account.

If you select **This user**, enter a valid user account that has permission to run SAS on your server.



Configuring Windows 2000 or XP with IIS 5 to Access a Remote DCOM IOM Server

For Windows 2000 and XP, IIS processes are configured as *COM+ Applications*. Therefore, you must configure an additional layer of security prior to accessing a remote IOM DCOM server from an ASP.

By default, an application in IIS 5 uses Medium (Pooled) application protection, and, as a result, it runs under the IIS Out of Process Pooled Applications COM+ application. In a typical IIS 5 installation, this application is launched by the IWAM_<machine_name> account.

The IWAM_<machine name> account exists on the \\<machine name>* domain on which IIS is running. But, when the IWAM_<machine name> attempts to authenticate on the remote server as the IWAM_<machine name> user, access is denied because the account does not exist on the remote server. The COM+ application must run under an account that exists on both machines. There are two ways to achieve this access:

- if the two computers are located under the same domain, you can use an account on the domain.
- you can use an account that exists locally on both computers if the passwords for the account match on both computers.

Important Note: It is recommended that you **DO NOT** change the launching user of the IIS Out of Process Pooled Applications. Changing the launching user will cause all of your pooled IIS applications to launch as a specific user and could cause problems. In addition, if you change the launching user from the IWAM account to another user, it is difficult to revert back to the IWAM account. You might want to revert back to the IWAM account if another application fails because you changed this launching user.

For these reasons, we recommend that you change to **High (Isolated) Application Protection** for the IIS Application that will access SAS using DCOM. This will create a new COM+ Application that you can configure independently, without affecting any other pooled applications. If you change the launching user of the IIS Out of Process Pooled Application, it is possible to revert back to the IWAM account. For more information about resetting the IWAM password, see [PRB: Configured Identity is Incorrect for IWAM Account \(Q297989\)](#) on the Microsoft Web site.

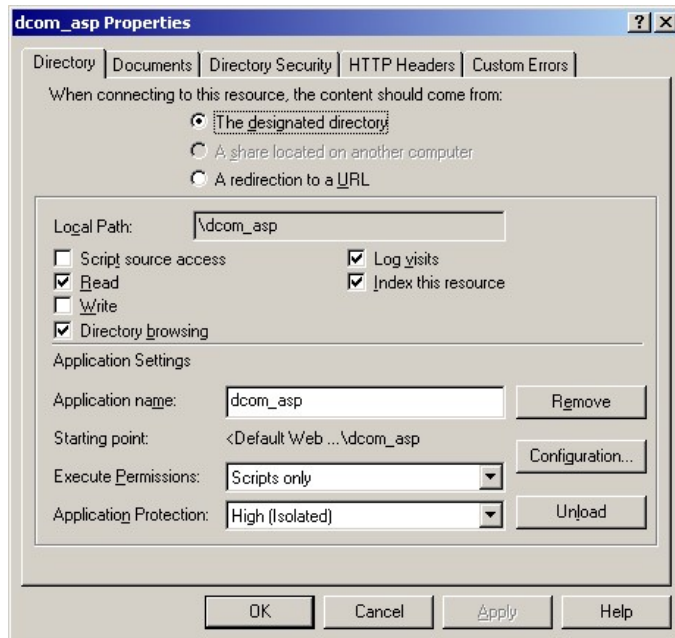
To set up remote DCOM and COM+:

1. Configure your IIS application to use High (Isolated) Application Protection.
2. Configure the IIS application to run as a specific user.
3. Set access and launch permissions for the user.

Configure your IIS Application to use High (Isolated) Application Protection

To run your application as an isolated process:

1. Start Internet Services Manager by clicking **Start ➤ Settings ➤ Control Panel**. Open **Administrative Tools** and click **Internet Services Manager**.
2. Select the directory where your ASP is located.
3. Right-click, and select **Properties** to view the properties for your directory.
4. On the Directory tab under Application Settings, change **Application Protection** to **High (Isolated)**.

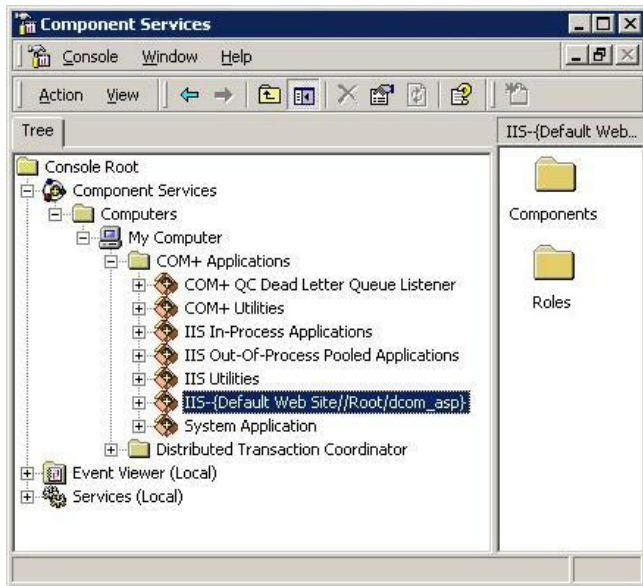


Configure your COM+ Application

Note: Be sure to read the Important Note under Configuring Windows 2000 or XP with IIS 5 Remote DCOM and COM+ Settings. It is recommended that you do NOT change the launching user of the **IIS Out-Of-Process Pooled Applications**.

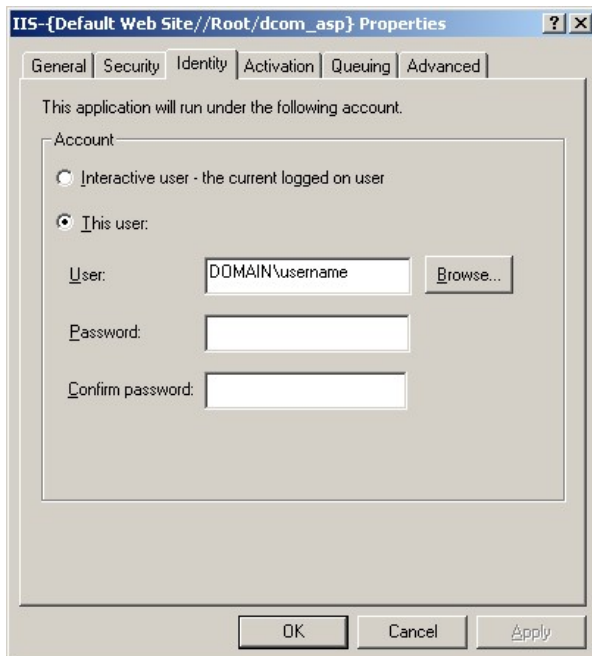
To configure the COM+ application:

1. Click **Start ➤ Settings ➤ Control Panel**.
2. Open **Administrative Tools** and click **Component Services**.
3. Expand the Component Services folder, expand the Computers folder, expand the My Computer folder, and then expand the COM+ Applications folder.



4. Find the newly created COM+ application for your IIS application. It will be named **IIS--{Default Web Site//Root/<iis_application>}** where *<iis_application>* is the name of your IIS application.
5. Right-click the appropriate COM+ application, and select **Properties**.
6. Select the Identity tab, and do one of the following:
 - ◆ Indicate a specific user account for the application.
 - ◆ Use the interactive user if the interactive user exists on both machines.

Note: If you have selected **The interactive user** and no users are logged onto the computer, the application will fail. It is recommended that you select **This user** and indicate a specific account.



Setting Access and Launch Permissions for the User

You must give the user who launches the IIS COM+ application permission to access and launch the remote IOM DCOM server. To set the permissions:

1. On your remote IOM DCOM server, start **dcomcnfg**.
2. Select **SAS.Workspace (SAS Version 9.1)**, and then select **Properties**.
3. Select the Security tab, and add launch and access permissions for the user who is launching your IIS COM+ application.
4. Add access permissions for

♦ Network (users accessing this object remotely)
found in the \\<machine name>* domain.

More Information

These COM/DCOM configurations will work for most simple setups. There are many other ways to configure IIS, DCOM and COM+ that might better suit your specific needs. The following documents and books on the World Wide Web provide additional information about IIS, DCOM, COM+ as well as information about developing ASP applications that use COM objects. There are also many other resources for Active Server Page developers available on the [MSDN](#) Web site.

- [Active Server Pages Developer Center](#)
- [ASP.NET Developer Center](#)
- [Building Secure Microsoft ASP.NET Applications](#)
- [Microsoft® Windows® 2000 Server Resource Kit: Microsoft Internet Information Services 5.0 Resource Guide](#)
- [COM+: Security, Communication, and Configuration](#)
- [HOWTO: Accessing Network Files from IIS Applications \(Q207671\)](#)

COM/DCOM

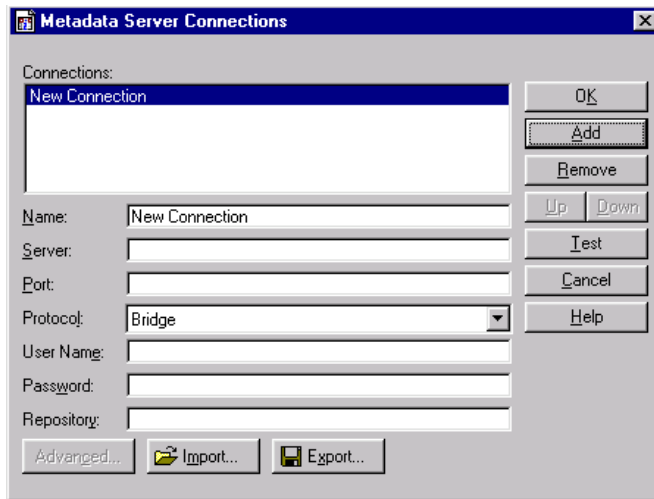
Creating a Metadata Configuration File in SAS

The Metadata Server Connections window in SAS enables you to:

- configure information for connecting to a SAS Metadata Server
- export the configuration to a metadata configuration file that you can use when connecting to a SAS Metadata Server from the Windows Object Manager.

To create a metadata configuration file in SAS, follow these steps:

1. Start SAS and enter the METACON command. The Metadata Server Connections window appears.



2. Click **Add** to create a new connection and complete the following fields:

Name

Specifies a name for the server connection.

Server

Specifies the fully-qualified name of the machine that the server runs on.

Port

Specifies the port that the server connection uses.

Protocol

Specifies whether the connection uses IOM Bridge protocol or COM protocol.

User Name

Specifies the user ID that is used to log on to the server. You might need to specify your authentication domain using the format *domain\user-ID*.

Password

Specifies the password that is used to log on to the server.

Repository

Specifies which metadata repository on the server to use.

3. To export the connection information as a metadata configuration file, click **Export**.

COM Servers

Using the SAS Integration Technologies Configuration Utility (ITConfig)

The SAS Integration Technologies configuration utility (ITConfig) enables you to generate metadata configuration files and test Integrated Object Model (IOM) connections between client machines and SAS. Using the ITConfig application, you can perform the following tasks:

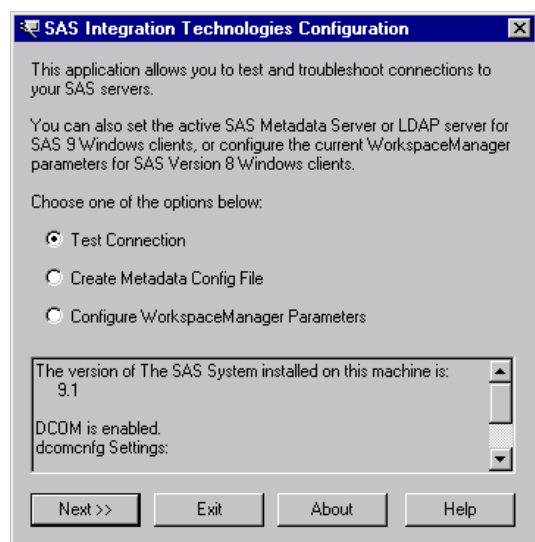
- create metadata configuration files that can be used to access an LDAP server or SAS Metadata Server
- test and diagnose IOM connections to SAS servers. The application can test COM, DCOM, and IOM Bridge connection types.
- set the registry parameters that are used by the workspace manager on an LDAP server.

Starting the Application

Select **Start ▶ Programs ▶ SAS ▶ SAS 9.1 Utilities ▶ Integration Technologies Configuration** to open ITConfig.

When the program starts, it checks the Windows program registry for unused SAS Integration Technologies entries. If any unused entries are found, the application gives you the option of removing the entries.

The SAS Integration Technologies Configuration window appears.



This window displays information about your current configuration, including the version of SAS installed, whether DCOM is installed and active, and DCOM configuration settings. Use this window to choose which task you want to perform:

- create metadata configuration files ([Create Metadata Config File](#))
- test the connection to a SAS Workspace Server or SAS Metadata Server ([Test Connection](#))
- view and change the LDAP parameters for the Workspace Manager (not used for the SAS Open Metadata Architecture).

COM/DCOM

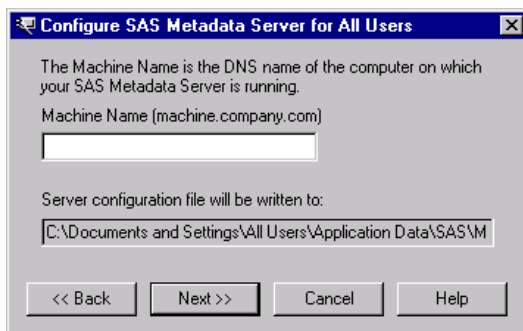
Using ITConfig to Create Metadata Configuration Files

To access definitions on a metadata server, you must first connect to the metadata server. For COM connections to the metadata server, the Object Manager and SAS use a metadata configuration file called the system configuration file. The system configuration file contains information about how to access the metadata server.

Note: For COM connections to the SAS Metadata Server, it is not possible to specify user information.

To create a system configuration file

1. Select **Create Metadata Config File** from the main ITConfig window. The Create SAS Metadata Config File window appears.
2. Select **SAS Metadata Server** and click **Next**. The Configure Metadata Server window appears.
3. Select **COM** for the connection type. For the configuration type, select **Current user** to create a user-specific configuration, or **All users on this machine** to create a configuration that is common to all users. Click **Next**. The Configure SAS Metadata Server window appears.



4. If metadata configuration files already exist on your machine, the information from those files will be included in this window. You can edit the existing configuration parameters.

Enter the following information:

Machine Name

specifies the fully-qualified name of the machine that the SAS Metadata Server runs on.

5. Click **Next**. The application connects directly to the SAS metadata server, retrieves the list of available repositories, and displays the SAS Metadata Server Repository Selection window. Select the repository that will be used for the metadata configuration and click **Next**.
6. The application writes the data to the metadata configuration file. The XML File Written dialog box appears.
7. To return to the main ITConfig window, click **OK**.

Name and Location for the Configuration File

The metadata configuration file is always stored with a default filename and path. The path is dependent on the version of Windows that you are using.

Default Paths for Windows NT:

Common system configuration file

\WINNT\Profiles\All Users\Application Data\SAS\
MetadataServer\oms_serverinfo.xml

User-specific system configuration file

```
\WINNT\Profiles\username\Application Data\SAS\
MetadataServer\oms_serverinfo.xml
```

Default Paths for Windows 2000, Windows XP, and Windows 2003 Server:

Common system configuration file

```
\Documents and Settings\All Users\Application Data\SAS\
MetadataServer\oms_serverinfo.xml
```

User-specific system configuration file

```
\Documents and Settings\username\Application Data\SAS\
MetadataServer\oms_serverinfo.xml
```

Note: The location and filename are displayed in the Configure SAS Metadata Server window and in the XML File Written dialog box.

Sample System Configuration File Format for a COM Connection

Use a text editor to edit your metadata configuration files. The following XML code shows a sample system configuration file for a COM connection to a SAS Metadata Server.

```
<?xml version="1.0" encoding="UTF-8" ?>
<Redirect>
  <LogicalServer Name="SAS Metadata Server"
    ClassIdentifier="2887E7D7-4780-11D4-879F-00C04F38F0DB">
    <UsingComponents>
      <ServerComponent Name="SAS Metadata Server"
        ClassIdentifier="2887E7D7-4780-11D4-879F-00C04F38F0DB">
        <SourceConnections>
          <COMConnection Name="SAS Metadata Server"
            HostName="aintserv.us.sas.com">
            <Properties>
              <Property Name="Repository"
                DefaultValue="Aintserv"
                PropertyName="Repository">
            </Property>
          </Properties>
        </COMConnection>
      </SourceConnections>
    </ServerComponent>
  </UsingComponents>
</LogicalServer>
</Redirect>
```

COM/DCOM

Using ITConfig to Test Connections

The SAS Integration Technologies configuration utility (ITConfig) allows you to test connections from your local machine to a SAS Workspace Server or SAS Metadata Server. The application can test a DCOM connection or a connection to a local machine. You can retrieve the server definition from a metadata server or define the server manually.

The test program used by the Integration Technologies Configuration Application is a small SAS program that verifies the following information about the server environment:

- events are returned
- the WORK data set is properly configured
- the location of the SASUSER directory
- the state of other SAS options.

Testing a Connection that is Defined on a Metadata Server

To test connections to a server that is defined on a metadata server:

1. Select **Test Connection** from the main Integration Technologies Configuration window and click **Next**. The Choose How to Test window appears.
2. Select **Retrieve logical server definitions from the currently configured metadata server**, then click **Next**. The Test window appears.
3. Select the **Logical Name** of the server connection that you wish to test.
4. Click **Test** to submit the test program through the connection. If the program establishes a connection to the specified server, the Connection Successful window appears.
5. Click **Copy Text** to copy the test results to the clipboard. Click **OK** to return to the main Integration Technologies Configuration window.

Testing a Local COM Connection

To test a local COM connection to a SAS server:

1. Select **Test Connection** from the main Integration Technologies Configuration window and click **Next**. The Choose How to Test window appears.
2. Select **Enter a SAS server definition manually** and click **Next**. The Run Tests window appears.
3. Select the type of server to test and select **Local Connection (COM)**, then click **Next** to submit the test program through the connection. If the program establishes a local COM connection, the Connection Successful window appears.
4. Click **Copy Text** to copy the test results to the clipboard. Click **OK** to return to the main Integration Technologies Configuration window.

Testing a Manually Defined DCOM Connection

To test a DCOM connection to a SAS server:

1. Select **Test Connection** from the main Integration Technologies Configuration window and click **Next**. The Choose How to Test window appears.
2. Select **Enter a SAS server definition manually** and click **Next**. The Run Tests window appears.

3. Select the type of server to test and select **Remote Connection (DCOM)**, then click **Next**. The DCOM Parameters window appears.
4. Enter the name of the machine for which you want to test a connection. Machine names are usually in the form machine.company.com.
5. Click **Test** to submit the test program through the connection. If the program establishes a DCOM connection to the specified server, the Connection Successful window appears.
6. Click **Copy Text** to copy the test results to the clipboard. Click **OK** to return to the main Integration Technologies Configuration window.

COM/DCOM

Troubleshooting a COM/DCOM Connection

The following tips provide assistance for troubleshooting a COM/DCOM connection.

- Make sure you observe COM/DCOM requirements:
 - ◆ You must use a SAS server to test a DCOM connection. You cannot test a DCOM configuration by trying to connect to a server on the same machine. This type of connection uses COM instead.
 - ◆ To obtain details about why a DCOM connection attempt failed, check the System Log using the Event Viewer on NT (**Start ▶ Programs ▶ Administrative Tools ▶ Event Viewer**). Double click on an event that has a source of DCOM.
 - ◆ In order to get two machines working with DCOM across untrusted domains, the AuthenticationLevel must be set to NONE on both machines. However, if you do this, the impersonation of the client will fail. There is also a requirement that the user names and passwords must be identical in both domains. In this case, Authentication can be enabled.
 - ◆ To determine if launch permissions or access permissions need to be fixed, use the control panel to assign a sound to for starting and ending processes. If you hear the sound, launch permissions are probably OK, but access permissions need to be adjusted. If you don't hear a sound, check your launch permissions. This is necessary because the server process may come and go faster than the NT task manager can update.

- Make sure the registry settings are correct:

- ◆ To reset application-specific dcomcnfg settings, edit the registry and remove the following keys:

```
HKEY_CLASSES_ROOT\AppID\SAS.EXE      (if it exists)
HKEY_CLASSES_ROOT\AppID\
    {440196D4-90F0-11D0-9F41-00A024BB830C}
```

Run dcomcnfg and view the (empty) access and launch permissions. When you press OK or Apply, the dcomcnfg utility will put in some values for access and launch permissions. You can see those values by viewing the access and launch permissions again through dcomcnfg.

- ◆ The Default security registry location is

```
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Ole
```

- ◆ DCOM registry settings affect local COM also. DefaultAccessPermissions (recommend Interactive and System), DefaultLaunchPermissions (recommend Interactive and System), and Impersonation (recommend IMPERSONATE) are all important for local COM. If you need to run local COM without a license, set the authentication level to CONNECT.
- ◆ Restart any affected server or client processes.
- ◆ Individual registry keys can be secured with regedt32, but not regedit.

- Make sure the working directory is correct:

- ◆ The current working directory for all programs (including SAS) started from the NT4 SCM is:

```
c:\winnt\system32
```

(This is the directory where rpcss.exe exists.) This means that files created by the SAS server (without a directory specified) will appear in this directory. To change the initial folder that is used after SAS starts, use the `-sasinitialfolder` option in your config file.

- Make sure the permissions are correct:
 - ◆ The NT Service Control Manager (SCM) runs in rpcss.exe. The SCM is responsible for launching SAS under both COM and DCOM.
 - ◆ If you do not have a license for the Integration Technologies product, the IOM server restricts incoming connections by allowing connections from the local machine only. As part of this verification, SAS System Version 8 servers must be able to impersonate the client. Because the SAS Workspace Manager will adjust the impersonation level settings when making a local connection to allow this check to work, if you are using Version 8 of the SAS System, then you should consider using the SAS Workspace Manager to initiate the client session. SAS System 9 and later servers can make this determination regardless of whether the client impersonation is enabled.
 - ◆ The system account must have launch and access permissions (the SCM runs under the system account).
 - ◆ A good technique to use to determine what user ID is being used to read/write files is to enable auditing on the file. To do this, first use the User Manager ➤ Policies ➤ Audit... to enable auditing for File and Object Access. At this point, nothing will actually be audited until the specific files that you want audited are enabled for auditing. Do this from the File Manager. Select Properties ➤ Security tab ➤ Auditing for each file you want to audit. (If you do this for a directory, you can specify all files under that directory.)

To view the audited information, use the Event Viewer and select Log ➤ Security. This will show you what user ID attempted to access the files specified through the user manager.

- ◆ An error message that states "Server execution failed" when trying to connect to the IOM server can be caused by many things including trying to connect to an IOM server with an expired license or having an invalid username/password in the dcomcnfg identity settings.
- ◆ Events work by having the IOM server make a call on an interface that the client provides to SAS. In order for SAS to make a call on that interface, the client must grant permission to SAS to make the call.

As another alternative, Microsoft has suggested setting the client's authentication level to None. For a C/C++ application, this can be controlled through CoInitializeSecurity. For a Visual Basic application, set the default authenticationLevel to None using dcomcnfg on the client side. Note that this implies that events cannot be encrypted, and that the only way to encrypt non-event data is through the server-side authenticationLevel settings in dcomcnfg.

- Make sure the authentication is correct:
 - ◆ On NT 4, the only authentication provided by default is NTLM, which uses RC4 for packet encryption (if you turn it on, of course).

COM/DCOM

AppIDs for Configuring DCOM

The following table lists the application name for each type of IOM server by SAS version.

SAS Version	Application Name	Description
8.0	SAS Workspace (Ver. 1.0)	SAS Workspace Server
8.1	SAS: Integrated Object Model (IOM) Server 1.0	SAS Workspace Server
8.2	SAS: IOM DCOM Servers	SAS Workspace Server
9.0	SAS.Workspace (SAS Version 9.0)	SAS Workspace Server
	SASOMI.OMI (SAS Version 9.0)	SAS Metadata Server
	SASMDX.Server (SAS Version 9.0)	SAS OLAP Server
9.1	SAS.Workspace (SAS Version 9.1)	SAS Workspace Server
	SASOMI.OMI (SAS Version 9.1)	SAS Metadata Server
	SASMDX.Server (SAS Version 9.1)	SAS OLAP Server

The following table shows the AppID for each type of IOM server. The AppIDs are the same for all versions of SAS.

Server Type	AppID
SAS Workspace Server	440196D4-90F0-11D0-9F41-00A024BB830C
SAS OLAP Server	F3F46472-1E31-11D5-87C2-00C04F38F9F6
SAS Metadata Server	2887E7D7-4780-11D4-879F-00C04F38F0DB

COM/DCOM

Object Server Parameters

All object server parameters are applicable on the command line that starts the server:

- For servers that are started by the object spawner, the object server parameters come from your server definition in the SAS Metadata Repository. (The server definition is located under the Server Manager plug-in of SAS Management Console. In the server definition, select the Options tab to locate the **Object Server Parameters** field).
- For servers that are not spawned (such as those that are run from command scripts, those that are run as Windows services, or those that are launched by COM), you use the OBJECTSERVERPARMS SAS option to specify the object server parameters on the command line.

To simplify the command that is needed to invoke an IOM server, the server startup sequence can also connect back to the metadata server in order to fetch additional information, including object server parameters. This feature involves use of the SERVER= and METAUTOINIT object server parameters. See [Specifying Metadata Connection Information](#) for details. The object server parameters that can be obtained in this way are indicated in the table below under the column "Can Be Fetched at Server Startup." These object server parameters can be specified in the server definition in SAS Management Console. In the server definition, select the Options tab to locate the **Object Server Parameters** field.

Important Note:

You can fetch object server parameters from metadata as follows:

- **When you start the server with a script**, some object server parameters cannot be obtained from the metadata. These parameters are designated as "No" in the "Can Be Fetched at Server Startup" column. Do not enter these object server parameters in your metadata.
- **When you start the server with a spawner**, all object server parameters can be obtained from the metadata (even those that are designated as "No" in the "Can Be Fetched at Server Startup" column).

Note: Object server parameters that are specified on the command line always override object server parameters obtained from a SAS Metadata Repository.

Object Server Parameter	Value	Connection Type	Can Be Fetched at Server Startup (When Starting a Server With a Script)	Definition
ANONYMOUSLOGINPOLICY	Deny Restrict	IOM Bridge	Yes	Specifies whether the server permits any access at all to connections that do not supply a user ID (in programming terms, ones that supply a zero-length user ID).

				<p>If you specify "restrict," then the server allows connections that do not have a user ID; however, the client only has restricted access to the IServerStatus interface (used primarily for querying basic server status).</p> <p>If you specify "deny," then the server completely disallows connections that do not provide a user ID. The default is "restrict." For details about ANONYMOUSLOGINPOLICY, see Setting Up Additional Server Security in the Security chapter.</p>
APPLEVEL	0, 1, 2, 3	IOM Bridge COM/DCOM	Yes	<p>Specifies the detail level of the trace that is written by the server application (such as the OLAP server, the SAS Metadata Server or the SAS Stored Process Server). The default value if APPLEVEL is omitted (1) enables logging at a level that is suitable for a production server; therefore, this parameter is optional.</p> <p>APPLEVEL=0 disables the application's logging and is discouraged because it suppresses useful diagnostic information. Higher APPLEVEL values can invoke additional tracing. The SAS Metadata Server, for example, defines additional logging levels. For details, see Enabling Logging of Authentication Events and SAS Metadata Server Logging Overview in the <i>SAS Metadata Server: Setup Guide</i>.</p>
CLASSFACTORY Alias: CLSID	36 character class identifier	IOM Bridge COM/DCOM	Yes	<p>Specifies the class ID number, which specifies the type of server to instantiate (for example, 2887E7D7-4780-11D4-879F-00C04F38F0DB specifies a SAS Metadata Server). An IOM server exposes one top-level class through its class identifier.</p> <p>By default, an IOM server hosts the Workspace class. If you want to specify an alternate class to expose as the top-level class, use the classfactory option to identify the class to IOM.</p> <p>When using the SERVER= objectserverparms suboption, the classfactory does not need to be specified because it is obtained from the logical server definition in the SAS Metadata Repository.</p> <p>This option is primarily used to start the SAS Metadata Server.</p>

CLIENTENCRYPTIONLEVEL Alias: CEL	none credentials everything	IOM Bridge	No	Specifies the degree of encryption to use when making outbound calls. This option is used only by the bridge protocol engine.
IOMLEVEL	0, 1, 2, or 3	IOM Bridge COM/DCOM	Yes	Specifies trace level for protocol-independent IOM events, particularly calls and the SAS LOG of workspaces. The default is 0. If IOMLEVEL is set to 1, then the calls that enter and leave the server are traced. This feature can be very helpful for identifying whether a problem arose in a client or in the server. Using IOMLEVEL=1 with the SAS Metadata Server will capture the input and output XML strings for metadata requests. For more information, see Capturing XML in the Log and SAS Metadata Server Logging Overview in the <i>SAS Metadata Server: Setup Guide</i> . For performance reasons, it is recommended that IOMLEVEL=1 be used only when diagnosing problems. Higher values of IOMLEVEL produce traces that are intended only for use by SAS Technical Support. Depending on the calls that are being traced, the JNLSTRMAX and JNLLINEMAX values may need to be increased to prevent truncation of long strings and long lines.
JNLARRELM	Numeric Value	IOM Bridge COM/DCOM	Yes	Specifies the maximum number of array elements to print out when an IOM array value is traced.
JNLLINEMAX	Numeric Value	IOM Bridge COM/DCOM	Yes	Specifies the maximum length of a line printed in the IOM server journal.
JNLSTRMAX	Numeric Value	IOM Bridge COM/DCOM	Yes	Specifies the maximum length of string printed in the IOM server journal. This option can be used to adjust the amount of material included in an IOM trace. A value greater than 500 is recommended.
LOGFILE Alias: LOG	Path in which to create the IOM server trace log	IOM Bridge COM/DCOM	Yes	Specifies an alternative file for the SAS log for IOM server trace output. Note: Using this option on a spawned server can prevent multiple servers from running simultaneously because they will all try to open the same log file. It is therefore recommended that this option be used only for specific diagnostic tasks. Note: The user who starts the server must have execute and write permissions for the log

				destination path.
METAAUTOINIT NOMETAAUTOINIT	N/A	IOM Bridge COM/DCOM	Yes	Specifies whether the IOM server should connect back to the SAS Metadata Server during startup in order to obtain additional configuration information such as object server parameters and pre-assigned libraries. When METAAUTOINIT is specified, the server uses the provided META* options to connect to the SAS Metadata Server. With NOMETAAUTOINIT, IOM server startup does not connect back to the SAS Metadata Server. The default depends on the type of server. For further details, see Server Startup Command . This option is applicable only if you have specified your logical server with the SERVER= object server parameter.
PELEVEL	0, 1, 2, or 3	IOM Bridge	No	Specifies trace protocol engine logic and packets. Level 3 specifies the most verbose output. The default is 0.
PORT	TCP/IP port number	IOM Bridge	Yes	Specifies the value for the bridge protocol engine to use as the port to start listening for client connections. Do not specify this option with spawned servers; it will be supplied automatically by the spawner.
PROTOCOL	bridge com (com,bridge)	IOM Bridge COM/DCOM	Yes	Specifies the protocol engines to launch in server mode. Server mode indicates that the protocol engines will listen for client connections. By default, Windows servers use the COM protocol engine and all other servers use the Bridge protocol engine. If you specify (com, bridge) then a multi-user server can simultaneously support clients using different protocols. Do not specify this option with spawned servers; it will be supplied automatically by the spawner.
SECURITY NOSECURITY	N/A	IOM Bridge COM/DCOM	No	Specifies whether client authentication is enabled. By default (SECURITY), clients must be authenticated; one exception is the use of ANONYMOUSLOGINPOLICY for public interfaces (see Setting Up Additional Server Security). When security is enabled, the bridge protocol engine requires a user name and password; the COM protocol engine is integrated with the single-signon security of the Windows networking environment. Authorization decisions are controlled by the server application. If NOSECURITY is specified, these security mechanisms are bypassed.

SERVER	Logical server name or OMSOBJ URI (object ID)	IOM Bridge COM/DCOM	No	Specifies the logical server name for the IOM run-time and server application to use to locate configuration information in a SAS Metadata Repository. The SERVER= option can be used to retrieve many of the OBJECTSERVERPARMS options (including PORT, PROTOCOL and CLASSFACTORY) from a SAS Metadata Repository. For details, see Specifying Metadata Connection Information .
SERVICE	TCP service name	IOM Bridge	Yes	Specifies the TCP service name (for example, from /etc/services on a UNIX system) for the port that the IOM Bridge protocol engine will use to listen for connections from clients. Do not specify this option with spawned servers; it will be supplied automatically by the spawner.
TRUSTSASPEER Alias: TSASPEER	N/A	IOM Bridge	Yes	Enables SAS peer sessions from IOM servers to connect as trusted peer sessions. For details, see Implementing Trusted Authentication Mechanisms .
V8ERRORTEXT	N/A	IOM Bridge COM/DCOM	Yes	Indicates that the MVA components should return Version 8 style error messages instead of the Version 9 XML style error messages.

COM/DCOM

Fields for the Server Definition

The server definition contains startup and connection information for an instance of a SAS server. The server is defined using the fields listed in the following table. For each field, the table shows the following information:

- the name that identifies the field in SAS Management Console. Under each field name, the table shows the corresponding properties tab and field name in the SAS Management Console application.
- "Required" or "Optional" to indicate whether the field is required.
- the type of server configuration (COM/DCOM or IOM Bridge) for which the field is used.
- a definition of the field.

For step-by-step instructions about defining the metadata for a server connection, refer to [Using SAS Management Console to Define Servers](#).

Fields for the Server Definition			
Field Name	Required Optional	Server Type	Definition
Availability Timeout <i>In SAS Management Console:</i> Options ➔ Advanced Options ➔ Load Balancing Properties: Availability Timeout	Optional	IOM Bridge	For load-balancing servers, the number of milliseconds to wait for a load-balancing server to become available. This parameter is used in the following situations: <ul style="list-style-type: none"> • when all servers have allocated the maximum number of clients per server. • when load balancing is waiting for a server to start and become available for its first client.
Command <i>In SAS Management Console:</i> Options ➔ Launch Commands: Command	Required	IOM Bridge	The command used to launch SAS as an object server. If the SAS executable is not already in your path, then specify the path to <code>sas.exe</code> . You can also specify additional options on the command line. For details, see Server Startup Command . This field is used only for spawned servers.
Description <i>In SAS Management Console:</i> General ➔ Description	Optional	COM/DCOM, IOM Bridge	Text to summarize why this definition exists.
Authentication Domain <i>In SAS Management Console:</i> <Connection> ➔	Required	IOM Bridge	The domain that is associated with a set of computing resources that use the same authentication process. In IOM Bridge servers configurations, the spawner definition must have the same authentication domain name as the server definition. The spawner uses the authentication domain name, along with the machine

Options ➤ Authentication Domain			name, to determine which servers it services.
Host Name <i>In SAS Management Console:</i> <Connection> ➤ Options ➤ Host Name	Required	COM/DCOM, IOM Bridge	<p>The <u>DNS name</u> or IP address for the machine on which this server definition can execute. The machine name must be the official network name (for example, <code>machine.corp.com</code>). The string <code>localhost</code> can be used to signify the host on which the associated spawner is executing.</p> <p>Note: If you use <code>localhost</code> in the configuration, it could cause clients to connect to their local machine instead of the machine that an administrator designates as <code>localhost</code>.</p>
Inactivity Timeout <i>In SAS Management Console:</i> Options ➤ Advanced Options ➤ Load Balancing Properties ➤ Inactivity Timeout <i>and</i> Options ➤ Advanced Options ➤ Pooling Properties ➤ Inactivity Timeout	Optional	COM/DCOM, IOM Bridge	<p>If you are using connection pooling (SAS Workspace Server only) or load balancing (SAS Stored Process Server only), specifies whether an idle server should always remain running, and if not, how long it should run before being shut down. If the check box is not selected, then idle servers remain running. If the check box is selected, then the servers run idle for the number of minutes specified in the field before being shut down. If the check box is selected and 0 is specified as the inactivity timeout, then the server behavior is as follows:</p> <ul style="list-style-type: none"> • for load balancing (IOM Bridge only), the server will shut down when the last client disconnects from the server. • for pooling, a connection returned to a pool by a user is disconnected immediately unless another user is waiting for a connection from the pool. <p>The maximum value is 1440.</p>
Login <i>In SAS Management Console:</i> Options ➤ Advanced Options ➤ Credentials ➤ Login	Optional	IOM Bridge	<p>For SAS Stored Process Servers, the login that provides the spawner with credentials to use when starting a multi-user SAS session.</p> <p>Note: If the server runs on Windows 2000 and Windows NT, for the user who is the owner of the multi-user login, define the "act as part of the operating system" user right.</p>
Major Version Number <i>In SAS Management</i>	Required	COM/DCOM, IOM Bridge	Specifies the major version number of the component.

<i>Console:</i> Options ➤ Major Version Number			
Minor Version Number <i>In SAS Management Console:</i> Options ➤ Minor Version Number	Required	COM/DCOM, IOM Bridge	Specifies the minor version number of the component.
Maximum Clients <i>In SAS Management Console:</i> Options ➤ Advanced Options ➤ Load Balancing Properties ➤ Maximum Clients <i>and</i> Options ➤ Advanced Options ➤ Pooling Properties ➤ Maximum Clients	Optional	COM/DCOM, IOM Bridge	<ul style="list-style-type: none"> • For Pooling (SAS Workspace Server), specifies the maximum number of simultaneous connections from the pool. • For Load Balancing (SAS Stored Process Servers and Response Time algorithm only), specifies the maximum number of simultaneous clients connected to this server.
Maximum Cost <i>In SAS Management Console:</i> Options ➤ Advanced Options ➤ Load Balancing Properties ➤ Maximum Cost	Optional	IOM Bridge	For load–balancing servers using the cost algorithm, the maximum cost allowed on each SAS server before requests to the server are denied.
Name <i>In SAS Management Console:</i> General ➤ Name	Required	COM/DCOM, IOM Bridge	The unique name for this server.
Object Server Parameters <i>In SAS Management</i>	Optional	IOM Bridge	For spawned servers, these object server parameters are added to others that are generated by the spawner and used to launch SAS. For servers that are not spawned, the values that you specify here can be used to

<i>Console:</i> Options ➔ Launch Commands: Object Server Parameters			supplement any that were supplied on the server invocation command line. Any command line parameters take precedence. For a list of object server parameters, see Object Server Parameters . For a more detailed explanation of object server parameter handling, see Server Startup Command .
Port Number <i>In SAS Management Console:</i> <Connection> ➔ Options ➔ Port Number	Required if server will have Java clients	IOM Bridge	<p>The port on which to connect to this server.</p> <p>If you specify a value for both port and service, then the value for service is used.</p> <p>If you are using a spawner and neither port nor service is specified, the spawner attempts to use the service name <code>sasobjspawn</code> as the service. If <code>sasobjspawn</code> has already been used, the spawner removes this service definition from its list.</p> <p>The port number is required if the server will have Java clients.</p> <p>The default port numbers are as follows:</p> <ul style="list-style-type: none"> • SAS Workspace Server: 8591 • SAS Stored Process Server: 8601 • SAS OLAP Server: 5451 • SAS Metadata Server: 8561
Protocol <i>In SAS Management Console:</i> <Connection> ➔ Protocol	Required	COM/DCOM, IOM Bridge	The protocol (Bridge or COM) that clients can use for connection. The protocol <code>bridge</code> must be used for servers that are serviced by the spawner. These include all servers other than Windows, as well as Windows servers that will be accessed by Java clients.
Recycle Activation Limit <i>In SAS Management Console:</i> Options ➔ Advanced Options ➔ Load Balancing Properties ➔ Recycle Activation Limit <i>and</i> Options ➔	Optional	COM/DCOM, IOM Bridge	<p>For pooling (SAS Workspace Servers only) and load balancing (SAS Stored Process Servers only), specifies the number of times a connection to the server will be reused in a pool before it is disconnected ("recycled"). If the value is 0, then there will be no limit on the number of times a connection to the server can be reused. This property is optional. The default value is 0.</p> <p>Note: For SAS Stored Process Servers, setting a Recycle Activation Limit can cause problems with sessions. If you create sessions, use the default value of 0 for Reaction Activation Limit.</p>

Advanced Options ➔ Pooling Properties ➔ Recycle Activation Limit			
Required Encryption Level <i>In SAS Management Console:</i> <Connection> ➔ Options ➔ Advanced Options ➔ Encryption ➔ Required Encryption Level	Optional	IOM Bridge	The level of encryption to be used between the client and the server. None means no encryption is performed; Credentials means that only user credentials (ID and password) are encrypted; and Everything means that all communications between the client and server are encrypted. The default is Credentials .
Server Encryption Algorithms <i>In SAS Management Console:</i> <Connection> ➔ Options ➔ Advanced Options ➔ Encryption ➔ Server Encryption Algorithms	Optional	IOM Bridge	The encryption algorithms that are supported by the launched object server. Valid values are: RC2, RC4, DES, TRIPLEDES, and SASPROPRIETARY, depending on the country in which the SAS software is licensed. See SAS/SECURE documentation for more information regarding this field. The default is SASPROPRIETARY.
Service <i>In SAS Management Console:</i> <Connection> ➔ Options ➔ Advanced Options ➔ Service	Optional	IOM Bridge	The service in which to connect to this server. If you specify a value for both port and service, then the value for service is used. If you are using a spawner and neither port nor service is specified, the spawner attempts to use the service name sasobjspawn as the service. If sasobjspawn has already been used, the spawner removes this service definition from its list. Note: If the server has Java clients, specify a port instead of a service.
Software Version <i>In SAS Management Console:</i> Options ➔ Software Version	Required	COM/DCOM, IOM Bridge	Specifies the version of the server software.

Start Size <i>In SAS Management Console:</i> Options ➤ Advanced Options ➤ Load Balancing Properties ➤ Start Size	Optional	IOM Bridge	For SAS Stored Process Servers, the number of Multibridge connections to start when the spawner starts.
Startup Cost <i>In SAS Management Console:</i> Options ➤ Advanced Options ➤ Load Balancing Properties ➤ Startup Cost	Optional	IOM Bridge	For load-balancing servers using the cost algorithm, the cost for starting a server.
Vendor <i>In SAS Management Console:</i> Options ➤ Vendor	Required	COM/DCOM, IOM Bridge	Specifies the vendor of the server software.

COM/DCOM

Fields for the Pooled Logical Server and Puddle Definitions

You can only convert SAS Workspace Servers to pooled logical servers.

The pooled logical server definition contains information for an instance of a pooled logical server. The pooled logical server is defined using the fields listed in the following table. For each field, the table shows the following information:

- the name that identifies the field in SAS Management Console. Under each field name, the table shows the location of the corresponding properties tab and field name in the SAS Management Console application.
- "Required" or "Optional" to indicate whether the field is required.
- a definition of the field.

For general information about the use of logical servers, refer to [Overview of Pooling](#).

Fields for Pooled Logical Server Definitions		
Field Name	Required/ Optional	Definition
Name <i>In SAS Management Console:</i> General ➔ Name	Required	Name of the pooled logical server.
Description <i>In SAS Management Console:</i> General ➔ Description	Optional	Text to summarize why this definition exists. This field is not used by the logical server.
Puddles <i>In SAS Management Console:</i> Options ➔ Puddles	Required	The puddles used for pooling. Click New to define a new puddle.

The puddle definition contains information for an instance of a puddle. The puddle is defined using the fields that are listed in the following table.

Note: For COM connections, only one puddle can be defined.

Fields for the Puddle Definition		
Field Name	Required/ Optional	Definition
Name <i>In SAS Management Console:</i> Options ➔ Puddles ➔	Required	Name of the puddle.

Name		
Minimum Available Servers <i>In SAS Management Console:</i> Options ➤ Puddles ➤ Minimum Available Servers	Required	The minimum number of connections using this login definition that need to be available. This value includes only idle connections.
Minimum Number of Servers <i>In SAS Management Console:</i> Options ➤ Puddles ➤ Minimum Number of Servers	Required	The minimum number of connections using this login definition that are created when the pool is created. This value includes both connections that are in use and connections that are idle. The default value is 0.
Login <i>In SAS Management Console:</i> Options ➤ Puddles ➤ Login	Required	<p>The user ID associated with the puddle. The user or group metadata identity that owns this login can also access the puddle.</p> <p>Note: The login field is used with IOM Bridge connections only.</p>
Grant Access to Group <i>In SAS Management Console:</i> Options ➤ Puddles ➤ Grant Access to Group	Optional	The group metadata identity that can access this puddle. The user or group metadata identities (and their associated logins) that are members of the group can also access this puddle.

IOM Bridge

Setting Up an IOM Bridge Connection

Introduction

An IOM Bridge connection enables client applications to access a server using the IOM Bridge for COM or IOM Bridge for Java.

The IOM Bridge for COM is a software component of SAS Integration Technologies that is used (transparently) to enable native COM/DCOM applications to access server data on either Windows platforms or on platforms other than Windows such as a UNIX or z/OS. The IOM Bridge for Java is used (transparently) when a Java client accesses an IOM server. This bridge allows developers to write Java applications that access server data.

For more information about the IOM Bridge for COM and the IOM Bridge for Java, refer to [Connecting Clients to IOM Servers](#) in the *SAS Integration Technologies Technical Overview*.

This section covers the following topics:

- [When to Use an IOM Bridge Connection](#)
- [Components of a Client–Server Configuration \(IOM Bridge Connection\)](#)
- [How Clients Use an IOM Bridge Connection to Access Servers](#)

When to Use an IOM Bridge Connection

You must use an IOM Bridge connection if any of the following is true:

- the server will run on a platform other than Windows (for example, UNIX)
- the server will be accessed by Java client applications
- the server will use load balancing.

You can also use an IOM Bridge connection if the server will run on a Windows machine and will be accessed by Windows clients. In this situation, clients will connect to the server using the IOM Bridge connection instead of the COM/DCOM connection.

Components of a Client–Server Configuration

When you configure a server with an IOM Bridge connection, the client–server configuration consists of the following:

- a server machine that hosts the Base SAS 9.1 software and the SAS 9.1 Integration Technologies software. In addition, if you are using a SAS Workspace Server or SAS Stored Process Server, then the spawner program (which is part of SAS Integration Technologies), must be running on the server machine in order for clients to obtain access. For information about best practices for setting up the four different types of IOM servers (SAS Workspace Server, SAS Stored Process Server, SAS OLAP Server, and SAS Metadata Server), see [Best Practices: Server and Spawner Setup](#).
- a client application, which can run on the same machine as the server or on a remote machine. To connect to the server via TCP/IP, client applications must use the IOM Bridge for COM or IOM Bridge for Java utilities provided with SAS Integration Technologies. To request specific services from the server, client applications

use Application Program Interfaces (APIs), also known as distributed objects, which are provided with SAS Integration Technologies.

- a SAS Metadata Server, which is a central repository that client and server software can access in order to obtain metadata (or configuration information) about the server. For IOM Bridge connections, the metadata includes definitions for server objects. Optionally, the metadata can also include definitions for spawner objects (which must be used for SAS Workspace Servers and SAS Stored Process Servers); user, group, and login objects (which can be used to provide credentials for various definitions); pooled logical server objects (which associate workspace servers and puddles for pooling security); and load balancing logical server objects (which associate workspace or stored process servers and spawner-to-spawner logins for load balancing).

Note: If your configuration is very simple (that is, consisting of only one or two servers and clients) and does not require strict security, you can supply the server parameters for the configuration directly in the application program.

How Clients Use an IOM Bridge Connection to Access Servers

When a client application uses an IOM Bridge connection to access a server:

1. The client application uses Integration Technologies distributed objects to request a server object. The requested server object can be defined in the client application, or a definition can be retrieved from a metadata server.
2. When the client application requests a server object, an IOM Bridge connection is made to the server.
 - ◆ If the server configuration uses a spawner, the spawner must be running. The spawner authenticates the client, launches the server, and connects the client to the server.
 - ◆ If the server configuration does not use a spawner, the server must be running. The server authenticates the client before continuing.After a connection is established, the server object is created.
3. The client application uses SAS Integration Technologies distributed objects to request services from the server object. The server object can provide services such as SAS language services and publishing services, depending on the type of server object.
4. When the client application is finished using the server object, it issues a request to close the object. Any server or spawner connections associated with the object are closed.

For details about other server and spawner setup, see [Best Practices: Server and Spawner Setup](#).

IOM Bridge

Best Practices: Server and Spawner Setup

Use the following best practices to configure and start your particular type of IOM server:

- **SAS Workspace Server**

1. Configure a spawner and SAS Workspace Server.
2. Start the workspace server as follows:
 - ◇ On Windows platforms, install the spawner as a service in order to listen for connection requests for the workspace server.
 - ◇ On UNIX, VMS, and z/OS platforms, start the spawner in order to listen for connection requests for the workspace server.

- **Load-Balancing Stored Process Server**

1. Configure a spawner and SAS Stored Process Server.

Note: In order for the SAS Stored Process Server to run as a multi-user server, the spawner must have credentials to use when launching the server as a multi-user process. Therefore, in the server definition, you must specify a multi-user login to use when launching a multi-user stored process server. If you do not specify a multi-user login for the stored process server, the stored process server will not run and a message similar to the following will be displayed:

```
This server (OMSOBJ:LOGICALSERVER/A5SRQ5Z5.AT00008E)
cannot be spawned without credentials which specify
the server process username.
```

2. Set up load balancing for the stored process logical server, spawner, and stored process server definitions.
3. Create MultiBridge connections for each stored process server definition.
4. Start the stored process server as follows:
 - ◇ On Windows platforms, install the spawner as a service in order to listen for connection requests for the stored process server.
 - ◇ On UNIX, VMS, and z/OS platforms, start the spawner in order to listen for connection requests for the stored process server.

- **OLAP server**

1. Configure a SAS OLAP server.
2. Start the OLAP server as follows:
 - ◇ On Windows platforms, configure and start the SAS OLAP Server as a service.
 - ◇ On UNIX and z/OS platforms, start the OLAP server with a SAS server startup command.

- **SAS Metadata server**

1. Configure a SAS Metadata Server.
2. Start the metadata server as follows:
 - ◇ On Windows platforms, configure and start the SAS Metadata Server as a service.
 - ◇ On UNIX and z/OS platforms, start the SAS Metadata Server with a SAS server startup command.

SAS® 9.1.2 Integration Technologies: Server Administrator's Guide

For details about configuring and starting the preceding servers, see [Summary of Setup Steps](#). To quickly configure a SAS Workspace Server and spawner and test the connection, see [Quick Start: Standard Workspace Server and Spawner](#). To configure a SAS Stored Process Server and spawner and test the connection, see [Quick Start: Load-Balancing Stored Process Server and Spawner](#).

IOM Bridge

Quick Start: Standard Workspace Server and Spawner

The following steps help you set up an IOM Bridge connection for a simple SAS Workspace Server that uses a spawner to start the server on a Windows or Unix platform. For details about setting up more complex configurations, see [Summary of Setup Steps \(IOM Bridge Connection\)](#).

Note: In this setup, if the client (that accesses the SAS Workspace Server) does not already have a user definition on the SAS Metadata Server, then the client is associated with the Public group on the SAS Metadata Server. The client then has access to objects on the SAS Metadata Server based on the Public group's permissions for that object.

To set up and test a standard server and spawner that use an IOM Bridge connection, complete the following steps:

1. Install SAS 9.1 (including SAS Integration Technologies and SAS Management Console) on the server machine. Refer to the SAS documentation for details about this procedure.
2. Determine the machine name and port for your SAS Metadata Server. Determine the fully-qualified user ID and password that you will use to connect to your SAS Metadata Server.
3. Set up and start your SAS Metadata Server. Then, connect to the server and register a repository. For details about setting up, starting, and connecting to the SAS Metadata Server, see the [SAS Metadata Server: Setup Guide](#).
4. Use SAS Management Console to create definitions for your SAS Workspace Server and spawner. To create a spawner definition, select the Server Manager in SAS Management Console and select **Actions ▶ New Server**. From the New Server wizard, select **Object Spawner** and click **Next**. Fill in the appropriate fields as follows:

- ◆ **Name.** A unique name for the spawner (for example, WSSpawner).
- ◆ **Associated Machine.** The machine on which the spawner runs.

Note: For this basic setup, accept the default operator **Login** of None.

- ◆ **Servers.** Click **New** to open the New Server Wizard.

To create a new server, fill in the appropriate fields as follows:

- ◇ **Name:** Unique name for the server.
- ◇ **Select the type of server:** The type of server. Select **Workspace Server**.

Note: When the New Server wizard prompts you to select the **Basic** or **Custom** configuration method, you must choose **Custom**.

- ◇ **Command:** The command to launch SAS. For example,
Windows

```
"c:\Program Files\SAS\SAS 9.1\sas"  
-config "c:\Program Files\SAS\SAS 9.1\sasv9.cfg"
```

Unix

```
/usr/local/bin/sas
```

- ◇ **Authentication Domain:** The authentication domain for your server and spawner, (for example, DefaultAuth).
- ◇ **Host:** The machine name on which your server will run. This should be the same machine name as the spawner's **Associated Machine**.
- ◇ **Port:** The unique port number of 8591.

For more information about defining servers, see [Using SAS Management Console to Define Servers](#).

When you have finished setting up the new server, its name appears in the **Selected servers** field.

- ◆ **Authentication Domain:** The authentication domain for your server and spawner (for example, DefaultAuth).
- ◆ **Host:** Machine name of the server that you created for the **Servers** field.
- ◆ **Port:** Unique port number of 8581.

For more information about defining object spawners, see [Using SAS Management Console to Define or Modify a Spawner](#).

5. (Windows only) Define Windows User Rights for each client.

- a. For each client that connects to the spawner, grant the **Log on as a batch job** user right. For detailed instructions about defining Windows user rights, see [Starting the Spawner on Windows](#).
- b. Restart Windows to apply the new user rights.

6. (Unix only) Set the `setuid` root bit for `sasrun`, `sasauth`, and `elssrv`. See [Changing the `setuid` Permissions to Root](#) for details.

7. Use SAS to create a metadata configuration file named `objspawn.xml` that contains information for accessing the SAS Metadata Server. Start SAS and enter the `METACON` command on the SAS command bar. The Metadata Server Connections window appears. Enter the following information:

- ◆ **Name:** A name for the server connection.
- ◆ **Server:** The machine name of your SAS Metadata Server.
- ◆ **Port:** The port number for your metadata server. Specify 8561.
- ◆ **Protocol:** The protocol to use. Select **Bridge**.
- ◆ **User Name:** The user ID that is used to log on to the metadata server.
- ◆ **Password:** The password that is used to log on to the metadata server.
- ◆ **Repository:** The name of the repository that you created in step 3.

Click **Export** to save the metadata configuration file to a directory. For example:

Windows

```
c:\program files\sas\servers\objectspawner\objspawn.xml
```

Unix

```
/users/myid/objspawn.xml
```

8. Start the object spawner with the metadata configuration file that you created in the previous step.

Windows

To install the object spawner as a service, enter the following command at a command prompt:

```
"c:\Program Files\SAS\SAS 9.1\objspawn" -sasSpawnercn "WSSpawner"
-install -saslogfile c:\objspawnlog.txt -xmlconfigfile
"c:\program files\sas\servers\objectspawner\objspawn.xml"
```

Note: You must specify the fully qualified path to the configuration file.

Use the Windows `net start` command to start the object spawner as a Windows service (case does not matter):

```
net start "sas object spawner daemon II"
```

Note: In the Windows Services utility, the object spawner service appears as the SAS Object Spawner Daemon II.

Unix

To start the object spawner, enter the following command at a system prompt:

SAS® 9.1.2 Integration Technologies: Server Administrator's Guide

```
/sasv91/utilities/bin/objspawn -sasSpawnercn "WSSpawner"  
-xmlconfigfile /users/myid/objspawn.xml
```

For the complete list of spawner invocation options, see [Spawner Invocation Options](#).

9. Test your server using SAS Management Console. For details, see [Using SAS Management Console to Test Server Connections](#).

IOM Bridge

Quick Start: Load–Balancing Stored Process Server and Spawner

The following steps help you set up an IOM Bridge connection for a load–balancing SAS Stored Process Server that uses a spawner to start the server on a Windows or Unix platform. For details about setting up more complex configurations, see [Summary of Setup Steps \(IOM Bridge Connection\)](#).

To set up and test a load–balancing stored process server and spawner that use an IOM Bridge connection, complete the following steps:

1. Install SAS 9.1 (including SAS Integration Technologies and SAS Management Console) on the server machine. Refer to the SAS documentation for details about this procedure.
2. Determine the machine name and port for your SAS Metadata Server. Determine the fully–qualified user ID and password that you will use to connect to your SAS Metadata Server.
3. Set up and start your SAS Metadata Server. Then, connect to the server and register a repository. For details about setting up, starting, and connecting to the SAS Metadata Server, see the [SAS Metadata Server: Setup Guide](#).
4. Create metadata definitions for a user, login, group, and group login to use for the load–balancing server configuration. For more information about load–balancing security with a stored process server, see [Spawner Security Scenario](#).

To define a user and login:

- a. In SAS Management Console, select User Manager and then select **Actions ➤ New ➤ User** to open the General tab of the New User Properties window.
- b. Enter a **Name** for the user (for example, User A).
- c. Select the Logins tab and click **New** to open the New Login Properties window. Enter the fully–qualified **User Id** (for example, PC101\usera), **Password**, and **Authentication Domain** (for example, DefaultAuth) for the user login.

To define a group and group login:

- a. In SAS Management Console, select User Manager and then select **Actions ➤ New ➤ Group** to open the General tab of the New Group Properties window.
- b. Enter a **Name** for the group, for example, Group ABC.
- c. Select the Members tab. Select the user that you defined in the previous step and click the arrow button to add it to the **Current Members** panel.
- d. Select the Logins tab and click **New** to open the New Login Properties window. Enter the fully–qualified **User Id** (for example, PC101\groupabc), **Password**, and **Authentication Domain** (for example, DefaultAuth) for the group login.

5. Create metadata definitions for your SAS Stored Process Server and spawner. To create a spawner definition, select the Server Manager in SAS Management Console and select **Actions ➤ New Server**. From the New Server wizard, select **Object Spawner** and click **Next**. Fill in the appropriate fields as follows:

- ◆ **Name:** Unique name for the spawner (for example, SPSPawner).
- ◆ **Associate Machine:** The machine on which the spawner runs.

Note: For this basic setup, accept the default operator **Login** of **None**.

- ◆ **Servers:** Click **New** to open the New Server Wizard.

To create a new server, fill in the appropriate fields as follows:

- ◇ **Name:** Unique name for the server (for example, SASMain).
- ◇ **Select the type of server:** Select **Stored Process Server**.
- ◇ **Command:** The command to launch SAS. For example,

Windows

```
"c:\Program Files\SAS\SAS 9.1\sas"
-config "c:\Program Files\SAS\SAS 9.1\sasv9.cfg"
```

Unix

```
/sasv91/utilities/bin/sas
-config /sasv91/sasv91.cfg
```

Note: When the SAS Management Console New Server Wizard prompts you to select the **Basic** or **Custom** configuration method, you must choose **Custom**.

- ◇ **Authentication Domain:** The authentication domain for your server and spawner (for example, DefaultAuth).
- ◇ **Host:** The machine name on which your server will run. This should be the same machine name as the spawner's **Associated Machine**.
- ◇ **Port:** The unique port number of 8601.

For more information about defining servers, see [Using SAS Management Console to Define Servers](#).

When you have finished setting up the new server, its name appears in the **Selected servers** field.

- ◆ **Authentication Domain:** The authentication domain for your server and spawner, for example, DefaultAuth.
- ◆ **Host:** Machine name of the server that you created for the **Servers** field.
- ◆ **Port:** Unique port number of 8581.

For more information about defining object spawners, see [Using SAS Management Console to Define or Modify a Spawner](#).

6. Set up load balancing for the server:

1. Convert the logical server to a load-balancing logical server. In SAS Management Console, select the logical stored process server definition that you created in step 5. Select **Actions ► Convert to ► Load Balancing** to open the Load Balancing Options window. From the **Load Balancing Credentials** drop-down list, select the group login that you created in step 4.
2. On the load-balancing logical server definition, grant the **Administer** permission to the user or group that owns the logical server credentials, (for example, GroupABC).
3. Create MultiBridge connections. In SAS Management Console, select the stored process server definition that you created in step 5. Select **Actions ► Add Connection** to open the New Connection wizard. Fill in the appropriate fields as follows:
 - ◇ **Authentication Domain:** The authentication domain for your server and spawner (for example, DefaultAuth).
 - ◇ **Host Name:** The machine name on which your server will run.
 - ◇ **Port Number:** The unique port number of 8611.

Create two additional Multibridge connections on ports 8621 and 8631.
4. Optionally, set additional load-balancing parameters for the server. For more information, see [Adding Pooling or Load Balancing Parameters](#).

For more information about setting load balancing options, see [Using SAS Management Console to Define a Load Balancing Logical Server](#).

7. (Windows only) Define the Windows user rights for each client.
 - a. For each client that connects to the spawner, specify **Log on as a batch job**. For detailed instructions about defining Windows user rights, see [Starting the Spawner on Windows](#).
 - b. Restart Windows to apply the new user rights.
8. (Unix only) Set the `setuid` root bit for `sasrun`, `sasauth`, and `elssrv`. To set the `setuid` root bit, see [Changing the `setuid` Permissions to Root](#).
9. Use SAS to create a metadata configuration file named `objspawn.xml` that contains information for accessing the SAS Metadata Server. Start SAS, and then enter the METACON command on the SAS command bar. The Metadata Server Connections window appears. Enter the following information:
 - ◆ **Name:** A name for the server connection.
 - ◆ **Server:** The machine name of your SAS Metadata Server.
 - ◆ **Port:** The port number for your metadata server. Specify 8561.
 - ◆ **Protocol:** Select **Bridge**.
 - ◆ **User Name:** The user ID that you specified in step 4 (for example, `PC101\usera`).
 - ◆ **Password:** The password that you specified in step 4.
 - ◆ **Repository:** The name of the repository that you created in step 3.

Click **Export** to save the metadata configuration file to a directory. For example:

Windows

```
c:\program files\sas\servers\objectspawner\objspawn.xml
```

Unix

```
/users/myid/objspawn.xml
```

10. Start the object spawner with the metadata configuration file that you created in the previous step.

Windows

To install the object spawner as a service, enter the following command at a command prompt:

```
"c:\Program Files\SAS\SAS 9.1\objspawn"
-sasSpawnercn "SPSpawner"
-install -saslogfile c:\objspawnlog.txt
-xmlconfigfile
"c:\program files\sas\servers\objectspawner\objspawn.xml "
```

Note: When you install the spawner as a Windows service, you must specify the fully qualified path to the configuration file. When the spawner is started as a Windows NT service, it will self configure using the options that are placed in the registry at install time.

Use the Windows `net start` command to start the object spawner as a Windows service (case does not matter):

```
net start "sas object spawner daemon II"
```

Note: In the Windows Services utility, the object spawner service appears as SAS Object Spawner Daemon II.

Unix

To start the object spawner, enter the following command at the system prompt:

```
/sasv91/utilities/bin/objspawn  
-sasSpawnercn "SPSpawner"  
-xmlconfigfile /users/myid/objspawn.xml
```

For the complete list of spawner invocation options, see [Spawner Invocation Options](#).

11. Test your server using SAS Management Console. For details, see [Using SAS Management Console to Test Server Connections](#).

IOM Bridge

Summary of Setup Steps (IOM Bridge)

To set up a server that is configured with an IOM Bridge connection:

1. Install SAS 9.1 (including SAS Integration Technologies) on the server machine. Refer to the SAS documentation for the details about this procedure.
2. Plan your users, groups, and logins for security. For details, see the appropriate sections in the [Security](#) chapter.
3. Set up and start your SAS Metadata Server. In addition, you must connect to the SAS Metadata Server and register a SAS Metadata Repository. For details about setting up, starting, and connecting to the SAS Metadata Server, see the [SAS Metadata Server: Setup Guide](#).
4. Create the necessary definitions (on the SAS Metadata Server) for servers, spawners, users, groups, logins, pooled logical servers, and load-balancing logical servers. (For SAS Workspace Servers, optionally set up [load balancing or pooling](#). For SAS Stored Process Servers, [set up load balancing](#) and [create MultiBridge connections](#).)

Note: You do not need to create a server definition for a SAS Metadata Server unless it is required for a replication or promotion job definition. See the [SAS Management Console User's Guide](#) for detailed instructions about how to promote and replicate repositories.

For planning details, see [Planning the Configuration Metadata](#) and its related sections.

For details about using SAS Management Console to create the metadata, see [Creating the Metadata with SAS Management Console](#).

5. Depending on whether you are using a spawner, start the server as follows:

Note: You must start the SAS Metadata Server that contains your SAS Metadata Repository before you attempt to start any other IOM servers.

- ◆ If you are using a spawner (required for SAS Workspace Servers and SAS Stored Process Servers), create the metadata configuration file that contains information for accessing the SAS Metadata Server. (Ensure that you have planned for the appropriate login information to specify in the metadata configuration file. For details, see [Planning the Spawner Security](#).) For details about generating the metadata configuration file, see [Metadata Configuration File](#).

If you are using a z/OS server, refer to [Configuring and Starting the Object Spawner on z/OS](#).

If you are not using a z/OS server, launch the spawner. Refer to [Invoking \(Starting\) the Spawner](#) for examples and special security considerations. The command syntax varies based on the server platform:

- ◇ If you are using a Windows server, refer to [Starting the Spawner on Windows](#).
- ◇ If you are using a UNIX server, refer to [Starting the Spawner on UNIX](#).
- ◇ If you are using a VMS server, refer to [Starting the Spawner on VMS](#).

For all platforms, refer to the list of [Spawner Invocation Options](#).

- ◆ If you are not using a spawner (for OLAP servers and other SAS Metadata Servers), create a startup command for the server. In addition, you might want to start the server as a service. For details, see [Starting Servers](#).

6. For SAS Workspace Servers and SAS Stored Process Servers, [test the server connection](#).

7. Install the necessary components on each client machine.

◆ For Windows Clients:

- ◇ Install the SAS Integration Technologies software for Windows clients. For instructions, refer to Developing Windows Clients in the *SAS Integration Technologies Developer's Guide*.

◆ For Java Clients:

- ◇ Install the SAS Integration Technologies software for Java clients. For instructions, refer to Developing Java Clients in the *SAS Integration Technologies Developer's Guide*.
- ◇ If you are using the Java Connection Factory interface of SAS Integration Technologies 9.1 and not using a SAS Metadata Server, you must also create a server definition in `com.sas.services.connection.BridgeServer`. This is necessary in order to obtain a reference to an IOM object, such as a workspace. Refer to Creating a Server Object with Java for an example. For more information, see Using the Java Connection Factory in the *SAS Integration Technologies Developer's Guide*.

This completes the basic configuration steps that are necessary to do client development on a Windows or Java platform. For information about developing applications that access servers using IOM Bridge connections, refer to Developing Java Clients and Developing Windows Clients in the *SAS Integration Technologies Developer's Guide*.

IOM Bridge

Spawner Overview

The Object Spawner is a program that can run on the server host and listen for requests. You must use a spawner to run SAS Workspace Servers and SAS Stored Process Servers.

Before you can run the spawner, you must create a [Metadata Configuration File](#) that contains information for accessing the metadata server. When you invoke the spawner, you use this metadata configuration file to connect to the SAS Metadata Server for configuration information; the spawner can then listen for requests for various [Spawner Tasks](#). (For details about starting a spawner, see [Invoking \(Starting\) the Spawner](#).)

Metadata Configuration File

A metadata configuration file contains information for accessing a metadata server. The spawner uses the information contained in the configuration file to connect to a metadata server and read the appropriate server definitions. In order for the spawner to connect to and read the appropriate metadata from a metadata server, you must specify the appropriate login information in the metadata configuration file. For details, see [Planning the Spawner Security](#).

To create the metadata configuration file, see [Creating a Metadata Configuration File in SAS](#).

Spawner Tasks

When a request is received, the spawner accepts the connection and performs the action that is associated with the port or service on which the connection was made. A connection to a spawner can do the following:

- **request a server.** When a connection is made on a port or service that is associated with a Server object, the spawner authenticates the client connection against the host authentication provider for the server's machine. The spawner then launches a server for use by the connecting client. To launch the server, the spawner locates the associated server definitions on the SAS Metadata Server.

When you define a server in SAS Management Console, you must specify a command that the spawner will use to start the server. For details about the server command, see [Server Startup Command](#). For SAS Stored Process Servers, on the server definition, you must also configure credentials for the spawner to use to start a [multi-user server](#).

Every connection to the server is authenticated (against the host authentication provider for the server's machine) with the credentials of the client; depending on the type of server, the process then runs under the following credentials:

- ◆ For SAS Workspace Servers, the credentials of the client.
- ◆ For SAS Stored Process Servers, the multi-user login credentials that are specified in the stored process server definition (Advanced Options ➤ Credential in the New Server Wizard) in SAS Management Console.

To understand the different security considerations for SAS Workspace and SAS Stored Process Servers, see [Planning Security on Workspace and Stored Process Servers](#).

You can use normal server security mechanisms to protect sensitive data. For more information about server security, see the [Security](#) chapter.

- **initiate the operator interface.** When a connection is made on the port or service that is identified as the operator port or operator service in the spawner definition, the spawner initiates the administration interface. Only one administrator can be active at a given time. For more information about the administration interface, see [Using Telnet to Administer the Spawner](#).
- **request a Universal Unique Identifier (UUID).** A spawner can be configured to support UUID generation; or, it can be configured solely as a UUID generator daemon (UUIDGEND). In either case, when a connection is made on the port or service that is identified as a UUID port or UUID service in the spawner definition, the spawner initiates UUID generation. For more information, see [Configuring a UUID Generator](#).

In addition, for stored process servers, you must configure the spawner for [Load Balancing](#). You can also configure load balancing for SAS Workspace Servers.

Multi-User Server

For stored process servers, you must specify a login on the Credentials tab of the server definition advanced options. The spawner that is associated with the server invokes a multiple user server that runs under this login. Other clients of this server definition can then connect to the server that is running.

Note: If you do not specify a multi-user login for the stored process server, the stored process server will not run and a message similar to the following will be displayed:

```
This server (OMSOBJ:LOGICALSERVER/A5SRQ5Z5.AT00008E)
cannot be spawned without credentials which specify
the server process username
```

Note: Because the load-balancing stored process server runs under the multi-user login credentials, the operating system account for these credentials must have access to any operating system resources used by stored processes that are hosted on this server.

Load Balancing

You can set up load balancing for workspace servers; you *must* set up load balancing for stored process servers.

A load balancer routine runs in the spawner and directs client requests to the SAS process (on a server) that is least loaded (busy) at the time the client request is made. Subsequent calls between the client and SAS are then direct calls. The load balancer uses a load-balancing algorithm (cost or response time) to determine which server is least loaded.

When launching a load-balancing spawner, you specify a [Metadata Configuration File](#) that contains information for accessing the SAS Metadata Server. The spawner then reads the load-balancing configuration metadata from the SAS Metadata Server and uses the metadata to determine what other machines or ports are in the load-balancing cluster. The spawner attempts to establish an IOM connection to each spawner in the cluster. Additional spawners can be added to a cluster at any time.

For an overview of load balancing, see [Load Balancing](#).

IOM Bridge

Spawner Requirements

Hardware Requirements

The spawner can be installed on a server machine that runs in one of the following operating environments:

- z/OS
- OpenVMS Alpha
- UNIX
 - ◆ AIX 64
 - ◆ HP-UX IPF
 - ◆ HP 64
 - ◆ Tru64 UNIX
 - ◆ Solaris 64
 - ◆ RedHat Linux on Intel
- Windows NT/XP/2000

Software Requirements

Install the following software on the server machine:

- SAS 9.1 (or later)
- SAS 9.1 Integration Technologies
- SAS/SECURE (optional)
- SAS Metadata Server

IOM Bridge

Planning the Configuration Metadata

To plan the server metadata, you must first plan your SAS Application Server and logical server definitions. To understand the SAS Application Server and logical server concepts, see [Planning the Metadata](#). To plan your SAS Application Server and logical servers, determine the following information:

- the number of SAS Application Servers
- the number and type of logical servers within each SAS Application Server.

To understand and plan server security metadata, see [Security Metadata Overview](#).

To plan the server configuration metadata, see the appropriate server metadata topic:

- [Standard SAS Workspace or SAS Stored Process Server Metadata](#)
- [Standard SAS OLAP Metadata](#)
- [Pooling Metadata \(SAS Workspace Servers only\)](#)
- [Load Balancing Metadata \(SAS Workspace and SAS Stored Process Servers only\)](#)

IOM Bridge

Security Metadata

This section provides an overview of where you can associate logins within a server configuration that uses an IOM Bridge connection. Depending on your IOM Bridge connection setup, there are several different areas where you might provide security through the association of login definitions.

To understand security, see the [Security](#) chapter.

User and Login Metadata

Each SAS login definition contains a fully qualified user ID, password, and authentication domain. The administrator can establish multiple login definitions for each user or group metadata identity. For each login instance of the user, you must specify the following information:

- the SAS login (fully qualified user ID) and password
- authentication domain name

You might also add users to groups and define login definitions for the groups.

For detailed information about users, groups, and login definitions, see [Defining Users, Groups, and Logins](#).

Standard, Pooled, and Load–Balancing Security Metadata

For OLAP servers, you only need to define a login for the user's server connection. For SAS Workspace and SAS Stored Process Servers, you must plan and specify several different types of login credentials. To understand security differences between SAS Stored Process Servers and SAS Workspace Servers, see [Planning the Workspace and Stored Process Server Security](#). For details about planning the spawner security, and pooling and load–balancing security, see the following topics:

- For spawner security, see [Planning the Spawner Security](#).
- For pooling security, see [Planning the Pooling Security \(IOM Bridge only\)](#).
- For load–balancing security, see [Planning the Load–Balancing Security \(IOM Bridge only\)](#).

The following table shows the login credentials that are required for standard, pooled, and load–balancing server configurations. For logins that are configured in SAS Management Console, the Login row links to the SAS Management Console location where the login must be specified.

Workspace and Stored Process Server Login Requirements				
Login	Description	SAS Workspace Server	Pooled SAS Workspace Server	Load–Balancing Stored Process or Workspace Servers
Logins for Users who Connect to Servers	Login definitions associated to users that request connections to a server. The authentication domain of the server definition	Yes	No	Yes

	must match the domain of the login definition. If a domain match for a login cannot be found within a user definition, the groups that the user belongs to are searched for a login that matches the domain of the server definition.			
Login for User ID in the Metadata Configuration File (for the Spawner or Windows Object Manager)	<p>User ID in the metadata configuration file. You must specify the login credentials that the spawner or Windows Object Manager will use to connect to the SAS Metadata Server. This user ID must be able to access the operator ID and if specified, the multi-user login definition.</p> <p>Important Note: DO NOT specify an <i>unrestricted user</i> for the user ID in the metadata configuration file.</p>	Yes	Yes	Yes
<u>Operator Login for Spawners (optional)</u>	<p>Administrator login definition to access the operator port of the spawner. The login definition must be one of the following:</p> <ul style="list-style-type: none"> • the login definition for the user ID that you specified in the metadata configuration file • a login definition that the user ID in the metadata configuration file can access 	Yes	Yes	Yes
<u>Multi-User Login for SAS Stored Process Servers</u>	<p>Login for the multi-user server. The launched SAS process runs under the process ID defined by this login. The login definition must be one of the following:</p> <ul style="list-style-type: none"> • the login definition for the user ID that you specified in the metadata configuration file • a login definition that the user ID in the metadata configuration file can access 	No	No	Yes, only for SAS Stored Process Servers

	Note: Because the load-balancing stored process server runs under the multi-user login credentials, the operating system account for these credentials must have access to any operating system resources used by stored processes that are hosted on this server.			
If METAAUTOINIT is specified (and the trustsaspeer option is not specified), Metaprofile User ID	User ID that is specified for the metadata connection profile option (or server's metadata configuration file) to enable the server to connect back to the SAS Metadata Server. For details about using METAAUTOINIT, see Server Startup Command .	Yes	Yes	Yes
For Pooling, Login for Pool Administrator	Login for pool administrator credentials supplied by the application. These credentials are used to connect to the SAS Metadata Server and read the puddle login definitions.	No	Yes	No
<u>For Pooling, Puddle Login</u>	Login definition that is used to establish the connection to the server for this puddle. You might decide to partition your pool into puddles in order to allow different login definitions for different puddles within the pool. When you define the puddle, you must associate a login with the puddle.	No	Yes	No
<u>For Pooling, Login Definitions for Users that are Members of a Group Granted Access to the Puddle</u>	Logins for users in a group that is granted access to a puddle. If you want a user to have access to a puddle in a pool, you can define the user and its login definitions, and then add the user to a group. You can then grant this group access to the puddle.	No	Yes	No
<u>For Load-Balancing, Login for the Logical Server Credentials</u>	Login definition that is used by spawners to connect to other spawners for load balancing. The login definition must be one of the following:	No	No	Yes

	<ul style="list-style-type: none">• the login definition for the user ID that you specified in the metadata configuration file• a login definition that the user ID in the metadata configuration file can access			
--	--	--	--	--

IOM Bridge

Standard Workspace or Stored Process Server Metadata

Before you can plan and set up servers and spawners, you must understand the following:

- the SAS Application Server and logical server definitions that contain the server definitions. For details, see [Planning the Metadata](#).
- the security implementation for spawners and servers. For details, see [Planning the Spawner Security](#).

You can then use the steps in the following section to plan the servers and spawners and link to instructions to set up servers and spawners.

To plan a standard server with an IOM Bridge connection, you must determine the following information:

- the number of servers you need. Decide how many servers you need for your implementation.
- the number of logical servers and SAS Application Servers you need. Decide which logical servers and SAS Application Servers will contain your server definitions.
- the number of spawners you need. Each server can only be associated with one specific spawner. You must use a spawner with SAS Workspace Servers and SAS Stored Process Servers.

To set up a standard server with an IOM Bridge connection, plan and set up metadata for the following:

1. **Plan the Logins**. You might need to plan the following logins:
 - ◆ metadata configuration file login for the spawner
 - ◆ operator login for the spawner
 - ◆ if using a SAS Stored Process Server with a spawner, you must plan a multi-user login for the spawner to use to start the server
 - ◆ logins for users that connect to the server
 2. **Plan the Servers**. You must plan the server definitions for servers that you will use to process client requests.
 3. **Plan the Spawners**. For each spawner, you must plan which servers to associate with the spawner in order to listen for requests for each server. Associate each server with a single spawner.
 4. **Set up Logins**. You must set up the appropriate logins:
 - ◆ metadata configuration file login for the spawner
 - ◆ operator login for the spawner
 - ◆ if using a SAS Stored Process Server with a spawner, you must set up a multi-user login for the spawner to use to start the server
 - ◆ logins for users that connect to the server
 5. **Set up Servers**. You must set up the server definitions for servers that you will use to process client requests.
 6. **Set up Spawners**. You must set up the spawner. Associate each server with a single spawner.
-

Step 1: Plan the Logins

You must determine the number and type of logins that you need to define. For the basic server and spawner configuration, determine how many separate logins you need for the following types of logins:

- **metadata configuration file login:** If you use spawners, you must plan and define a login to use in the metadata configuration file to connect to the SAS Metadata Server.
- **operator logins:** If you use spawners, for each spawner, you must plan and define a login to be used as the administrator (operator) login for the spawner. You must use one of the following:
 - ◆ the same login that you specified in the metadata configuration file

- ◆ a login that the login in the metadata configuration file can access
- **multi-user login (SAS Stored Process Server only):** For each SAS Stored Process Server that you start with a spawner, the multi-user login used by the spawner to start the server. You might use the same login to access different multi-user servers. This login must also be accessible by the login in the metadata configuration file.

Note: If you do not specify a multi-user login for the stored process server, the stored process server will not run and an error message will be displayed.

For details about how to plan the spawner and server configuration logins, see [Planning the Spawner Security](#).

In addition, you must determine which login definitions you need for users that request connections to a server. The authentication domain of the server definition must match the domain of the login definition. To understand how to plan your authentication domain, see [Understanding Security](#).

To plan each login definition, you must determine the following information:

- the user name of the user metadata identity
 - the domain-qualified user ID and password
 - the authentication domain name.
-

Step 2: Plan the Servers

To plan each server, you must determine which SAS Application Server and logical server will contain the server definition. You must also determine the following server parameters:

- server name
 - authentication domain
 - host name, and service or port for the bridge connection
 - type of encryption you will use
 - object server parameters, as required
 - for SAS Stored Process Servers, the multi-user login for the server
 - SAS startup command and options, as required. For details, see [Server Startup Command](#).
-

Step 3: Plan the Spawners

To plan each spawner, you must determine the following information

- the spawner name
- the name of the servers that the spawner is associated with
- the authentication domain (must match the associated server's authentication domain)
- the host name and operator port of the spawner in order to set up an operator connection.

In addition, you can plan to set up a UUID connection. See [Configuring a UUID Generator](#) for further information.

For detailed information about the fields included in the metadata for a spawner, see the [Fields for Spawner Definitions](#).

Note: You can only define one of each type of spawner connection (operator, UUID, or load balancing).

For detailed information about the fields included in the metadata for a server, see the [Fields for Server Definitions](#).

Step 4: Set up Logins

Use SAS Management Console to set up users, groups, and logins on the SAS Metadata Server. For detailed information about the fields included in the metadata for a user and login and how to set up users, groups and logins, see the [Defining Users, Groups, and Logins](#) in the Security chapter.

In addition, on Windows 2000 and Windows NT, for the user who is the owner of the multi-user login, define the "act as part of the operating system" user right.

Step 5: Set up Servers

Use SAS Management Console to set up the servers within the appropriate SAS Application Server and logical server. For detailed information about using SAS Management Console to set up a new server definition, see [Using the SAS Management Console to Define Servers](#).

Step 6: Set up Spawners

Use SAS Management Console to set up the spawner. For detailed information about using SAS Management Console to set up a spawner definition, see [Using SAS Management Console to Define a Spawner \(IOM Bridge\)](#).

IOM Bridge

Standard OLAP Server Metadata

To set up an OLAP server with an IOM Bridge connection, you must create metadata that describes your server configuration. For information about the SAS Application Server and logical server definitions that contain the server definitions, see [Planning the Metadata](#).

To plan a standard OLAP server with an IOM Bridge connection, you must determine the following:

- the number of servers you need. Decide how many servers you need for your implementation.
- the number of logical servers and SAS Application Servers you need. Decide which logical servers and SAS Application Servers will contain your server definitions.

To set up a standard OLAP server with an IOM Bridge connection, plan and set up metadata for the following:

1. **Logins.** You might need to plan and set up logins for users that connect to the server. The domain of the login definition and the authentication domain of the server definition must match in order to associate the server with the appropriate login credentials.
 2. **Servers.** You must plan and set up the server definitions for servers that you will use to process client requests.
-

Step 1: Plan and Set Up Logins

You must determine the number of logins that you need to define. For the basic server configuration, determine how many separate logins you need for logins associated with users that request connections to a server. The authentication domain of the server definition must match the domain of the login definition. To understand how to plan for your authentication domain, see [Overview of Domains](#).

To plan each login definition, you must determine the following information:

- the user name of the user metadata identity
- the fully qualified user ID and password
- the authentication domain name.

For detailed information about the fields included in the metadata for a user and login and how to set up SAS users, groups, and logins, see the [Defining Users, Groups, and Logins](#) in the Security chapter.

Step 2: Plan and Set Up Servers

To plan each server, you must determine which SAS Application Server and logical server will contain the server definition. You must also determine the following server parameters:

- server name
- authentication domain
- host name, and service or port for the bridge connection
- type of encryption you will use
- object server parameters, as required
- SAS startup command and options, as required. For details, see [Server Startup Command](#).

SAS® 9.1.2 Integration Technologies: Server Administrator's Guide

For detailed information about the fields included in the metadata for a server, see the [Fields for Server Definitions](#).

For detailed information about using SAS Management Console to set up a new server definition within the appropriate SAS Application Server and logical server, see [Using SAS Management Console to Define Servers](#).

IOM Bridge

Pooling Metadata

For SAS Workspace Servers, you can set up pooling for a group of servers within a logical server. Before you can plan and set up pooling, you must understand pooling and its security implementation. For details, see [Overview of Pooling](#) and [Planning the Pooling Security](#). You can then use the steps in the following section to plan pooling and link to instructions about how to set up pooling.

To plan for a pool, you must determine the following:

1. **The number of servers (one or more) in the pool.**
2. **The standard server metadata.** For each server in the pool, plan and set up the standard server, spawner, and login definitions. When you set up your standard server metadata, you must define the servers within the same logical server definition (that will be converted to a pooled logical server). For details about planning and setting up the standard server metadata, see [Standard Server Metadata](#).

Project Installation Note: For a project installation using SAS Configuration Wizard, you will already have a logical server named SASMain – Logical Workspace Server (that contains one or more server definitions), which you can then configure for pooling.

3. **The number of puddles to separate your pool into for security.** See [Planning the Pooling Security](#) for information about using puddles and security.
4. **The number of different logins you need based on Step 3.** You must define one login for each puddle. You might also want to define user metadata identities (and their associated logins) as members of the group metadata identities that you associate with puddles.

To set up a pool, you must plan and set up additional metadata for the following:

1. [Plan the Pooling Security](#). To set up pooling security, you must plan the logins that can access the SAS servers in the puddles, the group metadata identities that can access the puddle, and the pool administrator's user and group metadata identities. In addition, you must plan the appropriate access control for your pooling resources.
2. [Plan Pooled Logical Server and Puddles](#). To set up a pool, you must plan converting a standard logical server to a pooled logical server, puddles, pooling properties, an associated login for each puddle, and a group that is granted access to the puddle.
3. [Plan for Servers](#). To set up each server for pooling, you must plan pooling properties for each server.
4. [Set up Pooling Security](#). To set up pooling security, you must define the puddle logins, group metadata identities that are granted access to the puddles, and user or group metadata identity for the pool administrator. You must also implement authorization for the appropriate pooling resources.
5. [Set up Pooled Logical Server and Puddles](#). To set up a pool, you must convert a standard logical server to a pooled logical server, create the puddles, specify pooling properties, associate the login for each puddle, and associate a group that can access the puddle.
6. [Set up Servers](#). To set up each servers for pooling, on each server definition, you must specify pooling properties for the server.

Step 1: Plan the Pooling Security

To plan the pooling security, you must determine the user metadata identities and logins for the group metadata identities that can access the puddles in the pool. For puddle access to the pool, there are three types of logins you might define:

- Login that is used to establish the connection to the server for this puddle. All users of the puddle use this login when connecting to the SAS server. This login must be accessible to pool administrators. (Pool users are not required to have access to this login).
- Logins for the pool administrator in the metadata configuration file used with the Windows Object Manager.

Important Note: DO NOT specify an *unrestricted user* for the user ID of the pool administrator.

- Logins for user group metadata identities within the group that you grant access to the puddle.

To understand the login, user, and group definitions that can access the puddles, see [Overview of Pool and Puddle Configuration](#). To plan the login, user, and group definitions that can access the puddles and the pool administrators that can view the appropriate login definitions, see [Planning the Pool and Puddle Security](#).

Step 2: Plan the Pooled Logical Server and Puddles

To plan a pooled logical server, you need to determine how many puddles you want to use and which logins will be used to access each of the puddles. When you convert the logical server to a pooled logical server, you can then divide the pool into one or more puddles that associate the appropriate login definition and group metadata identity to use for access to the pool. The login for each puddle will be used to access the server. The user metadata identities in the group metadata identity granted access to the puddle, and the user or group metadata identity of the puddle login used to connect to SAS are also granted access to access the servers in the puddle.

The puddle parameters are specified on the logical server definition's properties in SAS Management Console (**Options ➤ Puddles**). Determine the following parameters for each puddle associated with the pooled logical server definition:

- [Name](#) of each puddle
 - [Minimum Available Servers](#)
 - [Minimum Number of Servers](#)
 - Puddle [Login](#)
 - SAS [Group](#) that is granted access to the puddle
-

Step 3: Plan the Pooled Servers

To plan server pooling, you must determine pooling properties for the servers that are contained in the pooled logical server.

The pooling properties are specified on the server definition's properties in SAS Management Console (**Options ➤ Advanced Options ➤ Pooling Properties**). For each server in the pooled logical server, determine the following appropriate pooling properties:

- [Maximum Clients](#)
 - [Recycle Activation Limit](#)
 - [Inactivity Timeout](#)
-

Step 4: Set up Pooling Security

To set up pooling security:

1. Set up your user, group, and login definitions for the users and groups that will access the pool. To understand user, group, and login definition structure, and how to set up a user (or group of users) and its associated logins on the SAS Metadata Server, see [Defining Users, Groups, and Logins](#).
2. Implement authorization (access control) for the group metadata identity that is granted access to the puddle. You must control access for who is authorized to update the group that is granted access to each puddle. To control who can update the group that is granted access to the puddle, in SAS Management Console, after you set up the group, you must use the Authorization tab for the group to do both of the following:
 - ◆ Deny "WriteMetadata" permission to the `Public` group.
 - ◆ Grant "WriteMetadata" permission to your metadata administrator.
3. Implement authorization (access control) for the logical server (that will be converted to a pooled logical server). You must control access for who is authorized to update the logical server. To control who can update the logical server, in SAS Management Console, you must use the Authorization tab for the logical server to do both of the following:
 - ◆ Deny "WriteMetadata" permission to the `Public` group.
 - ◆ Grant "WriteMetadata" permission to your metadata administrator.
4. Implement authorization (access control) for data on the server.

For details about setting up authorization (access controls), see the Authorization Manager Help in SAS Management Console.

Step 5: Set up Pooled Logical Servers

Use SAS Management Console to define a pooled logical server and puddles. To convert a logical server to a pooled logical server and define puddles, see [Using SAS Management Console to Define a Pooled Logical Server \(IOM Bridge\)](#).

Step 6: Set up Pooled Servers

For each server in the pool, use SAS Management Console to specify pooling properties on the server. To modify a server and set up the server pooling properties, see [Adding Pooling and Load-Balancing Parameters to an Existing Server's Properties](#). To define a server and set up the server pooling properties, see [Using SAS Management Console to Define Servers](#). *IOM Bridge*

Load Balancing Metadata

With SAS 9.1 Integration Technologies, you can configure load balancing for SAS Workspace Servers; you *must* configure load balancing for SAS Stored Process Servers. Before you can plan and set up load balancing, you must understand load balancing and its security implementation. For details, see [Overview of Load Balancing and Planning the Load Balancing Security](#). You can then use the steps in the following section to plan load balancing and link to instructions about how to set up load balancing.

To configure load balancing, you use SAS Management Console to set up load balancing for a cluster. In SAS Management Console, servers are grouped within logical servers. To set up load balancing, you convert the logical server group to a load balancing logical server group that is then equivalent to a load balancing cluster.

Project Installation Note: For a project installation using SAS Configuration Wizard, you will already have a logical server named SASMain – Logical Workspace Server or a load-balancing logical server named SASMain – Logical Stored Process Server. The load-balancing logical server SASMain – Logical Stored Process Server will contain at least one stored process server definition with three connections defined for the server.

To set up load balancing, you must plan and set up the following:

- the number of servers in the load-balancing logical server. Depending on the type of server, you can set up load balancing in one or both of the following ways:
 - ◆ For SAS Workspace Servers and SAS Stored Process Servers, add multiple physical servers to a load-balancing logical server.
 - ◆ For SAS Stored Process Servers, you must add multiple connections to each physical stored process server within a load-balancing logical stored process server.
- the standard server metadata. For each server within your load-balancing logical server, plan and set up the standard server, spawner, and login definitions. When you set up your standard server metadata, you must define the servers within the same logical server definition (that will be converted to a load-balancing logical server). You must then define spawners and associate each server with a spawner in order for the server to participate in load balancing. For details about planning for and setting up the standard server metadata, see [Standard Server Metadata](#).

If you have installed and configured with the project install and SAS Configuration Wizard, you will already have a logical server named SASMain – Logical Workspace Server or SASMain – Logical Stored Process Server (that contains one or more server definitions).

Important Note: Each client's credentials must be able to authenticate against any server in the load-balancing logical server (cluster). Therefore, when you define servers within a load-balancing logical server (cluster), you must use the same authentication domain for each server.

When you set up servers for load-balancing, in the server definition, you can specify either a port or a service for the server.

To set up load balancing, you must plan and set up additional metadata for the following:

1. **[Plan the Logins \(When Load Balancing Across Multiple Machines\)](#)**. If you have more than one spawner associated with the servers in your load-balancing logical server, you must plan for a login to use for connections between the different spawners.
2. **[Plan the Load-Balancing Logical Server](#)**. You must plan for the appropriate load-balancing properties. For more information, see [Determine the Load-Balancing Properties](#).

3. **Plan the Servers**. For each server in the load-balancing logical server, you must plan the appropriate load-balancing properties for the server definition. You might designate certain load-balancing properties on each server definition in order to increase performance for your implementation. For SAS Stored Process Servers, you must also plan multi-bridge connections for each server
 4. **Plan the Spawners**. You must plan a load-balancing connection for the spawner definition.
 5. **Set up Logins (When Load Balancing Across Multiple Machines)**. You must set up the login definition that you will specify on the load-balancing logical server definition (the spawner-to-spawner login) when you convert the logical server to a load-balancing logical server.
 6. **Set up Load-Balancing Logical Server**. You must convert the standard logical server to a load-balancing logical server definition. When you convert the server to a logical server, you must specify the logical server credentials (used to connect between multiple spawners) and load-balancing properties.
 7. **Set up Servers**. For each server in the load-balancing logical server, you must set up load-balancing properties on the server definition and, for SAS Stored Process Servers, define multi-bridge connections.
 8. **Set up Spawners**. You must set up a load-balancing connection for the spawner definition.
-

Determine the Load-Balancing Properties

SAS 9.1 Integration Technologies supports the cost algorithm for SAS Workspace Servers and SAS Stored Process Servers, and the response time algorithm for SAS Stored Process Servers only.

Cost Algorithm

When you use the cost algorithm, the spawner's load balancer looks at the cost values for all of the servers in the logical server. The cost algorithm picks the server that has the lowest cost value. The cost algorithm works differently for SAS Workspace Servers and SAS Stored Process Servers.

- **SAS Workspace Servers**. When a client connects it will be redirected to another (or possibly the same) server in the load-balancing cluster. When the client reconnects to the designated server, load balancing will increment that server's cost by a specified value (cost per client). When that client disconnects, load balancing will decrement that server's cost by the same value (cost per client).
- **SAS Stored Process Servers**. SAS Stored Process Servers process requests according to the amounts of work to be performed. For example, some clients might be performing intensive calculations and others might be connecting to a server only to retrieve data from it. SAS Stored Process Servers tell the load balancer their current cost value based on the number of clients and the amount of work each client is performing. The load balancer then uses these cost values to determine which server has the lowest cost; when a client makes a connection request, the load balancer directs the client to that server.

Cost Algorithm Example

For example, server A has a current cost of 300 and a maximum cost of 500. Server B has a startup cost of 200. The next client that connects is redirected to server B because server B's startup cost value is less than server A's current cost value ($200 < 300$). When server B starts, its current cost is set to 0 and then modified when the client connects to it.

Cost Algorithm Parameters

The cost algorithm uses the following cost parameters, which are treated as weighted values:

Cost Per Client (Field on the Load-Balancing Logical Server definition)

specifies the default amount of weight (cost) that each client adds (upon connection) or subtracts (upon disconnection) to the total cost of the server.

Startup Cost (Field on the Server definition)

specifies the startup cost of the server. When a request is made to the load balancer, the load balancer assigns this startup cost value to inactive servers. A new server is not started unless it is determined that its cost (the startup cost) is less than the rest of the servers in the cluster. This field enables the administrator to control the order in which servers are started. After a server is started, the cost value is 0. When a client connects to the server, the server's cost value is increased.

Maximum Cost (Field on the Server definition)

specifies the maximum cost value that each server can have. After a server reaches maximum cost, the load balancer will not redirect any more clients to the server until its cost value decreases.

Cost Utilizations

When you use the cost algorithm, you can specify the cost parameters in order to create a desired cost utilization:

Breadth—first utilization of servers

defined by specifying the Startup Cost to be less than the Cost Per Client. The load balancer will then pick a different connection for each new client, because it will cost less to start a new server (a different connection) than to add that client to an already running server. This parameter specification should cause all of the connections to have one connected client before any connection gets a second client.

Depth—first utilization of servers

defined by setting the Startup Cost to be greater than or equal to the Maximum Cost. The load balancer then continues to pick the same server (connection) for each new client, because it will cost more to start a new server (a different connection) than to add that client to an already running server. The load balancer continues to pick the same connection until the cost for that connection is greater than the Maximum Cost. This parameter specification should cause a connection to get a maximum number of clients (based on the values of Maximum Cost and Cost Per Client) before the load balancer attempts to start another server.

Maximum number of clients per server (SAS Workspace Server only)

defined by setting the Maximum Cost value equal to the Cost Per Client multiplied by the desired maximum number of clients, e.g., Maximum Cost= Cost Per Client *Maximum Clients.

Response Time Algorithm (SAS Stored Process Server only)

The response time algorithm uses a list of server response times in order to determine which server to use for the client's request. For each server in the load-balancing logical server, the load balancer maintains an ordered list of servers and their average response times. The load balancer reads this list (in a round-robin manner) and distributes clients across the servers in this list. The load balancer processes the next client request by using the machine at the top of the list. The load balancer updates the server response times periodically. You can specify the update frequency for the response time in the metadata (Response Refresh Time).

Note: Because each spawner's load balancer maintains its own list, if two different clients connect to two different spawners at the same time, they could get directed to the same multi-user server.

The response time algorithm uses the following parameters:

Response Refresh Rate (Field on the Load-Balancing Logical Server definition for SAS Stored Process Servers only)

specifies the length of the period in milliseconds that the load balancer will use the current response times. At the end of this period the load balancer updates the response times and reorders the servers for all the servers in the logical server.

Note: If this field is set to 0, the load balancer does not use the response time list to redirect clients to servers; instead, the load balancer redirects clients in a round-robin manner.

Max Clients (Field on the Server definition for SAS Stored Process Servers only)

specifies the maximum number of clients that a server can have. After a server reaches its maximum number of clients, the load balancer will not redirect any more clients to the server until a client disconnects.

Step 1: Plan the Logins (When Load Balancing Across Multiple Machines)

To enable load balancing between spawners:

1. Plan your logical server credentials login. When a spawner must connect to another spawner (within a load-balancing logical server group) to obtain load-balancing information, this login is used to authenticate the spawner-to-spawner connection. The login for the logical server credentials must be a login that each spawner can access in order to share load-balancing information.
 2. Plan to grant the `Administer` permission, on the load-balancing logical server definition, to the user or group metadata identity that owns the login for the logical server credentials. To plan the spawner and load-balancing logical server security, see [Planning the Spawner Security](#), and [Planning the Load Balancing Security](#).
-

Step 2: Plan a Load-Balancing Logical Server

To plan load balancing, you must plan to convert a logical server to a load-balancing logical server. To plan a load-balancing logical server, you must determine the following load-balancing properties for the load-balancing logical server definition. These load-balancing properties are specified on the Options tab of the load-balancing logical server definition's properties in SAS Management Console. For more information, see [Determine the Load-Balancing Properties](#).

- [Balancing Algorithm](#)
 - [Response Refresh Time \(SAS Stored Process Servers and Response Time Algorithm only\)](#)
 - [Cost Per Client](#)
 - [Login](#)
-

Step 3: Plan the Load-Balancing Servers

To plan load-balancing servers, for each server in the load-balancing logical server, you must determine the following load-balancing properties. (The load-balancing properties are specified on the server definition's properties in SAS Management Console (**Options** ➤ **Advanced Options** ➤ **Load Balancing Properties**):

- [Maximum Clients \(SAS Stored Process Servers only\)](#)
- [Maximum Cost](#)
- [Startup Cost](#)
- [Availability Timeout](#)
- [Start Size \(SAS Stored Process Servers with MultiBridge Connections only\)](#)
- [Recycle Activation Limit \(SAS Stored Process Servers only\)](#)
- [Inactivity Timeout \(SAS Stored Process Servers only\)](#)

For SAS Stored Process Servers, you must also plan MultiBridge connections. When you define a MultiBridge connection, you define a unique port for the connection. Each MultiBridge connection then defines a different process on your machine. For example, if you define a server definition with three MultiBridge connection definitions, you will have three processes. For an overview of MultiBridge connections, see [MultiBridge Connections](#).

Note: When you set up servers for load-balancing, in the server definition, you can specify either a port or a service for the server.

Step 4: Plan the Load-Balancing Spawners

You must plan to set up a load-balancing connection on the spawner definitions. This load-balancing connection is used to communicate between spawners for load balancing. To plan the load-balancing connection, determine the following information:

- the name for the connection
- the authentication domain (use the same authentication domain that is used for the load-balancing servers)
- the host name
- the port number (default is 8582).

For detailed information about the fields included in the metadata for a spawner, see the [Fields for Spawner Definitions](#).

Step 5: Set up Logins (When Load Balancing Across Multiple Machines)

If you are load-balancing between spawners:

1. Use SAS Management Console to set up the login (logical server credentials) for the load-balancing logical server. To understand user, group, and login definitions, and modify a user and its associated logins, see [Defining Users, Groups, and Logins](#).
 2. On the Authorization tab of the load-balancing logical server definition, grant the `Administer` permission to the user or group metadata identity that owns the login for the logical server credentials.
-

Step 6: Set up a Load-Balancing Logical Server

You must use SAS Management Console to set up a load-balancing logical server. To set up load balancing, you must convert a logical server to a load-balancing logical server and specify the load-balancing algorithm parameters (including the logical server credentials that are used to connect between multiple spawners). For details, see [Using SAS Management Console to Define a Load-Balancing Logical Server \(IOM Bridge\)](#).

Step 7: Set up Load-Balancing Servers

For each server in the load-balancing logical server, you must set up load-balancing properties for the server. To set up load-balancing servers:

1. Use SAS Management Console to specify load-balancing properties. To modify a server and set up the server

load-balancing properties, see [Adding Pooling or Load-Balancing Properties to an Existing Server's Properties](#). (To define a new server and set up the server load-balancing properties, you must first plan and set up the standard server, spawner, and login definitions specified in [Standard Server Metadata](#). For details about defining new servers, see [Using SAS Management Console to Define Servers](#)).

2. If you are setting up a SAS Stored Process server, you must use SAS Management Console to set up your MultiBridge connections. To modify a server and add MultiBridge connections, see [Adding a MultiBridge Connection to an Existing Server](#).
-

Step 8: Set up Load-Balancing Spawners

You must use SAS Management Console to modify a spawner and set up the spawner's load-balancing connection. For details, see [Using SAS Management Console to Add a Spawner Connection](#).

IOM Bridge

Creating the Metadata Using SAS Management Console

If you are using the SAS Metadata Server, you can use the SAS Management Console graphical user interface to create and modify the metadata for your server with an IOM Bridge connection. For information about SAS Management Console, from the SAS Management Console menu bar, select **Help ▶ Help on SAS Management Console**. For Help on the fields in a particular window, click **Help** in that window.

Before you can create definitions on your SAS Metadata Server, you must set up a SAS Metadata Server. You must also use SAS Management Console to create a repository. For details, see the [SAS Metadata Server: Setup Guide](#).

To understand how the server metadata is structured in SAS Management Console, see [Planning the Metadata](#). After you understand the metadata structure and have connected to a metadata repository, refer to the following sections for specific instructions about using SAS Management Console to set up a server configuration with an IOM Bridge connection:

- [Using SAS Management Console to Define Servers](#)
- [Using SAS Management Console to Modify Servers](#)
- [Using SAS Management Console to Define Custom Parameters for SAS Workspace or SAS Stored Process Servers \(IOM Bridge\)](#)
- [Using SAS Management Console to Define an OLAP Server \(IOM Bridge\)](#)
- [Using SAS Management Console to Define or Modify a Spawner \(IOM Bridge\)](#)
- [Using SAS Management Console to Modify Servers for Pooling or Load Balancing \(IOM Bridge\)](#)
- [Using SAS Management Console to Define a Pooled Logical Server \(IOM Bridge\)](#)
- [Using SAS Management Console to Define a Load-Balancing Logical Server \(IOM Bridge\)](#)

To understand and set up user, group, and login definitions for security, see [Defining Users, Groups, and Login Definitions](#)

For OLAP servers, to add a COM connection to an existing server, see [Adding a COM Connection](#) in the COM/DCOM chapter.

IOM Bridge

Using SAS Management Console to Define Servers

The SAS Management Console Server Manager provides a graphical user interface that allows you to create or modify a definition for the following servers that use an IOM Bridge connection:

- SAS Workspace Server
- SAS Stored Process Server
- SAS OLAP Server
- SAS Metadata Server (only for replication and promotion to other metadata servers. See the [SAS Management Console: User's Guide](#) for detailed instructions about how to promote and replicate repositories.)

For details about using SAS Management Console, from the SAS Management Console menu bar, select **Help ► Help on SAS Management Console**. For more information about the fields in the New Server Wizard, click **Help** from within the wizard.

Before you begin defining servers, you must have the following:

- a metadata profile for connecting to a metadata repository. For details about setting up this profile and a repository, see [SAS Metadata Server: Setup Guide](#).
- an understanding of the server metadata structure. For an overview of SAS Application Servers and logical server groupings, see [Planning the Metadata](#).
- the appropriate login, user, and group definitions for your server configuration. For details, see [Security Metadata](#) and the [Security](#) chapter.

Note: If you want a spawner to start the servers, after you define your servers you must define a spawner and designate which servers the spawner will start. If you subsequently define additional servers, you must modify the spawner and designate those additional servers that you want the spawner to start.

To define an IOM Bridge connection using SAS Management Console:

1. Start SAS Management Console and connect to a metadata repository.
2. Choose the appropriate scenario for defining your server:
 - ♦ **New SAS Application Server, logical server, and server.** To define a server and logical server in a new SAS Application Server, see [Defining a New Server and New Logical Server in a New SAS Application Server](#).
 - ♦ **New logical server and server.** To define a server in an existing SAS Application Server but within a new logical server, see [Defining a New Server and New Logical Server in an Existing SAS Application Server](#).
 - ♦ **New server.** To define a server in an existing SAS Application Server and logical server, see [Defining a Server in an Existing Logical Server](#).

Defining a New Server and New Logical Server in a New SAS Application Server

To define a new server, logical server, and SAS Application Server:

1. From the SAS Management Console navigation tree, select Server Manager, then select **Actions ➤ New Server** from the menu bar. The New Server Wizard appears. A list of resource templates is displayed.
2. Select the **SAS Application Server**. Click **Next**.
3. Enter the Name and Description. The name you provide will be the name of the SAS Application Server. Click **Next**.
4. Verify that the Minor Version Number, Major Version Number, Software Version, and the Vendor are correct. Click **Next**. A list of defined server resource templates is displayed. If your server type is not displayed, click **Cancel**.
5. Select the type of server you want to define. The type you choose will be the type of the first logical server and server in the SAS Application Server definition. For example, if you select workspace server as the server type, the SAS Application Server will contain a logical workspace server which in turn contains a workspace server. Click **Next**.
6. Select **Custom** and click **Next**.
7. Verify that the Minor Version Number, Major Version Number, Software Version, and the Vendor are correct.

For SAS Workspace Servers or SAS Stored Process Servers:

- ◆ Enter the command (Command) used to start the server. (For details, see Server Startup Command. Include the path relative to the directory in which the spawner will be started.)

Note: You must specify a command in order to launch the server with a spawner.

- ◆ Specify any additional object server parameters (Object Server Parameters) to use to launch SAS.

8. To continue configuring the server and add advanced options, see the appropriate topic for your type of server:

- ◆ Using SAS Management Console to Define Custom Parameters for Workspace or Stored Process Servers
- ◆ Using SAS Management Console to Define an OLAP Server

Defining a New Server and New Logical Server in an Existing SAS Application Server

To define a new server and new logical server in an existing SAS Application Server:

1. From the SAS Management Console navigation tree, select and expand the Server Manager and locate the SAS application server under which you want to add the new server. The SAS Application Servers are located one folder level below the Server Manager. Select the appropriate SAS Application Server, and then select **Actions ➤ Add Application Server Component** from the menu bar. The New Server Wizard displays. A list of server resource templates is displayed.

Note: A SAS Application Server can only contain one logical server of each of the following types: SAS Workspace Server, SAS Metadata Server, SAS Stored Process Server, and SAS OLAP Server.

2. Select the type of server you want to define. The type that you select will be the type of logical server and server defined. Click **Next**.
3. Select **Custom** and click **Next**.
4. Enter the Name and Description. The name you provide will be the name of the server. Click **Next**.

5. Verify that the Minor Version Number, Major Version Number, Software Version, and the Vendor are correct.

For SAS Workspace Servers or SAS Stored Process Servers:

- ◆ Enter the command (Command) used to start the server. (For details, see Server Startup Command. Include the path relative to the directory in which the spawner will be started.)

Note: You must specify a command in order to launch the server with a spawner.

- ◆ Specify any additional object server parameters (Object Server Parameters) to use to launch SAS.

6. To continue configuring the server and add advanced options, see the appropriate topic for your type of server:

- ◆ Using SAS Management Console to Define Custom Parameters for Workspace or Stored Process Servers
- ◆ Using SAS Management Console to Define an OLAP Server

Defining a New Server in an New Existing Logical Server

To define a new server in an existing logical server and SAS Application Server:

1. From the SAS Management Console navigation tree, select and expand the Server Manager, then select and expand the SAS Application Server that contains the logical server under which you want to add the new server. Then, select the appropriate logical server, and select **Actions ➤ Add Server** from the menu bar. The New Server Wizard displays.
2. Enter the Name and Description. The name you provide will be the name of your server. Click **Next**.
3. Verify that the Minor Version Number, Major Version Number, Software Version, and the Vendor are correct. Click **Next**.

For SAS Workspace Servers or SAS Stored Process Servers:

- ◆ Enter the command (Command) used to start the server. (For details, see Server Startup Command. Include the path relative to the directory in which the spawner will be started.)

Note: You must specify a command in order to launch the server with a spawner.

- ◆ Specify any additional object server parameters (Object Server Parameters) to use to launch SAS.

4. To continue configuring the server and add advanced options, see the appropriate topic for your type of server:

- ◆ Using SAS Management Console to Define Custom Parameters for Workspace or Stored Process Servers
- ◆ Using SAS Management Console to Define an OLAP Server

IOM Bridge

Using SAS Management Console to Modify Servers

SAS Management Console provides a graphical user interface that allows you to modify a definition for a server with an IOM Bridge connection. For details about using SAS Management Console, from the SAS Management Console menu bar, select **Help ▶ Help on SAS Management Console**. For more information about the fields in a particular window, click **Help** in that window.

Modifying an Existing Server's Properties

To modify a server definition (with an IOM Bridge connection) using SAS Management Console:

1. Start SAS Management Console and connect to a metadata repository.
2. In the SAS Management Console navigation tree, expand the Server Manager to find the server object that you want to modify.
3. Select the server object, and then select **File ▶ Properties** from the menu bar.
4. Select the appropriate tabs, and enter the necessary changes. For a description and location of the fields, refer to the [Fields for Server Definitions](#). When you are finished, click **OK** to return to the SAS Management Console main window.

Adding a Bridge Connection (SAS OLAP Servers only)

For SAS OLAP Servers, if you have defined a server with a COM connection, you might want to add an IOM Bridge connection to the server definition.

Note: You can only add one IOM Bridge connection to a server definition.

To add an IOM Bridge connection using SAS Management Console:

1. Start SAS Management Console and connect to a metadata repository.
2. In the SAS Management Console navigation tree, expand the Server Manager to find the server object that you want to modify.
3. Select the server object, and then select **Actions ▶ Add Connection** from the menu bar. The New Connection Wizard appears.
4. Enter a **Name** and optionally, a **Description** for the connection. Click **Next**. The Connection Options window appears.
5. Fill in the following fields:
 - a. For the **Authentication Domain**, enter the name of a domain ([Domain](#)). Note that if you define a spawner for this server, you must use an identical domain name in the spawner definition. Click **New** to add a new Authentication Domain:
 - ◇ Enter a [Domain](#).
 - ◇ Enter a description.
 - b. Enter the host name ([Host Name](#)) for the machine on which the server is to run.
 - c. Enter a unique port number ([Port Number](#)). The port number is required if the server will have Java clients.

If you want to enter service or encryption parameters, click **Advanced Options**.

- a.
On the Encryption tab, specify server encryption algorithms ([Server Encryption Algorithms](#)). Also make a selection to indicate what to encrypt ([Required Encryption Level](#)).
- b.
On the Service tab, enter the service ([Service](#)).

When you are finished entering information in the Advanced Options window, click **OK**. Click **Next**.

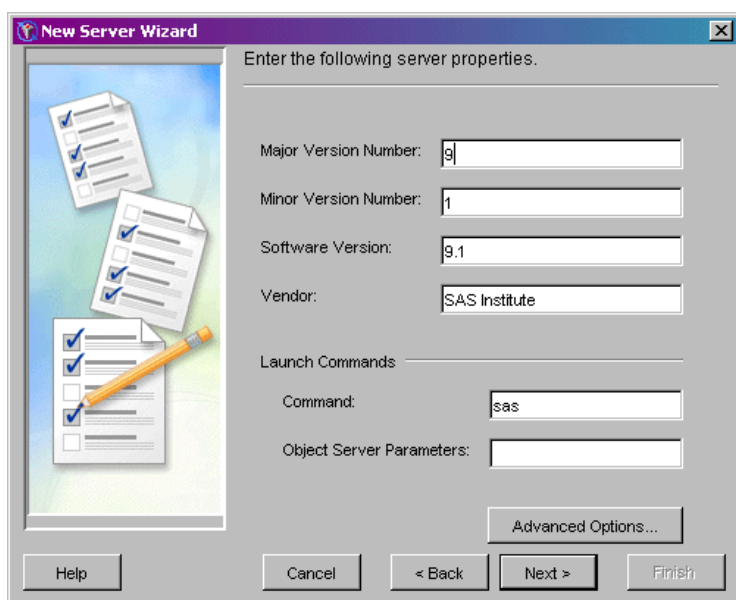
6.
Click **Finish** to define the connection and return to the SAS Management Console main window.

For a description and location of the fields, refer to the [Fields for Server Definitions](#).

IOM Bridge

Using SAS Management Console to Define Custom Parameters for Workspace or Stored Process Servers (IOM Bridge)

In order to define custom workspace or stored process server parameters, you must already have begun to add a server according to the instructions in [Using SAS Management Console to Define Servers](#). The New Server Wizard's Server Options window will be displayed as follows:



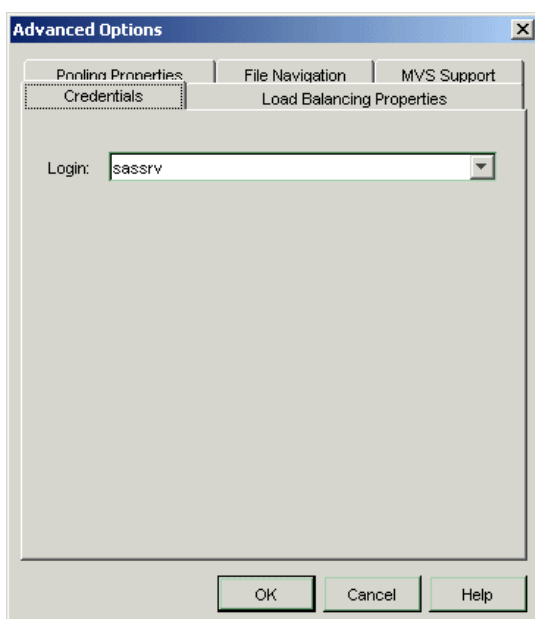
To continue defining a server with an IOM Bridge connection:

1. To enter multi-user login credentials, pooling, or load balancing, from the Server Options window, click **Advanced Options**.

Note: Multi-user login credentials and load balancing are required for stored process servers.

Choose the desired tab in the Advanced Options window in order to enter the necessary fields. Click the **Help** button on any tab to display entry instructions. Brief entry instructions are provided in these instructions.

- ◆ For SAS Stored Process Servers, select the Credentials tab and select a multi-user login (Login) from the drop-down list.



IMPORTANT NOTE: In order for the SAS Stored Process Server to run as a multi-user server, the spawner must have credentials to use when launching the server as a multi-user process. You must specify a multi-user login to use when launching a multi-user stored process server. In addition, on Windows 2000 and Windows NT, for the user who is the owner of the multi-user login, define the "act as part of the operating system" user right.

If you do not specify a multi-user login for the stored process server, the stored process server will not run and a message similar to the following will be displayed:

```
This server (OMSOBJ:LOGICALSERVER/A5SRQ5Z5.AT00008E)
cannot be spawned without credentials which specify
the server process username.
```

- ◆ For SAS Workspace Servers, to enter pooling parameters, select the Pooling Properties tab.

Specify the maximum number of clients in the pool (Maximum Clients) and the recycle activation limit (Recycle Activation Limit). If you want to shut down inactive servers, select the **Inactivity Timeout** check box and specify the inactivity timeout (Inactivity Timeout)

- ◆ For SAS Workspace Servers or SAS Stored Process Servers, to enter load-balancing parameters, select the Load Balancing Properties tab.

- ◇ For SAS Workspace Servers and SAS Stored Process Servers, specify the maximum cost for the server (Maximum Cost), the startup cost of the server (Startup Cost), and the availability timeout (Availability Timeout).

- ◇ For SAS Stored Process Servers, specify the following:

- If using the response time algorithm, the maximum number of clients for this server (Maximum Clients).
- The number of servers to start with the spawner (Start Size)
- The recycle activation limit (Recycle Activation Limit). If you wish to shutdown inactive servers, select the **Inactivity Timeout** checkbox and specify the inactivity timeout (Inactivity Timeout)

When you are finished entering information in the Advanced Options window, click **OK**. Click **Next**.

2. Select the **Bridge** connection. Click **Next**. The Connection Options window appears.
3. Fill in the following fields:

SAS® 9.1.2 Integration Technologies: Server Administrator's Guide

- a. For the **Authentication Domain**, enter the name of a domain (Domain). Note that if you define a spawner for this server, you must use an identical domain name in the spawner definition. Click **New** to add a new Authentication Domain:

◇ Enter a Domain.

◇ Enter a description.

- b. Enter the host name (Host Name) for the machine on which the server is to run.
- c. Enter a unique port number (Port Number). (The default is 8591 for SAS Workspace Servers and 8601 for SAS Stored Process Servers.) The port number is required if the server will have Java clients.

4. If you want to enter service or encryption parameters, click **Advanced Options**.

- a. On the Encryption tab, specify server encryption algorithms (Server Encryption Algorithms). Also make a selection to indicate what to encrypt (Required Encryption Level).
- b. On the Service tab, enter the service (Service).

When you are finished entering information in the Advanced Options window, click **OK**. Click **Next**.

5. Click **Finish** to create the server and return to the SAS Management Console main window.
6. Define the spawner. For instructions, see Using SAS Management Console to Define or Modify a Spawner.

IOM Bridge

Using SAS Management Console to Define an OLAP Server (IOM Bridge)

An OLAP server is a high-capacity, multi-user data manipulation engine specifically designed to support and operate on multi-dimensional data structures.

Use SAS Management Console to create the OLAP server definition. For details about SAS Management Console, from the SAS Management Console menu bar, select **Help ▶ Help on SAS Management Console**. For help on the fields in a particular window, click **Help** in that window.

The following documents provide additional information and help for the SAS OLAP Server:

- The [🌐 SAS OLAP Server Administrator's Guide](#)
- SAS OLAP Server Help
- SAS OLAP Administrator Online Help

IOM Bridge

Using SAS Management Console to Define or Modify a Spawner (IOM Bridge)

The Server Manager plug-in to SAS Management Console provides a graphical user interface that allows you to create or modify a definition for a server that uses an IOM Bridge connection. For details about using SAS Management Console, from the SAS Management Console menu bar, select **Help ▶ Help on SAS Management Console**. For more information about the fields in the New Server Wizard, click **Help** from within the wizard.

Before you begin defining spawners, you must have the following:

- a metadata profile for connecting to a metadata repository. For details about setting up this profile, see the [SAS Metadata Server: Setup Guide](#).
- the appropriate login, user, and group definitions for your server and spawner configuration. For details, see [Security Metadata](#) and the [Security](#) chapter.

For further information about launching the object spawner, see [Invoking \(Starting\) the Spawner](#).

To use SAS Management Console to define a spawner that starts a server with an IOM Bridge connection:

1. Start SAS Management Console and connect to a metadata repository.
2. From the SAS Management Console navigation tree, select the Server Manager, and then select **Actions ▶ New Server** from the menu bar. The New Server Wizard displays. A list of SAS Application Server resource templates is displayed.
3. Select the **Object Spawner** (located under Spawners). Click **Next**.
4. Enter the [Name](#) and [Description](#). Click **Next**.
5. Verify that the [Minor Version Number](#), [Major Version Number](#), and the [Software Version](#) are correct.

Specify the key length ([Encryption Key Length](#)).

In the **Associated Machine** drop-down list, select the [name of the machine](#) on which this spawner will run and listen for connection requests for the server.

When you are finished entering server properties, click **Next**. The Spawner Initialization window displays.

6. Select an operator login ([Login](#)). If you do not specify a login, the operator password defaults to sasobjspawn.

Select the check box to indicate whether you want to use verbose logging ([Verbose](#)), then specify the log file name and path ([LogFile](#)). If these options are specified on the object spawner command line, the object spawner command line values override these field values. Click **Next**.

7. Select the appropriate servers from the list of [Servers](#) that the spawner is permitted to start. To select the servers, highlight all of the servers that you want to associate with that spawner.

IMPORTANT NOTE: For SAS Stored Process Servers, the spawner must have credentials to use when launching the server as a multi-user process. Therefore, on the SAS Stored Process Server definition, you must specify a multi-user login to use when launching a multi-user stored process server. If you do not specify a multi-user login for the stored process server, the server runs as a single-user server; each

connection request for the server spawns a new stored process server which might not be properly shut down.

Click **Next**.

8. Select the **Operator Connection**. Click **Next**.

9. Fill in the following fields:

- a. For the **Authentication Domain**, enter a domain (Domain). The spawner must use the same domain as the server with which it connects. Click **New** to add a new Authentication Domain:

◇ Enter a Domain.

◇ Enter a description.

- b. The host name (Host Name) field contains the machine name on which the spawner is to run.

- c. Specify a unique port number (Port). The port number is required if the server will have Java clients.

When you are finished entering information in the fields, click **Next**.

10. Click **Back** to go back and change properties. Click **Finish** to define the spawner.

Note: If you are setting up load balancing, you must define a load–balancing connection for the spawner.

Adding a Load Balancing or UUID Connection

To add a connection to the spawner using SAS Management Console:

1. Start SAS Management Console and connect to a metadata repository.
2. In the SAS Management Console navigation tree, expand the Server Manager and locate the spawner object that you want to modify.
3. Select the spawner object that you want to modify, and then select **Actions ➤ Add Connection** from the menu bar. The New Connection Wizard displays.
4. Select the type of connection to add: **UUID** or **Load Balancing**. Click **Next**.
5. Enter a name and description for the connection. Click **Next**.
6. Fill in the following fields:

- a. For the **Authentication Domain**, enter a domain (Domain). The spawner must use the same domain as the server with which it connects. Click **New** to add a new Authentication Domain:

◇ Enter a Domain:

◇ Enter a description.

- b. The host name (Host Name) field contains the machine name on which the spawner is to run.

- c. Specify a unique port number (Port). The port number is required if the server will have Java clients.

- d. For UUID connections only, if you want to enter a service, click **Advanced Options**. Enter the service (Service).

When you are finished entering information in the fields, click **Next**. The parameters for the new spawner will be displayed.

7. Click **Finish** to define the connection and return to the SAS Management Console main window.

Modifying an Existing Spawner

To modify a spawner definition using SAS Management Console:

1. Start SAS Management Console and connect to a metadata repository.

2. In the SAS Management Console navigation tree, expand the Server Manager and locate the spawner object that you wish to modify.
3. Select the spawner object, and then select **File ► Properties** from the menu bar.
4. Select the appropriate tabs, and enter the necessary changes as follows:
 - ◆ Modify values on the Options tab to re-configure version information, encryption key length, MAC encryption, handshake timeout or the machine that on which the spawner runs.
 - ◆ Modify values on the Initialization tab to re-configure operator credentials and logging information.
 - ◆ Modify values on the Servers tab to re-configure which servers are associated with the spawner.

For a description of the fields, refer to the [Fields for Spawner Attributes Definitions](#).

IOM Bridge

Using SAS Management Console to Modify SAS Workspace or Stored Process Servers for Pooling or Load Balancing

SAS Management Console provides a graphical user interface that allows you to modify a definition for a server that uses an IOM Bridge connection. For details about using SAS Management Console, from the SAS Management Console menu bar, select **Help ➤ Help on SAS Management Console**. For more information about the fields in a particular window, click **Help** in that window.

Adding Pooling or Load–Balancing Properties to an Existing Server's Properties (Workspace and Stored Process Servers only)

For workspace or stored process servers, to add pooling or load balancing parameters to a server definition using SAS Management Console:

1. Start SAS Management Console and connect to a metadata repository.
2. In the SAS Management Console navigation tree, expand the Server Manager to find the server object that you want to modify.
3. Select the server object, and then select **File ➤ Properties** from the menu bar.
4. Select the Options tab, and then click **Advanced Options**.

To enter pooling or load–balancing parameters:

- ◆ For SAS Workspace Servers, to enter pooling parameters, select the Pooling Properties tab.

Specify the recycle activation limit (RecycleActivationLimit) and the maximum number of clients in the pool (Maximum Clients). If you want to shut down inactive servers, select the **Inactivity Timeout** check box and specify the inactivity timeout (Inactivity Timeout.)

- ◆ For SAS Workspace Servers or SAS Stored Process Servers, to enter load–balancing parameters, select the Load Balancing Properties tab.

◇ For SAS Workspace Servers and SAS Stored Process Servers, specify the availability timeout (Availability Timeout), maximum cost for the server (Maximum Cost), and the startup cost of the server (Startup Cost).

◇ For SAS Stored Process Servers, specify the following:

- If using the response time algorithm, the maximum number of clients for this server (Maximum Clients).
- The number of servers to start with the spawner (Start Size)
- The recycle activation limit (Recycle Activation Limit). If you want to shut down inactive servers, select the **Inactivity Timeout** check box and specify the inactivity timeout (Inactivity Timeout).

5. To save your changes and return to the SAS Management Console main window, click **OK**.

For a description and location of the fields, refer to the Fields for Server Definitions.

Adding a MultiBridge Connection (SAS Stored Process Servers only)

For stored process servers, you *must* add MultiBridge connections to enable the spawner to start processes. The number of MultiBridge connections is the maximum number of concurrent processes that the spawner will allow the stored process server to run.

To add a MultiBridge connection using SAS Management Console:

1. Start SAS Management Console and connect to a metadata repository.
2. In the SAS Management Console navigation tree, expand the Server Manager to find the server object that you want to modify.
3. Select the server object, and then select **Actions ➤ Add Connection** from the menu bar. The New Connection wizard appears. Click **Next**.
4. Enter a **Name** and optionally, a **Description** for the connection. Click **Next**. The Connection Options window appears.
5. Fill in the following fields:
 - a. For the **Authentication Domain**, enter the name of a domain (Domain). Note that you must use the same domain name for all servers and all connections in the load-balancing logical server.
 - b. Enter the host name (Host Name) for the machine on which the server is to run. This host name should be the same host name that was entered on the bridge connection definitions.
 - c. Enter a unique port number (Port). The default MultiBridge port is 8611, with subsequent default ports at 8621, 8631, 8641...8691.
6. Click **Next** and then **Finish** to add the MultiBridge connection.

For a description and location of the fields, refer to the [Fields for Server Definitions](#).

IOM Bridge

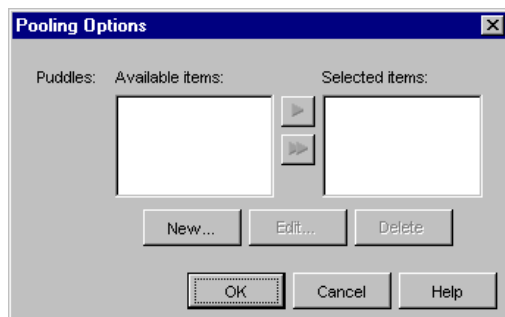
Using SAS Management Console to Define a Pooled Logical Server (IOM Bridge)

SAS Management Console provides a graphical user interface that allows you to convert a logical server to a pooled logical server. For details about using SAS Management Console, from the SAS Management Console menu bar, select **Help ➤ Help on SAS Management Console**. For more information about the fields in a particular window, click **Help** in that window.

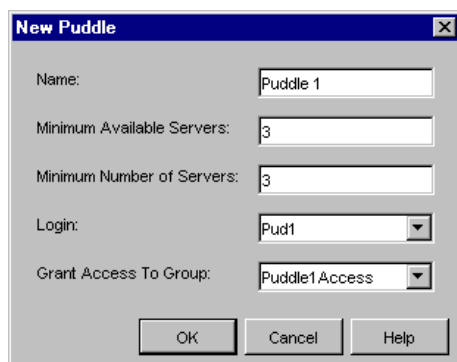
You can only convert workspace and stored process logical servers to pooled logical servers.

To convert a logical server to a pooled logical server using SAS Management Console:

1. Start SAS Management Console and connect to a metadata repository.
2. In the SAS Management Console navigation tree, select and expand the Server Manager to locate the logical server you wish to convert to a pooled logical server.
3. Select the logical server you want to convert to a pooled logical server, and then select **Actions ➤ Convert to ➤ Pooling** from the menu bar. The Information dialog box displays.
4. Click **Yes** to continue. The Pooling Options window displays.



5. Click **New** to create a new puddle. The New Puddle window displays.



For the new puddle, enter the

- ◆ name of the puddle (Name).
- ◆ minimum number of connections that need to be available (Minimum Available Servers).

- ◆ minimum number of connections to create when the pool is created (Minimum Number of Servers).
- ◆ user name, or login ID (Login).
- ◆ groups you want to grant access for puddle access (Group).

When you are finished entering the puddle parameters, click **OK**. The Pooling Options window appears and contains the new puddle.

6. From the Pooling Options window, do one of the following:

- ◆ Click **New** to create a new puddle.
- ◆ Select a puddle and click **Edit** to edit the puddle.
- ◆ Select a puddle and click **Delete** to delete the puddle.

Note: If you wish to delete a puddle, you must select the puddle and click **Delete**. Do not select a puddle and move it from the right side to the left side of the Pooling Options window. If you move a puddle to the left side of the Pooling Options window, when you click **OK** the puddle will no longer appear as available to the pool; however, the puddle will still be stored in a SAS Metadata Repository and will consume memory on the repository's machine.

When you are finished defining, editing, or deleting puddles, click **OK**.

For a description and location of the fields, refer to the [Fields for the Pooled Logical Server and Puddle Definitions](#).

IOM Bridge

Using SAS Management Console to Define a Load-Balancing Logical Server (IOM Bridge)

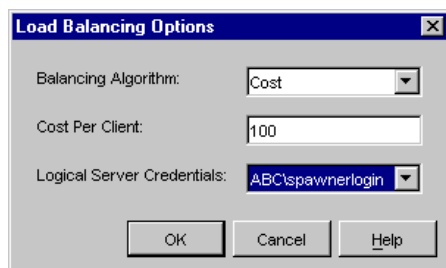
SAS Management Console provides a graphical user interface that allows you to convert a logical server to a load-balancing logical server.

For details about using SAS Management Console, from the SAS Management Console menu bar, select **Help ► Help on SAS Management Console**. For more information about the fields in a particular window, click **Help** in that window.

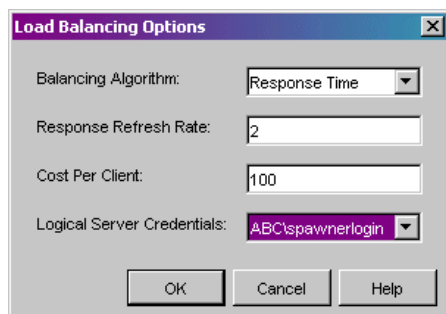
Note: You can only convert workspace logical servers and stored process logical servers to load-balancing logical servers.

To convert a logical server to a load-balancing logical server using SAS Management Console:

1. Start SAS Management Console and connect to a metadata repository.
2. In the SAS Management Console navigation tree, select and expand the Server Manager to locate the logical server you want to convert to a load-balancing logical server.
3. Select and right-click the logical server you want to convert to a load-balancing logical server, and then select **Actions ► Convert to ► Load Balancing** from the menu bar. The Information dialog box displays.
4. Click **Yes** to continue. The Load Balancing Options window displays.



Load Balancing Options for SAS Workspace Server



Load Balancing Options for SAS Stored Process Server

5. Enter the appropriate parameters for the following:
 - ◆ the type of algorithm to use for load balancing (Balancing Algorithm).
 - ◆ SAS Stored Process Servers and response time algorithm only, the response refresh time (Response Refresh Rate).

- ◆ the weighted cost per client (Cost Per Client).

Select the spawner-to-spawner login for load-balancing connections between spawners (Logical Server Credentials). You must define the appropriate login before converting your logical server to a load-balancing logical server.

When you are finished entering the load-balancing parameters, click **OK**.

For a description and location of the fields, refer to the Fields for the Load Balancing Logical Server Definition.

IOM Bridge

Configuring a UUID Generator

Currently, only SAS on Windows can generate unique UUIDs. The UUID Generator Daemon (UUIDGEND) generates unique UUIDs for SAS sessions that execute on hosts without native UUID generation support.

Installing UUIDGEND

If your SAS application executes on a platform other than Windows and your application requires unique UUIDs, install UUIDGEND and identify its location (see SAS UUIDGENDHOST and UUIDCOUNT options documentation) to your executing SAS application. If you install UUIDGEND on a host other than Windows, you need to contact SAS Technical Support to obtain a UUID node. The UUID node must be unique per UUIDGEND installation in order for UUIDGEND to guarantee truly unique UUIDs.

Configuring the Spawner for UUIDGEND

UUIDGEND is implemented in the spawner. You can execute a separate spawner to support UUIDGEND only, or you can update an existing spawner instance to support UUIDGEND along with its server definitions. To configure UUIDGEND, you must define a UUID connection on the spawner and specify port, service, and protocol fields for the UUID connection. All other spawner definition requirements must be met.

IOM Bridge

Configuring and Starting the Object Spawner on z/OS

On a z/OS server, the spawner starts a SAS server session in response to a request from a client. The client uses TCP/IP to communicate first with the spawner, and then with the object server. The object spawner runs as a started task; therefore, before the object spawner can handle client requests, you must start the spawner using a started task procedure.

If you used the project installation's z/OS configuration script to plan, install, and define your implementation, then you already have an initial z/OS spawner configuration. For details about the project installation, see [Getting Started With the SAS Configuration Wizard](#) and "Configuring SAS Servers on z/OS Systems" in the [SAS Intelligence Architecture: Planning and Administration Guide](#).

If you did *not* use the SAS Configuration Wizard, then the following setup tasks are required:

1. [Configure TCP/IP](#)
2. [Create the object spawner started task](#)
3. [Create a SAS startup command](#)

Note: This page is intended to serve as an outline of the process, rather than a step-by-step guide, for setting up a spawner on a z/OS platform.

Task 1: Configure TCP/IP

The overall configuration of TCP/IP is outside the scope of this discussion. Assuming that a functioning TCP/IP link is in place between the client and the z/OS server, the following additional step is required to support the object spawner:

- Verify that the SAS/C Transient Runtime Library (CTRANS), IBM TCPIP.DATA, and TCP/IP SERVICES configurations are available to both the object spawner and its object servers.

If you specify TCP/IP service names rather than ports in the spawner configuration, you must define the services in the TCP/IP services file. For example, the default spawner operator listen service name is `sasobjoper` and the default spawner server listen service name is `sasobjspawn`. To define these in the TCP/IP services file, add the following two lines:

```
sasobjoper      8582/tcp
sasobjspawn    8581/tcp
```

Task 2: Create the Object Spawner Started Task

The object spawner runs as a started task (STC). Its purpose is to listen for requests from clients and pass them to the startup command associated with the service/port in which there is activity. The startup command will start a server session. You must create a procedure in a system PROCLIB library (SYS1.PROCLIB, for example).

Create the Procedure

Because z/OS Job Control Language has a parameter line length restriction of 100 characters, you can use DDNames to identify filenames in object spawner parameters. When a file pathname is 8 characters or less, the file pathname is first checked to see if it matches a DDName. If so, the DDName is used. If DDNames are not used for the config file and log file, you need to specify a config file and log file in the UNIX file system.

If you need to specify more than 100 characters for command line parameters, put the additional parameters in a z/OS data set or UNIX file and reference it using the `=<//DDN:PARMS` parameter.

The following procedure explicitly specifies the pathname for the config file and uses a DDName to reference the log file in the command line parameters for the object spawner.

```
//OBJSPAWN PROC PROG=OBJSPAWN,
//  OPTIONS='-XMLCONFIGFILE /usr/lpp/SAS/objspawn.xml ',
//  OPT2='-SASVERBOSE -SASLOGFILE LOGFILE'
//OBJSPAWN EXEC PGM=&PROG,REGION=512M,
//      PARM='&OPTIONS &OPT2 =<//DDN:PARMS'
//STEPLIB DD DISP=SHR,DSN=SYS2.SAS.LIBRARY
//CTTRANS DD DISP=SHR,DSN=SYS2.SASC.TRANSLIB
//PARMS DD DISP=SHR,DSN=SYS2.OBJSPAWN.PARMS
//TKMVSJNL DD PATH='/tmp/objspawn/JNL.&LYYMMDD..&LHHMMSS..txt',
//      PATHMODE=(SIRUSR,SIWUSR,SIRGRP,SIROTH),
//      PATHOPTS=(OWRONLY,OCREAT,OTRUNC)
//LOGFILE DD PATH='/tmp/objspawn/LOG.&LYYMMDD..&LHHMMSS..txt',
//      PATHMODE=(SIRUSR,SIWUSR,SIRGRP,SIROTH),
//      PATHOPTS=(OWRONLY,OCREAT,OTRUNC)
```

Remember that the STC has access to the SAS/C Transient Runtime Library (CTTRANS).

The `-XMLCONFIGFILE` parameter identifies the SAS Metadata Server system configuration file that the spawner is to use.

The `-SASVERBOSE` and `-SASLOGFILE` options in the STC procedure provide useful information for diagnosing connection problems. It is a good idea to include these options until you are satisfied that everything is working correctly.

Define the Object Spawner System Security Configuration

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

If the following messages appear in the z/OS system log when a client attempts to connect, then a necessary library is not program controlled.

```
ICH420I PROGRAM program-name [FROM LIBRARY dsname] CAUSED THE
ENVIRONMENT TO BECOME UNCONTROLLED.
BPXP014I ENVIRONMENT MUST BE CONTROLLED FOR DAEMON (BPX.DAEMON) PROCESSING
```

Verify the Metadata Configuration File and SAS Management Console Definitions

If you have not already created a metadata configuration file, for information about creating the file, see [Metadata Configuration File](#).

You must also define your server and spawner using SAS Management Console. When you define the server, enter the following command in the **Command** field of the **Advanced Options ▶ Launch Commands** tab:

```
/usr/bin/startsas.sh --
```

For details about using SAS Management Console to create metadata, see [Creating the Metadata Using SAS Management Console](#).

Start the Object Spawner

After you have created the STC procedure, you can start the object spawner by issuing the following command:

```
START OBJSPAWN
```

For a list of all available spawner invocation options, see [Spawner Invocation Options](#). If there are no configuration errors, the object spawner will assume a listening state by entering a detected wait state (DW).

Task 3: Create a SAS Startup Command

Create the Startup Command

The startup command is meant to build a parameter string that is capable of launching SAS. The startup command in the spawner configuration must end with '--' to indicate the end of the user specified parameters. Here is a sample shell script (startsas.sh):

```
#!/bin/sh
#
# foundDashDash is a boolean.  When TRUE, we found the string
# "--" in our arguments.
#
foundDashDash=0

#
# Construct our arguments
#
args=''
for arg in "$@" ; do
    if [ "$arg" != "--" ]; then
        tmp="$arg "
    else
        tmp="SRVOPTS(' ');
        foundDashDash=1;
    fi
    args="$args$tmp"
done

#
# If we found a "--", we need to close the SRVOPTS option
```

```

#
if [[ $foundDashDash -ne 0 ]]; then
    args="$args '"
fi

#
# Construct the command line...
#
cmd="/bin/tso -t EX 'SYS2.TSO.CLIST(SPWNSAS)'"
cmd="$cmd 'nosasuser $args'"

#
# Set environment variables...
# Account data can be used to place SAS in the correct WLM
# service class. SYSPROC specifies the data set containing
# the SAS CLIST/REXX
#
export _BPX_ACCT_DATA=MYNAME1
export SYSPROC=SYS2.TSO.CLIST

#
# Start up SAS
#
exec $cmd

```

The sample invokes the `/bin/tso/` UNIX command to execute the CLIST `SYS2.TSO.CLIST(SPWNSAS)`. Replace the CLIST data set name `SYS2.TSO.CLIST` with the name appropriate to your site. The control (CNTL) data set that you created for your SAS installation contains an example CLIST for use in launching IOM server sessions.

Note: The SAS CLIST requires the following parameters:

- `NOSASUSER` to allow more than one concurrent SAS session per user. `NOSASUSER` suppresses allocation of a `SASUSER` data set.
- `SRVOPTS()` in order to pass in the objectserver options.

Specify Account Data

The IOM spawner on z/OS uses the Unix System Services spawn function to initiate a process to run an IOM server. This process runs in a USS initiator (BPXAS). By default, the process runs with the default Work Load Manager (WLM) service class that was assigned to OMVS work during installation. The default service class might have been defined with a goal of providing USS shell commands with good response times. This default service class assumes the requests are relatively short. Because work associated with IOM requests might require more time, it might be desirable to assign IOM servers to a different service class.

You can use MVS accounting data to assign the work to a specific Work Load Manager service class. To set the accounting data, use the `_BPX_ACCT_DATA` environment variable in the `startsas.sh` script that starts that SAS IOM server session. The server session then runs with the accounting data. For example:

```
export _BPX_ACCT_DATA=MYNAME1
```

To assign a Work Load Manager service class based on the accounting data, use the WLM AI classification rule. For example (in the WLM ISPF dialog):

Qualifier	Class
-----------	-------

SAS® 9.1.2 Integration Technologies: Server Administrator's Guide

Type	Name	Start	Service	Report
			DEFAULTS:	
1	AI	MYNAME1	1	OMVSSHRT
				OMVSLONG

For more information about using accounting information with USS processes, consult *Unix System Service Planning*. For information about defining WLM service classes with appropriate characteristics, and for information about specifying classification rules to use these classes, see *MVS Planning: Workload Management*.

Because you might define different IOM servers, in order to segregate different work loads, you can also specify that these servers run in different service classes. To specify different service classes, create a separate server definition for each class of work in the SAS Management Console configuration, and assign client requests to the listen port associated with each server.

IOM Bridge

Creating a Metadata Configuration File in SAS

The Metadata Server Connections window in SAS enables you to:

- Configure information for connecting to a SAS Metadata Server.
- Export the configuration to a metadata configuration file that you can use in the following situations:
 - when you are starting a spawner that connects to the SAS Metadata Server
 - when you are connecting to a SAS Metadata Server from the Windows Object Manager.

Preparing to Use METACON (z/OS Only)

Before you can use the METACON command on z/OS, you must complete these steps:

1. Allocate a file using the ALLOC z/OS host command. For example,

```
ALLOC F(METACFG) SPACE(10 10) TRACKS REUSE
```

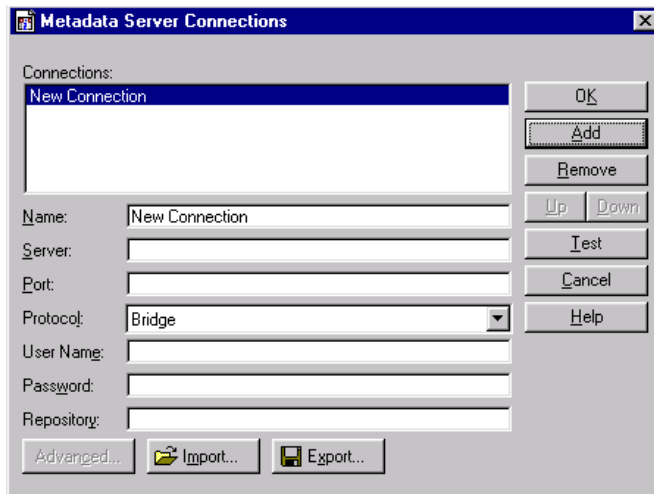
where *METACFG* is name of the file.

2. Use the –METAPROFILE option at SAS invocation to specify the file that you created with the ALLOC host command. For details about the –METAPROFILE system option, see [METAPROFILE= System Option](#) in the *SAS Language Reference: Dictionary*.

Using the METACON Command (All Hosts)

To create a metadata configuration file in SAS:

1. Start SAS and enter the METACON command. The Metadata Server Connections window appears.



2. Click **Add** to create a new connection and complete the following fields:

Name

specifies a name for the server connection.

Server

specifies the fully qualified name of the machine on which the server runs.

Port

specifies the port that the server connection uses.

Protocol

specifies whether the connection uses IOM Bridge protocol or COM protocol.

Note: If you are creating a configuration file for the object spawner, then you must specify Bridge.

User Name

specifies the user ID that is used to log on to the server. You might need to specify your authentication domain using the format *domain\user-ID*.

Password

specifies the password that is used to log on to the server.

Repository

specifies which metadata repository on the server to use.

3. To export the connection information as a metadata configuration file, click **Export**.

IOM Bridge

Using the SAS Integration Technologies Configuration Utility (ITConfig)

The SAS Integration Technologies configuration utility (ITConfig) enables you to generate metadata configuration files and test Integrated Object Model (IOM) connections between client machines and SAS. Using the ITConfig application, you can perform the following tasks:

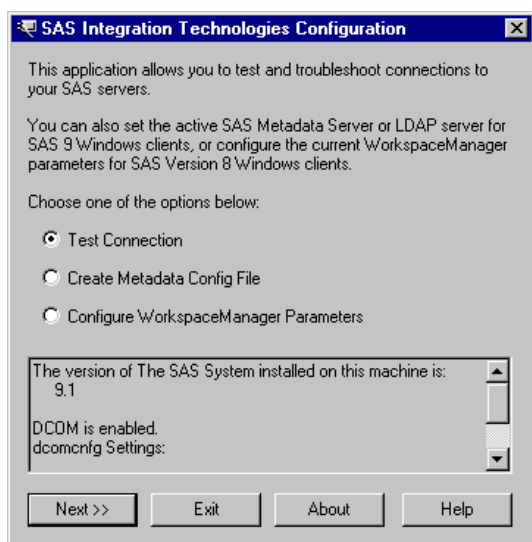
- create metadata configuration files that can be used to access an LDAP server or SAS Metadata Server
- test and diagnose IOM connections to SAS servers. The application can test COM, DCOM, and IOM Bridge connection types.
- set the registry parameters that are used by the workspace manager on an LDAP server.

Starting the Application

Select **Start ▶ Programs ▶ SAS ▶ SAS 9.1 Utilities ▶ Integration Technologies Configuration** to open ITConfig.

When the program starts, it checks the Windows program registry for unused SAS Integration Technologies entries. If any unused entries are found, the application gives you the option of removing the entries.

The SAS Integration Technologies Configuration window appears.



This window displays information about your current configuration, including the version of SAS installed, whether DCOM is installed and active, and DCOM configuration settings. Use this window to choose which task you want to perform:

- create metadata configuration files ([Create Metadata Config File](#))
- test the connection to a SAS Workspace Server or SAS Metadata Server ([Test Connection](#))
- view and change the LDAP parameters for the Workspace Manager (not used for the SAS Open Metadata Architecture).

IOM Bridge

Using ITConfig to Create Metadata Configuration Files

To access definitions on a metadata server, you must first connect to the metadata server. For IOM Bridge connections to the metadata server, the object manager, spawner, and SAS can use metadata configuration files that contain information about how to access the metadata server.

To create a metadata configuration file:

1. Select **Create Metadata Config File** in the main ITConfig window. The Create Metadata Config File window appears.
2. Select **SAS Metadata Server** and click **Next**. The Configure Metadata Server window appears.
3. Select **IOM Bridge** for the connection type. For the configuration type, select **Current user** to create a user-specific configuration, or **All users on this machine** to create a configuration that is common to all users. Click **Next**. The Configure SAS Metadata Server window appears.

Configure SAS Metadata Server for All Users

Server Information
The Machine Name is the DNS name of the computer on which your SAS Metadata Server is running.
Machine Name (machine.company.com)
[Text Box]
Choose either Port or Service and enter the appropriate value.
☒ Port [Text Box]
☐ Service [Text Box]
Server configuration will be stored here:
[Text Box: C:\Documents and Settings\All Users\Application Data\SAS\Me]

Login Information
Enter a valid username and password for your server.
Username: [Text Box] (domain\username) may be required
Password: [Text Box]
Authentication Domain (optional): [Text Box]
☒ Use this login information for all users.
☐ Use this login information for the current user only.
☐ Prompt each user for login information when they connect.
Please read the Help dialog if you will be configuring an object spawner with this metadata config file.

<< Back Next >> Cancel Help

4. If metadata configuration files already exist on your machine, the information from those files will be included in this window. You can edit the existing configuration parameters.

Enter the following information:

Machine Name

specifies the fully qualified name of the machine on which the SAS Metadata Server runs.

Port or Service

specifies the port or service to connect to on the server. If you are using a port to connect to your SAS Metadata Server, select **Port** and enter the TCP/IP port number. A typical port value is 8561.

If you are using a service name to connect to your SAS Metadata Server, select **Service** and enter the service name.

Username

specifies the user who will be accessing the SAS Metadata Server.

Password

specifies the password required for the specified user to log on to the SAS Metadata Server.

Authentication Domain

specifies the authentication domain associated with the credentials for the SAS Metadata Server.

5. If you specified **Current user** for the configuration type, select one of the following:

Use this login information each time you connect

writes the server and login information to a user-specific system configuration file.

Note: You must select this option if you plan to use your configuration file with the object spawner.

Prompt for login information each time you connect

writes the server information to a user-specific system configuration file.

If you specified **All users on this machine** for the configuration type, select one of the following:

Use this login information for all users

writes the server and login information to a common system configuration file.

Note: You must select this option if you plan to use your configuration file with the object spawner.

Use this login information for the current user only

writes the server information to a common system configuration file and writes the login information to a user-specific user configuration file.

Prompt each user for login information when they connect

writes only the server information to a common system configuration file.

6. Click **Next**. The application connects directly to the SAS Metadata Server, retrieves the list of available repositories, and displays the SAS Metadata Server Repository Selection window. Select the repository that will be used for the metadata configuration and click **Next**.

The ITConfig application writes the data to the metadata configuration files. The XML File Written dialog box appears.

7. To return to the main ITConfig window, click **OK**.

Names and Locations for Configuration Files

Metadata configuration files are always stored with a default filename and path. The path is dependent on the version of Windows that you are using.

Default paths for Windows NT:

Common system configuration file

\WINNT\Profiles\All Users\Application Data\SAS\
MetadataServer\oms_serverinfo.xml

User-specific system configuration file

```
\WINNT\Profiles\username\Application Data\SAS\
  MetadataServer\oms_serverinfo.xml
```

User configuration file

```
\WINNT\Profiles\username\Application Data\SAS\
  MetadataServer\oms_userinfo.xml
```

Default paths for Windows 2000, Windows XP, and Windows 2003 Server:

Common system configuration file

```
\Documents and Settings\All Users\Application Data\SAS\
  MetadataServer\oms_serverinfo.xml
```

User-specific system configuration file

```
\Documents and Settings\username\Application Data\SAS\
  MetadataServer\oms_serverinfo.xml
```

User configuration file

```
\Documents and Settings\username\Application Data\SAS\
  MetadataServer\oms_userinfo.xml
```

Note: The locations and filenames are displayed in the Configure SAS Metadata Server window and in the XML File Written dialog box.

Sample System Configuration File Format for an IOM Bridge Connection

Use a text editor to edit your metadata configuration files. The following XML code shows a sample system configuration file for an IOM Bridge connection to a SAS Metadata Server.

```
<?xml version="1.0" encoding="UTF-8" ?>
<Redirect>
  <LogicalServer Name="SAS Metadata Server"
    ClassIdentifier="2887E7D7-4780-11D4-879F-00C04F38F0DB">
    <UsingComponents>
      <ServerComponent Name="SAS Metadata Server"
        ClassIdentifier="2887E7D7-4780-11D4-879F-00C04F38F0DB" >
        <SourceConnections>
          <TCPIPConnection Name="SAS Metadata Server"
            Port="8561"
            HostName="server.us.alphaliteair.com"
            ApplicationProtocol="Bridge"
            CommunicationProtocol="TCP">
            <Properties>
              <Property Name="Repository"
                DefaultValue="intserv"
                PropertyName="Repository">
              </Property>
              <Property Name="Required Encryption Level"
                DefaultValue="none"
                PropertyName="Required Encryption Level">
              </Property>
            </Properties>
          </SourceConnections>
        </ServerComponent>
      </UsingComponents>
    </LogicalServer>
  </Redirect>
  <Domain>
    <AuthenticationDomain Name="domainName">
      <Logins>
        <Login Name="domainName\testuser"
          UserID="domainName\testuser"
          Password="{base64}cGFzc3dvcmQ=" />
        </Login>
      </Logins>
    </AuthenticationDomain>
  </Domain>
</Redirect>
```

```

        </AuthenticationDomain>
    </Domain>
</TCPIPConnection>
</SourceConnections>
</ServerComponent>
</UsingComponents>
</LogicalServer>
</Redirect>

```

Sample User Configuration File Format for an IOM Bridge Connection

Use a text editor to edit your metadata configuration files. The following XML code shows a sample user configuration file for an IOM bridge connection to a SAS Metadata Server.

```

<?xml version="1.0" encoding="UTF-8" ?>
<AuthenticationDomain Name="domainName">
    <Logins>
        <Login Name="Metadata Login"
            UserID="domainName\testuser"
            Password="{base64}cGFzc3dvcmQ=">
        </Login>
    </Logins>
</AuthenticationDomain>

```

IOM Bridge

Using ITConfig to Test Connections

The SAS Integration Technologies configuration utility (ITConfig) allows you to test IOM Bridge connections from your local machine to a SAS Workspace Server or SAS Metadata Server. You can retrieve the server definition from a metadata server or define the server manually.

The test program used by ITConfig is a small SAS program that verifies the following information about the server environment:

- events are returned
- the WORK data set is properly configured
- the location of the SASUSER directory
- the state of other SAS options.

Testing a Connection that is Defined on a Metadata Server

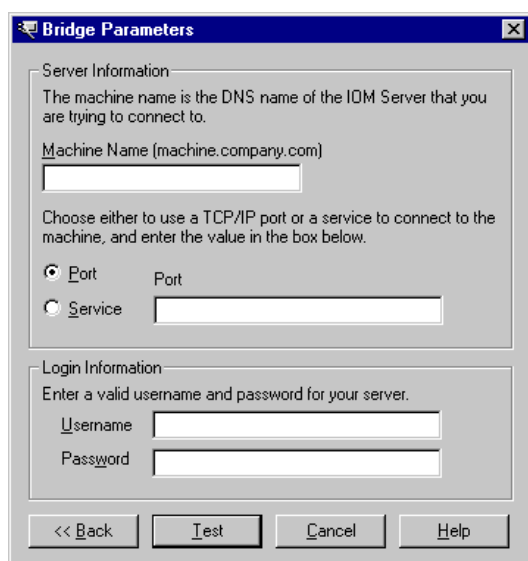
To test connections to an IOM server that is defined on a metadata server:

1. Select **Test Connection** from the main SAS Integration Technologies Configuration window and click **Next**. The Choose How to Test window appears.
2. Select **Retrieve logical server definitions from the currently configured metadata server**, then click **Next**. The Test window appears.
3. Select the **Logical Name** of the server connection that you want to test.
4. Enter a valid user name and password in the **Username** and **Password** fields.
5. Click **Test** to submit the test program through the connection. If the program establishes an IOM Bridge connection to the specified server, the Connection Successful window appears.
6. Click **Copy Text** to copy the test results to the clipboard. Click **OK** to return to the Bridge Parameters window.
7. Click **Test** to test the connection again, or click **Cancel** to return to the main SAS Integration Technologies Configuration window.

Testing a Manually Defined IOM Bridge Connection

To test an IOM Bridge connection:

1. Select **Test Connection** from the main SAS Integration Technologies Configuration window, then click **Next**. The Choose How to Test window appears.
2. Select **Enter a SAS server definition manually** and click **Next**. The Run Tests window appears.
3. Select the type of server to test and select **Bridge**, then click **Next**. The Bridge Parameters window appears.



The Bridge Parameters dialog box is divided into two sections: Server Information and Login Information. The Server Information section contains a text box for the Machine Name, a radio button for Port (selected), and a text box for the Port number. The Login Information section contains text boxes for Username and Password. At the bottom are buttons for Back, Test, Cancel, and Help.

Bridge Parameters

Server Information
 The machine name is the DNS name of the IOM Server that you are trying to connect to.
 Machine Name (machine.company.com)

 Choose either to use a TCP/IP port or a service to connect to the machine, and enter the value in the box below.
☒ Port Port
☐ Service

Login Information
 Enter a valid username and password for your server.
 Username
 Password

<< Back Test Cancel Help

4. Enter the fully qualified machine name in the **Machine Name** field. The following are examples of fully qualified names:
 - ◆ machine1.alphaliteair.com
 - ◆ server.us.alphaliteair.com
5. Select either **Port** or **Service** to specify the method used to connect to the server.
6. Enter either the port number or the service name in the **Port** or **Service Name** field. The title of the field changes depending on whether you selected Port or Service as the connection method.
7. Enter a valid user name and password in the **Username** and **Password** fields.
8. Click **Test** to submit the test program through the connection. If the program establishes an IOM Bridge connection to the specified server, the Connection Successful window appears.
9. Click **Copy Text** to copy the test results to the clipboard. Click **OK** to return to the Bridge Parameters window.
10. Click **Test** to test the connection again, or click **Cancel** to return to the main SAS Integration Technologies Configuration window.

IOM Bridge

Using SAS Management Console to Test Server Connections

SAS Management Console provides a graphical user interface that enables you to test connections to SAS Workspace Servers and SAS Stored Process Servers.

To test a connection:

1. Invoke the object spawner.
2. Start SAS Management Console and connect to a metadata repository.
3. In the SAS Management Console navigation tree, select and expand the Server Manager to locate the server definition you want to test. The connections that are associated with the server appear in the display area. Select the connection that you want to test in the display area, and select **Actions ► Test Connection** from the menu bar.
4. Provide user credentials. For stored process servers, the multi-user login is used automatically. For workspace servers, enter a valid **User Name** and **Password** in the Log on to (Server Name) window and click **OK**.

SAS Management Console attempts to connect to the server. If the connection is successful, a window appears with the message, "Test Connection Successful". If the connection was not successful, an error message appears.

IOM Bridge

Using Telnet to Administer the Spawner

The spawner can be controlled and monitored using a telnet client connected to the operator port or service.

Connecting to a Spawner

To connect to an executing spawner, telnet to the operator interface port or service that is specified in the spawner definition.

The following example, run on UNIX, assumes 6337 was specified as the port for the operator:

```
myHost> telnet serverhost 6337
Trying...
Connected to serverhost.
Escape character is '^]'.
```

After the telnet conversation is active, enter the operator password that is specified. If the operator password was not specified, use `sasobjspawn` as the password.

Note: You will not be prompted for the password. For example:

```
sasobjspawn
Operator conversation established
```

You can now interact with the executing spawner by issuing any of the [Available Commands](#).

Available Commands

The following is a list of commands that are available via the spawner's operator interface:

Command	Description
<code>btrace filename</code>	Begin trace. <code>filename</code> is a fully qualified path to the file in which to log spawner activity.
<code>bye</code>	Terminate the spawner execution. Note: You cannot shut down an object spawner while there are current or pending load-balancing tasks.
<code>cluster reset all</code> or <code><name or ID of load-balancing logical server (cluster)></code>	<ul style="list-style-type: none">• If <code>all</code> is specified, shuts down all multi-user servers associated with load-balancing logical servers (clusters) defined on the local machine. Note: This command only affects SAS Stored Process Servers that were launched from object spawner that you are currently administering. For example, to shutdown all servers in a cluster: <code>cluster reset all</code>

- If a load-balancing logical server (cluster) name or ID is specified, shuts down all multi-user servers on the local machine that are part of the named load-balancing logical server (cluster).

Note: If you use a character encoding other than Latin-1, you must specify the cluster using the object ID (for example, A5JJTGEQ.AX00005L).

For example:

- ◆ To shut down a cluster by specifying the name of a load-balancing logical server:

```
cluster reset "SASMain - Logical Stored Process Server"
```

To determine the name of the load-balancing logical server, in SAS Management Console, select the load-balancing logical server definition, and then select **File ▶ Properties** from the menu bar. Use the value in the **Name** field.

- ◆ To shut down a cluster by specifying the object ID of a load-balancing logical server:

```
cluster reset A5JJTGEQ.AX00005L
```

To determine the object ID of the load-balancing logical server definition, in SAS Management Console, select the load-balancing logical server definition, and then select **File ▶ Properties** from the menu bar. Use the value in the **ID** field.

To understand how to locate the logical server definition, see [Planning for Metadata Definitions](#) or the Server Manager online Help.

etrace	End trace.
help	List available operator commands.
list	List all known servers that are supported by this spawner.
quit	Exit operator conversation.

IOM Bridge

Spawner Error Messages

Here are error messages that might be reported by objspawn and explanations to correct their cause.

If you are still unable to correct the error, you might want the spawner to begin tracing its activity. See the [administrator command](#) section or use the `-slf` option to specify a log file when launching the spawner. For details, see [Invoking \(Starting\) the Spawner](#).

Note: If an error occurs when the `-slf` option is not in effect, the spawner sends error messages to the SAS Console Log. This is a host-specific output destination. For details about the SAS Console Log, see the SAS Companion for your operating environment.

[Service Name] is already installed as a service. Deinstall the service, then reissue the install request

Host: Windows

Explanation:

The spawner is already installed as a service.

Resolution:

Deinstall the spawner then reissue your install command.

A client that does not support redirection has connected to a server that requires redirection. The client connection will be closed.

Host: All

Explanation:

A down level IOM Bridge for Java client is attempting to connect to a server that has been defined within a load balancing cluster.

Resolution:

Upgrade the client's IOM Bridge for Java support.

A duplicate configuration option [duplicated option] was found.

Host: All

Explanation:

The displayed option was specified more than once.

Resolution:

Remove the redundant option and reissue your command.

A true socket handle cannot be obtained.

Host: All

Explanation:

The spawner was unable to retrieve the TCP/IP stack socket identifier from the runtime.

Resolution:

Contact SAS Technical Support.

A valid sasSpawner definition cannot be found.

Host: All

Explanation:

The spawner failed to find the named spawner definition. Or, if no name was given, a spawner definition that referenced the host in which the spawner is executing.

Resolution:

If a spawner name was specified at invocation, ensure the name is correct. Otherwise, correct the configuration source to define a valid spawner containing the correct host name.

Also known as:

Host: All

Explanation:

The host in which objspawn is executing is also known under the aliases listed.

Resolution:

N/A

An accepted client connection cannot be registered.

Host: All

Explanation:

The spawner was unable to place the socket associated with a connected client in a select.

Resolution:

Review the spawner log file to determine if the spawner is resource constrained. Ensure the client is still connected.

An attempt to communicate with the SAS Metadata Server failed. The error text associated with the failure is [error text describing failure].

Host: All

Explanation:

The spawner was unable to contact the SAS Metadata Server defined in the specified SAS Metadata Server configuration file.

Resolution:

Ensure that the SAS Metadata Server defined in the SAS Metadata Server configuration file is defined correctly and contains proper credentials. Also ensure that target SAS Metadata Server is running.

An error occurred while server [server name] was starting. Now attempting a different server.

Host: All

Explanation:

The load balancing implementation failed to connect to the server and will attempt to connect to a different server.

Resolution:

Ensure that the server port is not already in use.

An NLS pipeline ([encoding identifier] –to– [encoding identifier]) cannot be created.

Host: All

Explanation:

The spawner was unable to initialize an internal transcoding object.

Resolution:

Ensure the SAS installation is complete and correct.

An unknown option ([option name]) was specified.

Host: All

Explanation:

The spawner encountered an invocation option that is invalid.

Resolution:

Remove the invalid option and reissue the spawner command.

An unsupported UUID request version ([invalid version]) was received.

Host: All

Explanation:

A connection to the UUID listen port/service specified an invalid UUID protocol version.

Resolution:

Ensure that the IOM server clients are not connecting to the wrong port/service.

Cannot install objspawn with the NOSECURITY option.

Host: Windows

Explanation:

Due to the security exposure associated with the `nosecurity` option, the spawner will not install as a Windows service when `nosecurity` is specified.

Resolution:

Remove the `nosecurity` option and reissue the install command.

Communication cannot be established with the launched session.

Host: All

Explanation:

The spawner was unable to forward client information to the IOM server launched on behalf of the client.

Resolution:

Contact SAS Technical Support.

Configuration source ([source]) conflicts with the previously specified configuration source ([source]).

Host: All

Explanation:

More than one configuration source was specified.

Resolution:

Determine which configuration source is correct and remove the others from your spawner invocation.

Failed to launch the server (server name) on behalf of load balancing.

Host: All

Explanation:

The load balancing implementation requested that the spawner launch an IOM server. The spawner was unable to launch the named server.

Resolution:

Ensure that the server start command is correct. Also ensure that there is not a port/service conflict.

Failed to locate the server ([server name]) to launch on behalf of load balancing.

Host: All

Explanation:

The load balancing implementation requested that the spawner launch an IOM server. The spawner was unable to locate the named server definition.

Resolution:

Ensure that the load balancing cluster is defined correctly.

Failed to locate the server indicated in the kill request.

Host: All

Explanation:

The load balancing implementation requested that a server be stopped. The spawner was unable to locate the server in which to stop.

Resolution:

N/A

Load Balancing did not authorize server [server] to start and is disregarding the AddServer request.

Host: All

Explanation:

The spawner is using Load Balancing and started a server without Load Balancing instructing it to do so. This request is thrown out and the spawner should continue to function.

Resolution:

Review the configuration via SAS Management Console to ensure that all servers are set up correctly.

No configuration was specified.

Host: All

Explanation:

The spawner was invoked without a configuration source.

Resolution:

Reissue spawner command with a configuration source.

Objspawn cannot be deinstalled.

Host: Windows

Explanation:

The spawner was unable to deinstall as a Windows service.

Resolution:

Review the spawner log file to determine the cause of failure. Ensure the spawner is currently installed as a Windows service.

Objspawn cannot be installed.

Host: Windows

Explanation:

The spawner was unable to install as a Windows service.

Resolution:

Review the spawner log file to determine the cause of failure. Ensure the spawner is not currently installed as a Windows service.

Objspawn encountered [number of errors] error(s) during command-line processing.

Host: All

Explanation:

The spawner was unable to start.

Resolution:

Reissue the spawner invocation command with a valid log file destination. Review the contents of the generated log file to determine why the spawner failed to start.

Objspawn encountered errors during results processing.

Host: All

Explanation:

The spawner was unable to complete configuration processing.

Resolution:

Review the spawner log file to determine the configuration error details.

Objspawn encountered errors while attempting to start. To view the errors, define the DD name TKMVSJNL and restart objspawn with the sasVerbose option.

Host: z/OS

Explanation:

The spawner encountered errors and was unable to start.

Resolution:

Define the DD name TKMVSJNL and restart objspawn with the sasVerbose option to create a log file. Review the contents of the log file to determine why the spawner failed to start.

Objspawn encountered errors while attempting to start. View the application event log for the name of the log file containing the errors.

Host: Windows

Explanation:

The spawner encountered errors and was unable to start.

Resolution:

View the application event log to determine the name of the log file. Review the contents of the log file to determine why the spawner failed to start.

Objspawn failed to reinitiate multiuser server listen. Objspawn is removing server definition.

Host: All

Explanation:

The spawner was unable to restart a multi-user server listen when the previously launched multi-user server exited.

Resolution:

Ensure that there is not a port/service conflict.

Objspawn has completed initialization.

Host: All

Explanation:

The spawner is operational.

Resolution:

N/A

Objspawn has detected a bridge protocol over the operator conversation socket. Objspawn is closing the operator conversation with the peer (%s).

Host: All

Explanation:

An IOM Bridge client has connected to the operator listen port/service instead of a port/service belonging to a server definition.

Resolution:

Update the client to connect to the proper server definition port/service.

Objspawn is being terminated by the operating system.

Host: z/OS

Explanation:

The operator or operating system has requested that the spawner exit. The spawner will exit after this message is displayed.

Resolution:

N/A

Objspawn is executing on host [fully qualified host name] ([string IP address for fully qualified host name]).

Host: All

Explanation:

The host in which the spawner is executing returned the displayed fully qualified host name that resolved to the displayed IP address. These two strings plus the string "localhost", and any names/IP addresses listed after the alias message, are used by the spawner to locate the appropriate spawner and server definitions.

Resolution:

If the spawner fails to locate a spawner or server definition, ensure the spawner and/or server definitions specify one of the listed name or IP addresses.

Objspawn is exiting as a result of errors.

Host: All

Explanation:

The spawner was unable to start.

Resolution:

Reissue the spawner invocation command with a valid log file destination. Review the contents of the generated log file to determine why the spawner failed to start.

Objspawn lost connection with the launched session.

Host: All

Explanation:

The spawner was unable to complete startup of the launched IOM server.

Resolution:

If the message is identified as an error, contact SAS Technical Support.

Objspawn may not have been installed.

Host: Windows

Explanation:

The spawner was unable to deinstall as a Windows service. This might be due to the spawner not being installed as a Windows service.

Resolution:

Ensure that the spawner is installed as a Windows service.

Objspawn starting as service [service name].

Host: Windows

Explanation:

Indicates which service the spawner is starting as.

Objspawn service ([name of deinstalled spawner service]) was deinstalled successfully.

Host: Windows

Explanation:

The spawner is no longer installed as a Windows service.

Resolution:

N/A

Objspawn service ([name of installed spawner service]) was installed successfully.

Host: Windows

Explanation:

The spawner successfully installed as a Windows service. Subsequent boots of Windows will start the spawner automatically.

Resolution:

N/A

Objspawn version [major].[minor].[delta] is initializing.

Host: All

Explanation:

The version of the spawner being invoked.

Resolution:

N/A

Objspawn was unable to locate a server definition. Objspawn is exiting.

Host: All

Explanation:

The spawner was unable to find a server definition in the configuration source specified that was valid for this machine and the spawner definition's domain and logical name.

Resolution:

Ensure there is a valid server definition that meets the requirements stated. If you are using an LDIF configuration file and the configuration file contains a valid server definition, ensure that there are not two or more blank lines located before the server definition. In LDIF format, two contiguous blank lines signify the end of the definitions that will be used.

Objspawn was unable to open the configuration file ([file path]).

Host: All

Explanation:

The spawner was unable to open a configuration file at the specified location.

Resolution:

Ensure that the configuration file exists at the location specified. Ensure the configuration file is readable by the spawner.

Objspawn was unable to read data from the operator conversation socket. The returned error number is [errno], and the text associated with that error is ([errno description]).

Host: All

Explanation:

The spawner encountered a TCP/IP read error while attempting to converse with a connected operator.

Resolution:

Ensure the operator is still connected.

Objspawn was unable to send data over the operator conversation socket. The returned error number is [errno], and the text associated with that error is ([errno description]).

Host: All

Explanation:

The spawner encountered a TCP/IP write error while attempting to converse with the operator.

Resolution:

The operator might have terminated their connection.

Port [port number] will be ignored, and service [service name] will be used.

Host: All

Explanation:

The spawner encountered both port and service attributes in the current definition. The service definition takes precedence.

Resolution:

Remove the attribute that is redundant/incorrect.

The [attribute name/description] attribute is either missing or is mismatched.

Host: All

Explanation:

The spawner encountered an attribute that did not have a required value.

Resolution:

Correct the configuration.

The [attribute name] attribute requires an argument.

Host: All

Explanation:

An attribute present in the configuration requires a value.

Resolution:

Supply a value for the attribute and restart the spawner.

The [object class name] attribute [attribute name] is no longer supported and will be ignored.

Host: All

Explanation:

The spawner encountered an attribute within the specified configuration source that is no longer supported.

Resolution:

If the configuration source is not shared by earlier versions of the spawner, remove the named attribute from the configuration source.

The [option name] option requires an argument.

Host: All

Explanation:

The displayed option requires a value.

Resolution:

Reissue the spawner command specifying a value for the displayed option.

The [SAS Metadata Server method name] call of the SAS Metadata Server failed. The error ID associated with this failure is [hexadecimal error identifier].

Host: All

Explanation:

The SAS Metadata Server failed to process the spawner's request.

Resolution:

Ensure the SAS Metadata Server is still operating. Ensure the SAS Metadata Server defined in the SAS Metadata Server configuration file is the correct SAS Metadata Server in which to connect.

The [spawner utility name] service cannot be loaded.

Host: All

Explanation:

The spawner was unable to locate the specified support.

Resolution:

Ensure the SAS installation is complete/correct.

The [tracker name] resource tracker cannot be created.

Host: All

Explanation:

The spawner was unable to create an internal object repository.

Resolution:

Review the spawner log file to determine if the spawner is resource constrained.

The attribute [attribute name] will be ignored.

Host: All

Explanation:

The named attribute is not applicable to the spawner.

Resolution:

N/A

The client ([Client child process identifier]) specified by launched session could not be located.

Host: All

Explanation:

The spawner was unable to locate the connection information associated with the client definition in which an IOM server was launched.

Resolution:

Ensure that the command associated with the launched session is correct and that the IOM server is successfully launching.

The client definition cannot be created.

Host: All

Explanation:

The spawner was unable to allocate and initialize a descriptor for the connected client.

Resolution:

Review the spawner log file to determine if the spawner is resource constrained.

The configuration source [file name] is an objspawn log file. Objspawn is unable to process a log file as a configuration file.

Host: All

Explanation:

The value of the `-configFile` or `-xmlConfigFile` option specifies an objspawn log file.

Resolution:

Change the value of the configuration source option to specify a configuration file.

The connection with the UUID generator session was lost.

Host: All

Explanation:

The spawner lost contact with the UUID generator client.

Resolution:

Ensure the client did not terminate.

The duplicate [attribute name] attribute will be ignored.

Host: All

Explanation:

The named attribute was encountered more than once.

Resolution:

N/A

The entry ([object class name]) is no longer supported and will be ignored.

Host: All

Explanation:

The spawner encountered an object class within the specified configuration source that is no longer supported.

Resolution:

If the configuration source is not shared by earlier versions of the spawner, remove the named object class definition from the configuration source.

The entry ([object class name]) was defined incorrectly and will be ignored.

Host: All

Explanation:

The spawner encountered an object class within the specified configuration source that is not defined correctly.

Resolution:

Review the spawner log file to determine which values in the object class definition are invalid, then correct the object class definition.

The exit handler cannot be installed.

Host: z/OS

Explanation:

The spawner was unable to install an exit handler.

Resolution:

Contact SAS Technical Support.

The IOM run-time subsystem cannot be initialized.

Host: All

Explanation:

The spawner was unable to locate the IOM server runtime.

Resolution:

Ensure the SAS installation is complete/correct.

The IP address [string IP address] did not transcode.

Host: All

Explanation:

The load balancing implementation requested that the spawner redirect the connected client to the named IP address. The spawner was unable to transcode the IP address string to ASCII.

Resolution:

Ensure the IP address is valid in the load balancing cluster definition.

The launched session did not accept forwarded requirements. The reply is [reply error number].

Host: All

Explanation:

The launched IOM server could not process the client requirements presented.

Resolution:

Ensure that the spawner and the server being launched are compatible releases.

The load balancing instance [method name] call failed. The error text associated with the failure is [error string].

Host: All

Explanation:

The spawner was unable to communicate with its in process load balancing instance.

Resolution:

Contact SAS Technical Support.

The log file ([file path]) already exists. Please erase this file and restart.

Host: All

Explanation:

The spawner was unable to create a log file. A file, that is not a spawner log file, already exists at the named location.

Resolution:

Either delete the file at the named location or specify a different location for the spawner log file.

The log file ([file path]) cannot be created.

Host: All

Explanation:

The spawner was unable to create a log file at the given file path location.

Resolution:

Ensure the given file path is correct. Ensure that there is not a file at the specified location that is not a spawner log file.

The logged-in user does not have the appropriate user permissions to invoke [Windows service name].

Host: Windows

Explanation:

The spawner was not able to install/deinstall as a Windows service due to the launching user not having the appropriate Windows User Rights.

Resolution:

Ensure that the invoking user is an administrator on the Windows host and that the user holds the appropriate Windows User Rights.

The metadata for the SAS Metadata Server failed to process.

Host: All

Explanation:

The metadata received from the SAS Metadata Server is invalid for this spawner implementation.

Resolution:

Ensure that the spawner and SAS Metadata Server are compatible releases.

The multiuser login ([login identifier]) that was specified for the server ([server name]) cannot be found.

Host: All

Explanation:

The spawner was unable to locate the login definition associated with a multi-user server definition.

Resolution:

Correct the configuration source to properly define the missing login definition then reissue the spawner command.

The old client cannot be redirected as a result of IP address issues.

Host: All

Explanation:

The spawner cannot format the redirect IP address into a format suitable by a back level client.

Resolution:

Update the client IOM Bridge for COM or IOM Bridge for Java.

The operator communication buffer cannot be allocated.

Host: All

Explanation:

The spawner was unable to allocate a buffer in which to process operator conversations.

Resolution:

Review the spawner log file to determine if the spawner is resource constrained.

The operator conversation cannot be allocated.

Host: All

Explanation:

The spawner was unable to allocate a descriptor in which to process operator conversations.

Resolution:

Review the spawner log file to determine if the spawner is resource constrained.

The operator conversation was terminated by the peer.

Host: All

Explanation:

The administration session was disconnected by the administrator.

Resolution:

N/A

The operator listen definition cannot be allocated.

Host: All

Explanation:

The spawner was unable to allocate a descriptor in which to process the operator listen definition.

Resolution:

Review the spawner log file to determine if the spawner is resource constrained.

The operator listen socket cannot be created. The returned error number is [errno], and the text associated with that error is ([errno description]).

Host: All

Explanation:

The spawner was unable to create a TCP/IP socket for use as an operator listen socket.

Resolution:

Review the spawner log file to determine if the spawner is resource constrained.

The operator password specified by [string IP address] is invalid.

Host: All

Explanation:

The password received by a session originating from the displayed IP address was not correct.

Resolution:

Reissue operator session and specify the correct password.

The port or service for load balancing the TCP/IP definition is missing.

Host: All

Explanation:

The TCP/IP connection definition associated with the load balancing cluster did not contain a port or service definition.

Resolution:

Correct the TCP/IP connection definition.

The port or service for the UUID generator TCP/IP definition is missing.

Host: All

Explanation:

The TCP/IP connection definition associated with the UUID generation did not contain a port or service definition.

Resolution:

Correct the TCP/IP connection definition.

The process cannot be launched for client [client username].

Host: All

Explanation:

The spawner was unable to launch an IOM server on behalf of the named client.

Resolution:

Ensure the command associated with the server definition is correct. Review the spawner log file to determine the cause of failure.

The process definition cannot be tracked for the server [server name].

Host: All

Explanation:

The spawner was unable to insert a server definition object into its repository.

Resolution:

Review the spawner log file to determine if the spawner is resource constrained.

The repository information for the SAS Metadata Server failed to process.

Host: All

Explanation:

The repository metadata received from the SAS Metadata Server is invalid for this spawner implementation.

Resolution:

Ensure that the spawner and SAS Metadata Server are compatible releases.

The requested UUIDs cannot be generated.

Host: All

Explanation:

The spawner encountered an error while attempting to fulfill a UUID generator request.

Resolution:

Review the spawner log file to determine if the spawner is resource constrained.

The results extension of the SAS Metadata Server cannot be loaded.

Host: All

Explanation:

The spawner was unable to locate the SAS Metadata Server configuration support.

Resolution:

Ensure the SAS installation is complete/correct.

The SAS Metadata Server [SAS Metadata Server method name] call failed. The error text associated with the failure is [error string].

Host: All

Explanation:

The SAS Metadata Server failed to process the spawner's request.

Resolution:

Ensure the SAS Metadata Server is still operating. Ensure the SAS Metadata Server defined in the SAS Metadata Server configuration file is the correct SAS Metadata Server in which to connect.

The SAS Metadata Server configuration file failed to process.

Host: All

Explanation:

The SAS Metadata Server configuration file is invalid for this spawner implementation.

Resolution:

Review the spawner log file to determine the SAS Metadata Server configuration file error details.

The SAS Metadata Server repository [repository name] cannot be located.

Host: All

Explanation:

The response from the SAS Metadata Server did not contain the specified repository name.

Resolution:

Ensure the SAS Metadata Server configuration file identifies the correct repository name. Ensure that the SAS Metadata Server defined in the SAS Metadata Server configuration file hosts the given repository name.

The server [name of server] cannot be placed in a resource track.

Host: All

Explanation:

The spawner was unable to insert the internal server definition object in its repository.

Resolution:

Review the spawner log file to determine if the spawner is resource constrained.

The server [server name] listen cannot be registered.

Host: All

Explanation:

The spawner was unable to place the socket associated with a server listen in a select.

Resolution:

Review the spawner log file to determine if the spawner is resource constrained.

The server connection definition cannot be created.

Host: All

Explanation:

The spawner was unable to create an internal launched IOM server connection object.

Resolution:

Review the spawner log file to determine if the spawner is resource constrained.

The server name [server-name] is not unique. Therefore this server definition will not be included.

Host: All

Explanation:

The spawner was unable to process a server definition because another server definition has the same name.

Resolution:

Change the server name in the server definition.

The server definition cannot be allocated.

Host: All

Explanation:

The spawner was unable to allocate a server definition.

Resolution:

Review the spawner log file to determine if the spawner is resource constrained.

The server did not start in the specified amount of time.

Host: All

Explanation:

The spawner was unable to start the load-balanced server in the time specified by the Availability Timeout property.

Resolution:

Ensure that the SAS server can start properly. If appropriate, increase the value of the Availability Timeout property.

The server launch command cannot be allocated.

Host: All

Explanation:

The spawner was unable to allocate the server's launch command.

Resolution:

Review the spawner log file to determine if the spawner is resource constrained.

The server must define the available encryption algorithm(s) when an encryption level is set.

Host: All

Explanation:

The server definition specifies an encryption level, but does not specify which encryption algorithms are available.

Resolution:

Specify the available encryption algorithms.

The session socket for the UUID generator was not accepted.

Host: All

Explanation:

The spawner was unable to process a new UUID generator client.

Resolution:

Review the spawner log file to determine if the spawner is resource constrained.

The socket-access method handle cannot be acquired.

Host: All

Explanation:

The spawner was unable to locate the IOM protocol TCP/IP driver.

Resolution:

Ensure the SAS installation is complete/correct.

The specified [attribute name] value is invalid ([invalid attribute value]).

Host: All

Explanation:

The displayed value for the displayed attribute is not valid.

Resolution:

Correct the attribute value and reissue command.

The specified TCP/IP definition protocol is invalid.

Host: All

Explanation:

A TCP/IP connection definition specifies a protocol that is not supported by the spawner.

Resolution:

Correct the TCP/IP connection protocol attribute value.

The TCP/IP accept call failed to process the client connection.

Host: All

Explanation:

The spawner was unable to process a new client.

Resolution:

Review the spawner log file to determine if the spawner is resource constrained.

The TCP/IP accept call failed to process the operator connection. The returned error number is [errno], and the text associated with that error is ([errno description]).

Host: All

Explanation:

The spawner was unable to process a new operator.

Resolution:

Review the spawner log file to determine if the spawner is resource constrained.

The TCP/IP accept call failed to process the session conversation request.

Host: All

Explanation:

The spawner was unable to process a new IOM server connection.

Resolution:

Review the spawner log file to determine if the spawner is resource constrained.

The TCP/IP bind call for the operator listen port failed. The returned error number is [errno], and the text associated with that error is ([errno description]).

Host: All

Explanation:

The spawner was unable to establish the operator listen.

Resolution:

Ensure the port/service specified for use by the spawner is defined and not in use. If in use, ensure the spawner is not already executing.

The TCP/IP bind call for the server [server name] listen port failed. The text associated with that error is ([reason of failure]).

Host: All

Explanation:

The spawner was unable to establish the named server listen.

Resolution:

Ensure the port/service specified for use by the spawner is defined and not in use. If in use, ensure the spawner is not already executing.

The TCP/IP bind call for the session conversation port failed. The returned error number is [errno], and the text associated with that number is ([errno description]).

Host: All

Explanation:

The spawner was unable to bind to any port in order to establish a listen for use by launched IOM servers.

Resolution:

Contact SAS Technical Support.

The TCP/IP bind call for the UUID listen port failed. The returned error number is [errno], and the text associated with that error is ([errno description]).

Host: All

Explanation:

The spawner was unable to establish the UUID generator listen.

Resolution:

Ensure the port/service specified for use by the spawner is defined and not in use. If in use, ensure the spawner is not already executing.

The TCP/IP listen call for the operator listen port failed. The returned error number is [errno], and the text associated with that error is ([errno description]).

Host: All

Explanation:

The spawner was unable to establish the operator listen.

Resolution:

Ensure the port/service specified for use by the spawner is defined and not in use. If in use, ensure the spawner is not already executing.

The TCP/IP listen call for the server [server name] listen port failed. The text associated with that error is ([reason of failure]).

Host: All

Explanation:

The spawner was unable to establish the server listen.

Resolution:

Ensure the port/service specified for use by the spawner is defined and not in use. If in use, ensure the spawner is not already executing.

The TCP/IP listen call for the session conversation port failed. The returned error number is [errno], and the text associated with that error is ([errno description]).

Host: All

Explanation:

The spawner was unable to establish the launched IOM server listen.

Resolution:

Contact SAS Technical Support.

The TCP/IP listen call for the UUID listen port failed. The returned error number is [errno], and the text associated with that error is ([errno description]).

Host: All

Explanation:

The spawner was unable to establish the UUID generator listen.

Resolution:

Contact SAS Technical Support.

The URI support extension cannot be loaded.

Host: All

Explanation:

The spawner was unable to locate the URI parsing support.

Resolution:

Ensure the SAS installation is complete/correct.

The UUID listen definition cannot be created.

Host: All

Explanation:

The spawner was unable to create an internal UUID listen object.

Resolution:

Review the spawner log file to determine if the spawner is resource constrained.

The UUID service name ([service name]) cannot be resolved.

Host: All

Explanation:

The host TCP/IP stack was unable to resolve the displayed TCP/IP service name.

Resolution:

Ensure the given service name is correct and defined to the spawner host installation.

The wait event for the objspawn cannot be created.

Host: All

Explanation:

The spawner was unable to create an internal synchronization object.

Resolution:

Review the spawner log file to determine if the spawner is resource constrained.

The Windows [Windows routine name] call failed ([reason for failure]).

Host: All

Explanation:

The spawner encountered an Windows SDK error while invoking the given method.

Resolution:

Contact the system administrator to determine the meaning of the error text.

The Windows [Windows routine name] call failed. GetLastError() = [GetLastError() return value].

Host: All

Explanation:

The spawner encountered an Windows SDK error while invoking the given method.

Resolution:

Contact the system administrator to determine the meaning of the GetLastError() return code.

Unable to bind to the SAS Metadata Server because the [name of missing attribute] attribute is missing.

Host: All

Explanation:

The SAS Metadata Server configuration file did not specify the named attribute.

Resolution:

Update the SAS Metadata Server configuration file to include the missing attribute.

Unable to create the session conversation definition.

Host: All

Explanation:

The spawner was unable to create an internal launched IOM server conversation object.

Resolution:

Review the spawner log file to determine if the spawner is resource constrained.

Unable to obtain the session conversation port. The returned error number is [errno], and the text associated with that error is ([errno description]).

Host: All

Explanation:

The spawner was unable to retrieve the port associated with the session conversation listen.

Resolution:

Contact the system administrator to determine if there are issues with the TCP/IP implementation.

Unable to read the server ([server name]) client update information.

Host: All

Explanation:

The spawner encountered a TCP/IP read error while attempting to converse with a launched IOM server.

Resolution:

The server might have exited.

Unable to resolve "localhost".

Host: All

Explanation:

The spawner could not resolve the local IP address.

Resolution:

Ensure that your TCP/IP configuration settings are correct.

Unable to redirect the client request.

Host: All

Explanation:

The spawner failed to redirect the connection request to another server in the cluster.

Resolution:

Review the spawner log for more information.

You can only specify one of the following choices: install or deinstall.

Host: Windows

Explanation:

Both the `install` and `deinstall` commands were specified.

Resolution:

Remove the option that should not be specified.

IOM Bridge

Fields for the Server Definition

The server definition contains startup and connection information for an instance of a SAS server. The server is defined using the fields listed in the following table. For each field, the table shows the following information:

- the name that identifies the field in SAS Management Console. Under each field name, the table shows the corresponding properties tab and field name in the SAS Management Console application.
- "Required" or "Optional" to indicate whether the field is required.
- the type of server configuration (COM/DCOM or IOM Bridge) for which the field is used.
- a definition of the field.

For step-by-step instructions about defining the metadata for a server connection, refer to [Using SAS Management Console to Define Servers](#).

Fields for the Server Definition			
Field Name	Required Optional	Server Type	Definition
Availability Timeout <i>In SAS Management Console:</i> Options ➔ Advanced Options ➔ Load Balancing Properties: Availability Timeout	Optional	IOM Bridge	For load-balancing servers, the number of milliseconds to wait for a load-balancing server to become available. This parameter is used in the following situations: <ul style="list-style-type: none"> • when all servers have allocated the maximum number of clients per server. • when load balancing is waiting for a server to start and become available for its first client.
Command <i>In SAS Management Console:</i> Options ➔ Launch Commands: Command	Required	IOM Bridge	The command used to launch SAS as an object server. If the SAS executable is not already in your path, then specify the path to <code>sas.exe</code> . You can also specify additional options on the command line. For details, see Server Startup Command . This field is used only for spawned servers.
Description <i>In SAS Management Console:</i> General ➔ Description	Optional	COM/DCOM, IOM Bridge	Text to summarize why this definition exists.
Authentication Domain <i>In SAS Management Console:</i> <Connection> ➔	Required	IOM Bridge	The domain that is associated with a set of computing resources that use the same authentication process. In IOM Bridge servers configurations, the spawner definition must have the same authentication domain name as the server definition. The spawner uses the authentication domain name, along with the machine

Options ➤ Authentication Domain			name, to determine which servers it services.
Host Name <i>In SAS Management Console:</i> <Connection> ➤ Options ➤ Host Name	Required	COM/DCOM, IOM Bridge	<p>The <u>DNS name</u> or IP address for the machine on which this server definition can execute. The machine name must be the official network name (for example, <code>machine.corp.com</code>). The string <code>localhost</code> can be used to signify the host on which the associated spawner is executing.</p> <p>Note: If you use <code>localhost</code> in the configuration, it could cause clients to connect to their local machine instead of the machine that an administrator designates as <code>localhost</code>.</p>
Inactivity Timeout <i>In SAS Management Console:</i> Options ➤ Advanced Options ➤ Load Balancing Properties ➤ Inactivity Timeout <i>and</i> Options ➤ Advanced Options ➤ Pooling Properties ➤ Inactivity Timeout	Optional	COM/DCOM, IOM Bridge	<p>If you are using connection pooling (SAS Workspace Server only) or load balancing (SAS Stored Process Server only), specifies whether an idle server should always remain running, and if not, how long it should run before being shut down. If the check box is not selected, then idle servers remain running. If the check box is selected, then the servers run idle for the number of minutes specified in the field before being shut down. If the check box is selected and 0 is specified as the inactivity timeout, then the server behavior is as follows:</p> <ul style="list-style-type: none"> • for load balancing (IOM Bridge only), the server will shut down when the last client disconnects from the server. • for pooling, a connection returned to a pool by a user is disconnected immediately unless another user is waiting for a connection from the pool. <p>The maximum value is 1440.</p>
Login <i>In SAS Management Console:</i> Options ➤ Advanced Options ➤ Credentials ➤ Login	Optional	IOM Bridge	<p>For SAS Stored Process Servers, the login that provides the spawner with credentials to use when starting a multi-user SAS session.</p> <p>Note: If the server runs on Windows 2000 and Windows NT, for the user who is the owner of the multi-user login, define the "act as part of the operating system" user right.</p>
Major Version Number <i>In SAS Management</i>	Required	COM/DCOM, IOM Bridge	Specifies the major version number of the component.

<i>Console:</i> Options ➤ Major Version Number			
Minor Version Number <i>In SAS Management Console:</i> Options ➤ Minor Version Number	Required	COM/DCOM, IOM Bridge	Specifies the minor version number of the component.
Maximum Clients <i>In SAS Management Console:</i> Options ➤ Advanced Options ➤ Load Balancing Properties ➤ Maximum Clients <i>and</i> Options ➤ Advanced Options ➤ Pooling Properties ➤ Maximum Clients	Optional	COM/DCOM, IOM Bridge	<ul style="list-style-type: none"> • For Pooling (SAS Workspace Server), specifies the maximum number of simultaneous connections from the pool. • For Load Balancing (SAS Stored Process Servers and Response Time algorithm only), specifies the maximum number of simultaneous clients connected to this server.
Maximum Cost <i>In SAS Management Console:</i> Options ➤ Advanced Options ➤ Load Balancing Properties ➤ Maximum Cost	Optional	IOM Bridge	For load–balancing servers using the cost algorithm, the maximum cost allowed on each SAS server before requests to the server are denied.
Name <i>In SAS Management Console:</i> General ➤ Name	Required	COM/DCOM, IOM Bridge	The unique name for this server.
Object Server Parameters <i>In SAS Management</i>	Optional	IOM Bridge	For spawned servers, these object server parameters are added to others that are generated by the spawner and used to launch SAS. For servers that are not spawned, the values that you specify here can be used to

<i>Console:</i> Options ➔ Launch Commands: Object Server Parameters			supplement any that were supplied on the server invocation command line. Any command line parameters take precedence. For a list of object server parameters, see Object Server Parameters . For a more detailed explanation of object server parameter handling, see Server Startup Command .
Port Number <i>In SAS Management Console:</i> <Connection> ➔ Options ➔ Port Number	Required if server will have Java clients	IOM Bridge	<p>The port on which to connect to this server.</p> <p>If you specify a value for both port and service, then the value for service is used.</p> <p>If you are using a spawner and neither port nor service is specified, the spawner attempts to use the service name <code>sasobjspawn</code> as the service. If <code>sasobjspawn</code> has already been used, the spawner removes this service definition from its list.</p> <p>The port number is required if the server will have Java clients.</p> <p>The default port numbers are as follows:</p> <ul style="list-style-type: none"> • SAS Workspace Server: 8591 • SAS Stored Process Server: 8601 • SAS OLAP Server: 5451 • SAS Metadata Server: 8561
Protocol <i>In SAS Management Console:</i> <Connection> ➔ Protocol	Required	COM/DCOM, IOM Bridge	The protocol (Bridge or COM) that clients can use for connection. The protocol <code>bridge</code> must be used for servers that are serviced by the spawner. These include all servers other than Windows, as well as Windows servers that will be accessed by Java clients.
Recycle Activation Limit <i>In SAS Management Console:</i> Options ➔ Advanced Options ➔ Load Balancing Properties ➔ Recycle Activation Limit <i>and</i> Options ➔	Optional	COM/DCOM, IOM Bridge	<p>For pooling (SAS Workspace Servers only) and load balancing (SAS Stored Process Servers only), specifies the number of times a connection to the server will be reused in a pool before it is disconnected ("recycled"). If the value is 0, then there will be no limit on the number of times a connection to the server can be reused. This property is optional. The default value is 0.</p> <p>Note: For SAS Stored Process Servers, setting a Recycle Activation Limit can cause problems with sessions. If you create sessions, use the default value of 0 for Reaction Activation Limit.</p>

Advanced Options ➔ Pooling Properties ➔ Recycle Activation Limit			
Required Encryption Level <i>In SAS Management Console:</i> <Connection> ➔ Options ➔ Advanced Options ➔ Encryption ➔ Required Encryption Level	Optional	IOM Bridge	The level of encryption to be used between the client and the server. None means no encryption is performed; Credentials means that only user credentials (ID and password) are encrypted; and Everything means that all communications between the client and server are encrypted. The default is Credentials .
Server Encryption Algorithms <i>In SAS Management Console:</i> <Connection> ➔ Options ➔ Advanced Options ➔ Encryption ➔ Server Encryption Algorithms	Optional	IOM Bridge	The encryption algorithms that are supported by the launched object server. Valid values are: RC2, RC4, DES, TRIPLEDES, and SASPROPRIETARY, depending on the country in which the SAS software is licensed. See SAS/SECURE documentation for more information regarding this field. The default is SASPROPRIETARY.
Service <i>In SAS Management Console:</i> <Connection> ➔ Options ➔ Advanced Options ➔ Service	Optional	IOM Bridge	The service in which to connect to this server. If you specify a value for both port and service, then the value for service is used. If you are using a spawner and neither port nor service is specified, the spawner attempts to use the service name sasobjspawn as the service. If sasobjspawn has already been used, the spawner removes this service definition from its list. Note: If the server has Java clients, specify a port instead of a service.
Software Version <i>In SAS Management Console:</i> Options ➔ Software Version	Required	COM/DCOM, IOM Bridge	Specifies the version of the server software.

Start Size <i>In SAS Management Console:</i> Options ➤ Advanced Options ➤ Load Balancing Properties ➤ Start Size	Optional	IOM Bridge	For SAS Stored Process Servers, the number of Multibridge connections to start when the spawner starts.
Startup Cost <i>In SAS Management Console:</i> Options ➤ Advanced Options ➤ Load Balancing Properties ➤ Startup Cost	Optional	IOM Bridge	For load-balancing servers using the cost algorithm, the cost for starting a server.
Vendor <i>In SAS Management Console:</i> Options ➤ Vendor	Required	COM/DCOM, IOM Bridge	Specifies the vendor of the server software.

IOM Bridge

Object Server Parameters

All object server parameters are applicable on the command line that starts the server:

- For servers that are started by the object spawner, the object server parameters come from your server definition in the SAS Metadata Repository. (The server definition is located under the Server Manager plug-in of SAS Management Console. In the server definition, select the Options tab to locate the **Object Server Parameters** field).
- For servers that are not spawned (such as those that are run from command scripts, those that are run as Windows services, or those that are launched by COM), you use the OBJECTSERVERPARMS SAS option to specify the object server parameters on the command line.

To simplify the command that is needed to invoke an IOM server, the server startup sequence can also connect back to the metadata server in order to fetch additional information, including object server parameters. This feature involves use of the SERVER= and METAUTOINIT object server parameters. See [Specifying Metadata Connection Information](#) for details. The object server parameters that can be obtained in this way are indicated in the table below under the column "Can Be Fetched at Server Startup." These object server parameters can be specified in the server definition in SAS Management Console. In the server definition, select the Options tab to locate the **Object Server Parameters** field.

Important Note:

You can fetch object server parameters from metadata as follows:

- **When you start the server with a script**, some object server parameters cannot be obtained from the metadata. These parameters are designated as "No" in the "Can Be Fetched at Server Startup" column. Do not enter these object server parameters in your metadata.
- **When you start the server with a spawner**, all object server parameters can be obtained from the metadata (even those that are designated as "No" in the "Can Be Fetched at Server Startup" column).

Note: Object server parameters that are specified on the command line always override object server parameters obtained from a SAS Metadata Repository.

Object Server Parameter	Value	Connection Type	Can Be Fetched at Server Startup (When Starting a Server With a Script)	Definition
ANONYMOUSLOGINPOLICY	Deny Restrict	IOM Bridge	Yes	Specifies whether the server permits any access at all to connections that do not supply a user ID (in programming terms, ones that supply a zero-length user ID).

				<p>If you specify "restrict," then the server allows connections that do not have a user ID; however, the client only has restricted access to the IServerStatus interface (used primarily for querying basic server status).</p> <p>If you specify "deny," then the server completely disallows connections that do not provide a user ID. The default is "restrict." For details about ANONYMOUSLOGINPOLICY, see Setting Up Additional Server Security in the Security chapter.</p>
APPLEVEL	0, 1, 2, 3	IOM Bridge COM/DCOM	Yes	<p>Specifies the detail level of the trace that is written by the server application (such as the OLAP server, the SAS Metadata Server or the SAS Stored Process Server). The default value if APPLEVEL is omitted (1) enables logging at a level that is suitable for a production server; therefore, this parameter is optional.</p> <p>APPLEVEL=0 disables the application's logging and is discouraged because it suppresses useful diagnostic information. Higher APPLEVEL values can invoke additional tracing. The SAS Metadata Server, for example, defines additional logging levels. For details, see Enabling Logging of Authentication Events and SAS Metadata Server Logging Overview in the <i>SAS Metadata Server: Setup Guide</i>.</p>
CLASSFACTORY Alias: CLSID	36 character class identifier	IOM Bridge COM/DCOM	Yes	<p>Specifies the class ID number, which specifies the type of server to instantiate (for example, 2887E7D7-4780-11D4-879F-00C04F38F0DB specifies a SAS Metadata Server). An IOM server exposes one top-level class through its class identifier.</p> <p>By default, an IOM server hosts the Workspace class. If you want to specify an alternate class to expose as the top-level class, use the classfactory option to identify the class to IOM.</p> <p>When using the SERVER= objectserverparms suboption, the classfactory does not need to be specified because it is obtained from the logical server definition in the SAS Metadata Repository.</p> <p>This option is primarily used to start the SAS Metadata Server.</p>

CLIENTENCRYPTIONLEVEL Alias: CEL	none credentials everything	IOM Bridge	No	Specifies the degree of encryption to use when making outbound calls. This option is used only by the bridge protocol engine.
IOMLEVEL	0, 1, 2, or 3	IOM Bridge COM/DCOM	Yes	Specifies trace level for protocol-independent IOM events, particularly calls and the SAS LOG of workspaces. The default is 0. If IOMLEVEL is set to 1, then the calls that enter and leave the server are traced. This feature can be very helpful for identifying whether a problem arose in a client or in the server. Using IOMLEVEL=1 with the SAS Metadata Server will capture the input and output XML strings for metadata requests. For more information, see Capturing XML in the Log and SAS Metadata Server Logging Overview in the <i>SAS Metadata Server: Setup Guide</i> . For performance reasons, it is recommended that IOMLEVEL=1 be used only when diagnosing problems. Higher values of IOMLEVEL produce traces that are intended only for use by SAS Technical Support. Depending on the calls that are being traced, the JNLSTRMAX and JNLLINEMAX values may need to be increased to prevent truncation of long strings and long lines.
JNLARRELM	Numeric Value	IOM Bridge COM/DCOM	Yes	Specifies the maximum number of array elements to print out when an IOM array value is traced.
JNLLINEMAX	Numeric Value	IOM Bridge COM/DCOM	Yes	Specifies the maximum length of a line printed in the IOM server journal.
JNLSTRMAX	Numeric Value	IOM Bridge COM/DCOM	Yes	Specifies the maximum length of string printed in the IOM server journal. This option can be used to adjust the amount of material included in an IOM trace. A value greater than 500 is recommended.
LOGFILE Alias: LOG	Path in which to create the IOM server trace log	IOM Bridge COM/DCOM	Yes	Specifies an alternative file for the SAS log for IOM server trace output. Note: Using this option on a spawned server can prevent multiple servers from running simultaneously because they will all try to open the same log file. It is therefore recommended that this option be used only for specific diagnostic tasks. Note: The user who starts the server must have execute and write permissions for the log

				destination path.
METAAUTOINIT NOMETAAUTOINIT	N/A	IOM Bridge COM/DCOM	Yes	Specifies whether the IOM server should connect back to the SAS Metadata Server during startup in order to obtain additional configuration information such as object server parameters and pre-assigned libraries. When METAAUTOINIT is specified, the server uses the provided META* options to connect to the SAS Metadata Server. With NOMETAAUTOINIT, IOM server startup does not connect back to the SAS Metadata Server. The default depends on the type of server. For further details, see Server Startup Command . This option is applicable only if you have specified your logical server with the SERVER= object server parameter.
PELEVEL	0, 1, 2, or 3	IOM Bridge	No	Specifies trace protocol engine logic and packets. Level 3 specifies the most verbose output. The default is 0.
PORT	TCP/IP port number	IOM Bridge	Yes	Specifies the value for the bridge protocol engine to use as the port to start listening for client connections. Do not specify this option with spawned servers; it will be supplied automatically by the spawner.
PROTOCOL	bridge com (com,bridge)	IOM Bridge COM/DCOM	Yes	Specifies the protocol engines to launch in server mode. Server mode indicates that the protocol engines will listen for client connections. By default, Windows servers use the COM protocol engine and all other servers use the Bridge protocol engine. If you specify (com, bridge) then a multi-user server can simultaneously support clients using different protocols. Do not specify this option with spawned servers; it will be supplied automatically by the spawner.
SECURITY NOSECURITY	N/A	IOM Bridge COM/DCOM	No	<p>Specifies whether client authentication is enabled. By default (SECURITY), clients must be authenticated; one exception is the use of ANONYMOUSLOGINPOLICY for public interfaces (see Setting Up Additional Server Security).</p> <p>When security is enabled, the bridge protocol engine requires a user name and password; the COM protocol engine is integrated with the single-signon security of the Windows networking environment. Authorization decisions are controlled by the server application. If NOSECURITY is specified, these security mechanisms are bypassed.</p>

SERVER	Logical server name or OMSOBJ URI (object ID)	IOM Bridge COM/DCOM	No	Specifies the logical server name for the IOM run-time and server application to use to locate configuration information in a SAS Metadata Repository. The SERVER= option can be used to retrieve many of the OBJECTSERVERPARMS options (including PORT, PROTOCOL and CLASSFACTORY) from a SAS Metadata Repository. For details, see Specifying Metadata Connection Information .
SERVICE	TCP service name	IOM Bridge	Yes	Specifies the TCP service name (for example, from /etc/services on a UNIX system) for the port that the IOM Bridge protocol engine will use to listen for connections from clients. Do not specify this option with spawned servers; it will be supplied automatically by the spawner.
TRUSTSASPEER Alias: TSASPEER	N/A	IOM Bridge	Yes	Enables SAS peer sessions from IOM servers to connect as trusted peer sessions. For details, see Implementing Trusted Authentication Mechanisms .
V8ERRORTEXT	N/A	IOM Bridge COM/DCOM	Yes	Indicates that the MVA components should return Version 8 style error messages instead of the Version 9 XML style error messages.

IOM Bridge

Fields for the Spawner Definition

The spawner definition contains information for an instance of a SAS spawner (see [Spawner Overview](#)). The spawner is defined using the fields listed in the following table. For each option, the table shows the following information:

- the name that identifies the field name in SAS Management Console. Under each field's name, the table shows the corresponding properties tab and field name in the SAS Management Console application.
- "Required" or "Optional" to indicate whether the field is required.
- the type of server connection for which the field is used.

Note: Spawners are used only with servers that use an IOM Bridge connection. Therefore, IOM Bridge is listed as the connection type for each option.

- a definition of the field.

For step-by-step instructions about defining the metadata for a spawner, refer to [Using SAS Management Console to Define or Modify a Spawner](#).

Fields for the Spawner Definition			
Field Name	Required/Optional	Connection Type	Definition
Associated Machines <i>In SAS Management Console:</i> Options ➤ Associated Machines	Optional	IOM Bridge	The name of the machine on which this spawner will run and listen for connection requests for the server. (The list of machine names is created from the machine names for the servers that have already been defined in a metadata repository. If the desired machine name is not listed, then you must create a server definition for this machine. For details, see Using SAS Management Console to Define Servers .)
Authentication Domain <i>In SAS Management Console:</i> <Connection> ➤ Options ➤ Authentication Domain	Optional	IOM Bridge	The domain that is associated with a set of computing resources that use the same authentication process. The spawner definition must have the same authentication domain name as the server with which it connects. The spawner uses the authentication domain name, along with the machine name to determine which servers it services.
Description <i>In SAS Management Console:</i> General ➤ Description	Optional	IOM Bridge	Text to summarize why this definition exists.
Encryption Key Length	Optional	IOM Bridge	A numeric value (0, 40, or 128) that specifies the encryption key length. See SAS/SECURE documentation for more information regarding this field.

<i>In SAS Management Console:</i> Options ➤ Encryption Key Length			
Host Name <i>In SAS Management Console:</i> <Connection> ➤ Options ➤ Host Name	Required	IOM Bridge	The <u>DNS name</u> and IP address for the machine on which this spawner definition can execute. The machine name must be the official network name (for example, <code>machine.corp.com</code>). The string <code>localhost</code> can be used to signify the host on which the spawner is executing.
Log File <i>In SAS Management Console:</i> Initialization ➤ Log File	Optional	IOM Bridge	A fully qualified path to the file in which spawner activity is to be logged. Paths with blank spaces must be enclosed in quotation marks. On Windows, paths with embedded blank spaces must be enclosed in double quotation marks. On z/OS, specify filenames similar to UNIX file paths due to the requirement for z/OS Unix System Services.
Major Version Number <i>In SAS Management Console:</i> Options ➤ Major Version Number	Required	IOM Bridge	Specifies the major version number of the component.
Minor Version Number <i>In SAS Management Console:</i> Options ➤ Minor Version Number	Required	IOM Bridge	Specifies the minor version number of the component.
Name <i>In SAS Management Console:</i> General ➤ Name	Required	IOM Bridge	The unique name for this spawner. When specified at spawner invocation, its value identifies which spawner definition to use.
Operator Login <i>In SAS Management Console:</i> Initialization: Operator Login ➤ Operator Login	Required	IOM Bridge	The login that contains the password the spawner uses when starting a server as an operator connection. Click New to define a new login. If you do not specify a login, the operator password defaults to <code>sasobjspawn</code> .
	Required	IOM Bridge	

Port <i>In SAS Management Console:</i> <Connection> ➔ Options ➔ Port			<p>The port on which to connect to the spawner. If neither port nor service is specified, the service name sasobjspawn is used as the service. The type of port depends on the following values of Protocol:</p> <p><i>Protocol=Load Balancing</i> Port for a load balancing connection. The default is 8571.</p> <p><i>Protocol=UUID</i> Port for a UUID connection. The default is 8551.</p> <p><i>Protocol=Operator</i> Port for the operator connection. The default is 8581.</p>
Protocol <i>In SAS Management Console:</i> <Connection> ➔ Protocol	Optional	IOM Bridge	<p>The type of connection. Possible values are</p> <p><i>Load Balancing</i> Connection to a load balancing port or service.</p> <p><i>UUID</i> Connection to a UUID port or service.</p> <p><i>Operator</i> Connection to the operator port or service.</p>
Servers <i>In SAS Management Console:</i> Servers	Required	IOM Bridge	<p>The list of servers that this spawner is permitted to start. (The servers that are listed have been defined to run on the same host as the spawner.) Select the servers you want this spawner to start. Click New to define a new server that runs on the same host as the spawner.</p>
Service <i>In SAS Management Console:</i> <Connection> ➔ Advanced Options ➔ Service	Optional	IOM Bridge	<p>The service in which to connect to the spawner. If neither port nor service is specified, the service name sasobjspawn is used as the service. The type of service depends on the following values of Protocol.</p> <p><i>Protocol=Load Balancing</i> Service is the load balancing service.</p> <p><i>Protocol=UUID</i> Service is the UUID service.</p> <p><i>Protocol=Operator</i> Service is the operator service.</p>
Software Version <i>In SAS Management Console:</i> Options ➔ Software Version	Required	IOM Bridge	<p>Specifies the version of the spawner software.</p>
Verbose <i>In SAS Management Console:</i>	Optional	IOM Bridge	<p>When selected, this value causes the spawner to record more details in the log file (LogFile or slf).</p>

Initialization ➔			
Verbose			

IOM Bridge

Fields for the Pooled Logical Server and Puddle Definitions

You can only convert SAS Workspace Servers to pooled logical servers.

The pooled logical server definition contains information for an instance of a pooled logical server. The pooled logical server is defined using the fields listed in the following table. For each field, the table shows the following information:

- the name that identifies the field in SAS Management Console. Under each field name, the table shows the location of the corresponding properties tab and field name in the SAS Management Console application.
- "Required" or "Optional" to indicate whether the field is required.
- a definition of the field.

For general information about the use of logical servers, refer to [Overview of Pooling](#).

Fields for Pooled Logical Server Definitions		
Field Name	Required/ Optional	Definition
Name <i>In SAS Management Console:</i> General ➔ Name	Required	Name of the pooled logical server.
Description <i>In SAS Management Console:</i> General ➔ Description	Optional	Text to summarize why this definition exists. This field is not used by the logical server.
Puddles <i>In SAS Management Console:</i> Options ➔ Puddles	Required	The puddles used for pooling. Click New to define a new puddle.

The puddle definition contains information for an instance of a puddle. The puddle is defined using the fields that are listed in the following table.

Note: For COM connections, only one puddle can be defined.

Fields for the Puddle Definition		
Field Name	Required/ Optional	Definition
Name <i>In SAS Management Console:</i> Options ➔ Puddles ➔	Required	Name of the puddle.

Name		
Minimum Available Servers <i>In SAS Management Console:</i> Options ➤ Puddles ➤ Minimum Available Servers	Required	The minimum number of connections using this login definition that need to be available. This value includes only idle connections.
Minimum Number of Servers <i>In SAS Management Console:</i> Options ➤ Puddles ➤ Minimum Number of Servers	Required	The minimum number of connections using this login definition that are created when the pool is created. This value includes both connections that are in use and connections that are idle. The default value is 0.
Login <i>In SAS Management Console:</i> Options ➤ Puddles ➤ Login	Required	<p>The user ID associated with the puddle. The user or group metadata identity that owns this login can also access the puddle.</p> <p>Note: The login field is used with IOM Bridge connections only.</p>
Grant Access to Group <i>In SAS Management Console:</i> Options ➤ Puddles ➤ Grant Access to Group	Optional	The group metadata identity that can access this puddle. The user or group metadata identities (and their associated logins) that are members of the group can also access this puddle.

IOM Bridge

Fields for the Load–Balancing Logical Server Definition

For SAS Workspace Servers and SAS Stored Process Servers, the load–balancing logical server definition contains information for a load–balancing cluster. For each field, the table shows the following information:

- the name that identifies the field in SAS Management Console. Under each field name, the table shows the corresponding properties tab and field name in SAS Management Console.
- "Required" or "Optional" to indicate whether the field is required.
- the type of server connection for which the field is used. Load–balancing logical servers can only be configured for IOM Bridge connections.
- a definition of the field.

For information about load–balancing logical servers, refer to [Overview of Load Balancing](#) and [Load Balancing](#).

Fields for the Load Balancing Logical Server Definition		
Field Name	Required/ Optional	Definition
Name <i>In SAS Management Console:</i> General ➔ Name	Required	The logical server that is being defined.
Balancing Algorithm <i>In SAS Management Console:</i> Options ➔ Balancing Algorithm	Required	The type of balancing algorithm to use when load balancing the servers: <i>Cost (SAS Workspace Servers and SAS Stored Process Servers)</i> Performs load balancing based on the current cost or running servers and the startup cost of new servers. The cost algorithm takes the current cost of the servers and the startup cost of new servers into account and redirects the client to the server with the lowest cost. <i>Response Time (SAS Stored Process Servers only)</i> Performs load balancing based on servers' average response time and redirects the clients to a server by using a round–robin approach to the response time list. Response times are updated based on the response refresh rate.
Description <i>In SAS Management Console:</i> General ➔ Description	Optional	Text to summarize why this definition exists. This field is not used by the logical server.
Response Refresh Rate <i>In SAS Management Console:</i> Options ➔	Required	(SAS Stored Process Servers only) If the BalancingAlgorithm=Response Time, the length of time (in milliseconds) that a load balancer uses a set of response time values. At the end of this period the load balancer updates the response times and re–orders the servers for all of the servers in the load–balancing logical server.

Response Refresh Rate		Note: If this field is set to 0, the load balancer does not use the response time list to redirect clients to servers; instead, the load balancer redirects clients in a round-robin manner.
Cost Per Client <i>In SAS Management Console:</i> Options ➔ Cost Per Client	Required	If the BalancingAlgorithm=Cost, the default value of cost to add or subtract from a server's cost when a client connects or disconnects.
Logical Server Credentials <i>In SAS Management Console:</i> Options ➔ Logical Server Credentials	Required	The login that the load-balancing spawner uses to connect to other load-balancing spawners with servers in the same load-balancing logical server.

HTTP Servers

Administering HTTP Servers and WebDAV

HTTP servers use the HTTP protocol to provide read-only file access through the World Wide Web.

WebDAV is an extension to HTTP that provides write access, version control, and other features in addition to the basic features of HTTP. WebDAV is typically enabled only for specific folders on an HTTP server.

To define an HTTP server:

1. Set up a SAS Metadata Server and register a metadata repository. For instructions, see [🌐 SAS Metadata Server: Setup Guide](#).
2. Define the server using SAS Management Console. For details, see [Using SAS Management Console to Define an HTTP Server](#).

For the Xythos WFS WebDAV server, the SAS User Management Customization (provided with the Xythos WFS WebDAV server installation) enables the WebDAV server to use authentication and authorization metadata on the SAS Metadata Server. For more details about authentication and authorization with the Xythos WFS WebDAV server, see [Implementing Authentication and Authorization for Xythos WFS WebDAV](#) and [Scenario: Using the Xythos Administration GUI and SAS User Management Customization with the Portal](#).

HTTP Servers


Using SAS Management Console to Define an HTTP Server

To define an HTTP server using the SAS Management Console:

1. Start SAS Management Console and connect to a metadata repository.
2. From the navigation tree, select Server Manager. Then select **Actions ➤ New Server** from the menu bar. The New Server Wizard appears.
3. Select **Http Server** from the list of resource templates and click **Next**.
4. Enter a unique **Name** and a **Description** for this HTTP server. Click **Next**.
5. Enter the **Version** and **Vendor** of your server software.

Select or configure your **Base paths**:

- To create a new base path, click **New**. The New Base Path dialog box appears. Complete the following fields:

The image shows a 'New Base Path' dialog box with a title bar containing a close button. It has two text input fields: 'Base Path:' with a slash '/' entered, and 'Description:'. Below these fields is a checkbox labeled 'Supports WebDAV' which is currently unchecked. At the bottom are three buttons: 'OK', 'Cancel', and 'Help'.

Base Path

specifies the base path for the server.

Description

specifies an optional description for the base path.

Supports WebDAV

specifies whether the base path supports WebDAV.

Click **OK** to return to the New Server Wizard.

- To edit a base path, select the path and click **Edit**. The Edit Base Path dialog box appears. When you are finished editing the path, click **OK** to apply your changes.
- To delete a base path, select the path and click **Delete**. You will be prompted to confirm the deletion.

Note: You can select more than one base path. Typically, an HTTP server should specify "/" (all locations) as a base path, but not specify WebDAV support for that path. Folders with WebDAV content should be specified individually. For example, if your HTTP server contains two folders with WebDAV content (DAVinfo1 and DAVinfo2), you would specify three base paths:

- / – for standard HTTP content
- /DAVinfo1 – for WebDAV content
- /DAVinfo2 – for WebDAV content

When you are finished, click **Next**.

6. Enter your settings for the following fields:

Authentication Domain

specifies the domain that is associated with a set of computing resources that use the same authentication process. To add a new authentication domain, click **New** and enter the name and description for the domain in the New Authentication Domain dialog box.

Application Protocol

specifies the application protocol for the server. Valid values are `http` and `https`.

Host Name

specifies the host name used to access the server.

Port Number

specifies the port number used to access the server.

Proxy URL

specifies a proxy URL for use in accessing the server.

When you are finished, click **Next**.

7. Verify the server information. If any of the settings are incorrect, click **Back** to make changes.

When your settings are all correct, click **Finish** to complete the server definition and return to the SAS Management Console main window.

Starting Servers

Starting Servers

There are four methods of starting an IOM server:

- command line
- COM (in response to a client request)
- spawner
- as a service

The method that you use depends on the type of connection that is defined for the server, the type of server you are starting (OLAP, metadata, workspace, or stored process), and the operating environment. Use the following table as a guide to determine the available server start methods for your configuration. Additional information about specific configurations follows the table. For details about the order in which to start servers, see [Server and Services Startup Order](#)

Starting a Server			
Server Protocol	Operating Environment	Server Type	Available Start Methods
IOM Bridge	Windows	SAS Metadata Server	<ul style="list-style-type: none"> • Service (recommended) • Command line
		OLAP server	<ul style="list-style-type: none"> • Service (recommended) • Command line
		SAS Workspace Server	Spawner (required)
		SAS Stored Process Server	Spawner (required)
	UNIX z/OS VMS Alpha	SAS Metadata Server	Command line
		OLAP server	Command line
		SAS Workspace Server	Spawner (required)
		SAS Stored Process Server	Spawner (required)
COM	Windows only	SAS Metadata Server (experimental in SAS 9.1)	<ul style="list-style-type: none"> • Service (recommended) • Command line
		OLAP server	<ul style="list-style-type: none"> • Service (recommended) • Command line
		SAS Workspace Server	COM
IOM Bridge and COM	Windows only	SAS Metadata Server (COM experimental in SAS 9.1)	

			<ul style="list-style-type: none"> • Service (recommended) • Command line
		OLAP server	<ul style="list-style-type: none"> • Service (recommended) • Command line

Regardless of the method that you choose, you must construct a server startup command using appropriate SAS system options and object server parameters. See [Server Startup Command](#) for details.

SAS Workspace Servers (COM Connection)

For details about customizing the startup command for a SAS Workspace Server with a COM connection, see [Customizing the Server Command for COM/DCOM Connections](#).

SAS Workspace Servers and SAS Stored Process Servers (IOM Bridge Connection)

If you are starting a SAS Workspace Server or SAS Stored Process Server that uses an IOM Bridge connection, you must use a spawner to start the server. You must also create a metadata configuration file that contains information for accessing the SAS Metadata Server. See [Metadata Configuration File](#) for more information.

Verify that you have planned for the appropriate login information to specify in the metadata configuration file. For details, see [Planning the Spawner Security](#).

Project Installation Note: If you used the project installation and SAS Configuration Wizard, then the ObjectSpawner directory of your installation contains a metadata configuration file named `OMRConfig.xml`, and a startup script that is used to start the object spawner.

- For z/OS, refer to [Configuring and Starting the Object Spawner on z/OS](#).
- For other operating environments, refer to [Invoking \(Starting\) the Spawner](#) for examples and special security considerations.
- For Windows, refer to [Starting the Spawner on Windows](#).
- For UNIX, refer to [Starting the Spawner on UNIX](#).

For all operating environments, refer to the list of [Spawner Invocation Options](#).

SAS Metadata Servers and SAS OLAP Servers

To start a SAS OLAP Server or SAS Metadata Server you must create a server startup command or start the server as a service. For Windows platforms, it is recommended that you start the servers as services.

Project Installation Note: If you used the project installation and SAS Configuration Wizard, then the Metadata Server (SAS Metadata Server) and OLAPServer (SAS OLAP Server) directories of your installation contain the startup scripts (*server-type.extension*) for your installation.

- **For platforms other than Windows, to start servers**, see the following information:
 - ◆ To start an OLAP server, see: [🌐 Creating and Modifying the SAS OLAP Server Script](#) in the *SAS OLAP Server Administrator's Guide*.
 - ◆ To start a SAS Metadata Server that uses an IOM Bridge connection, see:
 - ◇ **UNIX**. To start a SAS Metadata Server that runs on UNIX, see [🌐 Start Command in a UNIX Environment](#) in the *SAS Metadata Server: Setup Guide*.
 - ◇ **z/OS**. To start a SAS Metadata Server that runs on z/OS, see [🌐 Starting a SAS Metadata Server on OS/390](#) in the *SAS Metadata Server: Setup Guide*.
- **For Windows platforms, to configure and start a server as a service**, you must use the SSCU utility to create a configuration file.
 - ◆ To start a SAS Metadata Server that uses an IOM Bridge connection as a service, see [🌐 Starting the Server as a Service](#) in the *SAS Metadata Server: Setup Guide*.
 - ◆ To start an OLAP server as a service, see [🌐 Starting the SAS OLAP Server as a Service](#) in the *SAS OLAP Server Administrator's Guide*.

Starting Servers

Server Startup Command

An IOM server is a noninteractive SAS session that is run with the OBJECTSERVER system option. Depending on how the server is run, the startup command might be stored in a script, in the Windows registry, or in the SAS Metadata Server. Furthermore, in order to make it easy to specify the command, the server can be started using a simple command with an option to connect back to the metadata server to obtain additional IOM-specific options.

You can specify the server startup command in several different locations:

- on the system command line
- in a script
- in the Command field in the server definition (located on the Options tab of the server definition in SAS Management Console)
- in the Windows registry (for COM connections only)

The general form of the server startup command is:

```
SAS-exec -objectserver <other-system-options>  
-objectserverparms "object-server-parameters"
```

- *SAS-exec* is the path to the SAS executable. The following table contains example values for *SAS-exec*:

Location	<i>SAS-exec</i>
system command line, script	Use the complete path to the SAS executable. Windows example: c:\program files\sas\sas 9.1\sas.exe UNIX example: /usr/local/bin/sas
<u>Command</u> field in the server definition (located on the Options tab of the server definition in SAS Management Console)	Use the name of the SAS executable. The complete path is not needed. Example: sas
Windows registry	Use the complete path to the SAS executable. You must use "8.3" (short) filenames. Example: c:\progra~1\sas\sas9~1.1\sas.exe

- *-objectserver* launches this SAS session as a server.
- *other-system-options* are other system options. System options that are typically used for servers include LOG, NOTERMAL, and NOLOGO. For complete information about system options, see the *SAS Language Reference: Dictionary*.
- *object-server-parameters* are IOM-specific options that are passed to the server by the OBJECTSERVERPARMS system option. For more information, see Object Server Parameters.

Note: For SAS Workspace Servers that run on UNIX, it is sometimes necessary to call the SAS startup command using a *wrapper script*. For more information, see Initializing UNIX Environment Variables for Workspace Servers.

The server startup command is obtained as follows:

- **When the server is started by a spawner, the startup command is stored in SAS metadata** (SAS Workspace and SAS Stored Process Servers with IOM Bridge connection). In the SAS metadata, there is one metadata field for the SAS startup command and system options, and another field for the object server parameters. The object spawner combines these two fields, along with connection information and some spawner internal object server parameters, to create the complete SAS command. The object spawner then passes this command to the operating environment.
- **When the server is started by a script or as a Windows service, or is launched by COM** (that is, a SAS Workspace Server with a COM connection, any OLAP server, or any SAS Metadata Server), the command that is passed to the operating environment is not determined by SAS metadata. However, SAS Workspace Servers with a COM connection, and any OLAP server can connect back to the SAS Metadata Server in order to obtain additional object server parameters and connection information (such as protocol engine and port number). (Note that some object server parameters cannot be obtained from the metadata).

When object server parameters are specified in the metadata, if there are any object server parameters that are also specified in the command, then the object server parameters in the command take precedence over those that are stored in the metadata. To enable the ability to connect back to the SAS Metadata Server for additional object server parameters and connection information, you specify the METAAUTOINIT and SERVER= object server parameters in the command. For more information, see [Specifying Metadata Connection Information](#).

Regardless of how the server is started, SAS Workspace Servers (with IOM Bridge or COM connections), SAS Stored Process Servers (IOM Bridge only), and SAS OLAP Servers can also connect back to the SAS Metadata Server in order to obtain configuration information, such as preassigned libraries, that is associated with the SAS Application Server. For example, if the SERVER= and METAAUTOINIT object server parameters are used, then the workspace, stored process, and OLAP servers will preassign libraries that are associated with the SAS Application Server definition. For more information, see [Specifying Metadata Connection Information](#).

The following table summarizes the ways that the SAS command, system options, and object server parameters can be specified for each type of IOM server.

Server Type	Launch with spawner	Use of SERVER= object server parameter	Can user specify METAAUTOINIT object server parameter?	Can server obtain command from SAS Metadata Server?	Can server obtain object server parameters from SAS Metadata Server?	Can server obtain librefs from SAS Metadata Server?
SAS Workspace Server with IOM Bridge connection	Required	Supplied by the spawner	Yes, if you want IOM to use librefs that are defined on the SAS Metadata Server	Yes (spawner retrieves)	Yes (spawner retrieves)	Yes, if METAAUTOINIT is specified
SAS Workspace Server with COM connection	Not allowed	Allowed	Yes, (with SERVER=) if you want IOM to use librefs that are defined on the SAS Metadata Server	No	Yes, they supplement the command-line object server parameters if both METAAUTOINIT and SERVER= are	Yes, if both METAAUTOINIT and SERVER= are specified

					specified	
SAS Stored Process Server	Required (load balanced)	Supplied by the spawner	Yes, if you want IOM to use librefs that are defined on the SAS Metadata Server	Yes (spawner retrieves)	Yes (spawner retrieves)	Yes, if METAAUTOINIT is specified
SAS OLAP server	Not allowed	Required	No, the default already specifies METAAUTOINIT	No	Yes, they supplement the command-line object server parameters if SERVER= is specified	Yes, if SERVER= is specified
SAS Metadata Server	Not allowed	Not allowed	No, not supported	No	No	Not supported

Important Note: When you start the server with a script, some object server parameters cannot be obtained from the metadata. For details, see the "Can Be Fetched at Server Startup" column in the [Object Server Parameters](#) section. Do not enter these object server parameters in your metadata.

In the server startup command, you can provide the following information:

- **SAS configuration file (required)**
- **Metadata Connection Information** (required when you specify the METAAUTOINIT object server parameter to enable a connection to the SAS Metadata Server)
- **SAS Autoexec File (optional)**
- **Logging Options (optional)**
- **Encoding and locale information (optional)**

For a workspace server with a COM connection, see [Customizing the Startup Command for Workspace Servers](#).

For workspace servers and stored process servers, see [Preventing Conflicts over the SASUSER Library](#).

Specifying a SAS configuration file (required)

To initialize SAS options, you must specify a SAS configuration file using the CONFIG system option in the server command. For example,

```
SAS-exec -config "C:\Program Files\SAS\SAS 9.1\sasv9.cfg"
```

The SAS configuration file contains SAS options that are automatically executed when SAS is invoked. The default configuration is located in the SAS installation directory; you can also create your own configuration file.

Specifying a SAS Autoexec File (optional)

To pre-assign server settings, specify a SAS autoexec file using the AUTOEXEC option in the server command. For example:

```
SAS-exec -autoexec "C:\Program Files\SAS\SAS 9.1\autoexec.sas"
```

A SAS autoexec file contains SAS statements that are executed as part of the SAS invocation. SAS autoexec files are particularly useful for pre-assigning librefs, filerefs, and macros. When multiple workspaces are used on the same server, each workspace inherits the server properties that are set by the autoexec file. Individual workspaces can override the properties that are inherited from the server by specifying new LIBNAME, FILENAME, or macro statements; however, these changes only affect the workspace where the new statements are submitted.

Note: Workspaces do not inherit the server WORK library that is used during autoexec processing.

To use a single autoexec file for both SAS sessions and IOM servers, you can set up conditional statements in your autoexec file. For example:

```
%macro autsetup;
%if %sysfunc(getoption(objectserver))=OBJECTSERVER
%then
%do;
    <IOM server autoexec statements>
%end;
%else
%do;
    <SAS session autoexec statements>
%end;
%mend autsetup;
%autsetup;
```

Important: For some SAS 9.1 hosts, IOM servers process a SAS autoexec file implicitly if the file is stored in the default location. This might cause compatibility issues for existing configurations because IOM servers did not process autoexec files in previous versions of SAS. You can suppress this behavior by specifying the NOAUTOEXEC option in the server command.

For more information about the AUTOEXEC system option, see the SAS documentation for your operating environment.

Specifying Logging Options (optional)

To diagnose server problems, specify the `-log` and `-logparm` logging options in the server command. Additional IOM-specific logging is available by specifying certain object server parameters. These object server parameters can be used to control the type and amount of information that is logged. For example, `IOMLEVEL=1` can be used to log all of the calls that are processed by the server. For details about object server parameters, see [Object Server Parameters](#).

When you specify the logging options, you can also configure the server to create a different log for each process, or switch logs during execution.

The following command (specified in the [Command](#) field of the server definition) creates a unique log file (in the server user's home directory) for each instance of this server definition.

```
SAS-exec -log "test%v.log" -logparm "rollover=session"
```

In the preceding example, when the spawner starts the first server, a log named `test1.log` is created; when the spawner starts the second server, a log named `test2.log` is created.

For information about system logging options, see *SAS Language Reference: Dictionary*.

If you are having trouble creating a log, then start the server from the system command line and specify the `TERMINAL` system option to see if additional messages are shown. Doing so can help diagnose problems such as an invalid log file path or a permission problem that prevents the creation of the log file.

Note: Specifying logging options can cause performance degradation in your server; therefore, you should specify logging options only to diagnose problems with your server connections.

Note: If you specify a log destination in the configuration metadata rather than the startup command, then you might miss some messages that are generated before the log destination is set.

Encoding and Locale Information (Optional)

If your server metadata contains characters other than those typically found in the English language, then you must be careful to start your server with an `ENCODING=` or `LOCALE=` system option that accommodates those characters. For example, a SAS server that is started with the default US English locale cannot read metadata that contains Japanese characters. SAS will fail to start and will log a message that indicates a transcoding failure.

In general, different SAS jobs or servers can run with different encodings (such as ASCII/EBCDIC or various Asian DBCS encodings) as long as the encoding that is used by the particular job or server can represent all of the characters for the data that is being processed. In the context of starting a server, this fact requires you to review the characters that are used in the metadata that describes your server (as indicated by the `SERVER= objectserverparm`) in order to ensure that SAS runs under an encoding that supports those characters.

Preventing Conflicts over the SASUSER Library

When multiple workspace servers or stored process servers are launched for the same user ID, the separate processes share a common SASUSER library. To prevent access conflicts, specify the `-RSASUSER` system option to make the SASUSER library read-only. You can specify the `-RSASUSER` option in the server command or in a SAS configuration file.

Note that some client applications might assume that the SASUSER library is writable. For example, Enterprise Guide 2.0 makes this assumption by default. Others clients, such as Web applications that use pooling, can potentially launch many workspace processes that would conflict over SASUSER. In order to support the requirements of both types of client, you might need to define a different workspace server configuration for use with each type.

Starting Servers

Specifying Metadata Connection Information (required if METAAUTOINIT is specified)

The metadata connection information is required when you specify the METAAUTOINIT object server parameter to enable a connection to the SAS Metadata Server. Note that for OLAP servers, METAAUTOINIT is specified by default. When you start a server with the METAAUTOINIT object server parameter, use of the SERVER= object server parameter enables you to pre-assign libraries to servers or to access server metadata:

- **pre-assign libraries to servers**

When a workspace, stored process, or OLAP server is started, the SERVER= object server parameter is used to obtain the library definitions that are defined in a SAS metadata repository for a server. (For workspace and stored process servers with an IOM bridge connection, the spawner automatically supplies the SERVER= parameter). When the server is started, it accesses the SAS Metadata Server to obtain the pre-assigned library definitions from the repository and assign the librefs for that server. The libref can then be used by all of the objects that are created on that server. For details about defining pre-assigned library definitions, see [Setting Up Libraries](#) in the Getting Started chapter.

- **access server metadata**

When a server is started, the SERVER= object server parameter is used on the server startup command in order to access the SAS Metadata Server and obtain the server metadata for that server. (For workspace and stored process servers with an IOM bridge connection, the spawner automatically supplies the SERVER= parameter). Use of the SERVER= object server parameter enables the server to obtain information about the type of server (CLASSFACTORY=) and its protocols and connections (PROTOCOL=, PORT=) from the metadata. This approach simplifies the server invocation command line. Using the SERVER= option enables the server to access additional object server parameters that might be specified in the metadata for the server definition.

When you use the METAAUTOINIT and SERVER= object server parameters, to enable a server to retrieve information from the SAS Metadata Server, you must also specify how to access the SAS Metadata Server. Therefore, when you launch the server, you must specify SAS Metadata Server connection information to enable the server to connect back to the SAS Metadata Server.

Note: By default, SAS Workspace Servers and SAS Stored Process Servers will not connect to the SAS Metadata Server (to retrieve the additional configuration metadata) unless you specify the METAAUTOINIT object server parameter.

The following table summarizes the locations where you specify the METAAUTOINIT and SERVER= parameters for each type of server.

Locations for METAAUTOINIT and SERVER= Parameters		
Server Type	METAAUTOINIT	SERVER=
Workspace Server with a COM connection*	Specify in Windows registry for server startup	Specify in Windows registry for server startup
Workspace Server	Command line or	Automatically supplied by the spawner

	In the SAS Management Console server definition: Options ➔ Launch Commands: Object Server Parameters	
Stored Process Server	Command line or In the SAS Management Console server definition: Options ➔ Launch Commands: Object Server Parameters	Automatically supplied by the spawner
OLAP Server	Automatically supplied as default	Specify in OLAP server startup script

***Note:** For details about customizing the server startup command for a workspace server with a COM connection, see [Customizing the Workspace Server Startup Command for COM/DCOM Connections](#).

For more details about the METAUTOINIT and SERVER= object server parameters, see [Object Server Parameters](#).

To use the METAUTOINIT and SERVER= object server parameter to obtain metadata configuration information:

1. **For SAS Workspace and SAS Stored Process Servers only**, specify the METAUTOINIT object server parameter on the command line or in the **Object Server Parameters** field of the server definition in SAS Management Console (found on the Options tab under Launch Commands).
2. **For SAS Workspace Servers with a COM connection and all SAS OLAP Servers**, specify the SERVER= object server parameter in the Windows registry for the server startup command (for workspace servers with COM) or in the object server parameters of the command that you use to start SAS (for OLAP servers). When you specify the SERVER= object server parameter, specify either a logical server name or the object URI:

Note: For SAS Stored Process and SAS Workspace Servers with an IOM Bridge connection, the spawner automatically supplies the SERVER= object server parameter.

- ◆ **logical server name:** Specify the logical server name of the SAS Metadata Server object definition. To determine the logical server name, in SAS Management Console select the logical server definition, and then select **File ➔ Properties** from the menu bar. Use the value in the **Name** field as the argument for the SERVER= object server parameter. For example:

```
SERVER="Sales - OLAP Logical Server"
```

- ◆ **URI:** You can also specify the generated object definition ID. To determine the generated ID, in SAS Management Console, select the logical server definition, and then select **File ➔ Properties** from the menu bar. Use the value in the **ID** field as the argument for the SERVER= object server parameter. For example,

```
SERVER="omsobj:LogicalServer/01234567.01234567"
```

The SAS Metadata Server determines which server to use based on the following, in this order:

1. The name of the logical server or the ID for the logical server definition. The SAS Metadata Server locates the server group that is defined in the logical server name or ID that you specify on the SERVER= object server parameter.
2. The host name on which you are starting the server. The SAS Metadata Server determines which server definition (within the logical server) to use based on the host name on which you are starting

your server.

When the logical server has been located, the associated actual server can be found for the machine on which the server is started. For IOM Bridge, in an advanced configuration where multiple bridge servers are located on the same machine, specifying the `PORT=` object server parameter when the server is launched indicates which server object is intended.

3. **Specify SAS Metadata Server Connection Information.** To enable the server to connect to the SAS Metadata Server, you must specify the appropriate security for the connection to the SAS Metadata Server as follows:

- ◆ If you specify the `trustsaspeer` option for the SAS Metadata Server startup command, the server connects to the SAS Metadata Server using the following user ID:
 - ◇ SAS Workspace Servers: the user ID of the workspace server's client
 - ◇ Pooled SAS Workspace Servers: the puddle login
 - ◇ SAS Stored Process Servers: the user ID that is defined on the Credentials tab of the server definition

For details about specifying the trusted peer option, see [Implementing Trusted Authentication Mechanisms](#).

- ◆ If you do not specify the `trustsaspeer` option, you must specify `META*` options for the SAS Metadata Server connection. When you specify the `META*` options for the credentials to connect to the metadata server, specify the user ID information as follows:
 - ◇ SAS Workspace Servers: the user ID of the workspace server's client
 - ◇ Pooled SAS Workspace Servers: the puddle login
 - ◇ SAS Stored Process Servers: the user ID that is defined on the Credentials tab of the server definition

For standard and pooled workspace servers, the `METAUSER` and `METAPASS` options defined in the workspace server definition cannot provide a different user ID and password for each login under which the workspace might be launched—all workspace users connect to the metadata server with the same credentials. If you need each user to be authenticated individually by the metadata server, use the `METAPROFILE` option to provide the user name and password for each user in a file in the user's home directory.

To understand the different security considerations for SAS Workspace Servers and SAS Stored Process Servers, see [Planning Security on Workspace and Stored Process Servers](#) (IOM Bridge Connection Only).

The following table summarizes the location where you specify the `METAAUTOINIT` and `SERVER=` parameters for each type of server.

Locations for Meta* Options		
Server Type	Meta* Options that are allowed on the command line or in the Command field of the server definition in SAS Management Console	Meta* Options that are allowed in a SAS config file
Workspace Server	METAPROFILE and METACONNECT	METAPROFILE and METACONNECT
	or METASERVER, METAPROTOCOL, METAPORT, METAUSER, and METAPASS	
Stored Process Server	METAPROFILE and METACONNECT	METAPROFILE and METACONNECT
	or	

	METASERVER, METAPROTOCOL, METAPORT, METAUSER, and METAPASS	
OLAP Server	METASERVER, METAPROTOCOL, METAPORT, METAUSER, and METAPASS	METAPROFILE and METACONNECT

You can specify the META* options in either of the following ways:

- ◆ Specify the META* options that contain the metadata server connection information on the command line or in the **Command** field (on the Options tab) of the server definition in SAS Management Console. Depending on your server type, you can use either of the following META* options:
 - ◇ For SAS Workspace Servers and SAS Stored Process Servers, the METAPROFILE and METACONNECT options. The following command specifies that the server will use the metadata configuration file `omr.xml` (located in the user's home directory) to connect to the SAS Metadata Server user connection profile named "SAS Metadata Server", and obtain metadata for the logical server named "My Server":

```
C:\Program Files\SAS\SAS 9.1\sas.exe -objectserver
-objectserverparms
    "METAAUTOINIT SERVER='My Server' "
-metaprofile omr.xml
-metacconnect "SAS Metadata Server Connection"
```

Note that, in SAS 9.1, the `-METAPROFILE` option does not honor environment variables (such as `SASROOT`) and, on Windows, is not relative to the setting of the `-SASINITIALFOLDER` option. Thus, in the above example, "omr.xml" is found in the current directory of the process, which will be the user's home directory in a spawned workspace.

To create the SAS metadata configuration file (XML file), see [Creating a Metadata Configuration File in SAS](#).

- ◇ For SAS Workspace Servers, SAS Stored Process Servers, and SAS OLAP Servers, the METASERVER, METAPROTOCOL, METAPORT, METAUSER, and METAPASS options. The following command specifies that the server will connect to the SAS Metadata Server on host `metaserver.unx.alphacorp.com` at port 9999 with the user ID "sasuser" and password "sasuser1", and obtain metadata logical server with an ID of "A3845545.04830224":

```
C:\Program Files\SAS\SAS 9.1\sas.exe -objectserver
-objectserverparms "METAAUTOINIT
    SERVER=omsobj:LogicalServer/A3845545.04830224"
-metaserver "metaserver.unx.alphacorp.com"
-metaport 9999 -metauser "sasuser"
-metapass "sasuser1" -metaprotocol bridge
```

- ◆ Specify the METAPROFILE and METACONNECT options (that contain the metadata server connection information) in your SAS configuration file.

For details about the META* options, see [SAS Metadata System Options](#) in the *SAS 9.1 Open Metadata Interface: Reference*.

Starting Servers

Customizing the Workspace Server Startup Command for COM/DCOM Connections

A workspace server is launched by COM in response to a `CoCreateInstance()` call (dim as new in Visual Basic) from a client. When COM launches a server, it checks the Windows registry for the launch command. The launch command is stored under the CLSID for the server type. There are two versions of the Workspace class: the original implementation from SAS 8 (Workspace Version 1.0) and a new version that was introduced in SAS 9 (Workspace Version 1.1). SAS 9 also provides a complete emulation of Workspace Version 1.0 that installs itself to be launched regardless of which version is requested.

A client can request the minimum version that it needs. Because most clients do not absolutely require any of the extra features that were introduced in SAS 9, they will typically request Workspace Version 1.0. However, the launch command that is used should be correct for both CLSIDs.

The registry locations for these are as follows:

Workspace Version 1.0:

```
HKEY_CLASSES_ROOT\CLSID\{440196D4-90F0-11D0-9F41-00A024BB830C}\LocalServer32
```

Workspace Version 1.1:

```
HKEY_CLASSES_ROOT\CLSID\{CF7BC7E6-C7E8-11D5-87E3-00C04F38F9F6}\LocalServer32
```

When SAS 9 is installed, or when you execute the `sas -regserver` command, SAS updates these keys to point to itself. The command that is set up by default is adequate for most purposes, but you can change it with the `regedit` utility if necessary.

If, for example, you want a workspace server that is launched by COM to contact a metadata repository in order to obtain additional pre-assigned libraries, then you can modify the launch command as follows:

```
C:\PROGRA~1\SAS\SAS9~1.1\SAS.EXE -config
"C:\Program Files\SAS\SAS 9.1\sasv9.cfg" -objectserver
-objectserverparms "metaautoinit server=
'Sales01 - Logical Workspace Server'"
-metaprofile c:\omr.xml -nologo -noterminal -noxcmd
-metacconnect "SAS Metadata Server Connection"
```

Note that COM launches do not accept long filenames for the EXE file and that they do not start in a well-defined initial directory unless you use the `SASINITIALFOLDER=` option. The preceding command uses the full path for the `METAPROFILE` option in order to compensate for the lack of a default directory. For more information, see [Specifying Metadata Connection Information](#). Note also that workspace servers require the `METAAUTOINIT` object server parameter as an indication that they should contact a SAS Metadata Repository.

Starting Servers

Initializing UNIX Environment Variables for SAS Workspace Servers

In UNIX environments, many third-party databases require access information such as the default server address to be set as environment variables. To make these environment variables available to a SAS Workspace Server, you must create the workspace using a *wrapper script* that defines the variables before invoking SAS.

The following code is an example script.

```
#!/bin/ksh -p

# Purpose: Runs database setup scripts before invoking SAS.
#          Called by objspawn.

# Restore quotation marks around arguments that have multiple tokens.

function quoteme { #arg

    if [[ $# -gt 1 ]]; then
        quoteme="\ "$*\ ""
    else
        quoteme=$1
    fi

    echo $quoteme
}

# Run database setup scripts or set required environment
# variables here.

<script calls or export commands>

# Reconstruct and execute the original SAS command.

cmd=''
for arg in "$@" ; do
    tmp="$(quoteme $arg)"
    cmd="$cmd $tmp"
done

eval exec $cmd
```

To use this script:

1. Add your `export` statements or script calls and save the file as `objspawn.setup` in a location where all users have read and execute permissions.
2. Set the execute bits for the file. You can do this using the following command:

```
chmod 755 objspawn.setup
```

3. Add `objspawn.setup` to the start of your `sas` command in the server definition. For example:

```
objspawn.setup sas
```

Starting Servers

Invoking (Starting) the Spawner

After you have created a [metadata configuration file](#) for the metadata server, you can then use the metadata configuration file to invoke and administer the defined spawner. Refer to the appropriate start-up procedures for your server platform:

- [Starting the Spawner on Windows](#)
- [Starting the Spawner on UNIX](#)
- [Starting the Spawner on Alpha/VMS](#)

As you use these instructions, refer to the list of [Spawner Invocation Options](#) that are available.

After you have started the spawner, you can connect to the spawner as an administrator (operator) to monitor and control the spawner's operation. For instructions, see [Monitoring the Spawner Using Telnet](#).

Security Considerations

The spawner can be launched with the `-noSecurity` option. However, this option should be used with caution, because it will allow any client connecting to the spawner to obtain a server using the same user ID that launched the spawner. This means that any client that can manipulate the host file system can obtain a server as if the client had the user ID that launched the spawner.

Note: If you use the `-noSecurity` option, the `-install` option is ignored.

Example Commands

In the following examples, `objspawn.xml` is the metadata configuration file that you created using the METACON command in SAS. The following are examples of the spawner command in the UNIX and Windows NT environments:

- UNIX example using a configuration file:

```
prompt> /sasv91/utilities/bin/objspawn
-sasSpawnercn "Spawner 1" -xmlconfigFile objspawn.xml
-sasLogFile "/sasv91/utilities/bin/objspawn.log"
```

- Windows NT example using a configuration file:

```
c:\program files\sas\sas 9.1> objspawn -sasSpawnercn Spawner1
-xmlConfigFile objspawn.xml -install
-sasLogFile "c:\logs\objspawn.log"
```

- Windows NT example using a configuration file and specifying not to use security:

```
c:\program files\sas\sas 9.1> objspawn -sasSpawnercn NameofSpawner
-xmlconfigFile objspawn.xml -nosecurity
-sasLogFile "c:\logs\objspawn.log"
```

Notes:

- In these examples, the command line options point to a spawner definition to use and a configuration file (`objspawn.xml`) where the configuration parameters are located.
- The invocation options vary depending on the platform. Refer to the [Spawner Invocation Options](#) for details.

- On Windows, in most cases you should install the spawner as a Windows NT service using the `-install` option.
- If you do not specify the `-sasSpawnerCn` option, the object spawner uses the first `sasSpawner` definition (on the metadata server) that has the same machine name as the current host.

Starting Servers

Starting the Spawner on Windows

To start the spawner on a Windows host:

1. **Note:** This step is only necessary if you are not starting the spawner as a service.

Define the user rights for the user who invokes the spawner. The user who invokes the spawner, in addition to being a Windows administrator, must have the following user rights:

- ◆ act as part of the operating system (Windows NT and Windows 2000).

This right needs to be held by the owner of a multi-user SAS session that will be authenticating connecting clients. This right is also required for the owner of the objspawn process. The Windows routine LogonUser() requires this user right for the process owner in order for it to authenticate other users.

- ◆ adjust memory quotas for a process (Windows XP only).

This right needs to be held by the owner of an objspawn process. The Windows routine CreateProcessAsUser() requires this user right for the process owner in order for that user to be able to launch SAS sessions on behalf of the connecting client.

- ◆ increase quotas (Windows NT and Windows 2000).

This right needs to be held by the owner of an objspawn process. The Windows routine CreateProcessAsUser() requires this user right for the process owner in order for that user to be able to launch SAS sessions on behalf of the connecting client.

- ◆ replace the process level token.

This right needs to be held by the owner of an objspawn process. The Windows routine CreateProcessAsUser() requires this user right for the process owner in order for that user to be able to launch SAS sessions on behalf of the connecting client.

To set the administrator's user rights on Windows NT:

- a. Select **Start ▶ Programs ▶ Administrative Tools ▶ User Manager**.
- b. From the Policies drop-down list, select **User Rights**.
- c. Select the **Show Advanced User Rights** check box.
- d. Add rights using the **Right** drop-down list.

To set the administrator's user rights on Windows 2000:

- a. Select **Start ▶ Settings ▶ Control Panel ▶ Administrative Tools ▶ Local Security Policy**.
- b. Select **Security Settings ▶ Local Policies ▶ User Rights Assignment**.
- c. Add rights by double-clicking each right and assigning the appropriate users.

To set the administrator's user rights on Windows XP:

- a. Select **Start ▶ Settings ▶ Control Panel ▶ Administrative Tools ▶ Local Security Policy**.
- b. Expand the tree for Local Policies and select **User Rights Assignment**.
- c. Add rights by double-clicking each right and assigning the appropriate users.

2. Define the user rights for each client that connects to the spawner. Similar to the administrator, each client that connects to the spawner must have the following user right: **log on as a batch job**.

The **log on as a batch job** user right needs to be held by every client that you want to connect into a multi-user SAS session or objspawn. The Windows routine LogonUser() requires this user right in order to authenticate the client's credentials.

3. Restart Windows to apply the new user rights.
4. Start the spawner program (called objspawn.exe) using a command that specifies the appropriate options. (You should have already created a metadata configuration file for the spawner to use to access the SAS Metadata Server.) In most cases, you should install the spawner as a service. Refer to the Spawner Invocation Options for a complete list of valid options for the command.

In the following examples, c:\sasv91 is the installed SAS folder and c:\objspawn.xml is the metadata configuration file that you created using the METACON command in SAS.

- ◆ The following command installs the spawner as a Windows NT service and updates the registry to hold the options that are specified (in this case -sasSpawnercn and -xmlConfigFile):

```
c:\program files\sas\sas 9.1\objspawn Spawner1
  -xmlconfigFile c:\objspawn.xml -install
```

When you install the spawner as a Windows NT service, you must specify the fully qualified path to the configuration file. When the spawner is started as a Windows NT service, it will self configure utilizing the options that are placed in the registry at install time.

- ◆ The following command installs the spawner as a Windows NT service, specifies service dependencies, and names the service:

```
c:\program files\sas\sas 9.1\objspawn -sasSpawnercn NameofSpawner
  -installDependencies "service1;service2"
  -name serviceName -xmlconfigfile c:\objspawn.xml
  -install
```

- ◆ The following command launches the spawner with the configuration file and specifies a log file:

```
c:\program files\sas\sas 9.1\objspawn -sasSpawnercn "Spawner 1"
  -xmlconfigFile c:\objspawn.xml -saslogfile c:\logs\objspawn.log
```

Note: After the spawner is started, a message is written to the application event log indicating whether objspawn initialization completed or failed.

Starting Servers

Starting the Spawner on UNIX

The SAS IOM server is launched in the client's home directory (as specified in the client's password entry). If the client has a directory in its home directory that is named the same as its user ID, SAS will use that directory as the SAS session's SASUSER path.

Note: If you are printing or using SAS/GRAPH procedures, you must set the DISPLAY environment variable to a running X server. For example, one of the following:

- `export DISPLAY=<machine name>:0.0`
- `-set DISPLAY=<machine name>:0.0`

Verify that `elssrv`, `sasauth`, and `sasrun` are owned by the root user and have the `setuid` bit enabled. If these settings are not enabled, the object spawner will not be able to launch SAS sessions. For instructions, see [Setting the File Permissions](#).

Start the spawner program (called `objspawn`) using a command that specifies the appropriate options. (You should have already created a [metadata configuration file](#) for the spawner to use to access the SAS Metadata Server.) Refer to the [Spawner Invocation Options](#) for a complete list of valid options for the command.

The following example uses `/sasv91/` as the directory in which SAS was installed and `objspawn.xml` as the name of the metadata configuration file that you created using the SAS Integration Technologies Configuration utility.

- The following command launches the spawner, specifying the `sasSpawner` definition to use, the log file, and the configuration file to access the SAS Metadata Server:

```
prompt> /sasv91/utilities/bin/objspawn
        -sasSpawnercn NameofSpawner -xmlConfigFile objspawn.xml
        -sasLogFile /sasv91/utilities/bin/objspawn.log
```

Note: After the spawner is started, an attempt is made to write a message to stdout indicating whether `objspawn` initialization completed or failed.

Setting the File Permissions

You can set the file permissions (enable the `setuid` bit and set the owner to **root**) for the files in `!SASROOT/utilities/bin` by using either of the following methods.

Method 1: Using SAS Setup

1. Log in to the root account.

```
$ su root
```

2. Run SAS Setup from `!SASROOT/sassetup`.
3. Select **Run Setup Utilities** from the SAS Setup Primary Menu.
4. Select **Perform SAS System Configuration**.
5. Select **Configure User Authorization**.

Method 2: Using the Command Line

From a Unix prompt, type the following:

```
$ su root
# cd !SASROOT/utilities/bin
# chown root elssrv sasauth sasperm sasrun
# chmod 4755 elssrv sasauth sasperm sasrun
# exit
```

Starting Servers

Starting a Spawner on Alpha/VMS

If the spawner is to service more than one client user ID, the spawner should run under an account that has the following privileges:

```
IMPERSONATE  NETMBX  READALL  TMPMBX
```

These privileges are required in order for the spawner to create a detached process with the connecting client as the owner.

If the spawner is to service one client, the spawner can be launched under that client's user ID.

Note: If you are printing or using SAS/GRAPH procedures, you must set the display to a machine running an X server. For example:

```
set display/create/transport=tcpip/node=  
  <ip address of machine running X server>
```

Included as part of the Base SAS installation are some sample DCL files that demonstrate how to start the daemon as a detached process. The files listed here are all located in SAS\$ROOT:[MISC.BASE]. Make a backup copy of these files before making any modifications.

OBJSPAWN_STARTUP.COM

executes OBJSPAWN.COM as a detached process.

OBJSPAWN.COM

runs the spawner. OBJSPAWN.COM also includes other commands that your site might need in order to run the appropriate version of the spawner, to set the display node, to define a process level logical pointing to a template DCL file (OBJSPAWN_TEMPLATE.COM), and perform any other actions needed before the spawner is started.

OBJSPAWN_TEMPLATE.COM

performs setup that is needed in order for the client process to execute. The spawner first checks to see if the logical SAS\$TKELS_TEMPLATE is defined. If SAS\$TKELS_TEMPLATE is defined, when the server first starts the corresponding template file is executed as a DCL command procedure. You are not required to define the template file.

OBJSPAWN_CONFIG.XML

provides a sample configuration file for the spawner.

Note: After the spawner is started, an attempt is made to write a message to stdout indicating whether objspawn initialization completed or failed.

Starting Servers

Server and Services Startup Order

To ensure proper operation of your implementation, you must start your servers and deploy the SAS Foundation Services in the appropriate order. The IOM servers and SAS Foundation Services have the following dependencies:

- The IOM servers are dependent on the SAS Metadata Server.
- The SAS Foundation Services deployment might be dependent on a service deployment configuration on the SAS Metadata Server.
- The servlet container might have a dependency on the remote SAS Foundation Services.

The following table shows the server and service dependencies:

Server	Dependency
SAS Metadata Server	none
Xythos WFS WebDAV Server	none
SAS Stored Process Server	SAS Metadata Server
SAS Workspace Server	SAS Metadata Server
SAS OLAP Server	SAS Metadata Server
SAS Foundation Services	SAS Metadata Server
Servlet Container / Application Server	SAS Metadata Server, Xythos WebDAV Server, SAS Stored Process Server, SAS Workspace Server, SAS Foundation Services deployment

Ensure that the servers are started in the following order:

1. Start the SAS Metadata Server.
2. Start the WebDAV server.
3. Depending on how you configured your SAS Workspace Server and SAS Stored Process Server, start these servers as follows:
 - ◆ If you set up one spawner to start both the SAS Workspace Server and SAS Stored Process Server, use that spawner to start both the stored process and workspace servers.
 - ◆ If you set up different spawners for your SAS Workspace Server and SAS Stored Process Server, use each spawner to start the respective servers.
4. Start the SAS OLAP Server.
5. Deploy your local and remote SAS Foundation Services. The remote SAS Foundation Services must be started and initialized before you start the servlet container.
6. Start your servlet container. If the servlet container is already running, you must restart it before you access any Web applications.

Starting Servers

Object Server Parameters

All object server parameters are applicable on the command line that starts the server:

- For servers that are started by the object spawner, the object server parameters come from your server definition in the SAS Metadata Repository. (The server definition is located under the Server Manager plug-in of SAS Management Console. In the server definition, select the Options tab to locate the **Object Server Parameters** field).
- For servers that are not spawned (such as those that are run from command scripts, those that are run as Windows services, or those that are launched by COM), you use the OBJECTSERVERPARMS SAS option to specify the object server parameters on the command line.

To simplify the command that is needed to invoke an IOM server, the server startup sequence can also connect back to the metadata server in order to fetch additional information, including object server parameters. This feature involves use of the SERVER= and METAUTOINIT object server parameters. See [Specifying Metadata Connection Information](#) for details. The object server parameters that can be obtained in this way are indicated in the table below under the column "Can Be Fetched at Server Startup." These object server parameters can be specified in the server definition in SAS Management Console. In the server definition, select the Options tab to locate the **Object Server Parameters** field.

Important Note:

You can fetch object server parameters from metadata as follows:

- **When you start the server with a script**, some object server parameters cannot be obtained from the metadata. These parameters are designated as "No" in the "Can Be Fetched at Server Startup" column. Do not enter these object server parameters in your metadata.
- **When you start the server with a spawner**, all object server parameters can be obtained from the metadata (even those that are designated as "No" in the "Can Be Fetched at Server Startup" column).

Note: Object server parameters that are specified on the command line always override object server parameters obtained from a SAS Metadata Repository.

Object Server Parameter	Value	Connection Type	Can Be Fetched at Server Startup (When Starting a Server With a Script)	Definition
ANONYMOUSLOGINPOLICY	Deny Restrict	IOM Bridge	Yes	Specifies whether the server permits any access at all to connections that do not supply a user ID (in programming terms, ones that supply a zero-length user ID).

				<p>If you specify "restrict," then the server allows connections that do not have a user ID; however, the client only has restricted access to the IServerStatus interface (used primarily for querying basic server status).</p> <p>If you specify "deny," then the server completely disallows connections that do not provide a user ID. The default is "restrict." For details about ANONYMOUSLOGINPOLICY, see Setting Up Additional Server Security in the Security chapter.</p>
APPLEVEL	0, 1, 2, 3	IOM Bridge COM/DCOM	Yes	<p>Specifies the detail level of the trace that is written by the server application (such as the OLAP server, the SAS Metadata Server or the SAS Stored Process Server). The default value if APPLEVEL is omitted (1) enables logging at a level that is suitable for a production server; therefore, this parameter is optional.</p> <p>APPLEVEL=0 disables the application's logging and is discouraged because it suppresses useful diagnostic information. Higher APPLEVEL values can invoke additional tracing. The SAS Metadata Server, for example, defines additional logging levels. For details, see Enabling Logging of Authentication Events and SAS Metadata Server Logging Overview in the <i>SAS Metadata Server: Setup Guide</i>.</p>
CLASSFACTORY Alias: CLSID	36 character class identifier	IOM Bridge COM/DCOM	Yes	<p>Specifies the class ID number, which specifies the type of server to instantiate (for example, 2887E7D7-4780-11D4-879F-00C04F38F0DB specifies a SAS Metadata Server). An IOM server exposes one top-level class through its class identifier.</p> <p>By default, an IOM server hosts the Workspace class. If you want to specify an alternate class to expose as the top-level class, use the classfactory option to identify the class to IOM.</p> <p>When using the SERVER= objectserverparms suboption, the classfactory does not need to be specified because it is obtained from the logical server definition in the SAS Metadata Repository.</p> <p>This option is primarily used to start the SAS Metadata Server.</p>

CLIENTENCRYPTIONLEVEL Alias: CEL	none credentials everything	IOM Bridge	No	Specifies the degree of encryption to use when making outbound calls. This option is used only by the bridge protocol engine.
IOMLEVEL	0, 1, 2, or 3	IOM Bridge COM/DCOM	Yes	Specifies trace level for protocol-independent IOM events, particularly calls and the SAS LOG of workspaces. The default is 0. If IOMLEVEL is set to 1, then the calls that enter and leave the server are traced. This feature can be very helpful for identifying whether a problem arose in a client or in the server. Using IOMLEVEL=1 with the SAS Metadata Server will capture the input and output XML strings for metadata requests. For more information, see Capturing XML in the Log and SAS Metadata Server Logging Overview in the <i>SAS Metadata Server: Setup Guide</i> . For performance reasons, it is recommended that IOMLEVEL=1 be used only when diagnosing problems. Higher values of IOMLEVEL produce traces that are intended only for use by SAS Technical Support. Depending on the calls that are being traced, the JNLSTRMAX and JNLLINEMAX values may need to be increased to prevent truncation of long strings and long lines.
JNLARRELM	Numeric Value	IOM Bridge COM/DCOM	Yes	Specifies the maximum number of array elements to print out when an IOM array value is traced.
JNLLINEMAX	Numeric Value	IOM Bridge COM/DCOM	Yes	Specifies the maximum length of a line printed in the IOM server journal.
JNLSTRMAX	Numeric Value	IOM Bridge COM/DCOM	Yes	Specifies the maximum length of string printed in the IOM server journal. This option can be used to adjust the amount of material included in an IOM trace. A value greater than 500 is recommended.
LOGFILE Alias: LOG	Path in which to create the IOM server trace log	IOM Bridge COM/DCOM	Yes	Specifies an alternative file for the SAS log for IOM server trace output. Note: Using this option on a spawned server can prevent multiple servers from running simultaneously because they will all try to open the same log file. It is therefore recommended that this option be used only for specific diagnostic tasks. Note: The user who starts the server must have execute and write permissions for the log

				destination path.
METAAUTOINIT NOMETAAUTOINIT	N/A	IOM Bridge COM/DCOM	Yes	Specifies whether the IOM server should connect back to the SAS Metadata Server during startup in order to obtain additional configuration information such as object server parameters and pre-assigned libraries. When METAAUTOINIT is specified, the server uses the provided META* options to connect to the SAS Metadata Server. With NOMETAAUTOINIT, IOM server startup does not connect back to the SAS Metadata Server. The default depends on the type of server. For further details, see Server Startup Command . This option is applicable only if you have specified your logical server with the SERVER= object server parameter.
PELEVEL	0, 1, 2, or 3	IOM Bridge	No	Specifies trace protocol engine logic and packets. Level 3 specifies the most verbose output. The default is 0.
PORT	TCP/IP port number	IOM Bridge	Yes	Specifies the value for the bridge protocol engine to use as the port to start listening for client connections. Do not specify this option with spawned servers; it will be supplied automatically by the spawner.
PROTOCOL	bridge com (com,bridge)	IOM Bridge COM/DCOM	Yes	Specifies the protocol engines to launch in server mode. Server mode indicates that the protocol engines will listen for client connections. By default, Windows servers use the COM protocol engine and all other servers use the Bridge protocol engine. If you specify (com, bridge) then a multi-user server can simultaneously support clients using different protocols. Do not specify this option with spawned servers; it will be supplied automatically by the spawner.
SECURITY NOSECURITY	N/A	IOM Bridge COM/DCOM	No	Specifies whether client authentication is enabled. By default (SECURITY), clients must be authenticated; one exception is the use of ANONYMOUSLOGINPOLICY for public interfaces (see Setting Up Additional Server Security). When security is enabled, the bridge protocol engine requires a user name and password; the COM protocol engine is integrated with the single-signon security of the Windows networking environment. Authorization decisions are controlled by the server application. If NOSECURITY is specified, these security mechanisms are bypassed.

SERVER	Logical server name or OMSOBJ URI (object ID)	IOM Bridge COM/DCOM	No	Specifies the logical server name for the IOM run-time and server application to use to locate configuration information in a SAS Metadata Repository. The SERVER= option can be used to retrieve many of the OBJECTSERVERPARMS options (including PORT, PROTOCOL and CLASSFACTORY) from a SAS Metadata Repository. For details, see Specifying Metadata Connection Information .
SERVICE	TCP service name	IOM Bridge	Yes	Specifies the TCP service name (for example, from /etc/services on a UNIX system) for the port that the IOM Bridge protocol engine will use to listen for connections from clients. Do not specify this option with spawned servers; it will be supplied automatically by the spawner.
TRUSTSASPEER Alias: TSASPEER	N/A	IOM Bridge	Yes	Enables SAS peer sessions from IOM servers to connect as trusted peer sessions. For details, see Implementing Trusted Authentication Mechanisms .
V8ERRORTEXT	N/A	IOM Bridge COM/DCOM	Yes	Indicates that the MVA components should return Version 8 style error messages instead of the Version 9 XML style error messages.

Starting Servers

Spawner Invocation Options

The following options can be used in the command to start up the spawner for a server with an IOM Bridge connection. Note that the spawner must be stopped and restarted in order to reflect configuration updates.

-allowxcmd

enables host commands and PIPE commands for all servers that are started by the spawner. By default, the spawner starts all servers with the `-NOXCMD` SAS system option. When you specify `-allowxcmd`, the spawner no longer specifies `-NOXCMD` when launching server sessions.

Caution: When you specify `-allowxcmd`, clients can use host commands to perform potentially harmful operations such as file deletion.

-authproviderdomain

because the spawner starts either a SAS Workspace Server or SAS Stored Process server, and workspace and stored process servers only authenticate against the host, the `-authproviderdomain` option can only be used to associate a domain with the host authentication provider. For example,

```
authproviderdomain (hostuser:Raleigh)
```

Note: The spawner always authenticates against the host environment.

The `-authproviderdomain` option has the following syntax:

```
authproviderdomain (provider:<domain>)
```

For more details about the `-authproviderdomain` option, see [Specifying Default Host Domains When Starting Servers That Only Use Host Authentication](#)

This option can be abbreviated as `-authpd`.

-conversationPort

specifies which port is used for communication between the spawner and the servers that the spawner launches. This option can be abbreviated as `-cp`.

-deinstall

Windows only. Instructs the spawner to deinstall as a Windows service. This option can be abbreviated as `-di`.

Note: If you specified a service name when you installed the spawner service, you must specify the same name when you deinstall the service.

-dnsName

specifies which IP stack is used for communication between the spawner and the servers that the spawner launches. This option can be abbreviated as `-dns`.

-install

Windows only. Instructs the spawner to install as a Windows service. This option can be abbreviated as `-i`. When asked to install as a service, the spawner records all options specified at install time in the registry under the following key:

```
"SYSTEM\CurrentControlSet\Services\service-name\Parameters"
```

You can also specify options in the Startup Parameters when you manually start the spawner service from the Services dialog box.

-installDependencies

Windows only. Specifies the Windows services that must be started before the spawner service starts. The *-installDependencies* option has the following syntax:

```
-installDependencies "service1<service2><service3>"
```

This option can be abbreviated as *-idep*.

-noSecurity

instructs the spawner not to authenticate clients. Clients will execute as the user that launched the spawner. This option is useful during development.

WARNING: Because clients connected to the *-noSecurity* spawner execute as the user that launched the spawner, it is strongly suggested that the host in which the spawner is executing not be connected to a network. Otherwise, data that is accessible by the user that launched the spawner is at risk.

Note: If you use the *-noSecurity* option, the *-install* option is ignored.

-name

Windows only. Specifies a service name to use when installing the spawner as a service. The default value is SAS Object Spawner Daemon II.

If you specify a service name that contains embedded blank spaces, you must enclose the name in quotation marks (" ").

Note: If you install more than one spawner as a service on the same machine, you must use the *-name* option to give each spawner service a unique name.

-sasLogFile

specifies a fully qualified path to the file in which to log spawner activity. Enclose paths with embedded blank spaces in quotation marks. On z/OS, specify filenames similar to UNIX file paths due to the requirement for z/OS Unix System Services. This option can be abbreviated as *-slf*.

Note: If you specify a log destination in the configuration metadata rather than the startup command, you might miss some messages that are generated before the log destination is set.

-sasSpawnerCn

specifies the name (used in the SAS Management Console configuration) of the spawner object to utilize for this spawner invocation configuration. If you do not specify *-sasSpawnerCn*, the object spawner uses the first spawner definition (on the metadata server) with the same machine name as the current host.

Note: If none of the spawner definitions contain a host name of the current host, you must specify the *-sasSpawnercn* option to designate which spawner definition to use.

If you specify a spawner name that contains embedded blank spaces, you must enclose the name in quotation marks (" "). This option can be abbreviated as *-ssc*.

-sasVerbose

when present, causes the spawner to record more detail in the log file (*sasLogFile*). This option can be abbreviated as *-sv*.

-servPass

Windows only. Specifies a password for the user name specified in the *-servUser* option. This option can be abbreviated as *-sp*.

-servUser

Windows only. Specifies a user name that the service will run under, when you also specify the `-install` option. This option can be abbreviated as `-su`.

-xmlConfigFile

specifies a fully qualified path to a metadata configuration file containing a SAS Metadata Server definition to connect to for the complete configuration. On Windows, enclose paths with embedded blank spaces in double quotation marks. On z/OS, specify filenames similar to UNIX file paths due to the requirement for z/OS Unix System Services.

For details about generating a metadata configuration file for the SAS Metadata Server, see [Metadata Configuration File](#).

This option can be abbreviated as `-xcf`.

Moving Servers

Moving Servers

After your initial installation, you might be required to move one or more of these servers to a different machine. Before you move servers to machines with different operating systems, be sure that you understand and have planned for your authentication domain(s). In addition, when you move servers to a new machine, you must update any permission statements for the servers in the appropriate applications' policy files.

For details about moving server locations, see the following sections:

- [Moving the SAS Stored Process Server](#)
- [Moving the SAS Workspace Server](#)
- [Moving the SAS OLAP Server](#)
- [Moving the SAS Stored Process Server and SAS Workspace Server to the Same New Machine](#)
- [Moving the SAS Stored Process Server and SAS Workspace Server to Separate Machines](#)

Moving Servers

Moving the SAS Stored Process Server

If you have used the project installation or basic installation to install the SAS Stored Process Server on a separate machine from the SAS Workspace Server, you can use the instructions in this section to move the server to a new machine.

Important Note: In addition to changing the machine name (and optionally, port number), if you move a server to a machine with a different operating system or to a machine with an operating system other than Windows, you might need to reconfigure the following:

- **accounts for authentication.** You might need to define accounts on the authentication provider for the new server machine.
- **metadata on the SAS Metadata Server.** The following metadata definitions might require reconfiguration or additional configuration:
 - ◆ **server definition.** On the server definition, you might need to use the Server Manager plug-in to SAS Management Console to change the following parameters:
 - ◇ **SAS startup command.** You might need to change the startup command for the new operating system.
 - ◇ **authentication domain.** When you move a server, you might need to set up an additional authentication domain.
 - ◆ **login definitions.** For the login definitions that access the server and the login definitions that are used in the load-balancing configuration (e.g. in the project installation, the SAS Guest user's login), you might need to use the User Manager plug-in to SAS Management Console to do one or more of the following:
 - ◇ define a new login definition. When you move a server, you might need to create a new login definition for the new authentication domain.
 - ◇ define a new login definition for a different authentication process. When you move a server, you might need to create a new login definition with credentials to access a server in a different operating system within the default authentication domain.
 - ◇ change the format of the user ID in the login definition. When you move a server, you might need to change the fully-qualified user ID for any login credentials used to access that server.
 - ◆ **stored process definitions.** You might need to use the Stored Process Manager plug-in to SAS Management Console to specify a new location for your stored process repository.
- **spawner startup command.** If you change operating systems when you move machines, you might need to change the spawner startup command. For details, see [Invoking \(Starting\) the Object Spawner](#).

To move the SAS Stored Process Server to a different machine, follow these steps:

1. Use SAS Management Console to reconfigure the server definition for the new machine:
 - a. Open SAS Management Console and connect to a metadata repository.
 - b. In the SAS Management Console navigation tree, locate and select the server object (e.g., Main – Stored Process Server) that you want to modify.
 - c. In the Display area, for each server connection, follow these steps:
 - i. Select and right-click the connection definition, then select **Properties** from the pop-up menu.
 - ii. Select the Options tab.
 - iii. Change the **Host Name** to the host name of the new machine for your server.
 - iv. If you are changing the port, change the **Port** to the new port for your server.

SAS® 9.1.2 Integration Technologies: Server Administrator's Guide

- v. If you need to change the authentication domain, click **New** and define the new **Authentication Domain** for your server.
 - vi. Click **OK**.
 - d. If you need to change the server startup command, select and right-click the server object, then select **Properties** from the pop-up menu. Select the Options tab and change the **Command** field.
 - e. Click **OK** to save the new configuration to the metadata repository.
 2. Use SAS Management Console to reconfigure the spawner definition for the new machine:
 - a. In the SAS Management Console navigation tree, locate the spawner object, then right-click the spawner definition, and select **Properties** from the pop-up menu.
 - b. If your spawner name contains the machine name, change the **Name** field to specify the name of the new machine.
 - c. Select the Options tab.
 - d. Change the **Associated Machine** to the host name of the new machine for your server.
 - e. If you have any other servers associated with the spawner, select the Servers tab. In the **Selected servers** list box, select the other servers and move them to the **Available servers** list box. Click **OK**. You must then define a new spawner for these servers.
 - f. If you are changing the port of either the operator connection or load-balancing connection, in the Display area, select the connection, then right-click and select **Properties** from the pop-up menu. Select the Options tab and change the **Port** to the new port for your spawner connection.

If you changed the server's authentication domain, select the same new **Authentication Domain** for your spawner.

 - g. Click **OK** to save the new configuration to the metadata repository.
 3. Edit the policy files for any applications that need to access the new server machine. (For details about editing policy files for the portal Web application and its components, see [Adding Permissions for Servers to Policy Files](#) in the *SAS Web Infrastructure Kit Administrator's Guide*).
 4. Install SAS 9.1 or higher and SAS Integration Technologies on the new machine.
 5. Copy your metadata configuration file (XML file) and spawner startup script from your spawner configuration directory to the same directory on the new machine. If necessary, change the spawner startup script for the new machine. For details, see [Invoking \(Starting\) the Object Spawner](#).
 6. Create a directory for stored process server log files. The recommended directory name is STPDemo, and the recommended location is the server home location that you specified when you ran the install program (for example, C:\Program Files\SAS\Servers\STPDemo.)

If you do not remember the server home location, see the \$STP_HOME\$ property in the `install.properties` file.

7. Ensure that the multi-user login (specified in the **Advanced Options** for the Stored Process Server definition, e.g. in the project installation, the login owned by the SAS General Server group) can authenticate against the host authentication provider for the SAS Stored Process Server's machine. On Windows NT and Windows 2000, give this account the Act as part of the Operating System user right.
 8. Give the shared account for the multi-user login (e.g. in the project installation, the SAS General Server group login) "Write" permission to the stored process log directory.
 9. Ensure that the spawner invoker (e.g., in the project installation, the SAS user) can authenticate against the host authentication provider for the machine.
10. Ensure that users who need to access the server are defined for the machine's host authentication provider.

When you are finished modifying the server and spawner definitions, do the following:

- if you have added a new authentication domain for the machine, do both of the following:

- ◆ use the User Manager plug-in to SAS Management Console to add a login definition for access to the server.
- ◆ use the User Manager plug-in to SAS Management Console to modify the login definition for the multi-user login (e.g., in the project installation, the SAS General Server group login). Modify the login definition to specify the new authentication, and, if required, the new user ID credentials.
- if you have changed operating systems and need to modify user credentials, use the User Manager plug-in to SAS Management Console to modify user and group login definition for the new user ID credentials of the new machine.
- if you need to move the stored process repositories to a different directory, use the Stored Process Manager plug-in to SAS Management Console to modify the stored process definition and change the **Source Repository** field on the Execution tab of the stored process definition.
- if your stored process definitions reference content on the old stored process or workspace server machine, you must add the content to the directory you defined in the stored process definition.

Moving Servers

Moving the SAS Workspace Server

If you have used the project installation or basic installation to install the SAS Workspace server on a machine separate from the SAS Stored Process Server, you can use the instructions in this section to move the server to a new machine.

Important Note: In addition to changing the machine name (and optionally, port number), if you move a server to a machine with a different operating system or to a machine with an operating system other than Windows, you might need to reconfigure the following:

- **accounts for authentication.** You might need to define accounts on the authentication provider for the new server machine.
- **metadata on the SAS Metadata Server.** The following metadata definitions might require re-configuration or additional configuration:
 - ◆ **server definition.** On the server definition, you might need to use the Server Manager plug-in to SAS Management Console to change the following parameters:
 - ◇ **SAS startup command.** You might need to change the startup command for the new operating system.
 - ◇ **authentication domain.** When you move a server, you might need to set up an additional authentication domain.
 - ◆ **login definitions.** For the login definitions that access the server and the login definitions that are used in the load-balancing configuration (e.g. in the project installation, the SAS Guest user's login), you might need to use the User Manager plug-in to SAS Management Console to do one or more of the following:
 - ◇ define a new login definition. When you move a server, you might need to create a new login definition for the new authentication domain.
 - ◇ define a new login definition for a different authentication process. When you move a server, you might need to create a new login definition with credentials to access a server in a different operating system within the default authentication domain.
 - ◇ change the format of the user ID in the login definition. When you move a server, you might need to change the fully-qualified user ID for any login credentials used to access that server.
 - ◆ **stored process definitions.** You might need to use the Stored Process Manager plug-in to SAS Management Console to specify a new location for your stored process repository.
- **spawner startup command.** If you change operating systems when you move machines, you might need to change the spawner startup command. For details, see [Invoking \(Starting\) the Object Spawner](#).

To move the SAS Workspace server to a different machine, follow these steps:

1. Use SAS Management Console to reconfigure the SAS Workspace Server definition for the new machine:
 - a. Open SAS Management Console and connect to a metadata repository.
 - b. In the SAS Management Console navigation tree, locate and select the server object (e.g., Main – Workspace Server) that you want to modify.
 - c. In the Display area, for each server connection, follow these steps:
 - i. Select and right-click the connection definition, then select **Properties** from the pop-up menu.
 - ii. Select the Options tab.
 - iii. Change the **Host Name** to the host name of the new machine for your server.
 - iv. If you are changing the port, change the **Port** to the port of the new port for your server.

SAS® 9.1.2 Integration Technologies: Server Administrator's Guide

- v. If you need to change the authentication domain, click **New** and define the new **Authentication Domain** for your server.
 - vi. Click **OK**.
 - d. If you need to change the server startup command, select and right-click the server object, then select **Properties** from the pop-up menu. Select the Options tab and change the **Command** field.
 - e. Click **OK** to save the new configuration to the metadata repository.
 2. Use SAS Management Console to reconfigure the spawner definition for the new machine:
 - a. In the SAS Management Console navigation tree, locate and select the spawner definition, then right-click and select **Properties** from the pop-up menu.
 - b. If your spawner name contains the machine name, change the **Name** field to specify the name of the new machine.
 - c. Select the Options tab.
 - d. Change the **Host Name** to the host name of the new machine for your server.
 - e. Click **OK**.
 - f. If you have any other servers associated with the spawner, select the Servers tab. In the **Selected servers** list box, select the other servers and move them to the **Available servers** list box. Click **OK**. You must then define a new spawner for these servers.
 - g. If you are changing the port of either the operator connection or load-balancing connection, in the Display area, select the connection, then right-click and select **Properties** from the pop-up menu. Select the Options tab and change the **Port** to the new port for your spawner connection.

If you changed the server's authentication domain, select the same new **Authentication Domain** for your spawner.

 - h. Click **OK** to save the new configuration to the metadata repository.
 3. Edit the policy files for any applications that need to access the new server machine. (For details about editing policy files for the portal Web application and its components, see [Adding Permissions for Servers to Policy Files](#) in the *SAS Web Infrastructure Kit Administrator's Guide*).
 4. Install SAS 9.1 or higher and SAS Integration Technologies on the new machine.
 5. Copy your metadata configuration file (XML file) and spawner startup script from your spawner configuration directory to the same directory on the new machine. If necessary, change the spawner startup script for the new machine. For details, see [Invoking \(Starting\) the Object Spawner](#).
 6. Ensure that the spawner invoker (e.g. in the project installation, the SAS user) can authenticate against the host authentication provider for the machine.
 7. Ensure that users who need to access the server are defined for the machine's host authentication provider.

When you are finished modifying the server and spawner definitions:

- if you have added a new authentication domain for the machine, use the User Manager plug-in to SAS Management Console to add a login definition for access to the server.
- if you have changed operating systems and need to modify user credentials, use the User Manager plug-in to SAS Management Console to modify user and group login definition for the new user ID credentials of the new machine.
- if you need to move the stored process repositories to a different directory, use the Stored Process Manager plug-in to SAS Management Console to modify the stored process definition and change the **Source Repository** field on the Execution tab of the stored process definition.
- if your stored process definitions reference content on the old stored process or workspace server machine, you must add the content to the directory you defined in the stored process definition.

Moving Servers

Moving the SAS OLAP Server

If you have used the project installation or basic installation to install the SAS OLAP Server, you can use the instructions in this section to move the server to a new machine.

Important Note: In addition to changing the machine name (and optionally, the port number), if you move a server to a machine with a different operating system or to a machine with an operating system other than Windows, you might need to reconfigure the following:

- **accounts for authentication.** You might need to define accounts on the authentication provider for the new server machine.
- **metadata on the SAS Metadata Server.** The following metadata definitions might require reconfiguration or additional configuration:
 - ◆ **server definition.** On the server definition, you might need to use the Server Manager plug-in to SAS Management Console to change the following parameters:
 - ◇ **SAS startup command.** You might need to change the startup command for the new operating system.
 - ◇ **authentication domain.** When you move a server, you might need to set up an additional authentication domain.
 - ◆ **login definitions.** For the login definitions that access the server and the login definitions that are used in the load-balancing configuration (e.g., in the project installation, the SAS Guest user's login), you might need to use the User Manager plug-in to SAS Management Console to do one or more of the following:
 - ◇ define a new login definition. When you move a server, you might need to create a new login definition for the new authentication domain.
 - ◇ define a new login definition for a different authentication process. When you move a server, you might need to create a new login definition with credentials to access a server in a different operating system within the default authentication domain.
 - ◇ change the format of the user ID in the login definition. When you move a server, you might need to change the fully-qualified user ID for any login credentials used to access that server.
- **server startup command.** You might need to change the server startup command depending on the operating system where you move the server configuration.

To move the SAS OLAP Server to a different machine, follow these steps:

1. Use SAS Management Console to reconfigure the SAS OLAP Server definition for the new machine:
 - a. Open SAS Management Console and connect to a metadata repository.
 - b. In the SAS Management Console navigation tree, locate and select the server object (Main – OLAP Server) that you want to modify.
 - c. In the Display area, for each server connection, follow these steps:
 - i. Select and right-click the connection definition, then select **Properties** from the pop-up menu.
 - ii. Select the Options tab.
 - iii. Change the **Host Name** to the host name of the new machine for your server.
 - iv. If you are changing the port, change the **Port** to the port of the new port for your server.
 - v. If you need to change the authentication domain, click **New** and define the new **Authentication Domain** for your server.
 - vi. Click **OK**.

- d. If you need to change the server startup command, select and right-click the server object, then select **Properties** from the pop-up menu. Select the Options tab and change the **Command** field.
- e. Click **OK** to save the new configuration to the metadata repository.
2. Edit the policy files for any applications that need to access the new server machine. (For details about editing policy files for the portal Web application and its components, see [Adding Permissions for Servers to Policy Files](#) in the *SAS Web Infrastructure Kit Administrator's Guide*).
3. On the new machine, install SAS 9.1 or higher and SAS Integration Technologies.
4. Depending on how your server startup is configured, do the following:
 - ◆ If you have a SAS OLAP Server startup script, copy your SAS OLAP Server startup script to the same directory on the new machine and modify as appropriate.
 - ◆ If the SAS OLAP Server is configured to start as a service, configure the new machine to start the SAS OLAP server as a service.
 - ◆ If you are not starting the server as a service, ensure that the user who starts the server (e.g., in the project installation, the SAS user) can authenticate against the host authentication provider for each machine.
 - ◆ Ensure that users who need to access the server are defined for the appropriate authentication provider.

When you are finished modifying the server and spawner definitions, do the following:

- if you have added a new authentication domain for the machine, use the User Manager plug-in to SAS Management Console to add a login definition for access to the server.
- if you have changed operating systems and need to modify user credentials, use the User Manager plug-in to SAS Management Console to modify user and group login definition for the new user ID credentials of the new machine.

Moving Servers

Moving Both the SAS Stored Process Server and SAS Workspace Server to the Same New Machine

If you have used the project installation or basic installation to install the SAS Stored Process Server and SAS Workspace Server on the same machine, you can use the instructions in this section to move the servers to another machine.

Important Note: In addition to changing the machine name (and optionally, the port number), if you move a server to a machine with a different operating system or to a machine with an operating system other than Windows, you might need to reconfigure the following:

- **accounts for authentication.** You might need to define accounts on the authentication provider for the new server machine.
- **metadata on the SAS Metadata Server.** The following metadata definitions might require reconfiguration, or additional configuration:
 - ◆ **server definition.** On the server definition, you might need to use the Server Manager plug-in to SAS Management Console to change the following parameters:
 - ◇ **SAS startup command.** You might need to change the startup command for the new operating system.
 - ◇ **authentication domain.** When you move a server, you might need to set up an additional authentication domain.
 - ◆ **login definitions.** For the login definitions that access the server and the login definitions that are used in the load-balancing configuration (e.g., the SAS Guest user's login), you might need to use the User Manager plug-in of the SAS Management Console to do one or more of the following:
 - ◇ define a new login definition. When you move a server, you might need to create a new login definition for the new authentication domain.
 - ◇ define a new login definition for a different authentication process. When you move a server, you might need to create a new login definition with credentials to access a server in a different operating system within the default authentication domain.
 - ◇ change the format of the user ID in the login definition. When you move a server, you might need to change the fully-qualified user ID for any login credentials used to access that server.
 - ◆ **stored process definitions.** You might need to use the Stored Process Manager plug-in to SAS Management Console to specify a new location for your stored process repository.
- **spawner startup command.** If you change operating systems when you move machines, you might need to change the spawner startup command. For details, see [Invoking \(Starting\) the Object Spawner](#).

To move both the SAS Stored Process Server and SAS Workspace Server to a new machine, follow these steps:

1. Use SAS Management Console to reconfigure the SAS Workspace Server definition for the new machine:
 - a. Open SAS Management Console and connect to a metadata repository.
 - b. In the SAS Management Console navigation tree, locate and select the server object (e.g., Main – Workspace Server) that you want to modify.
 - c. In the Display area, for each server connection, follow these steps:
 - i. Select and right-click the connection definition, then select **Properties** from the pop-up menu.
 - ii. Select the Options tab.
 - iii. Change the **Host Name** to the host name of the new machine for your server.

SAS® 9.1.2 Integration Technologies: Server Administrator's Guide

- iv. If you are changing the port, change the **Port** to the port of the new port for your server.
 - v. If you need to change the authentication domain, click **New** and define the new **Authentication Domain** for your server.
 - vi. Click **OK**.
 - d. If you need to change the server startup command, select and right-click the server object, then select **Properties** from the pop-up menu. Select the Options tab and change the **Command** field.
 - e. Click **OK** to save the new configuration to the metadata repository.
2. Use SAS Management Console to reconfigure the SAS Stored Process Server definition for the new machine:
 - a. Open SAS Management Console and connect to a metadata repository.
 - b. In the SAS Management Console navigation tree, locate and select the server object (e.g., Main – Stored Process Server) that you want to modify.
 - c. In the Display area, for each server connection, follow these steps:
 - i. Select and right-click the connection definition, then select **Properties** from the pop-up menu.
 - ii. Select the Options tab.
 - iii. Change the **Host Name** to the host name of the new machine for your server.
 - iv. If you are changing the port, change the **Port** to the port of the new port for your server.
 - v. If you need to change the authentication domain, click **New** and define the new **Authentication Domain** for your server.
 - vi. Click **OK**.
 - d. If you need to change the server startup command, select and right-click the server object, then select **Properties** from the pop-up menu. Select the Options tab and change the **Command** field.
 - e. Click **OK** to save the new configuration to the metadata repository.
3. Use SAS Management Console to reconfigure the spawner definition for the new machine:
 - a. In the SAS Management Console navigation tree, locate and select the spawner definition, then right-click and select **Properties** from the pop-up menu.
 - b. If your spawner name contains the machine name, change the **Name** field to specify the name of the new machine.
 - c. Select the Options tab.
 - d. Click **OK**.
 - e. Change the **Associated Machine** to the host name of the new machine for your server.
 - f. Click **OK**.
 - g. If you have any other servers associated with the spawner, select the Servers tab. In the **Selected servers** list box, select the other servers and move them to the **Available servers** list box. Click **OK**. You must then define a new spawner for these servers.
 - h. If you are changing the port of either the operator connection or load-balancing connection, in the Display area, select the connection, then right-click and select **Properties** from the pop-up menu. Select the Options tab and change the **Port** to the new port for your spawner connection.

If you changed the server's authentication domain, select the same new **Authentication Domain** for your spawner.

 - i. Click **OK** to save the new configuration to the metadata repository.
4. Edit the policy files for any applications that need to access the new server machine. (For details about editing policy files for the portal Web application and its components, see [Adding Permissions for Servers to Policy Files](#) in the *SAS Web Infrastructure Kit Administrator's Guide*).
5. On the new machine, install SAS 9.1 or higher and SAS Integration Technologies
6. Copy your metadata configuration file (XML file) and spawner startup script from your spawner configuration directory to the same directory on the new machine. If necessary, change the spawner startup script for the

new machine. For details, see [Invoking \(Starting\) the Object Spawner](#).

7. Create a directory for stored process server log files. The recommended directory name is STPDemo, and the recommended location is the server home location that you specified when you ran the install program (for example, C:\Program Files\SAS\Servers\STPDemo).

If you do not remember the server home location, see the \$STP_HOME\$ property in the `install.properties` file.

8. Ensure that the multi-user login (specified in the **Advanced Options** for the SAS Stored Process Server definition (e.g., in the project installation, the login owned by the SAS General Server group)) can authenticate against the host authentication provider for the SAS Stored Process Server's machine. On Windows NT and Windows 2000, give this account the Act as part of the Operating System user right.
9. Give the shared account for the multi-user login (e.g., in the project installation, the SAS General Server group login) "Write" permission to the stored process log directory.
10. Ensure that the spawner invoker (e.g., in the project installation, the SAS user) can authenticate against the host authentication provider for each machine.
11. Ensure that users who need to access the server are defined for the machine's host authentication provider.

When you are finished modifying the server and spawner definitions:

- if you have added a new authentication domain for the machine, do both of the following:
 - ◆ use the User Manager plug-in to SAS Management Console to add a login definition for access to the server.
 - ◆ use the User Manager plug-in to SAS Management Console to modify the login definition for the multi-user login (e.g., in the project installation, the SAS General Server group login). Modify the login definition to specify the new authentication, and, if required, the new user ID credentials.
- if you have changed operating systems and need to modify user credentials, use the User Manager plug-in to SAS Management Console to modify user and group login definition for the new user ID credentials of the new machine.
- if you need to move the stored process repositories to a different directory, use the Stored Process Manager plug-in to SAS Management Console to modify the stored process definition and change the **Source Repository** field on the Execution tab of the stored process definition.
- if your stored process definitions reference content on the old stored process or workspace server machine, you must add the content to the directory you defined in the stored process definition.

Moving Servers

Moving the SAS Stored Process Server and SAS Workspace Server to Separate Machines

If you have used the project installation or basic installation to install the SAS Stored Process Server and SAS Workspace Server on the same machine, you can use the instructions in this section to move the servers to new (and separate) machines.

Important Note: In addition to changing the machine name (and optionally, the port number), if you move a server to a machine with a different operating system or to a machine with an operating system other than Windows, you might need to reconfigure the following:

- **accounts for authentication.** You might need to define accounts on the authentication provider for the new server machine.
- **metadata on the SAS Metadata Server.** The following metadata definitions might require reconfiguration or additional configuration:
 - ◆ **server definition.** On the server definition, you might need to use the Server Manager plug-in to SAS Management Console to change the following parameters:
 - ◇ **SAS startup command.** You might need to change the startup command for the new operating system.
 - ◇ **authentication domain.** When you move a server, you might need to set up an additional authentication domain.
 - ◆ **login definitions.** For the login definitions that access the server and the login definitions that are used in the load-balancing configuration (e.g., in the project installation, the SAS Guest user's login), you might need to use the User Manager plug-in to SAS Management Console to do one or more of the following:
 - ◇ define a new login definition. When you move a server, you might need to create a new login definition for the new authentication domain.
 - ◇ define a new login definition for a different authentication process. When you move a server, you might need to create a new login definition with credentials to access a server in a different operating system within the default authentication domain.
 - ◇ change the format of the user ID in the login definition. When you move a server, you might need to change the fully-qualified user ID for any login credentials used to access that server.
 - ◆ **stored process definitions.** You might need to use the Stored Process Manager plug-in of the SAS Management Console to specify a new location for your stored process repository.
- **spawner startup command.** If you change operating systems when you move machines, you might need to change the spawner startup command. For details, see [Invoking \(Starting\) the Object Spawner](#).

To move both the SAS Stored Process Server and the SAS Workspace Server to separate machines, follow these steps:

1. Use SAS Management Console to reconfigure the SAS Stored Process Server definition for the new machine:
 - a. Open SAS Management Console and connect to a metadata repository.
 - b. In the SAS Management Console navigation tree, locate and select the server object that you want to modify.
 - c. In the Display area, for each server connection, follow these steps:
 - i. Select and right-click the connection definition, then select **Properties** from the pop-up menu.

- ii. Select the Options tab.
 - iii. Change the **Host Name** to the host name of the new machine for your server.
 - iv. If you are changing the port, change the **Port** to the port of the new port for your server.
 - v. If you need to change the authentication domain, click **New** and define the new **Authentication Domain** for your server.
 - vi. Click **OK**.
 - d. If you need to change the server startup command, select and right-click the server object, then select **Properties** from the pop-up menu. Select the Options tab and change the **Command** field.
 - e. Click **OK** to save the new configuration to the metadata repository.
2. Use SAS Management Console to reconfigure the spawner definition for the new SAS Stored Process Server's machine:

- a. Locate and select the spawner definition, then right-click and select **Properties** from the pop-up menu.
- b. If your spawner name contains the machine name, change the **Name** field to specify the name of the new machine.
- c. Select the Options tab.
- d. Change the **Associated Machine** to the host name of the new machine for your server.
- e. Click **OK**.
- f. Select the Servers tab.
- g. In the **Selected servers** list box, select the SAS Workspace Server named Main – Workspace server and move it the **Available servers** list box.

If you have any other servers associated with the spawner, select the other servers and move them to the **Available servers** list box. You must then define a new spawner for these servers.

- h. Click **OK**.
- i. If you are changing the port of either the operator connection or load-balancing connection, in the Display area, select the connection, then right-click and select **Properties** from the pop-up menu. Select the Options tab and change the **Port** to the new port for your spawner connection.

If you changed the server's authentication domain, select the same new **Authentication Domain** for your spawner.

- j. Click **OK** to save the new configuration to the metadata repository.
3. Use SAS Management Console to reconfigure the SAS Workspace Server definition for the new machine:

- a. Open SAS Management Console and connect to a metadata repository.
- b. In the SAS Management Console navigation tree, locate and select the server object that you want to modify.
- c. In the Display area, for each server connection, follow these steps:

- i. Select and right-click the connection definition, then select **Properties** from the pop-up menu.
- ii. Select the Options tab.
- iii. Change the **Host Name** to the host name of the new machine for your server.
- iv. If you are changing the port, change the **Port** to the port of the new port for your server.
- v. If you need to change the authentication domain, click **New** and define the new **Authentication Domain** for your server.
- vi. Click **OK**.

- d. Click **OK** to save the new configuration to the metadata repository.

4. Use SAS Management Console to define a spawner definition for the SAS Workspace Server's new machine. See [Using SAS Management Console to Define or Modify a Spawner](#) and fill in the appropriate fields as

follows:

- ◆ **Name.** Specify the name of the spawner, e.g. `<machine_name> Spawner`.
 - ◆ **Selected servers.** Add the name of the SAS Workspace Server, i.e. Main – Workspace Server.
 - ◆ **Authentication Domain.** Specify the spawner domain (must be the same as the server's authentication domain).
 - ◆ **Host Name.** Specify the machine name of the SAS Workspace Server.
 - ◆ **Port Number.** Specify the spawner port, default is 8581.
5. On the new machine for the SAS Workspace Server, follow these steps:
 - a. Install SAS 9.1 or higher and SAS Integration Technologies
 - b. Copy your metadata configuration file (XML file) and spawner startup script from your spawner configuration directory to the same directory on the new machine. If necessary, change the spawner startup script for the new machine. For details, see [Invoking \(Starting\) the Object Spawner](#).
 - c. Ensure that users who need to access the server are defined on the machine's host authentication provider.
 - d. Change the spawner startup script to specify the new spawner name for the spawner, e.g., `<machine_name> Spawner`.
 6. Edit the policy files for any applications that need to access the new server machine. (For details about editing policy files for the portal Web application and its components, see [Adding Permissions for Servers to Policy Files](#) in the *SAS Web Infrastructure Kit Administrator's Guide*).
 7. On the new machine for the SAS Stored Process Server, follow these steps:
 - a. Install SAS 9.1 or higher and SAS Integration Technologies.
 - b. Copy your metadata configuration file (XML file) and spawner startup script from your spawner configuration directory to the same directory on the new machine. If necessary, change the spawner startup script for the new machine. For details, see [Invoking \(Starting\) the Object Spawner](#).
 - c. Create a directory for stored process server log files. The recommended directory name is `STPDemo`, and the recommended location is the server home location that you specified when you ran the install program (for example, `C:\Program Files\SAS\Servers\STPDemo`).

If you do not remember the server home location, see the `STP_HOME` property in the `install.properties` file.

 - d. Ensure that the multi-user login (specified in the **Advanced Options** for the SAS Stored Process Server definition (e.g., in the project installation, the login owned by the SAS General Server group)) can authenticate against the host authentication provider for the SAS Stored Process Server's machine. On Windows NT and Windows 2000, give this account the Act as part of the Operating System user right.
 - e. Give the shared account for the multi-user login (e.g., in the project installation, the SAS General Server group login) "Write" permission to the stored process log directory.
 - f. Ensure that users who need to access the server are defined for each machine's host authentication provider.
 8. Ensure that the spawner invoker (e.g., in the project installation, the SAS user) can authenticate against the host authentication provider for each machine.

When you are finished modifying the server and spawner definitions:

- if you have added a new authentication domain for the machine, do both of the following:
 - ◆ use the User Manager plug-in to SAS Management Console to add a login definition for access to the server.

- ◆ use the User Manager plug-in to SAS Management Console to modify the login definition for the multi-user login (e.g., in the project installation, the SAS General Server group login). Modify the login definition to specify the new authentication, and, if required, the new user ID credentials.
- if you have changed operating systems and need to modify user credentials, use the User Manager plug-in to SAS Management Console to modify user and group login definition for the new user ID credentials of the new machine.
- if you need to move the stored process repositories to a different directory, use the Stored Process Manager plug-in to SAS Management Console to modify the stored process definition and change the **Source Repository** field on the Execution tab of the stored process definition.
- if your stored process definitions reference content on the old stored process or workspace server machine, you must add the content to the directory you defined in the stored process definition.

Security

Security

Authentication is the process of verifying the identity of a person or process within the guidelines of a specific security policy. *Authorization* is the process of evaluating whether a particular authenticated identity has permission to perform a task (such as read or write) on a particular resource. To understand, plan for, and implement authentication and authorization for the Open Metadata Architecture, see "Understanding the Security Concepts in the SAS Intelligence Architecture" and its related security chapters in the [SAS Intelligence Architecture: Planning and Administration Guide](#).

In addition to the security features that are provided with the SAS Open Metadata Architecture, SAS Integration Technologies provides other authentication and authorization mechanisms. These additional features enable you to implement the appropriate security for your enterprise. SAS Integration Technologies security provides additional mechanisms for authenticating users of IOM servers against an LDAP and Microsoft Active Directory server, and for providing authorized access to IOM Bridge servers. (For COM server connections, SAS Integration Technologies utilizes Windows security features. For details, see [Setting SAS Permissions on the Server \(COM/DCOM\)](#) and [Windows Client Security](#) in the *SAS Integration Technologies Developer's Guide*).

This chapter covers the following authentication and authorization topics:

- **Overview of Domains.** To understand the discussion of domains within this chapter, see [Overview of Domains](#).
- **Authentication Options.** For details about implementing authentication, see [Implementing Authentication](#).
- **User, Group, and Login Structure.** For details about the SAS Metadata Server user, group, and login structure, and how to specify login definitions for your user credentials, see [Defining Users, Groups, and Logins on the SAS Metadata Server](#).
- **Additional Server Security.** For details about additional server security, see [Setting up Additional Server Security](#).
- **Security for Spawner, Pooling, and Load–Balancing Configurations.** For details about the differences between SAS Workspace Server and SAS Stored Process Server security, spawner security, and pooling and load–balancing security, see
 - ◆ To understand security considerations for workspace servers, pooled workspace servers, load–balancing stored process servers, or load–balancing workspace servers, see [Planning for Workspace and Stored Process Server Security](#).
 - ◆ For spawner security, see [Planning the Spawner Security](#).
 - ◆ For pooling security, see [Planning the Pooling Security \(IOM Bridge only\)](#).
 - ◆ For load–balancing security, see [Planning the Load–Balancing Security \(IOM Bridge only\)](#).
- **Xythos WFS WebDAV Authentication and Authorization.** For details about the SAS Integration Technologies extension to the Xythos WFS WebDAV server, see [Implementing Authentication and Authorization for the Xythos WFS WebDAV Server](#).
- **Client Security and Encryption.** For details about implementing security in applications and implementing encryption, see [Implementing Security in Client Applications](#) and [Implementing Encryption](#).

Security

Overview of Domains

Within the host environment, SAS Open Metadata Architecture, and SAS Integration Technologies security, there are two types of domains used in basic security implementations. In addition, there is a third type of domain that is used for alternate authentication providers. In some cases, the domains names might be identical; however, it is important to distinguish between these different types of domains for the case where your implementation might require the different types of domains to be specified as different domain names.

- **security domains used by or associated with an authentication provider.** You can do both of the following:
 - ◆ define domains within the Windows operating system. For example, CARY and APEX.
 - ◆ when starting a server, specify a default domain to be used as the default security domain for the host operating system. For example, you might specify a default security domain APEX for the Unix operating system; when a user connects without a domain, the domain APEX is used to locate the correct fully qualified user ID (in a login definition) on the SAS Metadata Server. For details, see [Specifying Default Host Domains](#).
- **authentication domains specified in the SAS Metadata Server resource definitions.** Within the SAS Open Metadata Architecture, the authentication domain is a logical grouping that associates resources and logins (user credentials) together. An individual can use the same fully qualified user ID for any of the resources in the authentication domain. For an overview of authentication domains, see "Authentication Domains" in the [SAS Intelligence Architecture: Planning and Administration Guide](#).
- **authentication provider domain.** If you use an alternative authentication provider (such as LDAP or Microsoft Active Directory), you must specify an authentication provider domain in the user connection request. To authenticate to an alternative authentication provider (LDAP or Microsoft Active Directory), the connection request must specify an authentication provider domain that has been associated (on the server startup command AUTHPD option) with that authentication provider. For example, APEX\user@LDAP, where LDAP is the authentication provider domain. For details, see [Specifying Authentication Provider and Default Domains](#).

Security

Implementing Authentication

You can implement authentication with one or more of the following authentication mechanisms:

- **Host authentication provider (default)**: SAS Workspace Servers and SAS Stored Process Servers always authenticate against the host authentication provider. By default, SAS Metadata Servers and SAS OLAP Servers authenticate against the host authentication provider; however, you can set up trusted authentication mechanisms for the SAS Metadata Server or alternative authentication providers for either the SAS Metadata Server or SAS OLAP server. If the server authenticates against the host authentication provider, you must set up the appropriate accounts on the host authentication provider for the server's machine.
- **Trusted authentication mechanisms (for connections to the SAS Metadata Server only)**: You can set up *trusted user* or trusted peer session connections for the SAS Metadata Server.
- **Alternative authentication providers (for SAS Metadata Servers and SAS OLAP Servers only)**: You can set up your users to authenticate against an alternative authentication provider such as LDAP or Microsoft Active Directory.

The following table shows which types of authentication providers you can set up for each IOM server.

Authentication Providers for IOM Servers					
Type of Server	Host Authentication	Trusted Peer Authentication	Trusted User Authentication	LDAP Directory Server Authentication	Microsoft Active Directory Server Authentication
SAS Metadata Server	X	X	X	X	X
SAS OLAP Server	X			X	X
SAS Stored Process Server	X				
SAS Workspace Server	X				

Host Authentication Provider

By default, all IOM servers are authenticated by the host environment's authentication provider.

You must set up host authentication for the following user and group credentials:

- **For access to the servers, user or group credentials that connect to standard SAS Workspace Servers, SAS Stored Process Servers, or SAS OLAP Servers (that use host authentication)**. Users connect to the SAS Metadata Server and are initially authenticated against the SAS Metadata Server's authentication provider. To connect to the SAS Workspace, SAS Stored Process, or SAS OLAP Server, the appropriate credentials for the server are retrieved and returned. When the user (application) uses the appropriate credentials to connect to the SAS Workspace, SAS Stored Process Server, or SAS OLAP Server (if using host authentication), those user or group credentials are additionally authenticated by the host authentication provider for the SAS Workspace, SAS Stored Process, or SAS OLAP server's machine.

- **For a load–balancing SAS Stored Process Server configuration, the user or group credentials for the multi–user login definition.** The user or group credentials for the multi–user login definition are specified in the SAS Stored Process server definition. These credentials are authenticated against the host authentication provider for the SAS Stored Process Server's machine.
- **For a pooled server configuration, the user or group credentials for the puddle login(s) used to connect to the SAS Workspace Server(s).** The user or group credentials for the puddle login(s) are specified on the puddle definitions. These credentials are authenticated against the host authentication provider for the SAS Workspace Server's machine.
- **For a load–balancing configuration that load balances across more than one spawner, the user or group credentials that are used for connections between the load balancing spawners.** The user or group credentials are specified in the Logical Server Credentials field of the load balancing logical server definition. These credentials are authenticated against the host authentication provider for the server's machine.

To set up users for host authentication and to understand the host authentication process, see the following sections:

- [Implementing Host Authentication](#)
- [Specifying Default Host Domains](#)
- [How Hosts Handle Domains](#)

Trusted Authentication Mechanisms

The SAS Metadata Server supports two types of trusted connections: *trusted user* and trusted peer. Both represent a way to bypass authentication by the authentication provider for the SAS Metadata Server. They are provided in support of multiple back–end server environments where user IDs are authenticated by one server and must also be asserted on the metadata server.

- For SAS Metadata Servers, you can set up *trusted user* connections. The SAS Metadata Server views *trusted users* as already authenticated users. For details, see [Trusted User Connections](#).
- For SAS Metadata Servers, you can set up trusted peer session connections in order to allow SAS Workspace Servers, SAS Stored Process Servers, or SAS sessions to connect to the metadata server as trusted peers. For details, see [Trusted Peer Session Connections](#).

Alternate Authentication Providers

In addition, you can enable SAS Metadata Servers and SAS OLAP Servers to authenticate against alternative authentication providers (LDAP or Microsoft Active Directory). To set up users for authentication by an alternative authentication provider and to understand the authentication process, see the following sections:

- For details about setting up alternative authentication providers, see [Implementing Alternative Authentication Providers](#).
- For details about associating default domains or authentication provider domains, see [Specifying Authentication Provider and Default Domains](#).
- For details about how the server authenticates user credentials with or without authentication provider domains, see [How Servers Determine the Authentication Provider](#).
- For a description of an alternative authentication provider scenario, see [Scenario: Alternate Authentication Provider](#).

Defining Users for Host Authentication

By default, servers rely on the host environment to authenticate users. (SAS Workspace Servers and SAS Stored Process Servers always authenticate using the host environment). To implement host authentication for an IOM server, for every host user who needs to access a server or start a server, you must specify the following:

- **A valid user ID and password for the operating system account that provides access to the server's machine.** The procedure for adding host users varies depending on the operating system you are using.
- **System permissions for Windows and UNIX.**

For Windows systems, the following table shows the specific user rights (permissions) for server invokers and server accessors:

Required User Rights (Permissions) for Windows Operating System Accounts					
Type of Server and User	Act as part of the operating system	Adjust memory quotas for a process	Increase quotas	Replace the process level token	Log on as a batch job
SAS Metadata Server Invoker	Windows NT and 2000 only				
SAS OLAP Server Invoker	Windows NT and 2000 only				
Object Spawner Invoker for the SAS Stored Process Server*	Windows NT and 2000 only	Windows XP only	Windows NT and 2000 only	All Windows systems	
Object Spawner Invoker for the SAS Workspace Server*	Windows NT and 2000 only	Windows XP only	Windows NT and 2000 only	All Windows systems	
Accessors (clients) of SAS Metadata, OLAP, Stored Process, and Workspace Servers					**All Windows systems

***Note:** The object spawner invoker must also be a member of the Windows Administrators group.

****Note:** As an alternative, you might consider defining a **SAS Server Users** group and assign the **Log on as a batch job** user right to this group.

For details about setting user rights (permissions) on specific Windows systems, see these topics:

- ◆ [Setting System Access Permissions on Windows NT](#)
- ◆ [Setting System Access Permissions on Windows 2000](#)
- ◆ [Setting System Access Permissions on Windows XP](#)

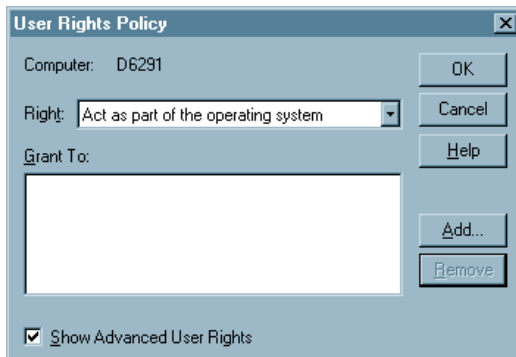
For UNIX systems, the servers require the SASPERM and SASAUTH files to be setuid and owned by root. See [Setting System Access Permissions on UNIX](#) for steps to ensure these permissions are set correctly.

When you use host authentication, you can also associate a default domain with the host; this domain is used for authorization purposes. For details, see [Specifying Default Host Domains](#).

Setting System Access Permissions on Windows NT

To set permissions on Windows NT, follow these steps:

1. Select **Start** ➤ **Programs** ➤ **Administrative Tools** ➤ **User Manager**.
2. On the Policies menu, select **User Rights**.
3. In the User Rights Policy window:
 - a. Select the **Show Advanced User Rights** check box.
 - b. Select the required permission from the **Right** drop-down list.



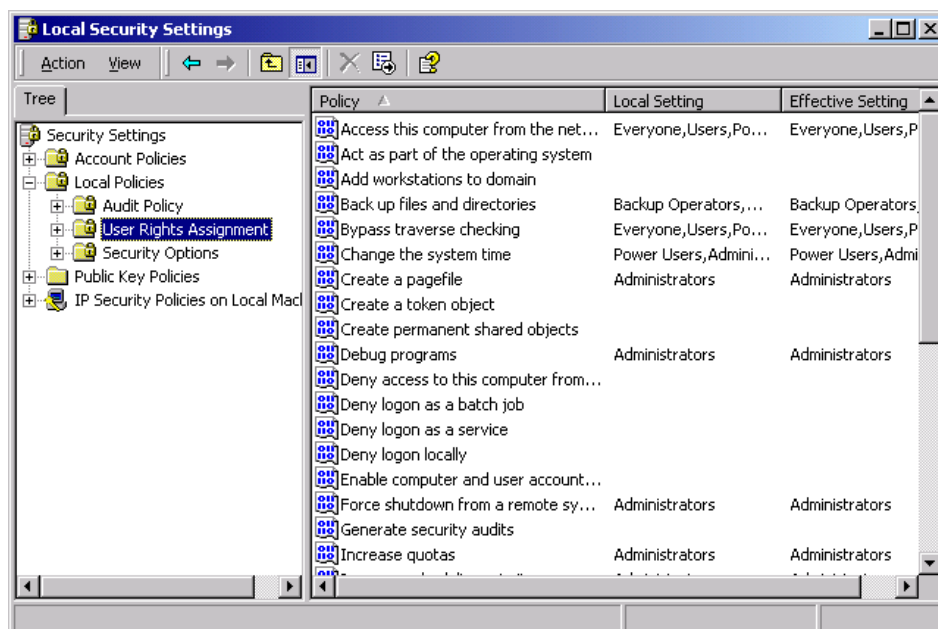
- c. Use the **Add** button to add the user ID for which you want to add the permission. The user ID is added to the Grant To box.
4. Click **OK**.
5. Restart the computer so that the updates can take effect.

Security

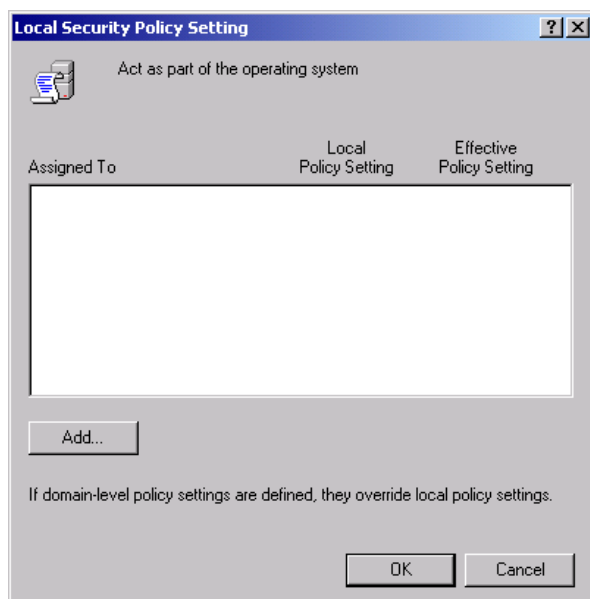
Setting System Access Permissions on Windows 2000

To set permissions on Windows 2000:

1. Select **Start** ➤ **Settings** ➤ **Control Panel** to open the Control Panel.
2. In the Control Panel, open **Administrative Tools**.
3. In Administrative Tools, open **Local Security Policy**.
4. Expand the tree for Local Policies, and then select **User Rights Assignment**.



5. Select and right-click the required user right to display a pop-up menu. From the pop-up menu, select **Security**. The following is an example of the window that is opened for the **Act as part of the operating system** user right.



To add permissions:

SAS® 9.1.2 Integration Technologies: Server Administrator's Guide

- a. Click **Add**. The Select Users or Groups window appears.
- b. In the Select Users or Groups window, type the user ID (that requires this permission) in the form:

domain\userid

- c. Click **OK**.

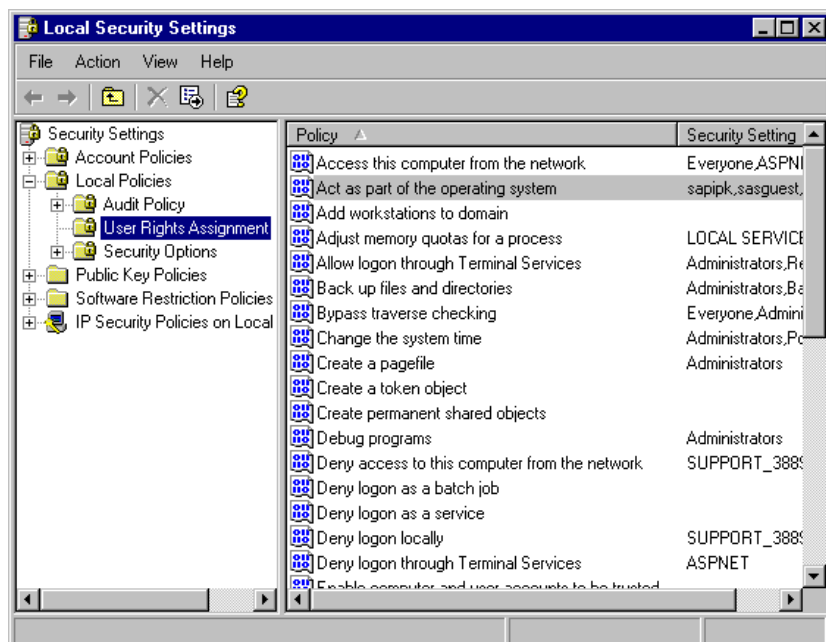
6. When you are finished adding permissions, restart the machine so that the updates will take effect.

Security

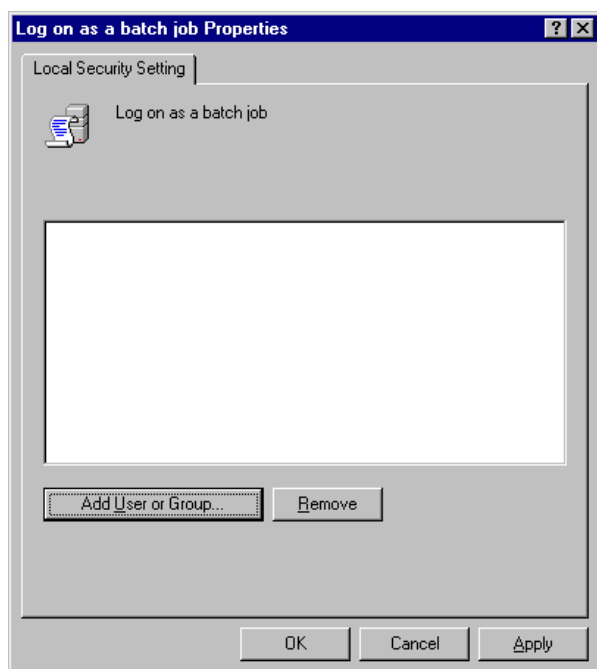
Setting System Access Permissions on Windows XP

To set permissions on Windows XP:

1. Select **Start ➔ Settings ➔ Control Panel**.
2. In the Control Panel, open **Administrative Tools**.
3. In Administrative Tools, open **Local Security Policy**.
4. In the Local Security Settings window, expand the tree for Local Policies and select **User Rights Assignment**.



5. Right-click the required user right, and select **Properties**. The appropriate properties window appears (for example, **Log on as a batch job**).



6. To add permissions:
 - a. Click **Add User or Group**. The software opens the Select Users or Groups window.
 - b. Specify the user ID that requires this permission.
 - c. Click **OK**.
 - d. Click **Apply** then **OK** to save your changes and close the Properties window.
7. When you are finished, restart the machine so that the updates will take effect.

Security

Setting System Access Permissions on UNIX

Like many other SAS processes, the IOM servers require that the SASPERM and SASAUTH files in the !SASROOT/utilities/bin directory be owned by the root user and have the setuid bit enabled. These permissions are typically set during SAS installation, using the setup utility. You might want to verify that the appropriate permissions are set. If they are not, then enable the setuid bit and change the owner to root by using one of the following methods.

Method 1: Using SAS Setup

1. Log in to the root account.

```
$ su root
```

2. Run SAS Setup from !SASROOT/sassetup.
3. From the SAS Setup primary menu, select **Run Setup Utilities**.
4. Select **Perform SAS System Configuration**.
5. Select **Configure User Authorization**.

Method 2: Using the Command Line

At a UNIX prompt, type the following:

```
$ su root
# cd !SASROOT/utilities/bin
# chown root sasauth sasperm sasrun
# chmod 4755 sasauth sasperm sasrun
# exit
```

Security

Specifying Default Host Domains When Starting Servers That Only Use Host Authentication

When you start a server, or a spawner that starts a server, you can use the AUTHPROVIDERDOMAIN startup option to associate a domain with the host authentication provider. (To understand the different types of domains used in the host environment, Open Metadata Architecture, and SAS Integration Technologies security, refer to [Overview of Domains](#)). When a user connects to the server without a domain, the server can use the domain association to determine a domain.

- On all hosts, when you associate a domain with the host authentication provider, if a user does not specify a domain in their credentials, the associated domain is used. For example, you might specify a default security domain APEX for the Unix operating system; when a user connects without a domain, the domain APEX is used to locate the correct fully qualified user ID (in a login definition) on the SAS Metadata Server.
- On hosts other than Windows, when you associate a domain with the host authentication provider, if a user specifies that domain with their credentials, the domain is removed from the credentials and the credentials are authenticated using the host authentication provider. If the user specifies a domain that is not the associated domain, the host authentication provider will not be able to authenticate the user.

When you specify a domain for hosts other than Windows, you allow multiple hosts to have their login definitions appear as identical. For example, when starting the servers xyz.iyi.abc.com and xyz2.iyi.abc.com, you can use the AUTHPROVIDERDOMAIN option to assign the domain name "abcunix". When users log in to either server, the domain will be returned and their user ID will look identical because both servers use the same domain name, e.g., "abcunix\abcmktg".

To associate a domain with the host authentication provider, on the SAS server or spawner startup command, specify the AUTHPROVIDERDOMAIN system option and associate a domain suffix with the host (HOSTUSER) authentication provider.

If you are only using host authentication to authenticate users that access the server, the AUTHPROVIDERDOMAIN option has the following syntax:

```
authproviderdomain = (HOSTUSER:domain)
```

Syntax Description

HOSTUSER

specifies that user IDs and passwords are authenticated by using the authentication processing that is provided by the host operating system.

domain

specifies a site-specific domain name. Quotation marks are required if the *domain* value contains blanks.

Note: The maximum length for the AUTHPROVIDERDOMAIN option value is 1,024 characters.

Operating Environment Information: Under the Windows operating environment, you can specify a authentication provider domain using either the AUTHPROVIDERDOMAIN system option or the AUTHSERVER system option. If both AUTHPROVIDERDOMAIN and AUTHSERVER are specified, the option that was specified first takes precedence.

How Hosts Handle Domains

When a user's credentials are authenticated, the domain allows the user credentials to be further qualified in order to determine an identity on the SAS Metadata Server. However, a user might not need to specify a domain (or machine name) when they logon:

- **For Windows host authentication**, your host users might or might not specify domains when they log in.
- **For host authentication for hosts other than Windows**, host users do not typically specify domains when they log in.

Depending on the type of authentication provider, domains are handled as follows:

Windows Host Authentication

For Windows host authentication:

- ◇ If users specify a domain when they log in, the Windows host returns that user domain (or machine name if it is a local account) for use in determining an identity on the SAS Metadata Server.
- ◇ If users do not specify a domain when they log in, the Windows host system handles the lack of domain as follows:
 - If the server was started with the AUTHPROVIDERDOMAIN system option to associate a domain with the HOSTUSER, the Windows host authentication returns this domain for use in determining an identity on the SAS Metadata Server.
 - If the server was not started with the AUTHPROVIDERDOMAIN system option, the host-authentication provider looks through all of the domains (searching the local machine first) for a match on the user ID. If a user ID match is found, the associated domain is returned.

Note: On Windows systems, if the AUTHSERVER option associates a domain with the HOSTUSER, the Windows host authentication returns this domain as the default domain. If both AUTHPROVIDERDOMAIN and AUTHSERVER are specified, the option that was specified first takes precedence.

Host Authentication for Hosts Other Than Windows

For host authentication for hosts other than Windows, users do not typically specify a domain when they logon. However, the non-Windows host can return a domain for use in determining an identity on the SAS Metadata Server.

- ◇ If the AUTHPROVIDERDOMAIN option was specified with a domain for the HOSTUSER, the host authentication returns this domain for use in determining an identity on the SAS Metadata Server.
- ◇ If the AUTHPROVIDERDOMAIN was not specified, the host authentication does not return a domain.

To understand how you define corresponding logins (fully qualified user IDs, passwords (optional), and authentication domains) for the SAS Metadata Server user and group definitions, see [Defining Users, Groups, and Logins on the SAS Metadata Server](#).

Security

Implementing Trusted Authentication Mechanisms

For multi-tier server environments where user IDs are already authenticated by a server or Web server's authentication mechanism and then must assert those identities on the metadata server, the authorization facility supports two types of trusted connections: *trusted user* connections and trusted peer session connections. User IDs that are used to connect to servers via the *trusted user* or trusted peer session mechanisms do not need to have an account on the authentication provider for the SAS Metadata Server's machine.

Trusted User Connections

The *trusted user* mechanism enables already authenticated users from peer back-end servers or Web-tier servers to connect to a SAS Metadata Server as *trusted user* connections. You must set up the appropriate authentication provider (host, LDAP, or Active Directory) for the *trusted user*; other users that connect via the *trusted user* do not require an account on the SAS Metadata Server's authentication provider as the SAS Metadata Server trusts that they have been authenticated at the back-end or web-tier server.

After you set up a *trusted user* in the `trustedUser.txt` file and on the appropriate authentication provider, the *trusted user* generates user passwords for users that have already been authenticated by a Web-tier or peer back-end server. From the viewpoint of the authorization facility, the *trusted user* represents an already authenticated connection to the SAS Metadata Server that can act on behalf of other users. If a user has already been authenticated on a Web-tier server, when they try to connect to the SAS Metadata Server, the *trusted user* can generate a password in order to allow them to connect.

For information about setting up *trusted user* for the SAS Metadata Server, see [Trusted User](#) in the *SAS Metadata Server: Setup Guide*.

Trusted Peer Session Connections

A trusted peer session connection enables a SAS process to establish a connection to a SAS Metadata Server without explicitly specifying the user ID and password to use for the connection. This feature enables the following:

- peer sessions can connect to the SAS Metadata Server without a password by using a user ID provided by the operating system
- applications that run jobs on SAS Stored Process Servers or SAS Workspace Servers can generate code without credentials
- batch jobs can run without explicit credentials.

For a SAS Metadata Server, you can allow a SAS Workspace Server or SAS Stored Process Server to connect to the metadata server as a trusted peer session.

The trusted peer connection works as follows:

1. The SAS Metadata Server is started with the `trustsaspeer` option. The `trustsaspeer` option specifies either
 - ♦ a file that contains a list of trusted domains for peer, back-end servers (or sessions) connecting from environments other than Windows.

Note: If your SAS Metadata Server is authenticating clients against an alternative authentication provider, you must specify a file that contains the trusted domains for the peer back-end servers (or sessions) connecting from an environment other than Windows.

- ◆ a blank or non-existent file.
- 2. A peer, back-end server, mid-tier server, or session uses a proprietary protocol to make a connection to the SAS Metadata Server.
- 3. If the SAS Metadata Server receives a connection with this proprietary protocol, it accepts the following:
 - ◆ non-domain qualified user IDs from hosts other than Windows.
 - ◆ domain-qualified user IDs from hosts other than Windows whose domains are specified in a trusted peer file (e.g., `trustedpeer.xml`) file.
 - ◆ user IDs from peer, back-end servers (or sessions) running on Windows.

Important Note: Use of this proprietary protocol implies that the SAS Metadata Server trusts the authentication mechanism of the connecting server. You must implement the appropriate security for your network to prevent untrusted machines and untrusted authentication that could compromise the SAS Metadata Server.

Setting up Trusted Peer Connections for SAS Sessions

You can set up trusted peer sessions for peer, back-end servers (or sessions) running on Windows and other systems. The following table shows the server (or session) environment from which you wish to connect, whether you use the AUTHPROVIDERDOMAIN (AUTHPD) option or AUTHSERVER option on the SAS Metadata Server startup command, how to specify the trusted peer option on the SAS Metadata Server startup command, and who can connect as a trusted peer connection when using the specified setup:

Trusted Peer Session Connection Setup(s)			
Connecting Environment for Trusted Peer Session	Is AUTHPD (or AUTHSERVER) Option Used on SAS Metadata Server Startup Command?	Trusted Peer Option To Use With SAS Metadata Server Startup Command	Who Can Connect
Windows peer back-end servers (or sessions)	Either YES or NO	<code>trustsaspeer=blankornonexist.xml</code> where <code>blankornonexist.xml</code> is a non-existent or empty trusted peer file	Peer SAS sessions from SAS Stored Process Servers, SAS Workspace Servers, or SAS batch sessions running on Windows. Peer SAS sessions from SAS Stored Process Servers, SAS Workspace Servers, or SAS batch sessions running on environments other than Windows if they DO NOT specify a domain with their user ID.
peer back-end servers (or sessions) not on Windows	NO	<code>trustsaspeer=blankornonexist.xml</code> where <code>blankornonexist.xml</code> is a non-existent or empty trusted peer file	Peer SAS sessions from SAS Stored Process Servers, SAS Workspace Servers, or SAS batch session that DO NOT specify a domain with their user ID

peer back-end servers (or sessions) not on Windows	YES	trustsaspeer= c:\config\trustedpeer.xml where trustedpeer.xml is a trusted peer file that contains the trusted domains. To create a trusted peer file, see Setting up a Trusted Peer File .	Peer SAS sessions from SAS Stored Process Servers, SAS Workspace Servers, or SAS batch sessions running on environments other than Windows if any of the following are true: <ul style="list-style-type: none"> the domain is specified in the credentials and is in the trusted peer file. a domain is not specified in the credentials and the domain specified by the AUTHPROVIDERDOMAIN (or AUTHSERVER) option is in the trusted peer file.
--	-----	---	---

Note: If the peer SAS session specifies a domain in its connection request (or has a domain associated to it by the AUTHPROVIDERDOMAIN (or AUTHSERVER) option), to allow that peer to connect, you must create a trusted peer file and include that domain as a trusted domain.

To understand the AUTHPROVIDERDOMAIN (or AUTHSERVER) option, if you are using host authentication, see [Specifying Default Host Domains When Starting Servers That Only Use Host Authentication](#). If you are using alternate authentication, see [Specifying Authentication Provider and Default Domains When Starting Servers](#).

Setting up a Trusted Peer File

To set up a trusted peer file, create a file (e.g., trustedpeer.xml) that contains a list of the trusted domains. For example:

```
<?xml version="1.0"?>
<!-- Specify which Windows Domain >
<!-- suffixes we will allow>
<TrustedSASDomains>
<!-- Allow the domain "Domain0" when >
<!-- peer SAS Session is executing on UNIX host>
<unix>Domain0</unix>
<!-- Allow the domains "Domain1" and "Domain2" when >
<!-- peer SAS Session is executing on z/OS host>
<os390>Domain1</os390>
<os390>Domain2</os390>
<!-- Allow the domain "Domain3" when >
<!-- peer SAS Session is executing on AlphaVMS host>
<vms>Domain3</vms>
</TrustedSASDomains>
```

Note: The trusted peer file is only required when the AUTHPROVIDERDOMAIN (or AUTHSERVER) option is specified upon startup of the SAS Metadata Server.

Example

The following is an example of a Windows SAS Metadata Server start command that specifies trusted peer support which enables peer, back-end Windows servers (or sessions) to connect as trusted peers:

```
"where_your_sas_is_installed\sas.exe"
-log "C:\sasoma\logs\sasoma.log" -logparm "write=immediate"
-linesize max -pagesize max -nosplash -noterminal -memsize 0
-objectserver -objectserverparms "protocol=bridge port=XXXX
  trustsaspeer=blank.xml
  classfactory=2887E7D7-4780-11D4-879F-00C04F38F0DB"
```

Note: Because Unix and MVS users can provide the domain information that is associated (by the AUTHPROVIDERDOMAIN option) as a default domain, any user who can executes SAS on a Unix or MVS system could supply a trusted peer domain. Therefore, if your network has separate Unix or MVS security domains with identical user IDs representing different actual users, it is unsafe to use the TRUSTSASPEER option. If users set the wrong domain value, they can easily be viewed as the identically named user in another domain. Data on the peer, back-end server or SAS Metadata Server could be compromised.

Security

Implementing Alternative Authentication Providers

To implement authentication, for SAS Metadata Servers or SAS OLAP Servers, you can implement one or both of the following alternative authentication providers:

- [LDAP directory server](#).
- [Microsoft Active Directory server](#).

LDAP Directory Server Authentication

When starting a server, you can enable LDAP users who specify a particular authentication provider domain to authenticate against an LDAP server instead of against the host. To implement authentication for LDAP, you must perform the following tasks:

1. **Ensure that LDAP users are defined.** Ensure that the appropriate user credentials are set up on an LDAP directory server. For details about setting up and administering an LDAP server, see [Setting up an LDAP Directory Server](#) in the *Administrator's Guide (LDAP)*.
2. **Start the server with the appropriate options for alternative authentication.** When starting the server, specify the following:
 - ◆ On the server start command or in the service configuration (if you run on Windows as a service), specify the AUTHPROVIDERDOMAIN option with the authentication provider domain to use for LDAP authentication. For example,

```
-authproviderdomain = (LDAP:Orion, HOSTUSER:'Orion2')
```

where Orion is the domain that will be specified when the user wishes to authenticate against LDAP and Orion2 is the default domain that will be associated with the host operating system.

For details, see [Specifying Authentication Provider and Default Domains When Starting Servers](#).

- ◆ Set the following environment variables (using the appropriate procedure for your operating system):

```
LDAP_PORT= <port number for LDAP. If LDAP_PORT is  
            not specified, default is 389.>  
LDAP_BASE= <base DN to use. For example:  
            o=my company,c=US>  
LDAP_HOST= <host name of the machine running LDAP>
```

In addition to these environment variables, you can set the LDAP_IDATTR environment variable to the name of the person entry LDAP attribute that stores the user ID if the attribute does not contain the default value, "uid".

Note: To set environment variables on the z/OS operating system, see [Environment Variables for the z/OS Operating System](#)

- ◆ If your users connect with a user ID instead of a distinguished name (DN), set the following environment variables:

```
LDAP_PRIV_DN= <privileged DN that is allowed to search  
              for users. For example, cn=useradmin>  
LDAP_PRIV_PW= <password for LDAP_PRIV_DN>
```

Note: If the LDAP server allows anonymous binds, you are not required to specify the LDAP_PRIV_DN and LDAP_PRIV_PW environment variables.

3. **Define login credentials on the SAS Metadata Server.** After authentication, the SAS Metadata Server searches for the user ID and associated user definition (identity) in the SAS Metadata Repository. Therefore, you must have a user and login definition (that contains the LDAP authentication credentials) in the appropriate SAS Metadata Repository. (For details, see [Defining Users, Groups, and Logins on the SAS Metadata Server](#)). For user IDs that authenticate against the LDAP server, create a login definition with a user ID that has the following format:

```
userid@AUTHPROVIDERDOMAIN
```

4. **Ensure that users connect with the appropriate credentials for alternative authentication.** When an LDAP user connects to the server, specify the authentication provider domain in the LDAP user connection request (in order to associate the authentication provider domain with the LDAP authentication provider). To authenticate against LDAP, the LDAP user must log in with the following format:

```
userid@AUTHPROVIDERDOMAIN
```

For example

```
Tom@Orion
```

where Orion is the authentication provider domain associated with the LDAP server.

If you have used the AUTHPD option to configure the LDAP server as an alternative authentication provider (e.g., LDAP: <AUTHPROVIDERDOMAIN>), all logins of the form userID@<domain> will be sent to the LDAP server (as opposed to the host authentication provider) for authentication.

Example

The following is an example of a Windows metadata server start command that specifies an alternative LDAP authentication provider:

```
"where_your_sas_is_installed\sas.exe"
-log "C:\sasoma\logs\sasoma.log" -logparm "write=immediate"
-linesize max -pagesize max -nosplash -noterminal
-memsize 0 -authproviderdomain (LDAP:MyLDAPDomain)
-objectserver -objectserverparms "protocol=bridge port=XXXX
classfactory=2887E7D7-4780-11D4-879F-00C04F38F0DB"
```

To be authenticated by this provider, a user would specify a user ID in the form:

```
userid@MyLDAPDomain
```

Microsoft Active Directory Authentication

When starting a server, you can enable Microsoft Active Directory users who specify a particular authentication provider domain to authenticate against a Microsoft Active Directory server instead of against the host. To implement authentication for Microsoft Active Directory, you must perform the following tasks:

1. **Ensure that Microsoft Active Directory users are defined.** Ensure that the appropriate user credentials are

set up on a Microsoft Active Directory server. For details, see the [Microsoft Active Directory](#) home page on the Microsoft Web site.

2. **Start the server with the appropriate options for alternative authentication.** When starting the server, specify the following:

- ◆ On the server start command or in the service configuration (if you run on Windows as a service), specify the AUTHPROVIDERDOMAIN option with the authentication provider domain to use for Microsoft Active Directory authentication. For example,

```
-authproviderdomain = (ADIR:Orion, HOSTUSER:'Orion2')
```

where Orion is the domain that will be specified when the user wishes to authenticate against Microsoft Active Directory and Orion2 is the default domain that will be associated with the host operating system.

Note: With Microsoft Active Directory alternative authentication, you can use your Windows network domain as the authentication provider domain. So, you could specify the following:

```
-authproviderdomain = (ADIR:Sales)
```

where Sales is the Windows network domain.

Note: For details, see [Specifying Authentication Provider and Default Domains When Starting Servers](#).

- ◆ Set the following environment variables:

```
AD_PORT= <Active Directory port number>
```

If AD_PORT is not specified, default is 389.

```
AD_HOST= <Active Directory host name>
```

Note: To set environment variables on the z/OS operating system, see [Environment Variables for the z/OS Operating System](#).

3. **Define login credentials on the SAS Metadata Server.** After authentication, the SAS Metadata Server searches for the user ID and associated user definition (identity) in the SAS Metadata Repository. Therefore, you must have a user and login definition (that contains the Microsoft Active Directory authentication credentials) in the appropriate SAS Metadata Repository. (For details, see [Defining Users, Groups, and Logins on the SAS Metadata Server](#)). For user IDs that authenticate against Microsoft Active Directory, create a login definition with a user ID that has one of the following formats:

```
domain\userid
domain/userid
userid@domain
```

4. **Ensure that users connect with the appropriate credentials for alternative authentication.** When a Microsoft Active Directory user connects to the server, specify the authentication provider domain in the user ID (bindDN) (in order to associate the domain with the Microsoft Active Directory authentication provider). To authenticate against Microsoft Active Directory, the Microsoft Active Directory user must log in with one of the following formats:

```
userid@AUTHPROVIDERDOMAIN or userid@domain
  (where domain=AUTHPROVIDERDOMAIN)
domain\userid@AUTHPROVIDERDOMAIN
domain/userid@AUTHPROVIDERDOMAIN
userid@domain@AUTHPROVIDERDOMAIN
```

For example:

```
ABC\Tom@Orion
```

where Orion is the authentication provider domain associated with the Microsoft Active Directory server.

Note: If you use the domain\userid format, you can take advantage of Microsoft Active Directory's trusted network of domain controllers.

With Microsoft Active Directory alternative authentication, you can also use your Windows network domain as the authentication provider domain. So, you could specify `userid@domain`, where domain equals the AUTHPROVIDERDOMAIN that you specified in step 2.

For example:

```
Tom@Sales
```

Note: If you have used the AUTHPD option to configure the Microsoft Active Directory alternative authentication provider (e.g., ADIR: <AUTHPROVIDERDOMAIN>), all logins of the form `userID@<domain>` will be sent to the Active Directory server (as opposed to the host authentication provider) for authentication.

Example

The following is an example of a Windows metadata server start command that specifies an alternative Microsoft Active Directory authentication provider:

```
"where_your_sas_is_installed\sas.exe"
-log "/sasoma/logs/sasoma.log" -logparm "write=immediate"
-linesize max -pagesize max -noterminal -memsize 0
-authproviderdomain (ADIR:MyADIRDomain) -objectserver
-objectserverparms "protocol=bridge port=XXXX
classfactory=2887E7D7-4780-11D4-879F-00C04F38F0DB"
```

To be authenticated by this provider, a user would specify a user ID in the form:

```
userid@MyADIRDomain
domain\userid@MyADIRDomain
domain/userid@MyADIRDomain
userid@domain@MyADIRDomain
```

Environment Variables for the z/OS Operating System

For the z/OS operating system, a TKMVSENV file is used to make a list of pseudo environment variables available. A TKMVSENV PDS is created at installation. To define the environment variables for the SAS Metadata Server or SAS OLAP Server, create a member in the PDS that specifies the necessary variables, then reference this PDS member in the TKMVSENV DD statement in your started task.

Security

Specifying Authentication Provider and Default Domains When Starting Servers

When you start a SAS Metadata Server or SAS OLAP server, you can use the AUTHPROVIDERDOMAIN startup option to associate domains with the host, LDAP, or Microsoft Active Directory authentication provider. When a user connects to the server, the server can use the domain associations to determine the appropriate authentication provider or associate a default domain with the host. When starting a SAS Metadata Server or SAS OLAP server, you can use the AUTHPROVIDERDOMAIN option to do the following:

- **associate specific domains with the LDAP or Microsoft Active Directory authentication provider.** When a user logs in using a particular domain, the user is authenticated by the authentication provider specified for that domain. If the domain is not associated with an authentication provider, host authentication is used as the default authentication provider.

To associate a domain with an authentication provider, on the SAS startup command, specify the AUTHPROVIDERDOMAIN system option and associate a domain suffix with the host (HOSTUSER), LDAP (LDAP), or ADIR (ADIR) authentication provider. This association allows the SAS server to choose the authentication provider by the domain name presented.

Note: To allow multiple security domains to authenticate to the same alternative authentication provider (LDAP or Microsoft Active Directory) you can associate a pseudo-domain name as the authentication provider domain name for that authentication provider. For example, the security domains RANDD and MKTG might both use the authentication provider domain of LDAP.

- **associate a domain with the host authentication provider.**
 - ◆ On all hosts, when you associate a domain with the host authentication provider, if a user does not specify a domain in their credentials, the associated domain is used.
 - ◆ On hosts other than Windows, when you associate a domain with the host authentication provider, if a user specifies that domain with their credentials, the domain is removed from the credentials and the credentials are authenticated using the host authentication provider. If the user specifies a domain that is not the associated domain, the host authentication provider will not be able to authenticate the user.

To associate a domain with the host authentication provider, on the SAS server startup command, specify the AUTHPROVIDERDOMAIN system option and associate a domain suffix with the host (HOSTUSER) authentication provider.

When using an alternative authentication provider, the AUTHPROVIDERDOMAIN option has the following syntax:

authproviderdomain = (*provider-1:domain-1*<, . . . *provider-n:domain-n*>)

Syntax Description

provider

specifies the authentication provider associated with a domain. Valid values for provider are as follows:

ADIR specifies that the authentication provider is a Microsoft Active Directory server that accepts a bind containing a user ID and password for authentication.

HOSTUSER

specifies that user IDs and passwords are authenticated by using the authentication processing that is provided by the host operating system.

Operating Environment Information: Under the Windows operating environment, assigning the authentication provider using the HOSTUSER domain is the same as assigning the authentication provider using the AUTHSERVER system option. You may want to use the AUTHPROVIDERDOMAIN system option when you specify multiple authentication providers.

LDAP specifies that the authentication provider uses an LDAP server by specifying either

- ◇ the bind distinguished name (BINDDN) and a password for authentication
- ◇ the default "uid" and enabling LDAP to search for the bind distinguished name (BINDDN) by setting the LDAP_PRIV_DN and LDAP_PRIV_PW environment variables.

domain

specifies a site-specific domain name. The domain name is a name supplied by the administrator to which authentication provider should be used to authenticate a user. Quotation marks are required if the domain value contains blanks.

The following examples show how to specify domain:

- `-authproviderdomain (LDAP:Domain Name)`
- `-authproviderdomain (HOSTUSER:Domain name, ADIR: Domain Name)`

The maximum length for the AUTHPROVIDERDOMAIN option value is 1,024 characters.

Security

How Servers Determine the Authentication Provider

When a user who requires authentication connects to an IOM Server, the server must determine the appropriate authentication provider to use for authentication. When a user connects, he or she might log in with credentials in any of the following formats:

```
userid  
userid@domain  
domain\userid  
userid@AUTHPROVIDERDOMAIN  
userid@domain@AUTHPROVIDERDOMAIN  
domain\userid@AUTHPROVIDERDOMAIN  
domain/userid@AUTHPROVIDERDOMAIN
```

The server determines the authentication provider as follows:

How Servers Determine the Authentication Provider	
Condition	Result
The server was started with the AUTHPROVIDERDOMAIN option and is a SAS Metadata Server or SAS OLAP Server.	<ul style="list-style-type: none">• if the user specified an authentication provider domain that matches an assigned authentication provider domain, the associated provider is used for authentication.• if the user specified an authentication provider domain that does not match an assigned provider domain or if the user did not specify an authentication provider domain, the host authentication provider for the server's machine is used.
The server was started without the AUTHPROVIDERDOMAIN option.	the host authentication provider for the server's machine is used.

Understanding How Authentication Providers Handle Domains

When a user's credentials are authenticated, the domain allows the user credentials to be further qualified in order to determine an identity for authorization purposes. However, a user might need to specify a domain (or machine name) when they log in:

- **For Windows host authentication**, your host users might specify domains when they log in.
- **For host authentication**, host users do not typically specify domains when they log in.
- **For LDAP and Microsoft Active Directory authentication**, the LDAP or Active Directory user must specify an authentication provider domain in order to associate that domain with an authentication provider. The server uses the AUTHPROVIDERDOMAIN option to enable LDAP or Active Directory users in that domain to use LDAP or Active Directory as the authentication provider. The user might also specify a security domain for the LDAP or Active Directory provider.

Depending on the type of authentication provider, domains are handled as follows:

Windows Host Authentication

For Windows host authentication:

- ◊ If users specify a domain when they log in, the Windows host returns that user domain (or machine name if it is a local account) for use in determining an identity for authorization.
- ◊ If users do not specify a domain when they log in, the Windows host system handles the lack of domain as follows:
 - If the server was started with the AUTHPROVIDERDOMAIN system option to associate a domain with the HOSTUSER, the Windows host authentication returns this domain for use in determining an identity for authorization.
 - If the server was not started with the AUTHPROVIDERDOMAIN system option, the host authentication provider looks through all of the domains (searching the local machine first) for a match on the user ID. If a user ID match is found, the associated domain is returned.

Note: On Windows systems, if the AUTHSERVER option associates a domain with the HOSTUSER, the Windows host authentication returns this domain as the default domain. If both AUTHPROVIDERDOMAIN and AUTHSERVER are specified, the option that was specified first takes precedence.

Host Authentication on Systems other than Windows

For host authentication on systems other than Windows, users do not typically specify a domain when they log in. However, the host can return a domain for use in determining an identity for authorization as follows:

- ◊ If the AUTHPROVIDERDOMAIN option was specified with a domain for the HOSTUSER, the host authentication returns this domain for use in determining an identity for authorization.
- ◊ If the AUTHPROVIDERDOMAIN was not specified, the host authentication does not return a domain.

LDAP or Microsoft Active Directory Authentication

For LDAP or Microsoft Active Directory authentication, if LDAP or Active Directory users do not specify a domain when they log in, the LDAP or Active Directory provider returns the domain as follows:

- ◊ If the user ID that is stored in LDAP or Active Directory contains a domain (i.e., ABC\Tom), that domain is returned for use in authorization.
- ◊ If the user ID that is stored in LDAP or Active Directory does not contain a domain (i.e., Tom), the LDAP or Active Directory domain that is specified on the AUTHPROVIDERDOMAIN option is returned for use in authorization.

To understand how you define corresponding logins (fully qualified user IDs, passwords (optional), and authentication domains) for the user and group definitions on the SAS Metadata Server, see [Defining Users, Groups, and Logins on the SAS Metadata Server](#).

Security

Scenario: Alternate Authentication Provider

One of the most beneficial ways to use alternative authentication providers is to run the SAS Metadata Server or OLAP server on UNIX or z/OS with SAS Workspace Servers and SAS Stored Process Servers deployed on a Windows machine. If you already have your users set up on a Microsoft Active Directory server, this type of setup might be a useful scenario to consider for user authentication. Authenticating users against the Microsoft Active Directory services minimizes the number of accounts you would be required to create on a UNIX or z/OS machine.

This type of scenario provides the following benefits:

- speed, flexibility, and excellent response time due to running the SAS Metadata Server on a large, multi-processor, 64-bit UNIX or z/OS server. An OLAP server can also authenticate against Microsoft Active Directory; therefore, it would also be beneficial to deploy an OLAP server on UNIX or z/OS for this scenario.
- with the exception of the user account definition for the invoker of the SAS Metadata Server, there will be no requirements for user accounts on the back-end UNIX or z/OS server.
- for each user, a requirement for only one Windows account definition; this account can also be used to host-authenticate users that connect to SAS Workspace Servers or SAS Stored Process Servers deployed on Windows.

The following scenario provides an example of how to configure such a setup by showing an example of how to enable the Microsoft Active Directory alternative authentication provider to authenticate users for a SAS Metadata Server on UNIX. The scenario consists of the following:

- a SAS Metadata Server that runs on a UNIX host system. Normally, in order to create users for host authentication, you would need to set up user accounts for your users on the UNIX host system. However, if your users are already defined in Active Directory, you can use a Microsoft Active Directory server to authenticate users of the SAS Metadata Server.
- Microsoft Active Directory server that contains users in the Raleigh domain.

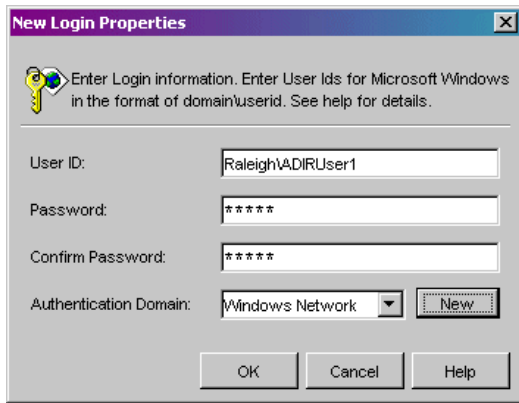
To configure this scenario, follow these steps:

1. Ensure that all users are defined on the Microsoft Active Directory server.
2. Start the SAS Metadata Server with the following startup script:

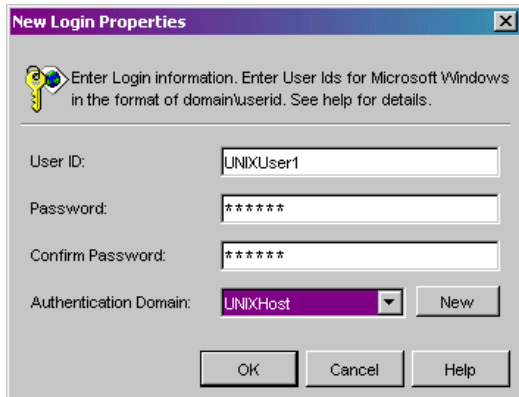
```
export AD_PORT=389
export AD_HOST=myMachine.myCompany.com
"/sasv91/sas.exe" -log "/sasoma/logs/sasoma.log"
-logparm "write=immediate" -linesize max
-pagesize max -noterminal -memsize 0
-authproviderdomain (ADIR: ADIRDomain)
-objectserver -objectserverparms "protocol=bridge
port=XXXX classfactory=2887E7D7-4780-11D4-879F-00C04F38F0DB"
```

3. Define users in the SAS Metadata Server as follows:

For Microsoft Active Directory authentication:



For host authentication:



4. Ensure that users log in with the appropriate login credentials:

- ◆ For Microsoft Active Directory authentication:

`domain\userid@ADIRDomain`

For example, `Raleigh\UNIXUser1@ADIRDomain`

- ◆ For host authentication:

`userid`

For example, `UNIXUser1`

The authentication process will then work as follows:

1. The SAS Metadata Server is started with the AUTHPD ADIR:ADIRDomain option.
2. A user logs in with the login credentials `Raleigh\ADIRUser1@ADIRDomain`.
3. The SAS Metadata Server (that was started with the AUTHPD ADIR:ADIRDomain option) determines that the @ADIRDOMAIN indicates Active Directory authentication.
4. The user `Raleigh\ADIRUser1` is authenticated against Microsoft Active Directory.
5. Another user logs in as `UNIXUser1`
6. The SAS Metadata Server determines that the lack of @domain indicates host authentication.
7. The user `UNIXUser1` is authenticated against the host authentication provider.

For further details about setting up Microsoft Active Directory authentication, see [Implementing Alternative](#)

Authentication Providers.

Security

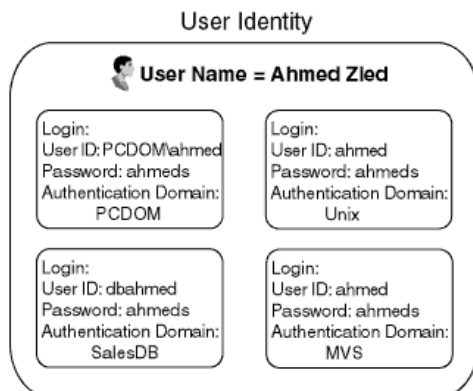
Defining Users, Groups, and Logins on the SAS Metadata Server

The User Manager plug-in of SAS Management Console provides centralized management of user information in a SAS metadata environment. The User Manager enables administrators to maintain user, group, and login definition information in a metadata repository. When you register an individual user or group in the User Manager, a SAS Open Metadata Architecture metadata identity is also created for the user or group. These definitions/identities are then used to do the following:

- authorize users or groups to access specific metadata or resources that the metadata describes.
- allow applications to retrieve appropriate login credentials for servers or other resources

Before you create the User Manager definitions, there are up to three types of domains which you must understand. To better understand the use of domains, refer to [Overview of Domains](#). The User Manager allows you to create these definitions:

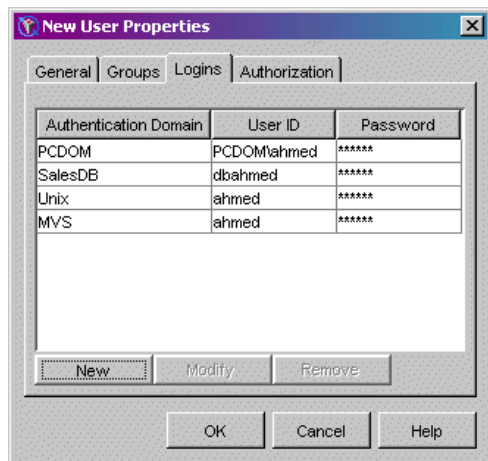
- **User Metadata Identity.** You can register user definitions and associate one or more login definitions with the user definition. The login definitions are then associated with the user metadata identity and this identity is used for authorization decisions. You can also add your user definitions to a group definition that is associated with a group metadata identity. The following diagram shows the relationship between a user metadata identity and its associated login definitions:



In the previous diagram, the user named Ahmed Zied contains login credentials for four different servers. These servers are each defined in different authentication domains:

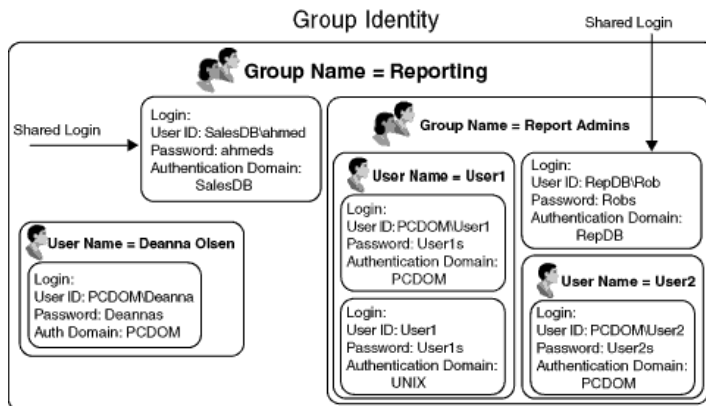
- ♦ the authentication domain that contains the server for the Windows network domain, PCDOM
- ♦ the authentication domain that contains Unix servers, Unix
- ♦ the authentication domain that contains database servers, salesdb
- ♦ the authentication domain that contains z/OS (MVS) servers, MVS

The following SAS Management Console screen shot shows the login definitions for the user Ahmed Zied:



To define users and login definitions on the SAS Metadata Server, see [Defining a User](#) in the *SAS Management Console: User's Guide*.

- Group Metadata Identity.** You can register group definitions and associate one or more user metadata identities and their login definitions with the group. When you add user metadata identities to a group, the users and their login definitions are then also associated with the group metadata identity. This association allows many different user metadata identities to use the same group metadata identity for authorization. The following diagram shows the relationship between group and user metadata identities, and their associated login definitions:



The following SAS Management Console screen shows the members of the SAS group named Reporting:



For each group, you can also define login definitions on the Logins tab of the group definition. These login definitions are then shared login definitions for the users and other groups defined as members of the group metadata identity.

To define groups and associated login definitions on the SAS Metadata Server, see [Defining a Group](#) in the *SAS Management Console: User's Guide*.

- Login Definitions.** A login definition contains the user credentials for a user account on a specific authentication provider. Multiple login definitions allow you to define different user credentials for different authentication providers. These different login credentials then belong to the same user metadata identity. For each login definition, you must define a fully qualified user ID, password (optional), and authentication domain. For each login definition, on the Logins tab of the user or group definition, enter the following fields as appropriate:

Authentication Domain

The authentication domain of the login definition must match the authentication domain of the resource you want to access with this login definition. In the **Authentication Domain** field of the login definition, enter the authentication domain name that is used in the **Authentication Domain** field of the resources (such as servers) that you want to access with this login definition.

Note: Applications use the name of the authentication domain that is associated with a server to locate login definitions that contain credentials to access the server. Choose an authentication domain name that is meaningful to the systems administrator.

For Windows users, the authentication domain name can (but is not required to) be the same as the Windows network domain name. Because applications use the authentication domain only to associate servers and login definitions, when you name the authentication domain the same as your Windows network domain, you still must enter a fully qualified user ID for the Windows system. The authentication domain is not used to construct the fully qualified user ID.

User ID

The user ID stored in the **User ID** field of a given login definition should exactly match the host user ID. You must specify a domain in the user ID of the login definition if you are authenticating against the following authentication providers:

- Windows host.

- Host other than Windows that is started using the AUTHPROVIDERDOMAIN option to specify a domain.
- LDAP directory server.
- Microsoft Active Directory server.

For each type of authentication provider, the following table gives information about how to specify the **user ID** field in a login definition:

Format for the User ID Field in the Login Definition			
Type of Authentication Provider Account	Qualifier for the User ID	Example	Additional Information
Windows local account	the name of the machine	If you access resources using a local Windows account that is named tara on a computer that is named mymachine.win.orionsports.com , you should have a login that includes a user ID of either mymachine\tara or tara@mymachine .	For details about how domains are handled, see Understanding How Hosts Handle Domains .
Windows network account	the name of the Windows network domain	If you access resources using a Windows network account that is named tara in a Windows network domain that is named WINNT , you should have a login that includes a user ID or either WINNT\tara or tara@WINNT .	For details about how domains are handled, see Understanding How Hosts Handle Domains .
Microsoft Active Directory account	the name of the Windows network domain	If you access resources using an Microsoft Active Directory account that is named tara in a Windows network domain that is named WINNT , you should have a login that includes a user ID or either WINNT\tara or tara@WINNT .	For details, see Specifying Authentication Providers Domains When Starting Servers .
LDAP Directory account	the name of the domain that is specified in the AUTHPROVIDERDOMAIN option when the target server is invoked	If you access resources using an LDAP account that is named tara and the target server is invoked using -authproviderdomain (LDAP:Sales) then you should have a login that includes a user ID of tara@Sales .	For details, see Specifying Authentication Providers Domains When Starting Servers .
Unix or z/OS account	none Note: If the	If you access resources using a Unix or z/OS operating system account that is named tara , you	For details about using the AUTHPROVIDERDOMAIN option, see Specifying

	AUTHPROVIDERDOMAIN option is used when the target server is invoked, you can qualify the user ID with the specified domain name. In most cases, this option is not specified for servers running on Unix or z/OS.	<p>should have a login that includes a user ID of tara.</p> <p>Note: If the target server is invoked using</p> <pre>-authproviderdomain (HOSTUSER:Sales)</pre> <p>then you should have a login that includes a user ID of either Sales\tara or tara@Sales.</p>	Authentication Providers Domains When Starting Servers.
Users Authenticated via Trusted User Mechanisms	a domain if one was passed from the Web server	<p>If you access resources using an account that is authenticated by a Web server's authentication provider,</p> <ul style="list-style-type: none"> · if the Web server passes credentials that contain a domain, specify a domain. For example, WINNT\tara. · if the Web server does not pass user credentials that contain a domain, do not specify a domain. For example, tara. 	For details, see Trusted User Connections
Trusted SAS Peer Sessions Authenticated via Trusted Peer Mechanisms	<p>If the SAS peer session connects from a Windows host, the Windows domain.</p> <p>If the AUTHPROVIDERDOMAIN option associates a default domain for a SAS peer session connection from a host other than Windows, the domain specified by AUTHPROVIDERDOMAIN</p> <p>If the SAS peer connection does not connect from Windows and the AUTHPROVIDERDOMAIN option is not used to associate a default domain, no qualifier is required.</p>	<p>If the session connects from a Windows host, then you should have a login that includes a domain, e.g., Sales\tara.</p> <p>If the target server is invoked using</p> <pre>-authproviderdomain (HOSTUSER:Sales)</pre> <p>then you should have a login that includes a domain, e.g., tara@Sales.</p>	For details, see Trusted Peer Session Connections .

Password

Enter the password in the following cases:

SAS® 9.1.2 Integration Technologies: Server Administrator's Guide

- Outbound login definitions: if the login definition is for credentials that applications can retrieve from a SAS Metadata Server and send to other systems that need to verify a user's identity, a password is required.
- WebDAV user's login definition for a WebDAV user that does one of the following:
 - uses DIGEST authentication
 - authenticates against a SAS Metadata Server that is in a different authentication domain than the WebDAV server.

Do not enter the password in the following cases:

- Inbound login definitions: if the login definition is used **ONLY** as an authenticated connection to the SAS Metadata Server in order to determine your metadata identity, a password is not required.
- WebDAV user's login definition for a WebDAV user that uses BASIC authentication and authenticates against a SAS Metadata Server in the same authentication domain as the WebDAV server.

When creating login definitions do the following as appropriate:

- ◆ If a user or group metadata identity has access to multiple authentication domains, create a separate Login definition for each authentication domain.
- ◆ If the same user ID and password combination exist in separate domains but within the same user or group metadata identity, create a separate Login definition for each domain.

Important Note: It is essential for the User Manager to resolve the fully qualified user ID to a single user or group metadata identity. For this reason, each user ID and domain combination within the metadata server must belong to the login definition for only one user or group metadata identity. While an identity can be associated with multiple fully qualified user IDs, each user ID and domain combination (domain qualified user ID) must be associated with only one user or group metadata identity.

Security

Implementing Authentication and Authorization for the Xythos WFS WebDAV Server

With SAS Integration Technologies, you might publish or subscribe to information stored on a Xythos WebFile Server (WFS) WebDAV server. In addition, if you use the SAS Information Delivery Portal, you might store file content on a Xythos WFS WebDAV server. Other products, such as SAS Web Report Studio, use the WebDAV server to store reports.

For security purposes, SAS Integration Technologies implements an extension, the SAS User Management Customization, that is an optional addition to the authentication mechanisms of the Xythos WFS WebDAV server. The extension enables the WebDAV server to use authentication and authorization metadata in the SAS Metadata Server as follows:

- **Authentication:** When using the Xythos WFS WebDAV server, WebDAV users can be authenticated against the SAS Metadata Server's authentication provider. In this case, you must define your WebDAV users on the appropriate authentication provider for the SAS Metadata Server. For details about authentication providers, see [Implementing Authentication](#). (In other cases, specific user login definitions can be used for authentication).
- **Authorization:** To authorize access to content on a Xythos WFS WebDAV server, administrators can specify users and groups that are defined in a SAS Metadata Repository. To set authorization (access control) for appropriate user or group metadata identities, administrators use the Xythos WFS Administration interface to control access to resources on the WebDAV server. Before you can associate access controls with a folder, you must complete these tasks:
 1. **Create folders on the WebDAV server.** Use the WebDAV tools to set up the appropriate folders.
 2. **Ensure that the appropriate user, group, and login definitions exist on the SAS Metadata Server for the WebDAV users and groups for whom you wish to control access to the folders:** Use the User Manager plug-in of the SAS Management Console to define the users, groups, and logins in a SAS Metadata Repository. Define a login as follows:
 - ◇ Specify the authentication domain name for the Xythos WebDAV server that you entered during installation of the SAS User Management Customization.
 - ◇ Specify the password field for the login definition based on the type of authentication setup that your WebDAV server uses. For details, see [Defining Users, Groups, and Logins on the SAS Metadata Server](#).

After you have created the WebDAV folders and have ensured that the appropriate user, group, and login definitions are created on the SAS Metadata Server, use the Xythos WFS WebDAV Administration interface to associate access controls with the folders. For an example of using the Administration interface with a portal publish and subscribe scenario, see [Scenario: Using the Xythos Administration GUI and SAS User Management Customization with the Portal](#). For further details about the Xythos administration tools, refer to the product documentation.

Security

Scenario: Using the Xythos Administration GUI and SAS User Management Customization with the Portal

When you administer the SAS Information Delivery Portal, you might want to set up WebDAV folders that enable group-based access to content. Using the SAS Customizations extensions for the Xythos WFS WebDAV server, you can grant users and groups (that are defined on the SAS Metadata Server) read and/or write access to folders on the Xythos WFS WebDAV Server. For example, within the portal implementation, you might utilize the publish and subscribe capabilities to publish (write) and subscribe to (read) group folders on a DAV-based publication channel. For details about the SAS Publishing Framework, see the [Publishing Framework](#) chapter in this guide, and [Publishing Framework](#) in the *SAS Integration Technologies Developer's Guide*.

The following scenario shows a portal's publish and subscribe setup for sales and executive teams that need different access to read (subscribe to) and write (publish) information that is stored in three different directories on the Xythos WFS WebDAV server. On the SAS Metadata Server, these teams are represented by two groups, `Americas Sales` and `Sales Executives`. In addition, the portal installation provides a group named `Portal Admins`, which has unrestricted access to the portal's metadata on the SAS Metadata Server. In this scenario, the `Portal Admins` group will also be given read, write, and delete access to all group-based directories on the Xythos WFS WebDAV server.

This publish and subscribe scenario has a requirement for three different content areas, or group folders on the WebDAV server:

- **Catalog Sales:** The `Catalog Sales` directory contains catalog sales information. The `Americas Sales` and `Sales Executives` groups can both read (subscribe to) and write (publish) information.
- **Field Sales:** The `Field Sales` directory contains direct sales information. The `Americas Sales` and `Sales Executives` groups can both read (subscribe to), but only the `Executives` group can write (publish) information.
- **Sales Execs:** The `Sales Execs` directory contains executive-level sales information and only the `Sales Executives` group can read (subscribe to) and write (publish) information.

Note: The `Portal Admins` group can also read (subscribe to), write (publish), and delete information all of the above directories.

The following table summarizes this scenario's group-based folders on the WebDAV server, and the permissions for each user:

Folder	Americas Sales	Sales Executives	Portal Admins
/Catalog Sales	Read, Write	Read, Write	Read, Write, Delete
/Field Sales	Read	Read, Write	Read, Write, Delete
/Sales Execs	(none)	Read, Write	Read, Write, Delete

To create this sample Xythos configuration, follow these steps:

1. [Install the Xythos WFS WebDAV server.](#)
2. [Create users, groups, and logins on the metadata server.](#)
3. [Create content folders on the Xythos server.](#)

4. Configure access permissions on the Xythos server.

Step 1: Install the Xythos WFS WebDAV server

Install Xythos WebFile Server and the SAS User Management Customization. For details, see the installation instructions on the Xythos Webfile Server CD. Enter the following values in the SAS User Management Customization installation screen:

Metadata Server hostname: your SAS Metadata Server machine name

Metadata Server port: SAS Metadata Server port

Metadata repository name: SAS Metadata Repository name, e.g., Foundation

Unrestricted user: an unrestricted user, e.g. sasadm

To understand and set up unrestricted access and server administrative privileges, see [Server Administrative Privileges](#) in the *SAS Metadata Server: Setup Guide*

Password: password for the unrestricted user.

Trusted user: the trusted user, e.g. sastrust.

To understand and set up a trusted user for the SAS Metadata Server, see [Trusted User](#) in the *SAS Metadata Server: Setup Guide*.

Authentication domain for SAS Metadata server: an authentication domain, e.g. DefaultAuth

Authentication domain for WFS WebDAV server: an authentication domain, e.g. DefaultAuth

Note: When you install the SAS User Management Customization, it is recommended that you specify the same authentication domain name for both the SAS Metadata Server and the Xythos WFS WebDAV server (for example, DefaultAuth). For details about when to specify different authentication domains for the SAS Metadata Server and Xythos WebDAV server, see the documentation for the SAS User Management Customization installation.

If you define a WebDAV server on the SAS Metadata Server, in the authentication domain field, specify the authentication domain that you specified for the Xythos WFS WebDAV server during the installation of the SAS User Management Customization.

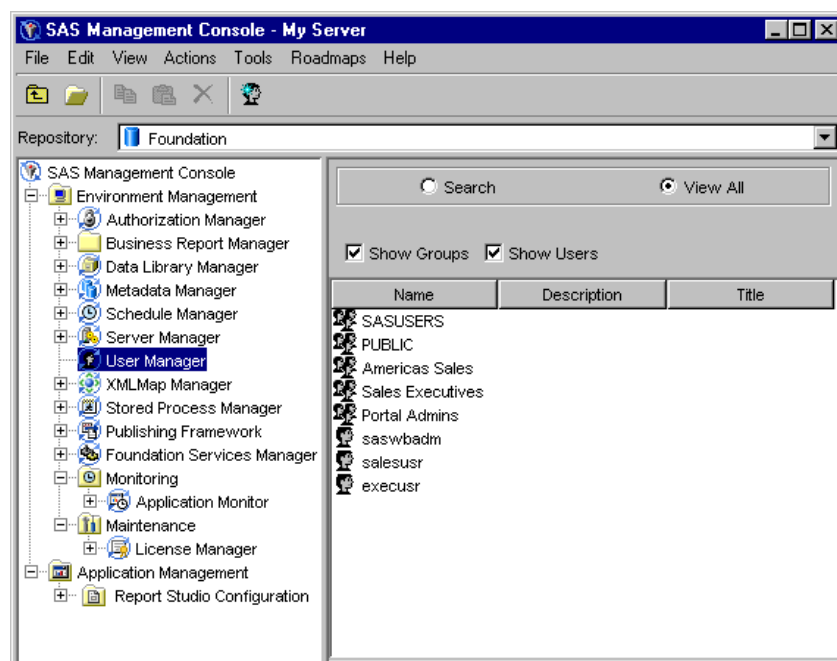
Step 2: Create Users, Groups, and Logins on the SAS Metadata Server

Define the users, groups, and login credentials that will access the WebDAV server. When you define login credentials, you must specify the same authentication domain name that you specified for the Xythos WFS WebDAV server during the SAS User Management Customization installation. For this example, define the following users, groups, and logins:

Group Metadata Identities	User Metadata Identities	Logins	
		User ID	Authentication Domain
Americas Sales	salesusr	salesusr	DefaultAuth

Scenario: Using the Xythos Administration GUI and SAS User Management Customization with the Portals

Portal Admins	saswbadm	saswbadm	DefaultAuth
Sales Executives	execusr	execusr	DefaultAuth



For details about configuring the metadata in SAS Management Console, see [Defining Users, Groups, and Logins on the SAS Metadata Server](#).

Step 3: Create Content Folders on the Xythos Server

To create the content folders on the Xythos Server, follow these steps:

1. Open the Xythos Administration interface in your Web browser. The default URL is `http://localhost:8300/xythosadmin`.
2. Enter your Xythos administrator username (default = "admin") and password (default = (nothing)).
3. Create three top-level directories: Catalog Sales, Field Sales, and Sales Execs.

To create a top-level directory, follow these steps:

- a. Click **FILE SYSTEM ➤ Add a Top-Level Directory**. The Add New Top-Level Directory page appears.

Add New Top-Level Directory

Name: Virtual Server:

Document Store:

Owner ID: Owner Location:

Trash Can: Bandwidth (MB):

Quota: ☐ Unlimited ☒ Limited MB Protect Quota:

To:

- b. Specify a **Name** for the new directory and click **Create Top-Level Directory** to create the new directory.

Note: Ignore any messages that state "The directory does not have an owner"—directory ownership is not a requirement for the SAS User Management Customization.

Step 4: Configure Access Permissions on the Xythos Server

To configure the access permissions for the content folders, follow these steps:

1. In the Xythos Administration GUI, click **FILE SYSTEM ▶ Directory & File Admin**. The Directory Administration page appears.
2. Click **Find Top-Level Directory** to display a list of top-level directories on the server. The Directory Administration: Top-Level Directories page appears.

Directory Administration: defaultVirtualServer

Top-Level Directories

Directory Name	Size	Available Quota	Properties	Permissions	Delete
/Catalog Sales	0	20.00M			
/Field Sales	0	20.00M			
/Sales Execs	0	20.00M			

3. Set the access permissions for each directory:
 - a. On the Directory Administration: Top-Level Directories page, click the icon for the directory for which you want to set access permissions. The Directory Administration: Access Permissions page appears.

- b. Click **Search for Users and Groups**. The Find Users and Groups: Access Permissions page appears.
- c. Click **OK** to display a list of users and groups defined on the SAS Metadata Server.

Find Users and Groups:

"defaultVirtualServer/Catalog Sales" Access Permissions

Choose From These Users/Groups

ID	Location	Display Name
<input type="checkbox"/> admin	defaultVirtualServer	admin
<input type="checkbox"/> saswbadm	defaultVirtualServer	saswbadm
<input type="checkbox"/> salesusr	defaultVirtualServer	salesusr
<input type="checkbox"/> executr	defaultVirtualServer	executr
<input type="checkbox"/> SASUSERS	defaultVirtualServer	SASUSERS
<input type="checkbox"/> PUBLIC	defaultVirtualServer	PUBLIC
<input type="checkbox"/> Americas Sales	defaultVirtualServer	Americas Sales
<input type="checkbox"/> Sales Executives	defaultVirtualServer	Sales Executives
<input type="checkbox"/> Portal Admins	defaultVirtualServer	Portal Admins

- d. Select the check boxes for Americas Sales, Sales Executives, and Portal Admins groups and click **OK** to return to the Directory Administration: Access Permissions page.

Directory Administration: "defaultVirtualServer/Catalog Sales" Access Permissions

Parent directory path: /

Access Control Entry

Grant to:	Location	Read	Write	Delete	Permissions	Inherit Read	Inherit Write	Inherit Delete
Owner (nobody)	N/A	Yes	<input type="button" value="Yes"/>	<input type="button" value="Yes"/>	Yes	<input type="button" value="Yes"/>	<input type="button" value="Yes"/>	<input type="button" value="Yes"/>
Users with accounts	N/A	<input type="button" value="No"/>	<input type="button" value="No"/>	<input type="button" value="No"/>	<input type="button" value="No"/>	<input type="button" value="No"/>	<input type="button" value="No"/>	<input type="button" value="No"/>
Public	N/A	<input type="button" value="No"/>	<input type="button" value="No"/>	<input type="button" value="No"/>	<input type="button" value="No"/>	<input type="button" value="No"/>	<input type="button" value="No"/>	<input type="button" value="No"/>
Americas Sales	defaultVirtualServer	<input type="button" value="Yes"/>	<input type="button" value="Yes"/>	<input type="button" value="No"/>	<input type="button" value="No"/>	<input type="button" value="Yes"/>	<input type="button" value="Yes"/>	<input type="button" value="No"/>
Sales Executives	defaultVirtualServer	<input type="button" value="Yes"/>	<input type="button" value="Yes"/>	<input type="button" value="No"/>	<input type="button" value="No"/>	<input type="button" value="Yes"/>	<input type="button" value="Yes"/>	<input type="button" value="No"/>
Portal Admins	defaultVirtualServer	<input type="button" value="Yes"/>	<input type="button" value="Yes"/>	<input type="button" value="Yes"/>	<input type="button" value="No"/>	<input type="button" value="Yes"/>	<input type="button" value="Yes"/>	<input type="button" value="Yes"/>
Add user:	<input type="text" value="defaultVirtualServer"/>	<input type="button" value="No"/>	<input type="button" value="No"/>	<input type="button" value="No"/>	<input type="button" value="No"/>	<input type="button" value="No"/>	<input type="button" value="No"/>	<input type="button" value="No"/>
Add group:	<input type="text" value="defaultVirtualServer"/>	<input type="button" value="No"/>	<input type="button" value="No"/>	<input type="button" value="No"/>	<input type="button" value="No"/>	<input type="button" value="No"/>	<input type="button" value="No"/>	<input type="button" value="No"/>

☒ Apply changed settings to sub-directories and files
☐ Overwrite all permissions on all sub-directories and files
☐ Only apply to this directory

- e. Set the access permissions as appropriate for the directory:

Group	Permissions for Catalog Sales					
	Read	Write	Delete	Inherit Read	Inherit Write	Inherit Delete
Americas Sales	Yes	Yes	No	Yes	Yes	No
Portal Admins	Yes	Yes	Yes	Yes	Yes	Yes
Sales Executives	Yes	Yes	No	Yes	Yes	No

Group	Permissions for Field Sales					
	Read	Write	Delete	Inherit Read	Inherit Write	Inherit Delete

Americas Sales	Yes	No	No	Yes	No	No
Portal Admins	Yes	Yes	Yes	Yes	Yes	Yes
Sales Executives	Yes	Yes	No	Yes	Yes	No

Group	Permissions for Sales Execs					
	Read	Write	Delete	Inherit Read	Inherit Write	Inherit Delete
Americas Sales	No	No	No	No	No	No
Portal Admins	Yes	Yes	Yes	Yes	Yes	Yes
Sales Executives	Yes	Yes	No	Yes	Yes	No

Note: In addition to the basic **Read**, **Write**, and **Delete** permissions, you should also set the corresponding *inherit permissions*. Inherit permissions apply to any new files that are created in the directory. For example, if a user has the **Read** permission for a directory, but does not have the **Inherit Read** permission, the user can read the directory itself, but cannot necessarily read the files in the directory.

- f. Click **Save Changes** to apply the new access permissions.

Security

Implementing Encryption with Integration Technologies

You can implement encryption for COM/DCOM and IOM Bridge Server connections:

- **For COM/DCOM connections**, encryption is enabled by using an *AuthenticationLevel* of *Packet Privacy*. By default, DCOM uses the RC2 encryption algorithm. You can set the authentication level for a DCOM object using the Windows dcomcnfg utility.
- **For IOM Bridge Server connections**, the IOM Bridge for Java and IOM Bridge for COM have the ability to encrypt all messages exchanged with the IOM server, using a two-tiered security solution. The first tier is the SASProprietary encryption algorithm. The second tier is made up of standards-based RC2, RC4, DES, and Triple DES encryption algorithms.
 - ◆ The first-tier encryption algorithm, the SAS proprietary encryption algorithm (SASProprietary), is appropriate for use in applications where you want to prevent accidental exposure of information while it is being transmitted over a network between an IOM Bridge and an IOM server. Access to this encryption algorithm is included with your Base SAS license, and the Java and Windows implementations are integrated into the IOM Bridge for Java and the IOM Bridge for COM.
 - ◆ The second-tier encryption algorithms are appropriate for use in applications where you want to prevent exposure of secret information. Using these algorithms makes it extremely difficult to discover the content of messages exchanged between an IOM Bridge for Java (or IOM Bridge for COM) and an IOM server. To use these algorithms you must license the SAS/SECURE software.

Specifying Server Encryption Settings for IOM Bridge Connections

To enable encryption for an IOM Bridge connection, you must specify an encryption algorithm and an encryption level.

Specifying the Encryption Algorithm.

Depending on how your server is configured, do one of the following:

- **For servers that are not configured using SAS Management Console**, specify an encryption algorithm using the NETENCRYPTALGORITHM system option in the server startup command. The NETENCRYPTALGORITHM option can also be specified as NETENCALG. The syntax for this option is

```
-NETENCRYPTALGORITHM "algorithm" | ("algorithm", "algorithm" ...)
```

Where *algorithm* is one of the following values:

- ◆ SASProprietary
- ◆ RC2
- ◆ RC4
- ◆ DES
- ◆ TripleDES

Note: If you do not have a license for SAS/SECURE, you can only specify the SASProprietary algorithm.

There is no default encryption algorithm for servers that are not configured using SAS Management Console.

- **For servers that are configured using SAS Management Console**, you can specify an encryption algorithm using either the NETENCRYPTALGORITHM system option or the Server Encryption Algorithms field (in

SAS Management Console: <**Connection**> ➔ **Options** ➔ **Advanced Options** ➔ **Encryption** ➔ **Server Encryption Algorithms**). If you specify a value both in the server command and in the Server Encryption Algorithms field, the value from the server command is used.

The default algorithm for servers that are configured using SAS Management Console is SASPROPRIETARY.

Specifying the Encryption Level

Depending on how your server is configured, do one of the following:

- **For servers that are not configured using SAS Management Console**, specify the encryption level using the CLIENTENCRYPTIONLEVEL object server parameter. You can specify the following values:

NONE

nothing is encrypted.

CREDENTIALS

the login credentials are encrypted

EVERYTHING

all client–server communications are encrypted

Note: CLIENTENCRYPTIONLEVEL can also be specified as CEL.

Servers that are not configured using the SAS Management Console have a default encryption level of **none**.

- **For servers that are configured using SAS Management Console**, you can specify the encryption level using either the CLIENTENCRYPTIONLEVEL object server parameter or the Required Encryption Level field (in SAS Management Console: <**Connection**> ➔ **Options** ➔ **Encryption** ➔ **Required Encryption Level**) . If you specify a value both in the server command and in the Required Encryption Level field, the value from the server command is used.

Servers that are configured using the SAS Management Console have a default encryption level of **credentials**.

Specifying Server Encryption Settings for DCOM Connections

Encryption for DCOM connections is dependent on your Windows DCOM settings. If you enable encryption for a DCOM connection, all communications between the client and server are encrypted using the RC2 algorithm. SAS/SECURE is not required to use RC2 with DCOM.

To enable encryption for DCOM connections, perform the following steps:

Windows NT/2000

1. From the Windows taskbar, select **Start** ➔ **Run**.
2. Type dcomcnfg and click **OK**. The Distributed COM Configuration Properties dialog box appears.
3. Select the Applications tab. This tab displays a list of AppIDs. To determine which AppID corresponds to your IOM server, see AppIDs for Configuring DCOM.
4. Select the AppID for the type of IOM server that you wish to set encryption for. Click **Properties**. The Properties dialog box for the selected IOM server appears.
5. On the General tab, expand the Authentication Level drop–down list and select **Packet Privacy**.

- Click **Apply** to apply the settings and **OK** to close the dialog box.

Windows XP

- From the Windows taskbar, select **Start ➤ Run**.
- Type `dcomcnfg` and click **OK**. Component Services window appears.
- In the left panel, expand the entries as follows: **Component Services ➤ Computers ➤ My Computer ➤ DCOM Config**.
- From the left panel, select **DCOM Config**. In the right panel, a list of AppIDs appears. To determine which AppID corresponds to your IOM server, see [AppIDs for Configuring DCOM](#).
- Select the AppID for the type of IOM server that you wish to set encryption for. Click **Properties**. The Properties dialog box for the selected IOM server appears.
- On the General tab, expand the Authentication Level drop-down list and select **Packet Privacy**.
- Click **Apply** to apply the settings and **OK** to close the dialog box.

Specifying Client Encryption Settings

Depending on which type of client you are configuring, see the appropriate security section for client encryption settings:

- For Java clients, see the [com.sas.services.connection](#) class documentation in the *SAS Integration Technologies Developer's Guide* for details about how to use the encryption features.
- For Windows clients, see [Windows Client Security](#) in the Windows Clients chapter of the *SAS Integration Technologies Developer's Guide* for details on how to use encryption.

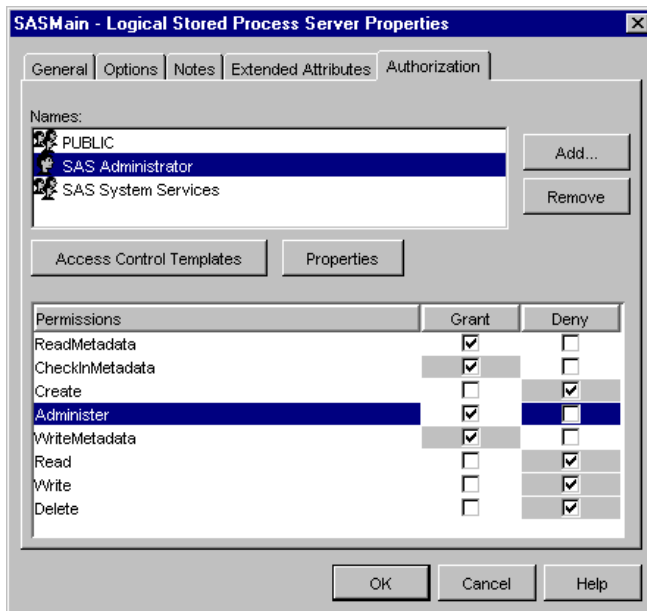
Security

Setting Up Additional Server Security

Depending on your security implementation, you might want to enable additional users to perform administrative functions or allow additional users access to public interfaces on the servers. You can set up the following additional security features for your servers and spawners:

- **Administrative Privileges (SAS Metadata Server only).** The user ID that starts the metadata server has unrestricted access to all metadata on the server with no additional configuration required. (This user is called the *unrestricted user*). You can also enable other user IDs to have unrestricted access to the server (as an *unrestricted user*) or additional administrative privileges for some metadata (as an *administrative user*). To understand and set up unrestricted access and server administrative privileges, see [Server Administrative Privileges](#) in the *SAS Metadata Server: Setup Guide*
- **Server-level Administer Permissions (SAS Stored Process and SAS OLAP Servers only).** The user who starts the server has permission to stop, pause, and resume a server. To enable another user to stop, pause, and resume a server, grant the "Administer" permission to that user on the Authorization tab of the logical server definition (in SAS Management Console). Note that if you grant the "Administer" permission on the server definition (rather than the logical server definition), the user will not be able to administer the server.

The following image shows the "Administer" permission being set in SAS Management Console:



- **Anonymous Login Capability (IOM Bridge connections for multi-user servers only).** An anonymous user is a user who does not provide a user ID when connecting to the server. You can allow or deny anonymous login credentials access to the IServerStatus interface of a multi-user IOM server (OLAP, Stored Process or SAS Metadata Server). To allow or deny anonymous login credentials, specify "restrict" or "deny" for the anonymousLoginPolicy option in one of the following places:
 - ♦ on the **Object Server Parameters** field of the server definition's Advanced Options ▶ Launch Commands tab.
 - ♦ in the SAS startup command's `-objectserverparms` option.For example,

`anonymousLoginPolicy=deny`

For details about object server parameters, see [Object Server Parameters](#)

The default for the `anonymousLoginPolicy` option is `restrict`.

Security

Planning Security on Workspace and Stored Process Servers (IOM Bridge Connection Only)

You might choose whether to run a workspace server, pooled workspace server, load-balancing stored process server, or load-balancing workspace server based on your security considerations. (For an overview of the user IDs specified in the configuration, see [Security Metadata](#)). The following table shows several aspects of security for workspace servers, pooled workspace servers, and load-balanced stored process servers:

Workspace and Stored Process Security Considerations				
Security Features	SAS Workspace Server	Pooled SAS Workspace Server	Load-Balancing SAS Stored Process Server	Load-Balancing SAS Workspace Server
Server Reuse	dedicated server per client	sequential reuse (of the server) by clients	efficient (scalable) reuse (of the server) by many simultaneous clients	dedicated server per client
User ID Under Which The Server Runs	client's user ID	<p>puddle login; all users in a puddle run under the puddle login's user ID.</p> <p>CAUTION: A stored process that runs on a pooled workspace server accesses data using the account under which the server is running (i.e. the puddle login). Because your account is not being used to access the data, your permissions to the data are not relevant. In these circumstances, it is particularly important to set appropriate access controls to secure the stored process.</p>	<p>multi-user login; all users for a server run under the multi-user login's user ID.</p> <p>Note: Because the load-balancing stored process server runs under the multi-user login credentials, the operating system account for these credentials must have access to any operating system resources used by stored processes that are hosted on the stored process server.</p> <p>CAUTION: A stored process that runs on a stored process server accesses data using the account under which the server is running (i.e. the multi-user login). Because your account is not being used to access the data, your</p>	client's user ID

			permissions to the data are not relevant. In these circumstances, it is particularly important to set appropriate access controls to secure the stored process.	
Client Authentication	client's credentials must be valid on the server's host authentication provider	clients mapped to puddles of servers; clients' user IDs must be valid on the SAS Metadata Server's authentication provider	client's credentials must be valid on the server's host authentication provider	client's credentials must be valid on the server's host authentication provider
Metadata Access Requirements for User IDs Important Note: DO NOT specify an <i>unrestricted user</i> for either the user ID in the spawner's metadata configuration file or the user ID for the pool administrator.	user ID in the spawner's metadata configuration file must be able to view the following user ID: <ul style="list-style-type: none"> operator login, if one is specified. 	user ID in the spawner's metadata configuration file must be able to view the following user ID: <ul style="list-style-type: none"> operator login, if one is specified. user ID in the pool's metadata configuration file or pooling connection request (the pool administrator's credentials) must be able to view the following user ID: <ul style="list-style-type: none"> puddle login 	user ID in the spawner's metadata configuration file must be able to view the following user IDs: <ul style="list-style-type: none"> operator login, if one is specified. multi-user login logical server credentials 	user ID in the spawner's metadata configuration file must be able to view the following user ID: <ul style="list-style-type: none"> operator login, if one is specified. logical server credentials
Use of METAAUTOINIT to Connect Back to the SAS Metadata Server	allowed, not specified by default	allowed, specified by default for COM and not specified by default for IOM Bridge	allowed, not specified by default	allowed, not specified by default
When using METAUTOINIT, Server Security for Connecting Back to the SAS Metadata Server	if the trustsaspeer option is specified, connects using the client's user ID	if the trustsaspeer option is specified, connects using the puddle login	if the trustsaspeer option is specified, connects using the multi-user login	if the trustsaspeer option is specified, connects using the client's user

	if the trustsaspeer option is NOT specified, use the required META* options to specify the client user ID	if the trustsaspeer option is NOT specified, use the required META* options to specify the puddle login	if the trustsaspeer option is NOT specified, use the required META* options to specify the multi-user login	ID if the trustsaspeer option is NOT specified, use the required META* options to specify the client user ID
--	---	---	---	---

For details about the use of METAAUTOINIT and how to specify security, see [Specifying Metadata Connection Information](#).

Security

Planning the Spawner Security

When you set up a spawner configuration, you specify login credentials or definitions in two locations:

- **login definitions in the server and spawner configuration.** When you configure the spawner and server definitions on the SAS Metadata Server, you might specify certain login definitions in the configuration.
- **login credentials in the spawner's metadata configuration file.** When you use a spawner to start a server, you specify a metadata configuration file that contains information to allow the spawner to access the SAS Metadata Server for server and spawner metadata information. When you create a metadata configuration file for the spawner to use to access the SAS Metadata Server, you specify a fully qualified user ID and password to use to connect to the SAS Metadata Server.

Note: You can use the same credentials that you use for accessing the SAS Metadata Server with the SAS servers "-metaprofile" options or Windows Object Manager credentials. Using common credentials allows for easier configuration.

The login credentials that you specify in the spawner's metadata configuration file must be able to do both of the following:

- access the SAS Metadata Server
- view login definitions that are specified in the spawner and associated server definitions on the SAS Metadata Server.

Therefore, you must plan appropriately for the user ID in the spawner's metadata configuration file, and for login definitions in the server and spawner configuration. In addition, you must define these login credentials on the appropriate authentication provider. For details, see [Understanding Spawner Authentication](#)

Understanding Spawner/Server Login Configuration and Access

In the spawner and server definitions (on the SAS Metadata Server), you can specify the following login definitions:

- operator login definition for the spawner (specified in the spawner definition).
- for SAS Stored Process Servers, multi-user login definition (specified on the Credentials tab of the server definition).

The login credentials that are used to access the SAS Metadata Server (i.e. the user ID in the spawner's metadata configuration file) must be able to access the previously mentioned server and spawner login definition in the configuration's SAS Metadata Repository. The SAS Metadata Server allows a user ID to read login definitions if one of the following conditions are true:

- the login definitions are owned by the user ID's user or group metadata identity.
- the login definitions are group (shared) login definitions that the user ID can access as part of a group metadata identity.

Important Note: DO NOT specify an *unrestricted user* for the user ID in the spawner's metadata configuration file.

The following table summarizes the credentials required for spawner security configuration:

Locations Where Credentials Are Specified for Spawner Configuration
--

User ID or Login Definition	Location Where Credentials Are Specified	Description
User ID in the Spawner's Metadata Configuration File	Metadata Configuration File Project Installation Note: For a project installation (using SAS Configuration Wizard), the metadata configuration file named <code>OMRConfig.xml</code> (located in the <code>ObjectSpawner</code> directory) contains the SAS Trusted User credentials.	Login credentials in the metadata configuration file. You must specify the login credentials that the spawner will use to connect to the SAS Metadata Server. This user ID must be able to access the operator login (ID) and if specified, the multi-user login definition. Important Note: DO NOT specify an <i>unrestricted user</i> for the user ID in the metadata configuration file.
Operator Login for Spawners (for spawner, optional)	<i>In the SAS Management Console spawner definition:</i> Initialization: Operator Login ➔ Operator Login Project Installation Note: For a project installation (using SAS Configuration Wizard), the operator login is not specified by default.	Administrator login definition to access the operator port of the spawner. The login definition must be one of the following: <ul style="list-style-type: none"> the login definition for the user ID that you specified in the metadata configuration file a login definition that the user ID in the metadata configuration file can access
Multi-User Login for SAS Stored Process Servers	<i>In SAS Management Console stored process server definition:</i> Options ➔ Advanced Options ➔ Credentials ➔ Login Project Installation Note: For a project installation (using SAS Configuration Wizard), the login for the SAS General Servers group is specified.	Login for the multi-user server. The launched SAS process runs under the process ID defined by this login. The login definition must be one of the following: <ul style="list-style-type: none"> the login definition for the user ID that you specified in the metadata configuration file a login definition that the user ID in the metadata configuration file can access

Enabling the User ID in the Spawner's Metadata Configuration File to View Spawner/Server Login Definitions

To enable the user ID in the spawner's metadata configuration file to access the spawner and server configuration login definitions on the SAS Metadata Server, the user ID in the metadata configuration file must be one of the following:

Important Note: DO NOT specify an *unrestricted user* for the user ID in the spawner's metadata configuration file.

- the same user ID as the user ID of the operator login definition (in the spawner definition) and, for SAS Stored Process Servers, the same user ID as the multi-user login definition (in the server definition).
- a member of a group metadata identity in which the multi-user login definition (SAS Stored Process Servers

only) and the operator login definition are login definitions owned by the group (or groups). You can do either of the following:

- ◆ create a group metadata identity and use the same group (shared) login definition for the multi-user (SAS Stored Process Servers only) and operator login definition.
- ◆ for SAS Stored Process Servers only, create a group metadata identity with a group (shared) login definition (e.g., for the multi-user login definition) and then add that group metadata identity to another group with a group (shared) login definition (e.g., for the operator login definition). The second group metadata identity must also contain a login definition for the user ID specified in the spawner's metadata configuration file.

To create a group metadata identity with a group (shared) login definition:

1. Create a group metadata identity.
2. Create a login definition (that uses the shared user ID) for the new group.
3. Add any user metadata identities (or another group with a group (shared) login definition) to the group as members.

In addition, if you are setting up load balancing, the user ID in the spawner's metadata configuration file must be able to access (under one of the above three conditions) the user ID that you specify for the logical server credentials login definition (on the load-balancing logical server definition). For details, see [Planning the Load Balancing Security](#).

Project Installation Note: For a project installation (using SAS Configuration Wizard), the login in the spawner's metadata configuration file (e.g. `sastrust`) can view the multi-user login definition for the stored process server (e.g. `sassrv`) because the SAS Trusted User (which owns `sastrust`) is a member of the SAS General Server group that owns the multi-user login definition (`sassrv`). With the project installation, no operator login is specified.

Understanding Spawner Authentication

When you implement a spawner and server configuration, the login definitions in the spawner and server configuration, and the clients who connect to the servers must be authenticated against the appropriate authentication provider. Depending on the type of spawner and server setup, spawner and client authentication works as follows:

Locations Where Credentials Are Authenticated		
Type of Credentials	User ID Role	Authentication Location
Standard Spawner and Server Configuration	user ID of the operator login	no authentication
	for SAS Stored Process Servers, the user ID of the multi-user login	host authentication provider on the SAS Stored Process Server's machine
Connections to Standard Spawner and Server	the client user ID	host authentication provider on the SAS Workspace or SAS Stored Process Server's machine
Connections to Pooled Server	user ID of the puddle login	host authentication provider on the SAS Workspace Server's machine
	user IDs associated with the user metadata identities that are members of the group metadata identity that is granted access to	SAS Metadata Server's authentication provider

	the pool	
	user IDs associated with the pool administrator	SAS Metadata Server's authentication provider
Load–Balancing Logical Server Configuration	user ID of the load–balancing logical server credentials	host authentication provider on the SAS Workspace or SAS Stored Process Server's machine and the host authentication provider on the machine of the other spawners to which it connects
Connections to Load–Balancing Server	the client user ID	host authentication provider on the SAS Workspace or SAS Stored Process Server's machine

For details about defining users for authentication, see [Implementing Authentication](#)

When the spawner starts the server process, the process runs under the following credentials

- for SAS Workspace Servers, the credentials of the connecting client
- for SAS Stored Process Servers, the multi–user login credentials that are specified in the stored process server definition (Advanced Options ➤ Credential) in SAS Management Console.

IOM Bridge

Spawner Security Scenario

The following scenario shows a recommended setup for spawner and server security. In this scenario, the object spawner runs on the server host, monitors client requests for the stored process and workspace server, and connects clients to the appropriate server process.

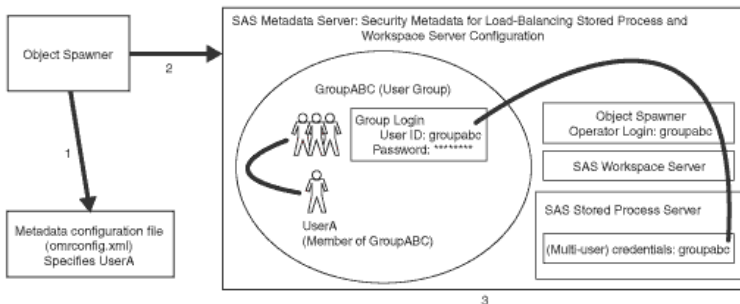
The SAS Metadata Server contains the spawner, server, and security metadata for the load-balancing stored process server and workspace server configuration. The object spawner must connect to the SAS Metadata Server, and the metadata must be appropriately configured to enable the spawner to start the load-balancing stored process server or workspace server.

Project Installation Note: For a project installation (using SAS Configuration Wizard), within this scenario:

- UserA and `usera` correspond to the SAS Trusted User and its user ID (e.g. `sastrust`).
- GroupABC and `groupabc` correspond to the SAS General Servers group and its user ID (e.g. `sassrv`).

The following diagram shows the initial security setup and process flow for the load-balancing stored process server, workspace server, and spawner configuration:

Note: On Windows, all user IDs would be machine- or domain-qualified. For example, `europa\usera`.



In the previous diagram, the Object Spawner obtains the metadata information to start a load-balancing stored process server or workspace server as follows:

1. When the spawner is started, it reads a metadata configuration file (`omrconfig.xml`) to access the SAS Metadata Server. This metadata configuration file specifies the location of the SAS Metadata Server and the user ID that the spawner will use to connect to the metadata server.

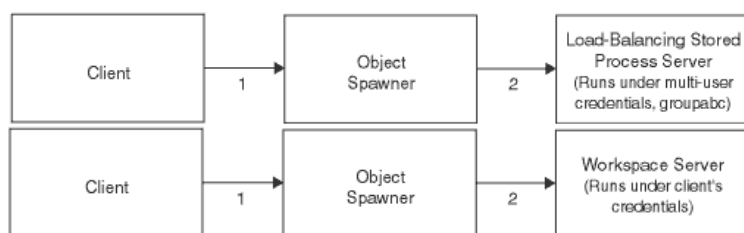
In this example, the `omrconfig.xml` file contains the user ID `usera`, which is owned by the UserA user.

2. The object spawner connects to the SAS Metadata Server using the user ID that is specified in `omrconfig.xml`. UserA's credentials are authenticated against the SAS Metadata Server's authentication provider.
3. On the SAS Metadata Server, the connection from the object spawner is associated with the user metadata identity that owns the `usera` user ID, UserA. The spawner (as UserA) reads the metadata information for the server and spawner configurations.

Note: The user metadata identity UserA can view both the stored process server's multi-user login credentials and the operator login (`groupabc`) because UserA is a member of the group metadata identity GroupABC, and GroupABC owns both the server's multi-user login credentials and operator login (`groupabc`).

The object spawner then has the necessary metadata to launch a workspace or stored process server. The following

diagrams show the flow for a client request and a stored process server or workspace server launch.



1. When a client requests a server, the client is authenticated against the host authentication provider for the server.
2. If the object spawner needs to launch a new stored process server, the object spawner uses the server's multi-user login credentials (`groupabc`) to launch the load-balancing stored process server.

If the object spawner needs to launch a new workspace server, the object spawner uses the client's credentials to launch the workspace server. All further communications between the client and the server are direct, rather than through the object spawner.

Note: Because the stored process server runs under the credentials for the multi-user stored process server, each client can only access information for which the multi-user credentials are authorized.

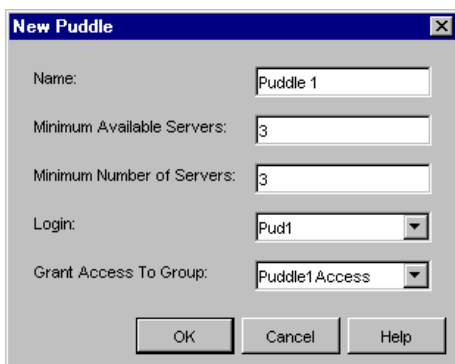
Security

Planning the Pooling Security (IOM Bridge only)

Note: For SAS Integration Technologies 9.1, you can only set up pooling for SAS Workspace Servers.

For an overview of pooling, see [Overview of Pooling](#). To set up pooling metadata on the SAS Metadata Server, see [Pooling Metadata](#).

A pool consists of one or more puddles. You use the Server Manager, User Manager, and Authorization Manager plug-ins of the SAS Management Console to create the pool metadata on the SAS Metadata Server; the pool metadata consists of a pooled logical server definition that contains server definitions and puddle definitions. The puddle definitions specify which group metadata identity can use each puddle in the pool. In addition, for each puddle, you also set up a specific user ID and password (i.e., the puddle login) to connect to a SAS IOM server and create a pooled connection. The following screen shot shows the SAS Management Console dialog box for a puddle definition:



Pooling works as follows:

1. An application is authorized to use a pool by obtaining access to the credentials of the pool administrator.
2. The application uses the pool administrator's credentials to connect and authenticate itself to the SAS Metadata Server. The pool administrator's credentials must be able to view the metadata for all the logins (puddle logins) that are used to make connections for the pool.
3. The pool administrator uses the puddle login credentials to create pooled connections to server objects.
4. The pooled connections are shared and reused by multiple clients. Each puddle has a set of clients (users that are members of the group metadata identity that is granted access to the puddle) who are permitted to use the puddle.

Note: The pool administrator cannot view the login definitions of the clients using the pool.

The SAS Metadata Server administrator might choose to partition a pool of connections into several puddles to control the data that users are authorized to access. Because the SAS server uses the puddle login credentials to both connect to the metadata and run the server process, this authorization (access control) can be applied in the metadata or on the actual data (using filesystem authorization). For example, the metadata administrator might give one puddle read and write access to a table on an IOM server, while giving another puddle only read access. In such a case, the Java Connection Factory or Windows Object Manager automatically routes a user's request for a connection to a puddle that the user is authorized to use. (It is not important to distinguish between pools made up of several puddles and pools made up of only one puddle).

Note: In addition, Windows and Java clients have an option to check the caller's credentials to verify that the caller has been granted permission to use the credentials that were used to establish the connection to SAS.

Overview of Pool and Puddle Configuration

To configure a pooled logical server, you must use SAS Management Console to set up one or more puddles for the pool. A puddle consists of the servers within the pooled logical server and the puddle's pooling parameters. To configure each puddle's security, you specify:

- **a puddle login definition that is used to connect to the SAS IOM server.** Because each puddle can have only one login definition, you must define a puddle for each domain that needs to access the pool. When you define a login definition for the puddle, the user or group metadata identity that is associated with the login definition also has access to the puddle.
- **a group metadata identity whose members (user metadata identities or other group metadata identities with associated login definitions) can also access the puddle (optional).** The login definitions associated with the group (and users that are members of the group) are not required to have the same authentication domain as the servers in the puddle.

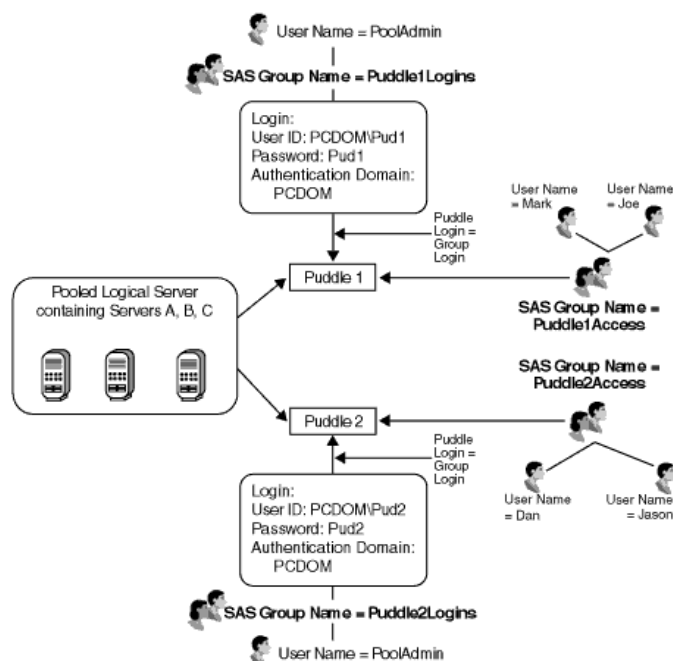
In addition, you must set up a user metadata identity who will be the pool administrator (used in the Windows Object Manager metadata configuration file) and enable him or her to view all the puddle login(s) definitions for the pool. (No other users need to be able to view these login definitions).

The following table summarizes the credentials required for pooling security configuration:

Locations Where Credentials Are Specified for Pooling		
User ID or Login Definition	Location Where Credentials Are Specified	Description
User ID in the Spawner's Metadata Configuration File	Metadata Configuration File Project Installation Note: For a project installation (using SAS Configuration Wizard), the metadata configuration file named <code>OMRConfig.xml</code> (located in the <code>ObjectSpawner</code> directory) contains the SAS Trusted User credentials.	Login credentials in the metadata configuration file. You must specify the login credentials that the spawner will use to connect to the SAS Metadata Server. This user ID must be able to access the operator login (ID) and if specified, the multi-user login definition. Important Note: DO NOT specify an <i>unrestricted user</i> for the user ID in the metadata configuration file.
Operator Login for Spawners (for spawner, optional)	<i>In the SAS Management Console spawner definition:</i> Initialization: Operator Login ➔ Operator Login Project Installation Note: For a project installation (using SAS Configuration Wizard), the operator login is not specified by	Administrator login definition to access the operator port of the spawner. The login definition must be one of the following: <ul style="list-style-type: none"> • the login definition for the user ID that you specified in the

	default.	metadata configuration file <ul style="list-style-type: none"> • a login definition that the user ID in the metadata configuration file can access
Login for Pool Administrator	Supplied by Application and <i>In SAS Management Console login definition</i>	Administrator login (credentials) supplied by application to connect to the SAS Metadata Server and read the puddle login definitions.
Puddle Login	<i>In SAS Management Console pooled logical server definition:</i> Options ➤ Puddles ➤ Login	Login definition that is used to establish the connection to the server for this puddle.
Logins for Users in the Group that is Granted Access to the Puddle	<i>In SAS Management Console pooled logical server definition:</i> Options ➤ Puddles ➤ Grant Access to Group	Logins for users in a SAS group that is granted access to a puddle.

Pool and Puddle Security Scenario: Recommended Configuration



The diagram shows a pool that consists of the following server and security definitions on the SAS Metadata Server:

- **a pooled logical server with two puddle definitions and three server definitions.** The pool consists of a pooled logical server that has two puddle definitions, Puddle1 and Puddle2; each puddle contains Servers A, B, and C.

- **a user metadata identity for a pool administrator, PoolAdmin**, whose login credentials (not shown) are used by the application to access the puddle logins.
- **two group metadata identities for puddle administration**. The two groups, Puddle1Logins and Puddle2Logins, each contain a unique login definition that is used as the puddle login credentials to connect to SAS for Puddle1 or Puddle2. The pool administrator, PoolAdmin, belongs to both the Puddle1Logins and Puddle2Logins group. Because PoolAdmin is a member of each of the groups that contain the puddle login definitions for Puddle 1 and Puddle 2, PoolAdmin can access both of these puddle login definitions. Note that the login definitions for the puddle logins must have the same authentication domain as the servers in the pool.
- **two group metadata identities for group access to each puddle**. A group named Puddle1Access is granted access to Puddle1. A group named Puddle2Access is granted access to Puddle2. The users who are members of Puddle1Access can access Puddle1; the users who are members of Puddle2Access can access Puddle2. Note that the login definitions for the users DO NOT need to have the same authentication domain as the servers in the pool.

When Joe connects to the logical server, because he is a member of the group Puddle1Access, he connects to Puddle 1. When PoolAdmin connects to the pooled logical server, because he can access puddle logins for both puddles, he might be directed to either Puddle 1 or Puddle 2, depending on which puddle is available.

Planning the Pool and Puddle Security

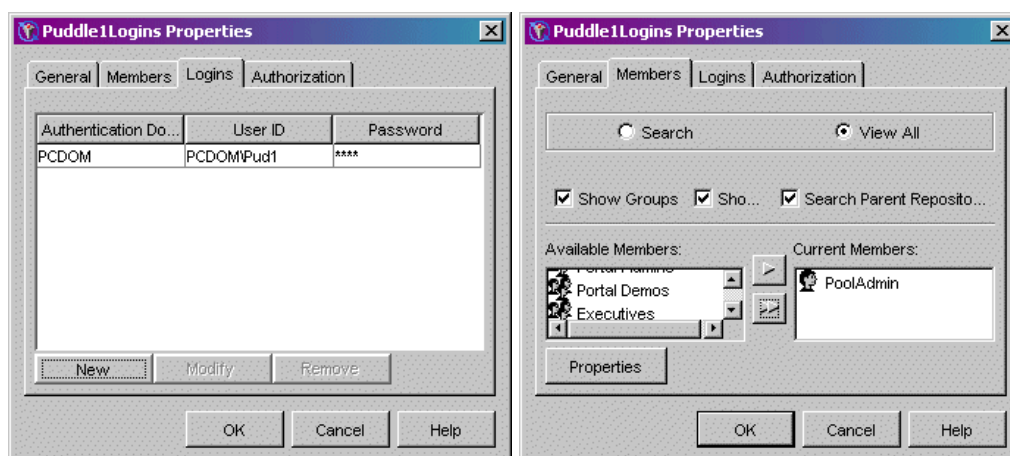
On the SAS Metadata Server, you must configure the puddle login definitions such that the pool administrator can access all of the puddle login definitions in the pool. In addition, you must restrict who can update the user metadata identities who are members of the group metadata identity that is granted access to each puddle. You must also restrict who can access data on the server. To plan for appropriate pooling security, follow these steps:

1. **For each puddle, plan for the puddle login definition and the group metadata identity for the puddle administrator group.** To set up puddle login definitions for each puddle, you must enable the pool administrator to view all of the puddle login definitions for the pool. For details about which user IDs can view other login definitions, see [Enabling the User ID in the Spawner's Metadata Configuration File to View Spawner/Server Login Definitions](#). To set up a puddle login definition and pool administrator for a puddle, plan to implement the following SAS login definition, user, and group structure:
 - a. A group metadata identity that has the puddle login definition as the group (shared) login definition
 - b. Depending on which user ID is used for the pool administrator, a pool administrator that is set up as follows:

Important Note: DO NOT set up an *unrestricted user* as the pool administrator.

- ◇ If the pool administrator's user ID is the same user ID as the puddle login definition's user ID, no additional user and group setup is required.
- ◇ If the pool administrator's user ID is not the puddle login's user ID, set up a user metadata identity for the pool administrator. Add this user as a member of the group (from Step a) that contains the puddle login definition as the group (shared) login definition. (You can also create a group metadata identity for a group of pool administrators and add that group to the group that contains the puddle login definition as the group (shared) login).

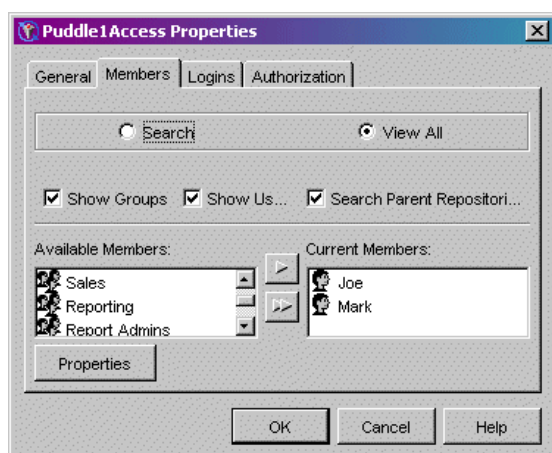
For example, the following screen shots show the Puddle1Logins group with a group (shared) login as the puddle login for Puddle 1 and the PoolAdmin user as a member of the group:



For each puddle, implement the previously described user, group and login definition structure on the SAS Metadata Server.

2. **For each puddle, plan for the group metadata identity and members (user metadata identities) of the group that are granted access to the puddle.** You must set up the group of users that are granted access to the puddle.

For example, the following screen shot shows the Puddle1Access group with the members (users Joe and Mark) who can access Puddle 1:



3. **For each puddle, plan to control access to the group metadata identity that is granted access to the puddle.** You must control access for who is authorized to update the group metadata identity that is granted access to each puddle. To control who can update the group metadata identity that is granted access to the puddle, in SAS Management Console, after you set up the group metadata identity, use the Authorization tab for the group to do both of the following:
 - ◆ Deny "WriteMetadata" permission to the Public group.
 - ◆ Grant "WriteMetadata" permission to your metadata administrator.
4. **Plan to control access to the pooled logical server.** You must control access for who is authorized to update the pooled logical server. To control who can update the pooled logical server, in SAS Management Console, you must use the Authorization tab for the pooled logical server to do both of the following:
 - ◆ Deny "WriteMetadata" permission to the Public group.
 - ◆ Grant "WriteMetadata" permission to your metadata administrator.
5. **For each puddle, plan to control access to the data on the servers.** For each server you must control access to the data on the server, either through authorization (access control) metadata (see the Authorization Manager and User Manager plug-in help in the SAS Management Console) or file system access on your host

system.

Note: SAS Stored Process definitions are accessed under the credentials of the pool administrator; therefore, you cannot implement access control for SAS Stored Process definitions that are accessed by users of the puddle.

How Users Are Authenticated for the Pool

When you implement a pooled configuration, the pool administrator, the puddle login credentials, and the users who access the pool must be authenticated against the appropriate authentication provider. The following table shows the locations where the credentials are authenticated:

Locations Where Credentials Are Authenticated	
User ID Role	Authentication Location
user ID of the puddle login	host authentication provider for the SAS Workspace Server's machine
user IDs of metadata identities that are members of the group metadata identity that is granted access to the pool	SAS Metadata Server's authentication provider
user ID of the pool administrator credentials	SAS Metadata Server's authentication provider

How Users are Authorized to Access the Puddles

When users request a connection from a pool, they are authenticated against the SAS Metadata Server's authentication provider and the pool administrator uses the SAS Metadata Server to obtain the requesting user's metadata identity (user or group). The pool administrator then allocates a connection from the pool to the requesting user as follows:

1. The pool administrator selects a puddle where the requesting user ID matches one of the following:
 - ◆ the puddle login's user ID, or is associated with the user or group metadata identity of the puddle login's user ID.
 - ◆ a user ID associated with the user or group metadata identities that are members of the group granted access to each puddle.
2. The pool administrator returns a connection to the selected puddle as follows:
 - ◆ if a connection is available, the pool administrator returns a connection to the requesting user. The user uses the connection as long as required.
 - ◆ if there are no available connections to the selected puddle, the pool administrator uses the puddle login credentials to create a new connection to a SAS server.
 - ◆ if the selected puddle has already established the maximum allowed number of connections, the requesting user waits for a connection to become available.
3. When the user is finished with a connection, the user releases the connection so it can then be used by a subsequent user.

Security

Planning the Load Balancing Security (IOM Bridge only)

Note: For SAS Integration Technologies 9.1, you can only set up load balancing for SAS Stored Process and SAS Workspace Servers.

For an overview of load balancing, see [Overview of Load Balancing](#). To set up load balancing metadata on the SAS Metadata Server, see [Load Balancing Metadata](#).

A load-balancing configuration uses a spawner to start servers. When you configure the spawner, you must plan for the appropriate login definitions. (For details, see [Planning the Spawner Security](#)). In addition, for load-balancing across spawners, you must plan for additional configuration security; this security will be implemented differently depending upon whether you use a SAS Workspace Server or SAS Stored Process Server:

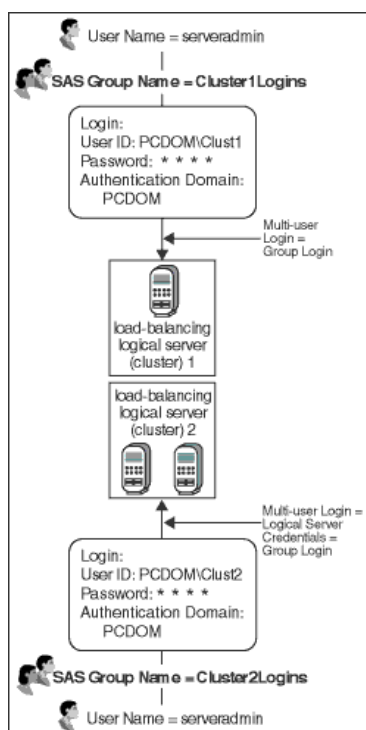
- **For load-balancing SAS Stored Process Servers**, the user ID in the spawner's metadata configuration file must be able to access the multi-user login (specified on the Credentials tab of the server definition) and, if specified, the operator user ID (specified in the spawner definition). In addition, when you configure load balancing across different spawners, you must specify a login definition (the logical server credentials) within the load-balancing configuration; the user ID in the spawner's metadata configuration file must also be able to access this login definition (the logical server credentials).
- **For load-balancing SAS Workspace Servers**, the user ID in the metadata configuration file must be able to access the operator user ID, if it is specified. In addition, when you configure load balancing across different spawners, you must specify a login definition (the logical server credentials) within the load-balancing configuration; the user ID in the spawner's metadata configuration file must also be able to access this login definition (the logical server credentials).

The following table summarizes the credentials required for load-balancing security configuration:

Locations Where Credentials Are Specified for Load Balancing		
User ID or Login Definition	Location Where Credentials Are Specified	Description
Login for User ID in the Spawner's Metadata Configuration File	Metadata Configuration File Project Installation Note: For a project installation (using SAS Configuration Wizard), the metadata configuration file named <code>OMRConfig.xml</code> (located in the <code>ObjectSpawner</code> directory) contains the SAS Trusted User credentials.	Login credentials in the metadata configuration file. You must specify the login credentials that the spawner will use to connect to the SAS Metadata Server. This user ID must be able to access the operator login (ID) and if specified, the multi-user login definition. Important Note: DO NOT specify an <i>unrestricted user</i> for the user ID in the metadata configuration file.
Operator Login for Spawners (for spawner, optional)	<i>In the SAS Management Console spawner definition:</i> Initialization: Operator Login ➔ Operator Login Project Installation Note: For a project installation (using SAS Configuration	Administrator login definition to access the operator port of the spawner. The login definition must be one of the following: <ul style="list-style-type: none">• the login definition for the user ID that you specified in the

	Wizard), the operator login is not specified by default.	<p>metadata configuration file</p> <ul style="list-style-type: none"> • a login definition that the user ID in the metadata configuration file can access
Multi–User Login for SAS Stored Process Servers	<p><i>In SAS Management Console stored process server definition:</i></p> <p>Options ➤ Advanced Options ➤ Credentials ➤ Login</p> <p>Project Installation Note: For a project installation (using SAS Configuration Wizard), the login for the SAS General Servers group is specified.</p>	<p>Login for the multi–user server. The launched SAS process runs under the process ID defined by this login. The login definition must be one of the following:</p> <ul style="list-style-type: none"> • the login definition for the user ID that you specified in the metadata configuration file • a login definition that the user ID in the metadata configuration file can access
Logical Server Credentials (for load–balancing across multiple spawners)	<p><i>In SAS Management Console logical server definition:</i></p> <p>Options ➤ Logical Server Credentials</p>	<p>Login definition that is used by spawners to connect to other spawners for load balancing. The login definition must be one of the following:</p> <ul style="list-style-type: none"> • the login definition for the user ID that you specified in the metadata configuration file • a login definition that the user ID in the metadata configuration file can access

Load–Balancing Security Scenario: Recommended Configuration



The previous diagram shows a recommended load-balancing configuration that consists of the following server and security definitions on the SAS Metadata Server:

- **two clusters (load-balancing logical servers): Cluster 1, which contains 1 server, and Cluster 2, which contains 2 servers.** Cluster 1 contains 1 server and spawner on the same machine; Cluster 2 contains 2 servers and spawners on different machines.
- **a user metadata identity for the server administrator, ServerAdmin,** whose login credentials (not shown) are specified in the spawners' metadata configuration file and are used to access the multi-user login (for SAS Stored Process Servers), the logical server credentials (when load balancing across spawners), and the operator ID, if one is specified.
- **two group metadata identities for server administration.** The two group metadata identities, Cluster1Logins and Cluster2Logins, each own a group (shared) login definition that is used as follows:
 - ◆ For SAS Stored Process Servers, as the multi-user login credentials to start the SAS servers in Cluster 1 and Cluster 2.
 - ◆ For SAS Stored Process or SAS Workspace Servers that load-balance across spawners, as the logical server credentials.

The server administrator, ServerAdmin, belongs to both the Cluster1Logins and the Cluster2Logins groups. Because ServerAdmin is a member of each of the groups that contain the multi-user and logical server credential login definitions for Cluster1 and Cluster 2, ServerAdmin can access all of the multi-user and logical server credential login definitions. Note that the login definitions for the multi-user logins must have the same authentication domain as the servers in their respective clusters.

Project Installation Note: If you set up a load-balancing SAS Stored Process Server using the SAS project installation (and SAS Configuration Wizard), you have a load-balancing configuration with one or more SAS Stored Process Servers (and three multibrige connections on each server) in one cluster (e.g., Cluster 1) named SASMain – Logical Stored Process Server and security configured as follows:

- the user metadata identity for server administration is the SAS Trusted User. The SAS Trusted User's credentials (e.g. `sastrust`) are specified in the spawner's metadata configuration file.
- the group metadata identity for server administration is SAS General Servers group. The multi-user login definition (configured in the stored process server definition) is the login for the SAS General Servers group (e.g. `sassrv`).

You can then add additional servers and spawners to the load-balancing logical server (cluster) or add another load-balancing logical server (cluster) in a new SAS Application Server, as diagrammed in the recommended configuration above.

Planning the Security

To plan for the appropriate load-balancing security, for each load-balancing logical server (cluster), follow these steps:

1. **For SAS Stored Process Servers, plan for the multi-user login definition (on the SAS Metadata Server) and user ID for the spawner's metadata configuration file.** It is recommended that you specify the same multi-user login for each server in a cluster. To set up multi-user login definitions for each cluster, you must enable the user ID in the spawner's metadata configuration file to view the multi-user login definitions on the SAS Metadata Server. For details, see [Enabling the User ID in the Spawner's Metadata Configuration File to View Spawner/Server Login Definitions](#). For each cluster, to set up a multi-user login definition and user ID for the spawner's metadata configuration file, plan to implement the following login definition, user, and group structure on the SAS Metadata Server:

- a. Define a group metadata identity.
- b. Define a multi-user login definition as the group (shared) login definition for the group in step a. Depending on which user ID is used for the user ID in the spawner's metadata configuration file, set up the user ID for the spawner's metadata configuration file as follows:
 - ◊ If the user ID in the spawner's metadata configuration file is the same user ID as the multi-user login definition's user ID, no additional user and group setup is required.
 - ◊ If the user ID in the spawner's metadata configuration file is not the multi-user login's user ID, set up a user metadata identity and login definition for the login credentials (user ID and password) in the spawner's metadata configuration file. Add this user metadata identity as a member of the group metadata identity (from Step a) that contains the multi-user login definition as the group (shared) login definition. (You can also create a group of spawner administrators and add that group to the group that contains the multi-user login definition as the group (shared) login).

Note: Because the load-balancing stored process server runs under the multi-user login credentials, the operating system account for these credentials must have access to any operating system resources used by stored processes that are hosted on this server.

2. **If you implement more than one spawner, plan for the logical server credentials** (on the load-balancing logical server definition on the SAS Metadata Server) for the spawner to use to access other spawners for load balancing. The user ID specified in the spawner's metadata configuration file must also be able to access the login definition that is specified on the logical server credentials in the load-balancing logical server definition (on the SAS Metadata Server). For details, see [Planning the Spawner Security](#).

- ◆ For SAS Stored Process Servers, the recommended configuration specifies the same user ID for the multi-user and logical server credentials.
- ◆ For SAS Workspace Servers, the recommended configuration specifies the logical server credentials as a group login for the group of users who are the server administrators. Plan to implement the

following login definition, user, and group structure on the SAS Metadata Server:

- a. Define a group metadata identity.
- b. Define the logical server credentials login definition as the group (shared) login definition.
Depending on which user ID is used for the user ID in the spawner's metadata configuration file, set up the user ID for the logical server credentials login as follows:
 - If the user ID in the spawner's metadata configuration file is the same user ID as the logical server credentials login definition's user ID, no additional user and group setup is required.
 - If the user ID in the spawner's metadata configuration file is not the logical server credentials user ID, set up a user metadata identity and login definition for the login credentials (user ID and password ID) in the metadata configuration file. Add this user as a member of the group (from Step a) that contains the logical server credentials login definition as the group (shared) login definition. (You can also create a group metadata identity of spawner administrators and add that group to the group that contains the logical server credentials as the group (shared) login definition).

If multiple spawners are used with load balancing (e.g., when multiple machines are used in load balancing) and the spawners connect to the metadata server using different user or group metadata identities, you must add the user or group metadata identities to the group you use for the group (shared) logical server credentials and multi-user login definition (SAS Stored Process Servers only).

3. **Plan to grant the "Administer" permission (on the load-balancing logical server definition) to the user or group metadata identity that owns the logical server credentials.** On the load-balancing logical server definition, you must plan to grant the "Administer" permission to the user (or group) metadata identity that owns the logical server credential's login definition.
4. **Plan to control access to the load-balancing logical server.** You must control access for who is authorized to update the load-balancing logical server. To control who can update the load-balancing logical server, in SAS Management Console, you must use the Authorization tab for the load-balancing logical server to do both of the following:
 - ◆ Deny "WriteMetadata" permission to the Public group.
 - ◆ Grant "WriteMetadata" permission to your metadata administrator.
5. **Plan to control access to the data on the servers.** For each server you must control access to the data on the server, either through authorization (access control) metadata (see the Authorization Manager and User Manager plug-in help in the SAS Management Console) or file system access on your host system.

Understanding Authentication

When you implement a load-balancing spawner and server configuration, the user who starts the server, the login definitions specified in the spawner, server, and load-balancing logical server configuration, and the clients who connect to the servers must be authenticated against the appropriate authentication provider. The following table shows the location where the credentials are authenticated:

Locations Where Credentials Are Authenticated		
Type of Credentials	User ID Role	Authentication Location
Standard Spawner and Server Configuration	the user ID for the operator login	no authentication
	for SAS Stored Process Servers, user ID for the	host authentication provider for the SAS Stored Process Server's machine

	multi-user login	
Load Balancing Logical Server Configuration	user ID of the logical server credentials (on the load-balancing logical server definition)	host authentication provider for the local server's machine and the host authentication provider on the machine of the other spawners to which it connects. You can use a network account to provide host authentication for the appropriate machines.
Connections to Load-Balancing Spawner and Server	the client user ID	host authentication provider for the SAS Stored Process or SAS Workspace Server's machine

Security

Implementing Security in Client Applications

To connect to and access data on a server, clients provide a fully qualified user ID and password. In a SAS Metadata Repository, the server, user, group, and login definition (which corresponds to a user's credentials within a security domain) metadata defines which users are allowed access to a server as follows:

- For SAS Metadata Servers, login credentials defined on the authentication provider for the SAS Metadata Server's machine.
- For IOM servers, login definitions defined in the same authentication domain as the server.
- For IOM pooled servers:
 - ◆ the login definition (and its user or group metadata identity) that is associated with a puddle defined for a pooled logical server
 - ◆ the login definitions defined for the user metadata identities that are members of a group metadata identity that is granted access to a puddle.

Important Note: Do not connect to a server as the *unrestricted user*. To understand unrestricted access for *unrestricted users*, see [🌐 Server Administrative Privileges](#) in the *SAS Metadata Server: Setup Guide*.

Applications can specify credentials in the following ways:

- **provide credentials to connect to servers.** Your application can directly supply the necessary fully qualified user ID and password that is required to connect to the server.
- **retrieve credentials from the SAS Metadata Server in order to connect to servers.** Your application can access the SAS Metadata Server and retrieve server and login (user credential) information in order to connect to a server. The application must then connect to the server using the retrieved credentials.
- **retrieve credentials from other applications by sharing session or user contexts (Java clients only).** Java clients can use the User Service to retrieve and share user information between applications. When one application is accessed from another application, the first application passes the second application its user or group metadata identity (via a shared session and user context). This identity can then be used for authorization purposes or to retrieve user credentials to access particular resources. This context-sharing feature enables single sign-on to be seamlessly implemented between applications. For detailed information about context sharing, see the SAS Foundation Services class documentation for the [🌐 User Service](#).
- **connect to downstream servers by providing credentials or by retrieving credentials from the SAS Metadata Server.** When connecting to an FTP, HTTP, or WebDAV server,
 - ◆ if the client or SAS Metadata Server provides a set of credentials to use for the WebDAV, FTP, or HTTP server, those credentials are used for connection to the downstream server.
 - ◆ if the client or SAS Metadata Server does not provide a set of credentials, anonymous access is used for connection to the downstream server.

For information about coding client applications, refer to the following:

- For Java clients, [Developing Java Clients](#) in the *SAS Integration Technologies Developer's Guide* and the SAS Foundation Services class documentation.
- For Windows clients, [Developing Windows Clients](#) in the *SAS Integration Technologies Developer's Guide* and the Windows Object Manager class documentation.

Authenticating Clients

When a client connects to a server, the server authenticates the client against the appropriate authentication provider or trusted authentication mechanism. For details, see [Implementing Authentication](#).

Retrieving and Enforcing Authorization Decisions

In order to secure access to a resource, your application must do the following:

1. Retrieve authorization metadata for a particular user's action on a resource.
2. Enforce the authorization decisions for a particular user's action on a resource.

The SAS Open Metadata Architecture provides the `ISecurity` class for authorizing access both to metadata and the data that is represented by the metadata. For details, see [ISecurity Class](#) in the *SAS Open Metadata Interface: Reference*.