



SAS Publishing

Administrator for Enterprise Clients: User's Guide

Second Edition



The correct bibliographic citation for this manual is as follows: SAS Institute Inc. 2002. *Administrator for Enterprise Clients: User's Guide, Second Edition*. Cary, NC: SAS Institute Inc.

Administrator for Enterprise Clients: User's Guide, Second Edition

Copyright © 2002 by SAS Institute Inc., Cary, NC, USA

ISBN 1-59047-003-6

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

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

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

1st printing, July 2002

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 www.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.

IBM® and all other International Business Machines Corporation product or service names are registered trademarks or trademarks of International Business Machines Corporation in the USA and other countries.

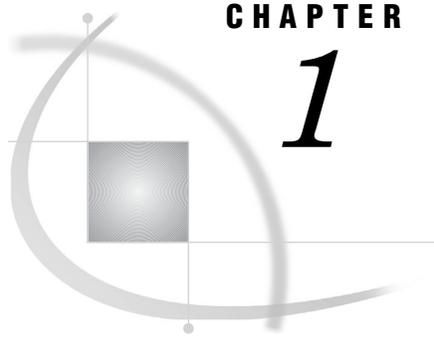
Oracle® and all other Oracle Corporation product or service names are registered trademarks or trademarks of Oracle Corporation in the USA and other countries.

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

Contents

Chapter 1	△ Welcome to Enterprise Guide Administrator	1
	What Is Enterprise Guide Administrator?	1
	What Are Enterprise Clients?	2
	Using This Guide	2
	Updates to This Guide	3
	Getting Help	3
Chapter 2	△ Installing Enterprise Clients and SAS Servers	5
	Installing Enterprise Clients	5
	Preserving the Repository during an Upgrade	9
	Installing SAS Servers	10
	Troubleshooting the Installation	12
Chapter 3	△ Setting Up Enterprise Clients	13
	Overview	13
	Setting Up the Administrator Metadata Repository	13
	Maintaining Repositories	25
Chapter 4	△ Configuring an IOM Bridge Connection	33
	Introduction	33
	How an IOM Bridge Connection Works	33
	Setting Up an Object Spawner	36
Chapter 5	△ Configuring a DCOM Connection	47
	Overview	47
	Requirements	47
	Step 1 — Enabling DCOM on the Client and the Server	48
	Step 2 — Creating Client Groups	49
	Step 3 — Setting SAS Launch Policies	50
Chapter 6	△ Preparing a Setup Image for Users	61
	Preparing a Setup Image for Users in a Networked Environment	61
Chapter 7	△ Tasks That You Perform in Enterprise Guide Administrator	69
	Changing Repositories	69
	Viewing the Objects	70
	Managing Servers	72
	Managing Libraries	81
	Managing Binders	90
	Managing Groups	95
	Managing Users	99
	Adding a Format	104
	Importing Server Definitions	108

Chapter 8	△ Examples of Creating Database Libraries	115
Creating an Oracle Library		115
Creating a DB2 Library		117
Chapter 9	△ Administering Enterprise Guide Tasks	119
Administering Tasks		119
Importing Task Definitions		119
Enabling Tasks for User Groups		122
Enabling Task Administration		124
Appendix 1	△ Object Spawner Details	125
Object Spawner Overview		125
Requirements		126
Information You Need before Configuring		126
Modifying the Services File		126
Creating the Spawner Configuration File		127
Setting Up and Invoking the Object Spawner		130
Administering a Running Spawner		138
Error Messages		139
Appendix 2	△ Testing Server Connections with Integration Technologies Configuration	143
Index		147

**CHAPTER****1****Welcome to Enterprise Guide Administrator**

What Is Enterprise Guide Administrator? 1

What Are Enterprise Clients? 2

Using This Guide 2

Updates to This Guide 3

Getting Help 3

What Is Enterprise Guide Administrator?

Enterprise Guide Administrator is a Windows application for network and PC administrators that serves as a central resource manager for Enterprise Client applications. With Enterprise Guide Administrator, you can provide Enterprise Client users with transparent access to their data. In other words, you can enable users to access their data without their having to remember the exact locations of the data. Enterprise Clients provide an easy-to-use Windows interface to a SAS System session that is running either locally or on a remote machine. For more information about Enterprise Clients, see “What Are Enterprise Clients?” on page 2.

With Enterprise Guide Administrator you can

- identify SAS servers on multiple platforms
- set up virtual folders with shortcuts to multiplatform data
- set up groups of users to provide high-security access to resources, or bypass groups to provide low-security access.

For example, you could add a SAS server that runs on UNIX, specify which users have access to the server, and set up SAS libraries on that server. The specified users could then access the libraries from their Enterprise Client applications.

Enterprise Guide Administrator runs on Windows NT, Windows 95, and Windows 98, and has an easy-to-use interface that is like Windows Explorer. You do not need to install SAS on your local machine to run Enterprise Guide Administrator.

What Are Enterprise Clients?

Enterprise Clients are thin-client Windows applications that provide a point-and-click desktop interface to the SAS System (Version 8 or later). Enterprise Clients can be used on various server platforms, including UNIX, OS/390, and those that run Windows 95 or Windows 98 as a local server. Enterprise Clients help users to produce results easily in a few minutes, regardless of their SAS knowledge.

Here are two examples of Enterprise Clients:

Enterprise Guide

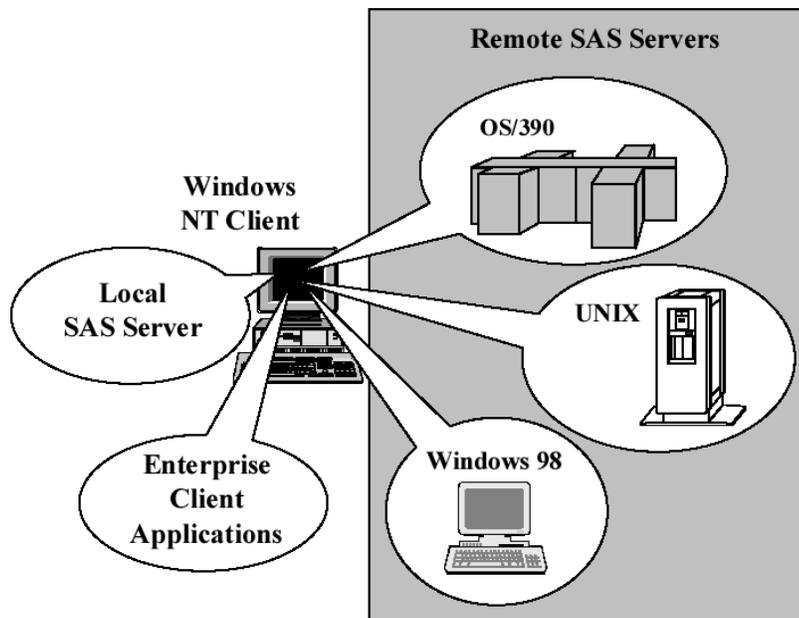
communicates with the SAS System to access data, perform analysis, and generate results. From Enterprise Guide you can access and analyze many types of data, such as SAS data sets, Excel spreadsheets, and Oracle databases. You can either use a set of task dialog boxes or write your own SAS code to perform your analysis.

Enterprise Reporter (Version 3.0 and later)

is a powerful desktop reporting tool that provides transparent access to data from PC, SAS, MDDB, ODBC, and OLE DB data providers. You get a WYSIWYG (what you see is what you get) report-creation interface, including the ability to drag and drop live data on tables and charts. Enterprise Reporter is designed especially for the business user who wants to create reports easily, but who handles both simple, static reports and reports with complex calculations.

Enterprise Clients use a distributed computing model of networked servers that enable users to access data on multiple platforms quickly and easily.

Figure 1.1 Enterprise Clients in a Distributed Computing Environment



Using This Guide

This guide shows you how to perform tasks in Enterprise Guide Administrator and how to perform some setup tasks outside of Enterprise Guide Administrator. The sections in this guide are arranged to reflect the order in which you might use them.

Enterprise Client and SAS Server Installation
walks you through installing SAS on server platforms.

Enterprise Client Setup
discusses setting up the Administrator metadata repository (a type of database that holds information about the Enterprise Client objects) and preparing an Enterprise Client image for users to install.

Tasks That You Perform in Enterprise Guide Administrator
includes both background information and step-by-step instructions for tasks such as viewing objects in the Enterprise Guide Administrator Viewer, adding SAS servers, and adding SAS libraries.

Examples of Creating Database Libraries
gives examples of how to set up libraries of Oracle and DB2 for use with Enterprise Clients applications.

If you are setting up Enterprise Guide Administrator, SAS, and the other Enterprise Clients applications for the first time, read the applicable sections completely before installing or using any of the software. The list below will help you determine which sections of this guide are most important for your setup.

- If you are running the Enterprise Clients and SAS on one local machine, then read
 - “What Are Enterprise Clients?” on page 2
 - “Installing Enterprise Clients” on page 5
 - Chapter 3, “Setting Up Enterprise Clients,” on page 13.
- If you are running Enterprise Clients locally and running SAS on a networked Windows server, then read
 - “What Are Enterprise Clients?” on page 2
 - Chapter 2, “Installing Enterprise Clients and SAS Servers,” on page 5
 - Chapter 3, “Setting Up Enterprise Clients,” on page 13.
- If you are running Enterprise Clients locally and running SAS on a networked server that does not run Windows, then read
 - Chapter 2, “Installing Enterprise Clients and SAS Servers,” on page 5
 - Chapter 3, “Setting Up Enterprise Clients,” on page 13
 - Chapter 4, “Configuring an IOM Bridge Connection,” on page 33
 - Chapter 5, “Configuring a DCOM Connection,” on page 47
 - Chapter 6, “Preparing a Setup Image for Users,” on page 61
 - Appendix 1, “Object Spawner Details,” on page 125.

Updates to This Guide

Updates to this guide are available from the SAS Institute Web site at www.sas.com.

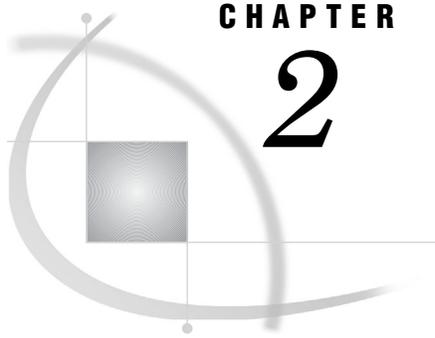
Getting Help

To get started with Enterprise Guide Administrator, from the Enterprise Guide Administrator main menu, click

Help ► Administrator Help

Or click the **Help** button in any Enterprise Guide Administrator dialog box to get help for that dialog box.

For technical assistance, contact SAS Technical Support at 919-677-8008 or visit the Web site at www.sas.com.



CHAPTER

2

Installing Enterprise Clients and SAS Servers

<i>Installing Enterprise Clients</i>	5
<i>Using a Silent Setup to Install Enterprise Guide</i>	5
<i>Recording an Installation</i>	6
<i>Playing Back an Installation</i>	6
<i>Troubleshooting</i>	6
<i>Installing Enterprise Guide on a Citrix Server</i>	7
<i>Install Enterprise Guide and Enterprise Guide Administrator</i>	7
<i>Create an Administrator Account</i>	7
<i>Set File Permissions for Sdslocal</i>	8
<i>Set DCOM Permissions</i>	8
<i>Specify the User That Will Run the Repository Process</i>	9
<i>Edit the Registry</i>	9
<i>Preserving the Repository during an Upgrade</i>	9
<i>Installing SAS Servers</i>	10
<i>Troubleshooting the Installation</i>	12

Installing Enterprise Clients

System requirements and instructions for installing SAS Enterprise Clients applications are described in the installation document that is included with each Enterprise Clients application software CD.

The Enterprise Clients software is installed on Microsoft Windows platforms and provides access to SAS services and data that are located on your network. A SAS server can be located on Windows NT, OS/390, or UNIX (AIX 4.2, HP-UX 10.20, or Solaris 2.6) platforms. Specific installation procedures are provided with the Enterprise Clients software.

Using a Silent Setup to Install Enterprise Guide

The silent setup program enables users to “record” an installation and later to “play back” the recorded installation in order to install Enterprise Guide on other machines. The silent setup is a timesaving tool for system administrators who have to install software on many machines.

Note: Be sure that the machine on which you are installing the software has enough free disk space before you begin the installation. △

Recording an Installation

When you record an installation, you are recording all your dialog box settings in a setup response file. As you proceed through the dialog boxes, read all of the available online Help before continuing.

To record an installation:

- 1 Determine the location of the source Enterprise Guide image (that is, the image from which you want to install). Make this location the current directory.
- 2 (Optional) Determine the name and location of the setup response file that you are creating. For this example, it is **C:\myEGfile.iss**.

Note: If you do not provide the pathname and filename for the setup response file, then setup.iss will be created in your WINDIR, which is the full pathname where Windows is installed. Δ

- 3 Start the silent setup in record mode by using the **/r** command line argument (**-r** is also accepted). The setup will record the options that you select in the file **C:\winnt\setup.iss**. (In this example, **C:\winnt** is the WINDIR.) For example, you could issue the command **m:\eg> setup /r**.

If you are specifying the setup response filename and pathname (optional), then use the **/f1** command line option. Note that there is no space between **/f1** and the file specification.

```
m:\eg> setup /r /f1c:\myEGfile.iss
```

- 4 Proceed with the record mode setup by making the selections for your installation, including the target destination and additional setup options.
- 5 When the setup is complete, you can copy, move, or rename the setup response file that was created. Note that if a setup response file of the same name exists, then it will be overwritten.

Playing Back an Installation

After you record your setup response file, you can use it to repeat the same installation without using or seeing the setup dialog boxes.

To play back an installation:

- 1 Determine the location of the source Enterprise Guide, and then make that the location of the current directory.
- 2 Determine the name and the location of the setup response file that you want to use. For this example, it is **C:\myEGfile.iss**.

Note: If you do not provide the pathname and filename for the setup response file, then the setup program tries to use setup.iss in the directory where setup.exe resides. Δ

- 3 Run the setup program by using the **/s** and **/f1** command line options to specify both silent mode and the setup response file that should be used for input. The syntax of this command is

```
m:\eg> setup /s /f1c:\myEGfile.iss
```

Troubleshooting

If the silent setup program encounters a problem during playback, then the program terminates. When the program terminates, the status dialog box disappears and all of the changes that were made to your system are reversed. The system is thus left in its original state.

Here are two reasons why the program could terminate during playback, along with suggestions for solving the problem.

- 1 *The drive to which you are installing does not contain enough free disk space.* Free up some disk space and try again.
- 2 *The order of dialog box prompts that you recorded in the setup response file (setup.iss) is not the same as the order of dialog box prompts that the silent setup is encountering (silently) during playback.* This is the most common problem. Try running a normal interactive setup on the machine in question, and note any dialog box prompts that were not recorded in the original setup response file. You might need to record a new setup response file to use on machines similar to the one that is exhibiting the problem.

Installing Enterprise Guide on a Citrix Server

Note: Enterprise Guide is not tested or officially supported for the Citrix environments. Before proceeding, please read the terms of your license agreement to confirm that this configuration is permitted. △

Installing Enterprise Guide and Enterprise Guide Administrator on a Citrix server presents special challenges. In a Citrix environment, Enterprise Guide and Enterprise Guide Administrator are installed on only one machine, and then all users access that machine. Use the following steps to install and configure Enterprise Guide Administrator on a Citrix server.

Install Enterprise Guide and Enterprise Guide Administrator

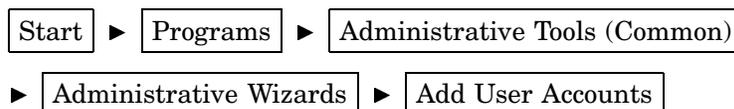
To install Enterprise Guide and Enterprise Guide Administrator for a Windows/NT account that has administrator privileges:

- 1 From the Windows taskbar, select **Start Programs Command Prompt**.
- 2 At the command prompt, change to the `/install` directory for the user account for which the installation is to be performed. Close the Command Prompt window.
- 3 Select **Start Run**.
- 4 In the Run window, enter the command to start the Enterprise Guide installation program. The program is named `setup.exe`.
- 5 Complete the Enterprise Guide installation process.

Create an Administrator Account

To create an administrator user account on the machine on which you installed Enterprise Guide:

- 1 Select



- 2 Enter the following information in the windows that open:

Table 2.1 Administrator Account Information

Domain name	Domain of current machine (for example, INSTMACH)
User's full name	Name of account owner
Unique name to identify the user	Account ID (for example, sdsrep)
Description	Account description
Password	Account password (must be confirmed)
Password expiration	Select this password never expires
Available groups	Select group Users

- 3 Select **Finish** to create the account.

Set File Permissions for Sdslocal

To set the file permissions for the database file Sdslocal in order to permit full control for the administrator account that you just created:

- 1 From the Windows desktop, select **My Computer**, and then locate the directory **\Program Files\SAS Institute\Shared Files\SAS Directory Services**.
- 2 Click the right mouse button on the file Sdslocal, and then select **Properties** from the shortcut menu. The Sdslocal Properties window opens.
- 3 Select the **Security** tab, and then click **Permissions**. The File Permissions window opens.
- 4 Select the group **Everyone**. From the **Type of Access** menu, select **Read**.
- 5 Click **Add**. The Add Users and Groups window opens.
- 6 Select the administrator account that you added earlier. From the **Type of Access** menu, select **Full Access**.
- 7 Click **OK** in all open windows.

Set DCOM Permissions

Next, you must set DCOM permissions. This step is described in more detail in "Creating Shared Access to the Repository Process" on page 18.

- 1 From the Windows taskbar, select **Start Run**. In the Run window, type **dcomcnfg** and then click **OK**.
- 2 From the Distributed COM Configuration Properties window, select the **Applications** tab, select **SAS Directory Services Repository**, and then click **Properties**. The SAS Directory Services Repository Properties window opens.
- 3 From the Properties window, select the **Security** tab.
- 4 Select **Use custom configuration permissions**, and then click **Edit**. The Registry Value Permissions window opens.
- 5 From the Permissions window, click **Add**. The Add Users and Groups window opens.
- 6 In the **Add Names** field, type **Everyone**. Click **OK** to close the Add Users and Groups window, then click **OK** to close the Permissions window.
- 7 From the Properties window, select **Use custom launch permissions** and then click **Edit**. The Registry Value Permissions window opens.
- 8 In the Permissions window, click **Add**. The Add Users and Groups window opens.

- 9 In the Add Users and Groups window, type **Everyone** in the **Add Names** field. Click **OK** to close the Add Users and Groups window, then click **OK** to close the Permissions window. Leave the Properties window open.

Specify the User That Will Run the Repository Process

To specify the user that will run the repository process:

- 1 From the SAS Directory Services Properties window, select the **Identity** tab.
- 2 Select **This user**.
- 3 Click **Browse** to open the Browse for Users window.
- 4 From the **List Names From** pull-down menu, select the domain that you specified when you created the administrator account.
- 5 In the **Names** list, select the user ID of the administrator account that you created earlier. The **User** field displays the machine and user ID (for example, **INSTMACH\sdsrep**).
- 6 Type and confirm the password for the user ID.
- 7 Click **OK** to close the SAS Directory Services Repository Properties window. Then click **OK** to close the Distributed COM Configuration Properties window.

Edit the Registry

To edit the registry to set the server to the name of the Citrix server:

- 1 From the Windows taskbar, open the Run window by selecting



In the **Run** field, type **regedit** and click **OK**. The Registry Editor window opens.

- 2 From the Registry Editor window, expand the following tree:



- 3 Click the right mouse button on the Server entry and select **Modify** from the shortcut menu. The Edit String window opens.
- 4 In the **Value Data** field, enter the fully qualified name of the machine that you specified when you created the administrator account (for example, **INSTMACH**).
- 5 Click **OK** to close the Edit String window, and then close the Registry Editor window.

Preserving the Repository during an Upgrade

If you are upgrading from Enterprise Guide Administrator, Version 1.0 or Release 1.1 or 1.1.1, then you must back up your current repository before uninstalling Enterprise Guide Administrator. If you do not back up the repository, then the uninstall process will remove the repository along with the application, resulting in loss of data.

To back up the repository:

- 1 Select **File Repository Maintenance**. The Repository Maintenance window displays the path of the repository.
- 2 Make a copy of the repository file in a different location.
- 3 Uninstall the current copy of Enterprise Guide Administrator.

- 4 Return the backup repository to its original location.
- 5 Install the new release of Enterprise Guide Administrator.

If you install the new release of Enterprise Guide Administrator before restoring the repository file to its original location, then the new schema will not be applied to the repository. To manually upgrade the schema, from the Windows Run dialog box or an MS-DOS prompt, type the command `sdsdmmnt.exe repository_pathname`, where `repository_pathname` is the pathname of the repository file.

Installing SAS Servers

This section describes the configuration of a SAS server for use with Enterprise Clients. These instructions complement the normal SAS installation. For system requirements for installing SAS on server hosts, see the installation instructions that are included with the SAS installation CD.

- 1 Install the required SAS software.

Using the SAS installation procedures that are included with the software package, install Version 8 or a later version of SAS software on the server host. These products are required when you install SAS software:

- base SAS
- SAS Core
- SAS Integration Technologies.

- 2 Install additional SAS products.

Based on your business needs, install any additional SAS products by using the installation procedures that are provided with the software package. The following is a sample list of products that you can install on the server. See the system requirements for the specific Enterprise Clients for more information.

- SAS/ACCESS (for accessing database management system data, such as data that is housed in Oracle or DB2)
- SAS/GRAPH
- SAS/QC
- SAS/MDDB
- SAS/STAT.

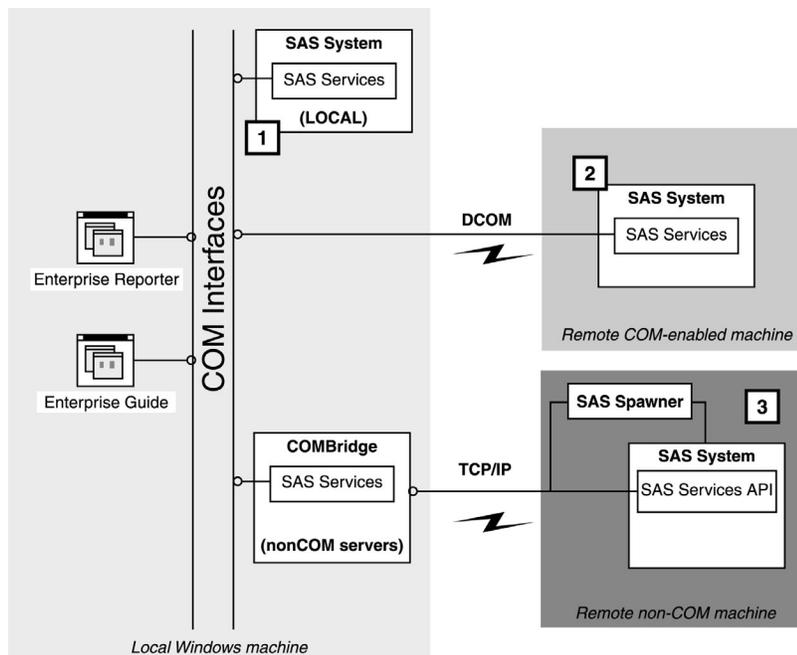
- 3 Configure the object server.

After you install SAS, you need to configure a SAS object server so that it can communicate with the Enterprise Clients. The configuration procedure that you follow is based on the protocol that is used to communicate between the Enterprise Clients and the object server, which depends on the server's host environment. The following list and figure will help you determine the correct configuration procedure to follow.

Connection Type	Configuration Procedure
COM connection to a local Windows machine	After you have installed the Enterprise Clients and SAS software on the Windows computer, you are ready to use the Enterprise Clients software.
DCOM connection to a remote Windows NT machine	Follow the procedures in Chapter 5, “Configuring a DCOM Connection,” on page 47.
IOM Bridge connection to OS/390, UNIX, or Windows NT	Follow the general procedures in Chapter 4, “Configuring an IOM Bridge Connection,” on page 33 and specific information for your platform in Appendix 1, “Object Spawner Details,” on page 125.

Figure 2.1 on page 11 illustrates a scenario in which the Enterprise Clients are running locally on a Windows computer and accessing three different object servers. Refer to the figure to further help you decide how to configure the SAS object server.

Figure 2.1 Enterprise Client Applications and SAS Object Servers



- In the first case (1), the Windows machine functions as both the client and the object server because a *local Windows machine* has both the Enterprise Clients and SAS software installed. The Microsoft Component Object Model (COM) is used to communicate between client and object server, and no additional configuration is required.
- The second object server in the scenario (2) is located on a *remote Windows machine*. Because both the object server and the client use the Windows operating environment, the distributed version of the component object model (DCOM) is used as the communication protocol. Configuring for a DCOM session includes

setting security parameters and user rights, and ensuring that DCOM is enabled both on the machine where the object server will run and on the client machine.

Note: If you want to use a Windows 95 machine as a DCOM client, you must verify that Microsoft's DCOM95 Version 1.2 upgrade to enable DCOM has been installed on the machine. If the upgrade has not been installed, you must install it yourself. DCOM95 Version 1.2 is available from Microsoft. \triangle

- The third object server (3) is installed on a *non-COM-enabled remote machine* that has SAS installed on it. This is a host, such as OS/390, UNIX (AIX 4.2, HP-UX 10.20, or Solaris 2.6), or Windows NT that uses TCP/IP as its communication protocol. The way to configure this type of server is to start an object spawner (objspawn) on the server host. The object spawner software is installed as part of the SAS Integration Technologies software and is designed to detect a client that is requesting access to the SAS object server. After the object spawner receives a request, it launches a SAS object server.

Troubleshooting the Installation

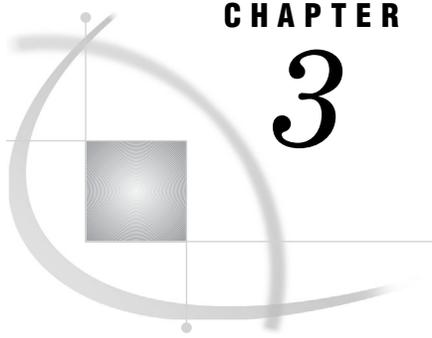
After you install either Enterprise Guide or Enterprise Guide Administrator, you must verify that the installation was successful. To verify the installation, perform one of the following actions:

- Start the Enterprise Guide Administrator application. If Enterprise Guide Administrator displays an error message that indicates that the application cannot connect to the local repository, then contact either your local SAS representative or SAS Technical Support before continuing.

This is the preferred method of verification.

- Start Enterprise Guide. If Enterprise Guide displays an error message that indicates that the application cannot connect to the local repository, then contact either your local SAS representative or SAS Technical Support before continuing. Use this method of verification only if the Enterprise Guide Administrator application is not installed.

Performing either of these actions will identify any problem machines that are not found by the installation wizard.



CHAPTER

3

Setting Up Enterprise Clients

<i>Overview</i>	13
<i>Setting Up the Administrator Metadata Repository</i>	13
<i>What Is a Metadata Repository?</i>	13
<i>Running the Repository Process Locally</i>	14
<i>Running the Repository Process on a Remote Server</i>	15
<i>Working with Local and Remote Repositories</i>	17
<i>Creating Shared Access to the Repository Process</i>	18
<i>Maintaining Repositories</i>	25
<i>Changing the Path</i>	26
<i>Taking a Shared Repository Offline</i>	26
<i>Compacting</i>	27
<i>Registering MDDBs</i>	27

Overview

Setting up the Enterprise Clients for yourself and for your users includes these general steps. You can perform these steps in this order or in the order that works best for your situation.

- 1 If you have not already done so, install the Enterprise Client applications on your machine. See Chapter 2, “Installing Enterprise Clients and SAS Servers,” on page 5.
- 2 Set up the administrator metadata repository. See “Setting Up the Administrator Metadata Repository” on page 13.
- 3 Prepare a setup image for users. See Chapter 6, “Preparing a Setup Image for Users,” on page 61.

Setting Up the Administrator Metadata Repository

What Is a Metadata Repository?

Using Enterprise Guide Administrator, you create object definitions that are used by the Enterprise Clients — objects such as servers, libraries, and users. The Administrator metadata repository is a type of database that holds information about the object definitions that you create. The Administrator metadata repository is an essential part of Enterprise Guide Administrator because each Enterprise Client needs

to access at least one repository, whether the repository is stored locally or on a networked Windows server.

Enterprise Guide Administrator (and other Enterprise Clients) can access one metadata repository at a time, but you can create multiple repositories and then switch between them as needed. Multiple repositories are used

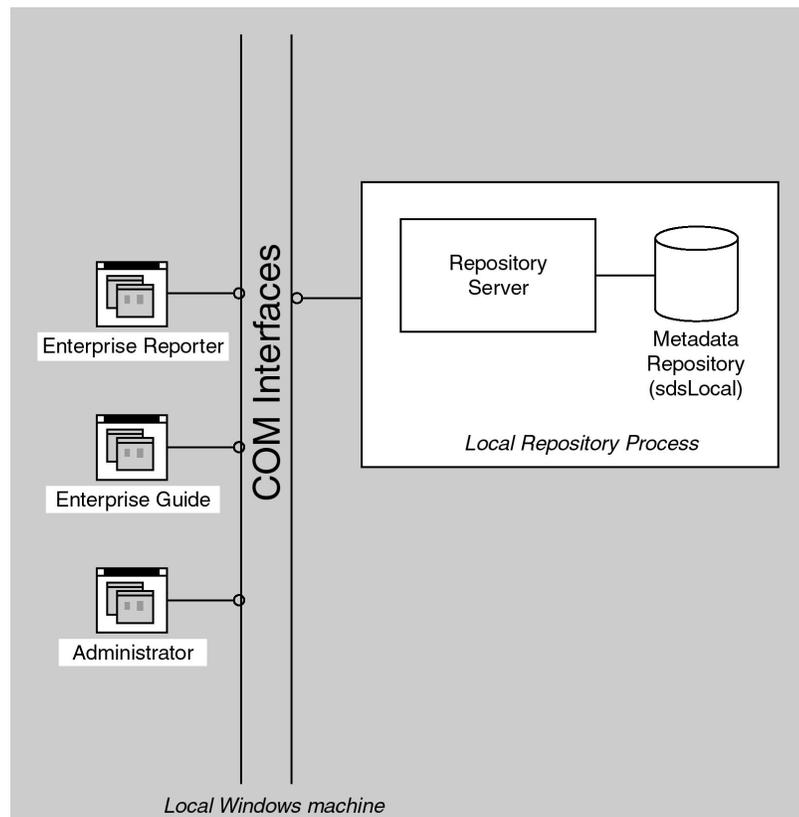
- to create tailored repositories for different groups of users, giving each group a different scenario for sharing access to servers and data.
- to work with test metadata repositories.

Each metadata repository is accessed through a metadata repository process so that multiple Enterprise Client applications can access the same metadata repository if necessary. The repository process is located with a metadata repository either on your local machine or on a networked Windows server.

Running the Repository Process Locally

At installation, Enterprise Guide Administrator, along with a repository process, is installed on your local machine with the Enterprise Client application. Local is a location that depends on the vantage point of the application. You can run the repository process locally on a server or on a desktop PC, and either machine can be configured for access from other Windows machines.

Figure 3.1 The Repository Process Running on a Local Machine



A default metadata repository is created in this location:

```
C:\Program Files\SAS Institute\Shared Files\  
    SAS Directory  
    Services\SdsLocal
```

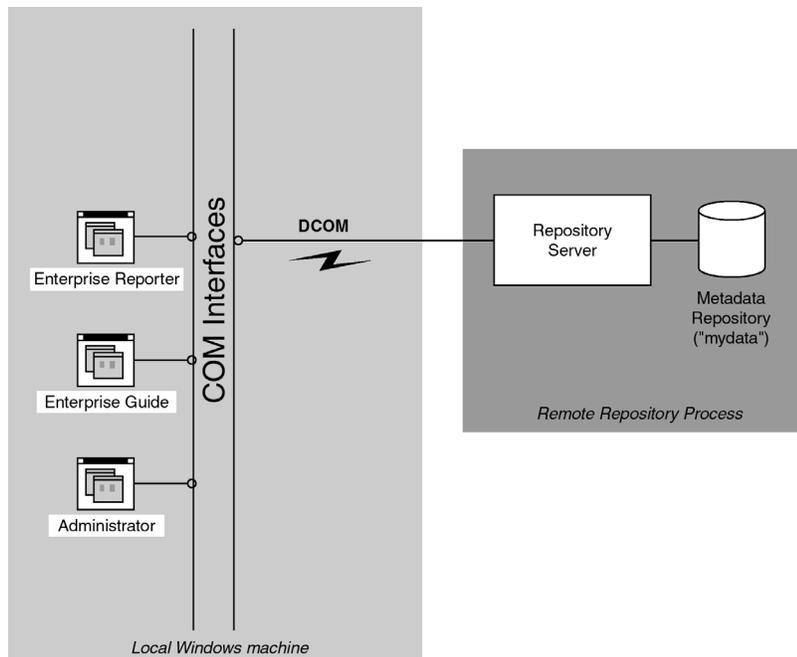
where **C:** is the drive on which the Enterprise Clients are installed on your machine. The default objects that are defined in the default repository indicate that you are running both SAS and the Enterprise Clients locally. It reflects the simplest installation of the Enterprise Clients.

The objects that are defined in the SdsLocal file are

Server	Local
User	[Unknown User]
Group	General
Binders	Sample

The default object definitions indicate that all users are [Unknown User], belong to the General group, and have access to both the Local server and the Sample binder. You can change the default settings for each object in Enterprise Guide Administrator, and you can enter new object definitions.

Figure 3.2 When Files Are Merged



Running the Repository Process on a Remote Server

You can set up the repository process to run on a server that is remote from the Enterprise Clients applications. On a server you can choose to install only Enterprise Guide Administrator and the repository process, also referred to as SAS Directory Services (SDS), without installing any Enterprise Client applications.

To install SDS on the remote server machine:

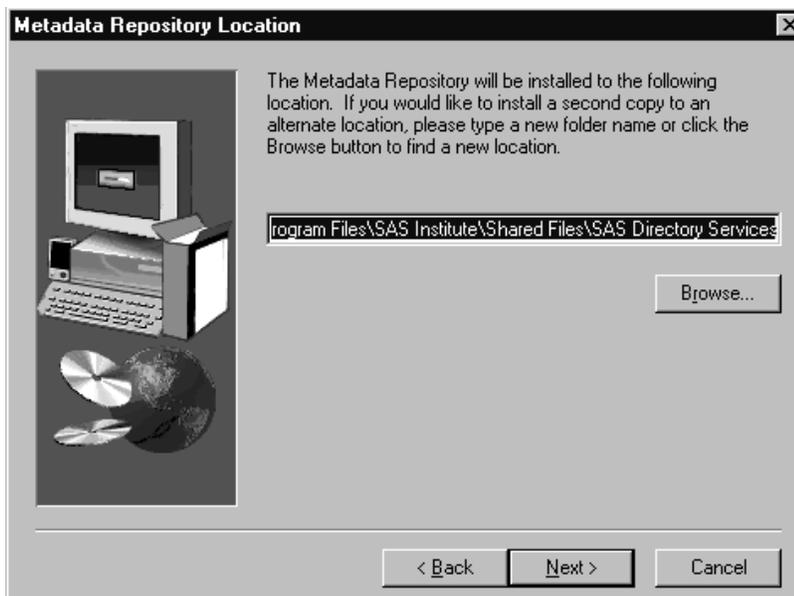
- 1 Execute this file from the installation CD:

```
[CD ROOT]\bundles\sds\setup.exe
```



Click **Next** to continue.

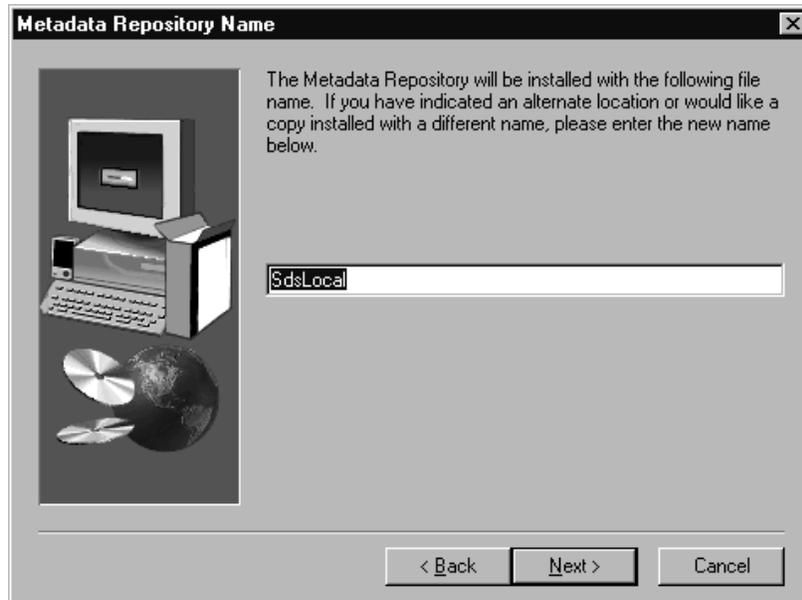
- 2 Select the location where you want Enterprise Guide Administrator and the repository process to be installed.



Click **Next** to continue.

- 3 Select a name for the repository. The default repository database is located on the server in

```
C:\Program Files\SAS Institute\Shared Files\  
SAS Directory Services\SdsLocal
```



Click **Next** to continue.

- 4 Verify your entries in the next dialog box, and then click **Finish** in the last dialog box to complete the installation.

Working with Local and Remote Repositories

Before you add object definitions in the repository, add yourself as a user with administrator privileges and set a password.

To work with a repository for the first time:

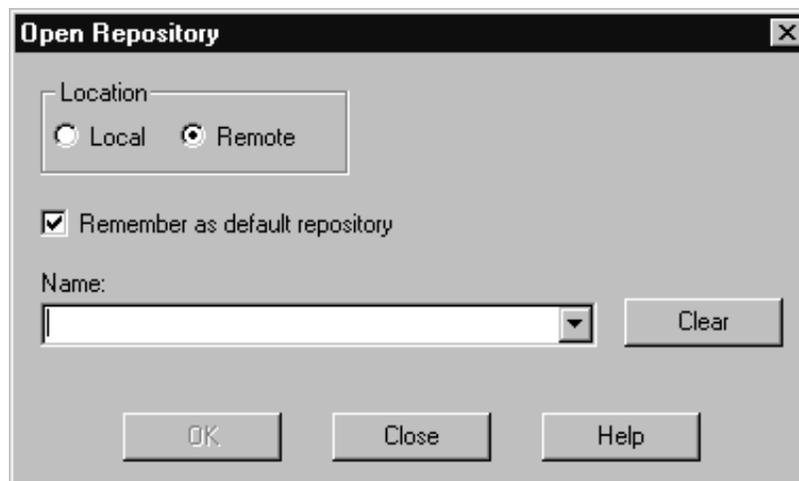
- 1 From your Windows desktop, select

Start ► Programs ► The SAS System ► Administrator

From the **Administrator** pull-down menu, select

File ► Open Repository

or select the Open icon  from the toolbar.



- 2 Indicate whether you will use the local repository (sdsLocal) or a shared remote repository. If you choose Remote, then enter the remote server name to begin using it. Opening a repository both opens it so that you can make updates and, if you check **Remember as default repository**, makes it the active repository when you open an Enterprise Client application. To change the name of the local repository or to designate a repository offline for maintenance, choose

File ► Repository Maintenance

(See “Maintaining Repositories” on page 25.)

- 3 Add yourself as a user with administrator privileges. See “Adding a User” on page 100.
- 4 Add security by setting a password for the active repository. (Security is optional for local repositories.) After you set a password, only users with administrator privileges can make changes to the repository. A user can gain administrator privileges temporarily by using the password. (This is also a way to add an administrator if none exists.) From the pull-down menu, select

Mode ► Change Password

Type the password and verify it.

The image shows a standard Windows-style dialog box titled "Change administrator password". It features a title bar with a close button (X) on the right. The main area contains three text input fields, each preceded by a label: "Current password:", "New password:", and "Verify new password:". At the bottom of the dialog, there are two buttons: "OK" and "Cancel".

- 5 “Populate” the repository; that is, add servers, libraries, users, and other object definitions. You can add the objects in the order in which they appear in this document or in the order that works best for your situation. To add SAS servers and libraries and other object definitions, see Chapter 7, “Tasks That You Perform in Enterprise Guide Administrator,” on page 69.

Creating Shared Access to the Repository Process

The repository process is a multiple-user process (called sasdsdb.exe). Only one repository process will run on each machine, and that one process will service all of the repository users on that machine. The repository process starts on demand from the first Enterprise Client application (for example, Enterprise Guide or Enterprise Guide Administrator) that tries to access the repository. It ends when the last such application ends.

To enable multiple users who are logged on to different machines to share information in one repository, configure the repository process as a Distributed COM (DCOM) server. A repository process can run either on your local machine or on a shared remote machine. Use the Windows `dcomcnfg` utility on the machine that will be accessed remotely by other machines. If a user's desktop machine does not have a metadata repository that needs to be shared with other users who are logged on to other machines, the Enterprise Clients or Enterprise Guide Administrator can access the local repository.

When you configure the repository process for DCOM access, the process must be set to run under the authority of a user ID account on the machine or a domain user ID that is allowed to run on the machine. The machine can be a Windows NT server that is shared by many users, or it can be any desktop machine capable of running a DCOM server process (such as an NT Workstation).

A current `dcomcnfg` problem on some Windows machines causes the following problem to occur. If a domain user ID is specified to run the repository server and there is a local machine user ID that has the same name, then the repository process execution will fail. `dcomcnfg` correctly sets the domain user ID to be used to run the process, but it incorrectly gives the local user ID rather than the domain user ID the needed user right to "Logon as a batch job." To avoid this problem, do not enter a domain user ID for the repository process identity that is the same user ID as a local machine user ID, because they are unrelated accounts. Or, after you run `dcomcnfg`, follow these steps:

- 1 Run the User Manager application. From the Start menu, select

Programs \blacktriangleright Administrative Tools \blacktriangleright User Manager

- 2 From the main menu select

Policies \blacktriangleright User Rights

Check the **Advanced User Rights** check box.

- 3 Go to **Logon as a batch job** and set the domain user ID (not the local user ID account) to have this right.

Because of this limitation, it is recommended that a new user ID account be added to the machine on which the shared repository process is to run. The account needs to be only a member of the Users group, but it also must have full-control file permission access to the repository database that is installed as part of the Enterprise Client or Enterprise Guide Administrator setup.

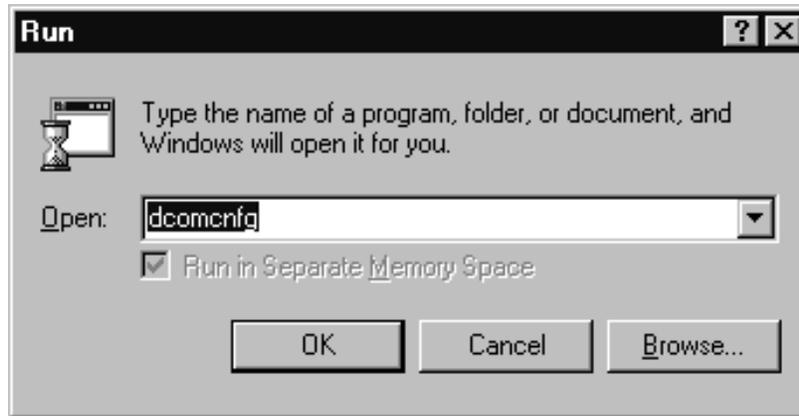
If you want to keep the repository process running at all times, then you can keep a copy of the Enterprise Guide Administrator running while you access the metadata repository. This is especially useful if you use Windows 95 or 98 machines as DCOM servers for the metadata repository process because, on these machines, there is no way to instantiate a DCOM server upon first request.

To configure a DCOM server process for sharing the metadata repository process:

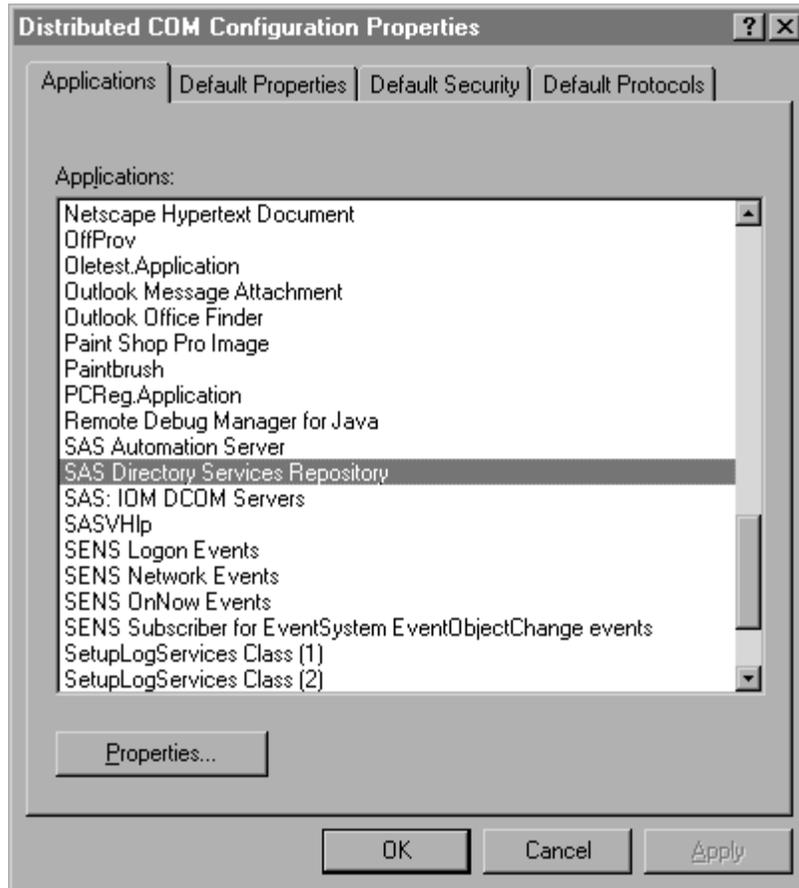
- 1 From the Windows taskbar on the server machine, select

Start \blacktriangleright Run

- 2 Type `dcomcnfg`, and then select **OK**.

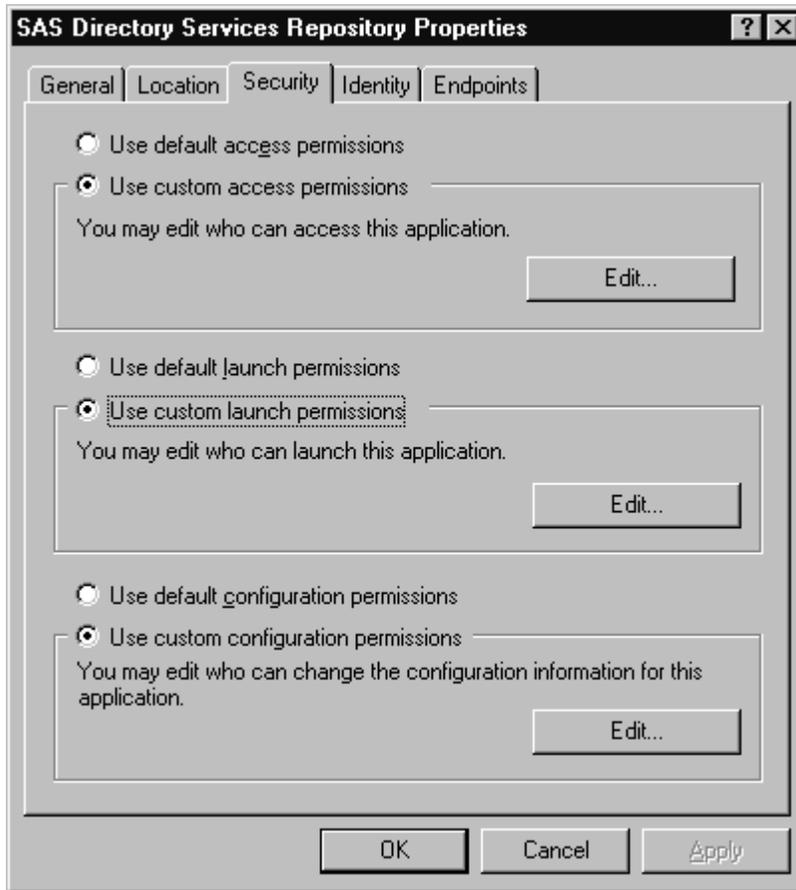


- 3 From the Distributed COM Configuration Properties window, select the **Applications** tab, select **SAS Directory Services Repository**, and then click **Properties**.

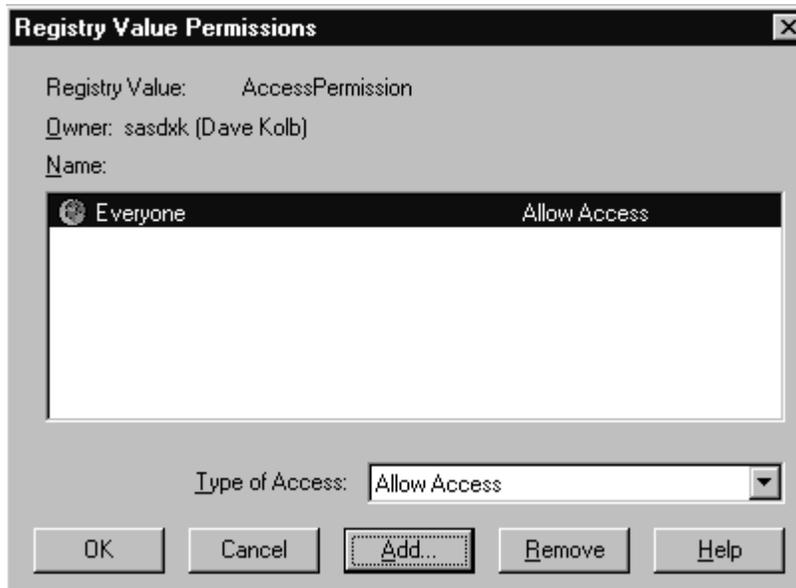


Note: The dcomcnfg dialog boxes might look slightly different depending on what version of Windows you are running and which service pack you have applied. Δ

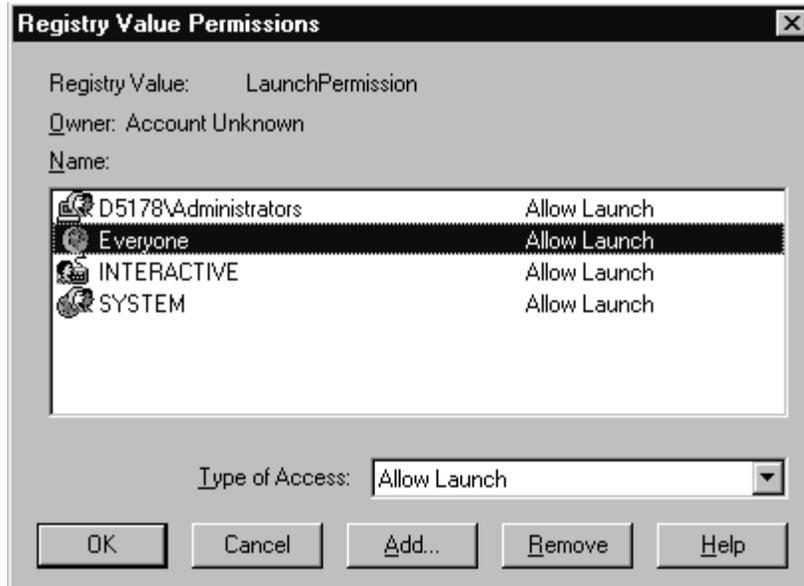
- 4 Select the **Security** tab. Select the **Use custom access permissions** option, and then click **Edit** to grant custom launch permissions.



Select **Add** in the Registry Value Permissions window, and either add the group called Everyone (you must type in the word **Everyone**) or add the specific groups and/or users to whom you want to grant or deny access. Then click **OK**.

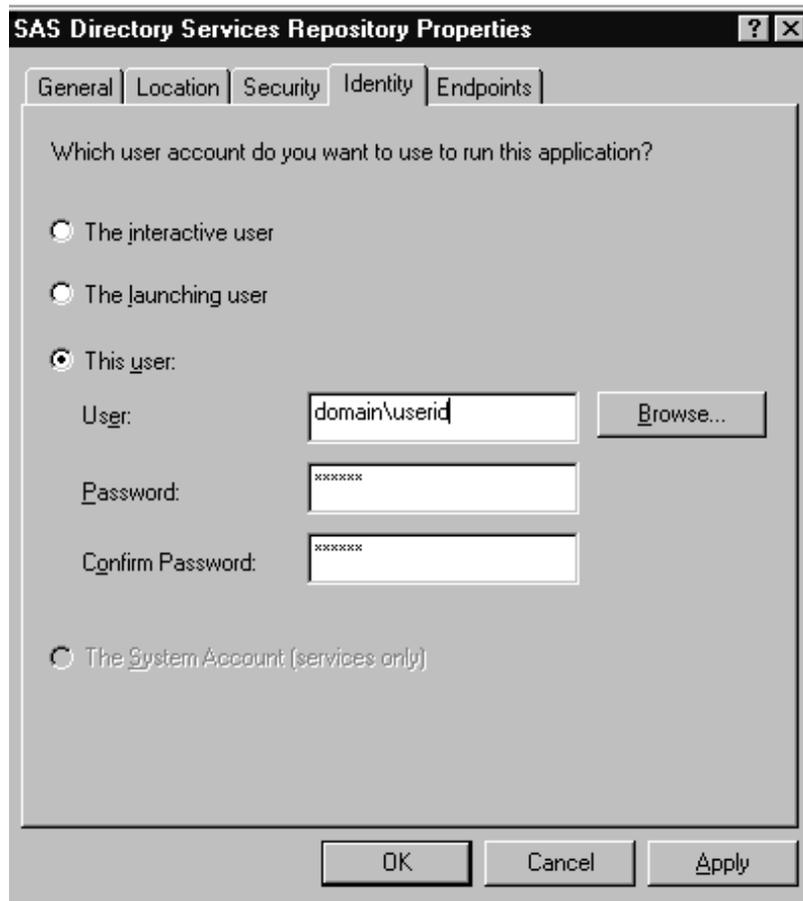


- 5 Repeat step 4 for **Use custom launch permissions**. (You can set **Everyone** for access and launch permissions because only users who are defined as administrators in the Enterprise Guide Administrator software will be allowed to make changes to the repository. If you choose, however, you can restrict certain users and/or groups.)



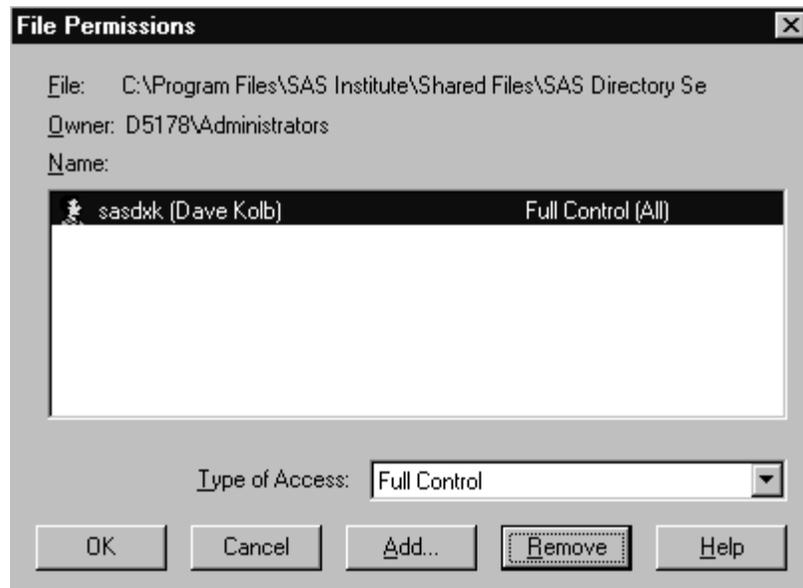
- 6 From the SAS Directory Services Repository Properties window, select the **Identity** tab. Enter a user ID (and domain) to be used to run the repository process, and then click **OK**.

Note: Due to a current DCOM limitation, this user ID should not be the same as the user ID for the interactive user who might run an Enterprise Client application on this server. Δ



- 7 At least this user must have full control of the repository database that was installed previously (either the name that was specified at installation or the name that was specified by the administrator in the Repository Maintenance Path window). If the Everyone group is shown, it can be removed because no other users need to have access to this file. To ensure full control in the NT file system, select the file in the My Computer directory. Click the right mouse button to access the shortcut menu. Select **Properties**. Click the **Security** tab, and then click **Permissions**. Either add a group that the user ID (used to run the repository DCOM server) belongs to, or add the actual user ID. In the **Type of Access** field, select **Full Control**, and then click **OK**.

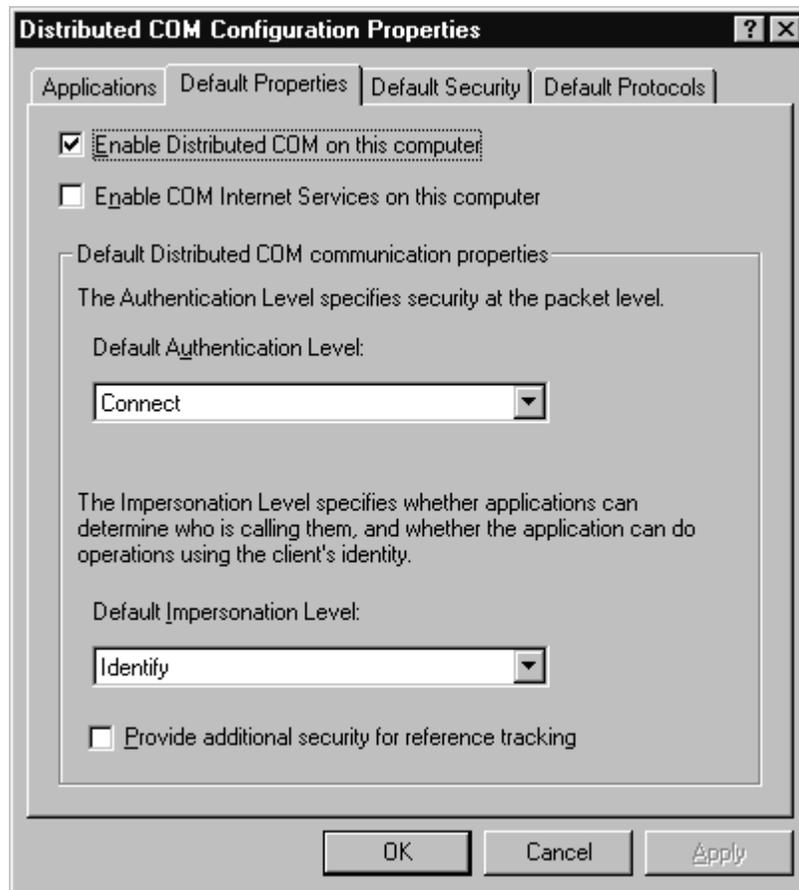
To control access on a FAT drive, do not make the repository database file available for read/write access in a shared directory on a network.



- 8 Restart the machine so that all the settings take effect.
- 9 Finally, if DCOM is not enabled on the server machine, then enable it now by selecting

Start ► Run ► dcomcnfg

Select the **Default Properties** tab from the Distributed COM Configuration Properties window. Select **Enable Distributed COM on this computer**.



Maintaining Repositories

As part of repository maintenance you can

- change the path of the local repository that you are using
- take a repository offline
- compact the repository
- repair the repository.

You must be in Admin Mode to perform maintenance. Select

File ► **Repository Maintenance**

The Repository Maintenance dialog box appears.

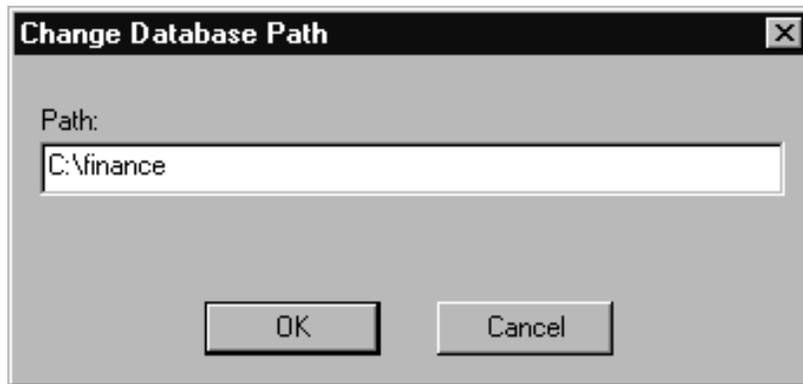


Changing the Path

To change the path that the server process is pointing to, click **Path** in the Repository Maintenance window and type the pathname. Note that the path is on the server that is specified in

File ► **Open Repository**

This capability is useful in the following offline scenario, where changes to the repository database are made all at once and then the database path is changed to point to the new database. Note that changing the path for a repository server changes the path for all users of that repository server.



Taking a Shared Repository Offline

A good reason for momentarily taking a shared repository offline is so that you copy it and make many additions and corrections to the copy rather than to the current working repository database. With the correct repository active, click **Offline** in the Repository Maintenance window. If users try to access the repository while it is offline, then they get a message that the repository is not available. You can then make a copy of the repository in the file system and click **Online** to make the repository available for users again while you make updates to the copy after moving it to another machine.

Since only one metadata repository database can be active on a machine, the copy of the database must be moved to another machine. The machine must have Enterprise Guide Administrator (or another Enterprise Client application) installed, and it must not share the repository with other machines. Typically this machine would be the administrator's desktop machine.

The database should be kept offline for as little time as possible and should not be offline when you want to switch to a different database. Copy the database file to a machine where a repository server is installed. Typically this would be the administrator's local machine that is not being used as a shared server, so that no users would be affected by changes that are made to the repository database.

Follow these steps to make changes to the copy of the repository database:

- 1 To access the local repository or the machine where the copy was placed, select

File ► **Open**

- 2 To change the path to the copy of the repository database, select

File ► **Repository Maintenance**

- 3 Make updates to the data in Enterprise Guide Administrator.
- 4 Place the current repository offline so that the modified repository database can be copied back to the original server machine with a name different from that machine's current repository database.
- 5 To access the shared repository server that will get the new repository database, select

File ► Open

- 6 To change the path to the new repository database, select

File ► Repository Maintenance

The new repository database will be effective immediately for all users who are using this repository server.

Compacting

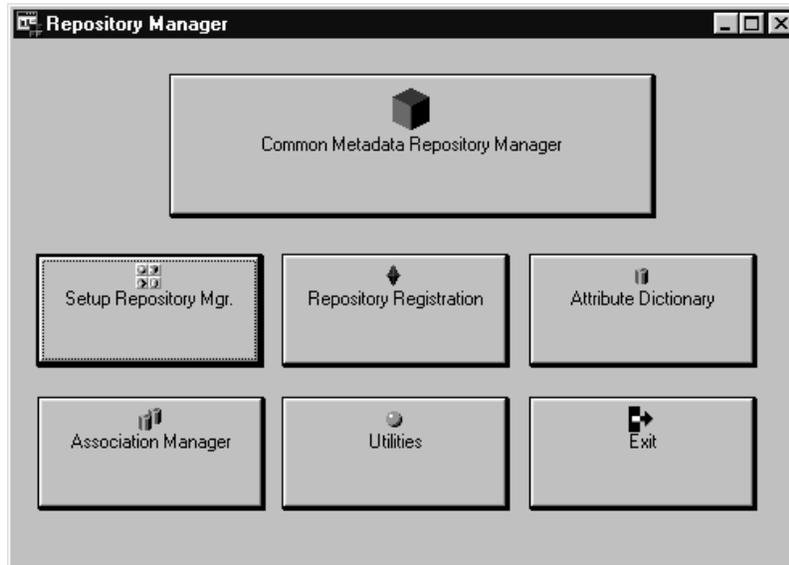
After defining or deleting many objects in a metadata repository, you can compact the repository to make it smaller and to help it run faster. Compacting also fixes a repository that has been damaged. From the Repository Maintenance window, select **Compact**. You do not need to take the database offline before you begin compacting.

Registering MDDBs

In order to use existing MDDBs (also referred to as cubes) with Enterprise Client applications such as Enterprise Guide, you must first register them on the SAS server on which the MDDBs reside. The SAS server must be running SAS Release 8.1 or later releases, and it can be either a local or remote server that is running on Windows NT, UNIX, or a mainframe.

To register existing MDDBs:

- 1 Invoke SAS (Release 8.1 or later releases) on the server on which the MDDBs reside.
- 2 If a library reference does not exist, then create a reference to the existing MDDBs.
- 3 Type **reposmgr** in the SAS command window to start the Common Metadata Repository Manager.



- 4 Click **Setup Repository Mgr** to open the Repository Manager Setup window.



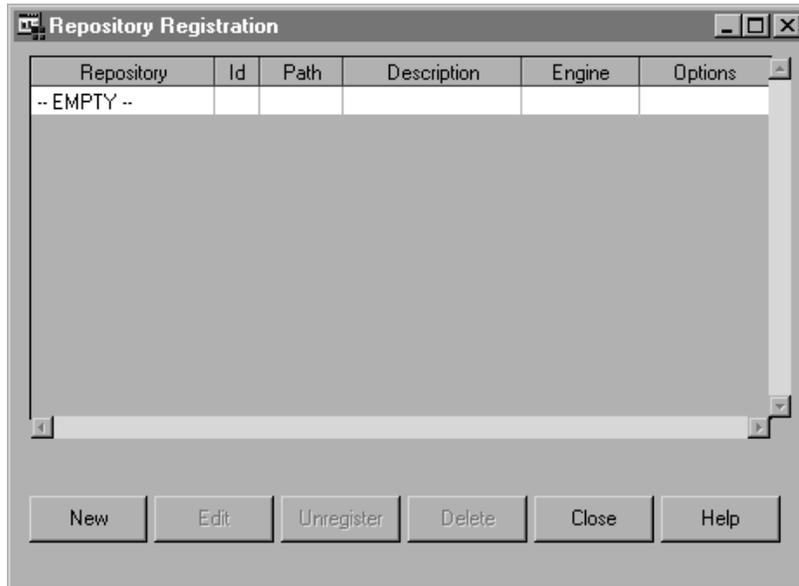
- 5 In the Repository Manager Setup window, fill in the fields as follows:

Library	Leave RPOSMGR (the default value).
Path	Type the location of an existing but empty directory in the server's file system.
Write values to system registry	Select this check box so that multiple users can view the cube and users can access the cube from a remote server. If this option is grayed out, then another SAS process is running on the server. You must stop all SAS processes on the server to make this option selectable. You can also add the RPOSMGR libref to the startup statement for the server, as in this example:

```
libname RPOSMGR 'c:\myrepmgr'
```

Adding this statement makes Enterprise Client applications aware of the location of the repository manager each time the server is started.

- 6 Click **OK** to close the window. A dialog box asks if you want to install all the repository system files. Click **Yes** and then click **OK**.
- 7 From the Repository Manager window, click **Repository Registration**. The Repository Registration window opens.



- 8 Click **New** to open the Register Repository window.

Register Repository

Repository Name:

Path:

Description:

Engine:

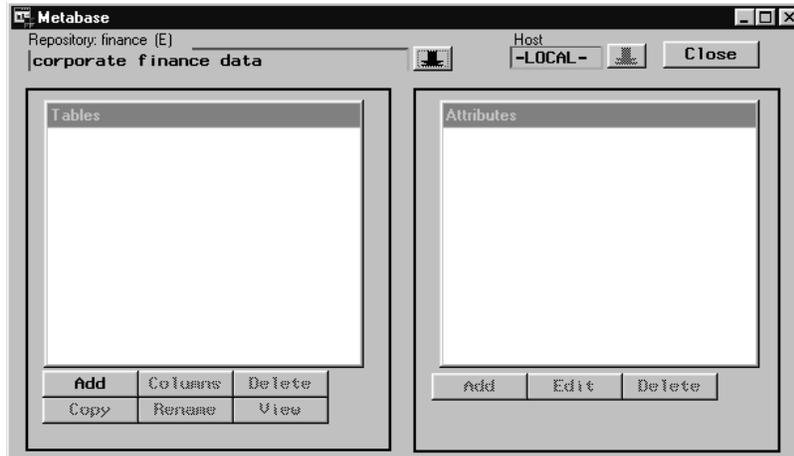
Library Options:

Readonly access.

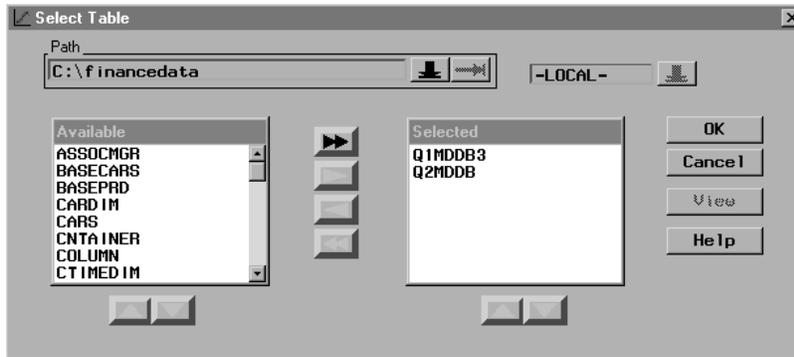
- 9 Type the appropriate information in the fields. The **Repository Name** field is case sensitive. The **Path** field must point to an existing directory. Do not select the **Readonly access** check box.
- 10 Click **OK** to close the Register Repository window. Click **Close** to close the Repository Registration window. Click **Exit** to close the Repository Manager window.

Next, you must use the Metabase to specify the MDDB data that you want to register. You must also add the Open OLAP server attribute so that the MDDB cube can be accessed by Enterprise Client applications.

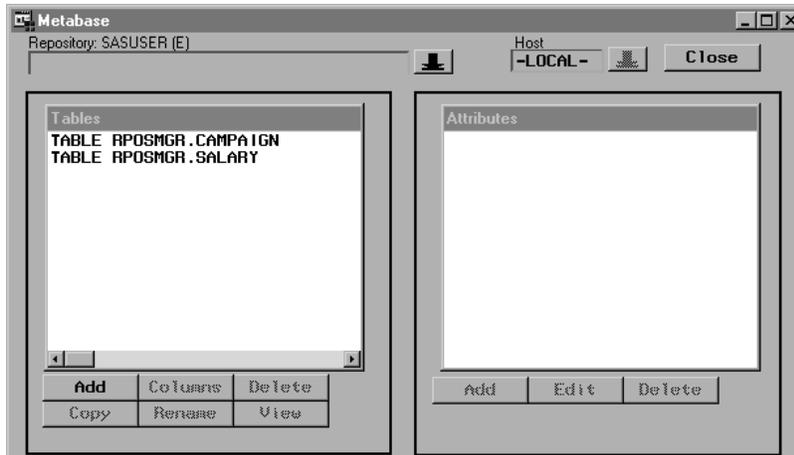
- 1 Type **metabase** on the SAS command line to open the Metabase window. Specify the path for your repository.



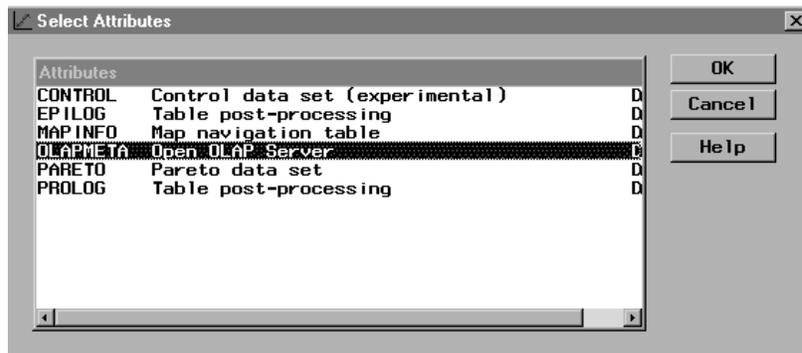
- 2 Select **add** to open the Select Table window.



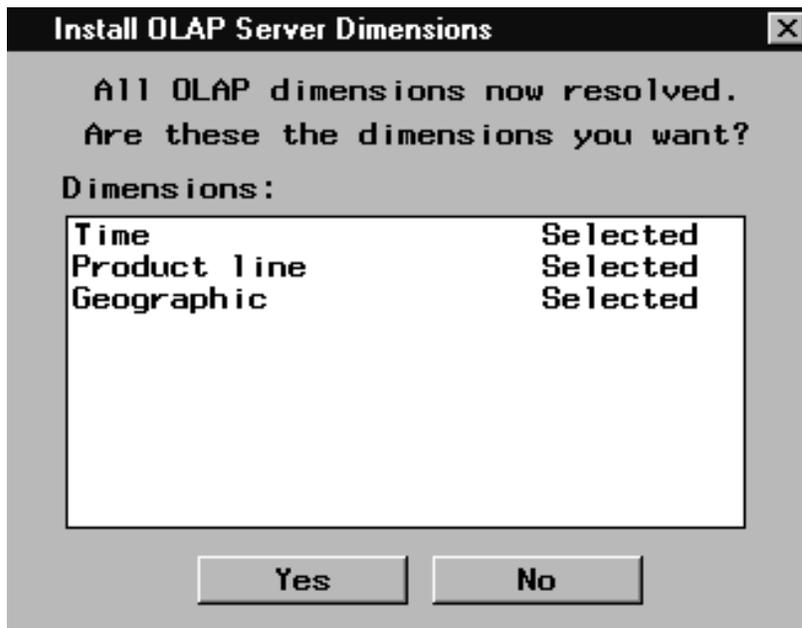
- 3 In the **Available** list box, select the MDDBs that you want to register. Click the arrow to move the MDDBs to the **Selected** list box. When you finish, click **OK** to return to the Metabase window.



- 4 Select an MDDB from the **Tables** list box. Below the **Attributes** list box, click **Add**. The Select Attributes window opens.



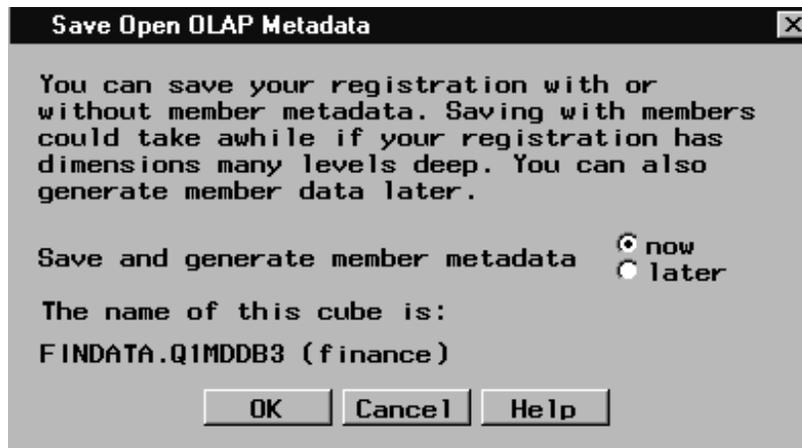
- 5 Select **OLAPMETA** in the Select Attributes window and then click **OK**.
- 6 A dialog box asks whether you want to base the Open OLAP Server dimensions on the existing SAS metabase hierarchies. Click **Yes**. The Install OLAP Server Dimensions dialog box opens.



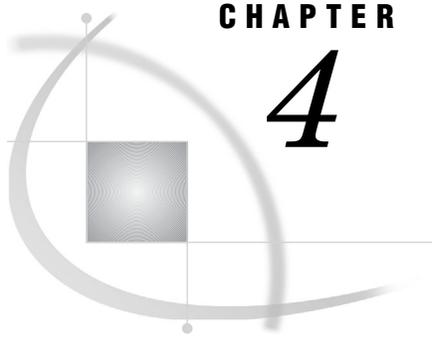
- 7 Click **yes** to install all of the OLAP dimensions. The Open OLAP Server cube registration window opens.



- 8 Click **OK** to accept the registration. The Save Open OLAP Metadata window opens.



- 9 Click **OK** to save the registration with the metadata, then click **Close** in the Metabase window. You can now close your SAS session.



CHAPTER

4

Configuring an IOM Bridge Connection

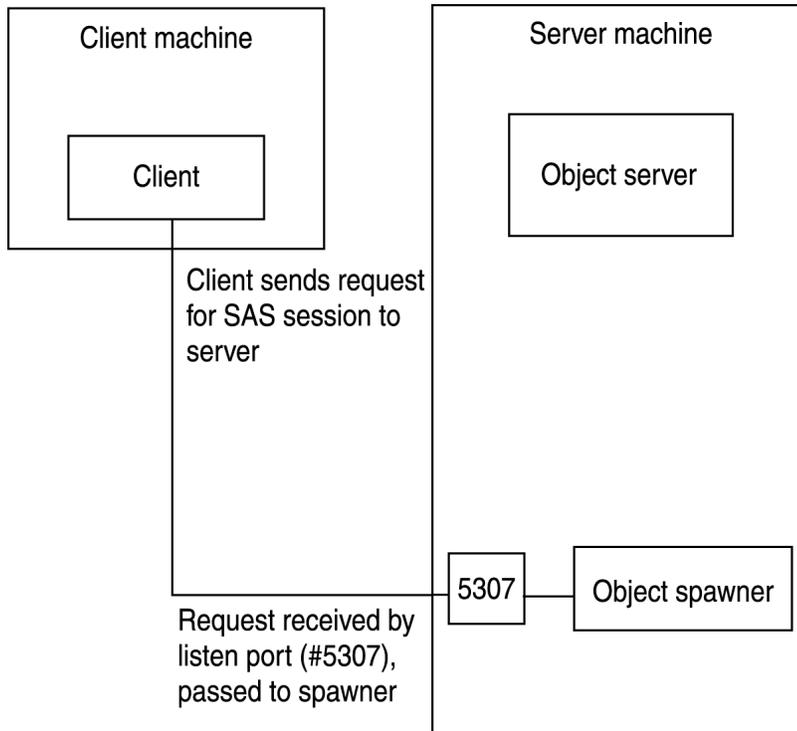
<i>Introduction</i>	33
<i>How an IOM Bridge Connection Works</i>	33
<i>Setting Up an Object Spawner</i>	36
<i>Step 1 — Setting Up a Spawner Login</i>	36
<i>Step 2 — Identifying Client Users</i>	38
<i>Step 3 — Choosing and Defining Ports</i>	39
<i>Step 4 — Creating the Spawner Configuration File</i>	39
<i>Step 5 — Installing and Starting the Spawner</i>	44
<i>Installing and Starting the Spawner on Windows</i>	44
<i>Installing and Starting the Spawner on UNIX</i>	45
<i>Step 6 — Testing the Server Definition</i>	46
<i>Step 7 — Setting Up the Production Environment</i>	46

Introduction

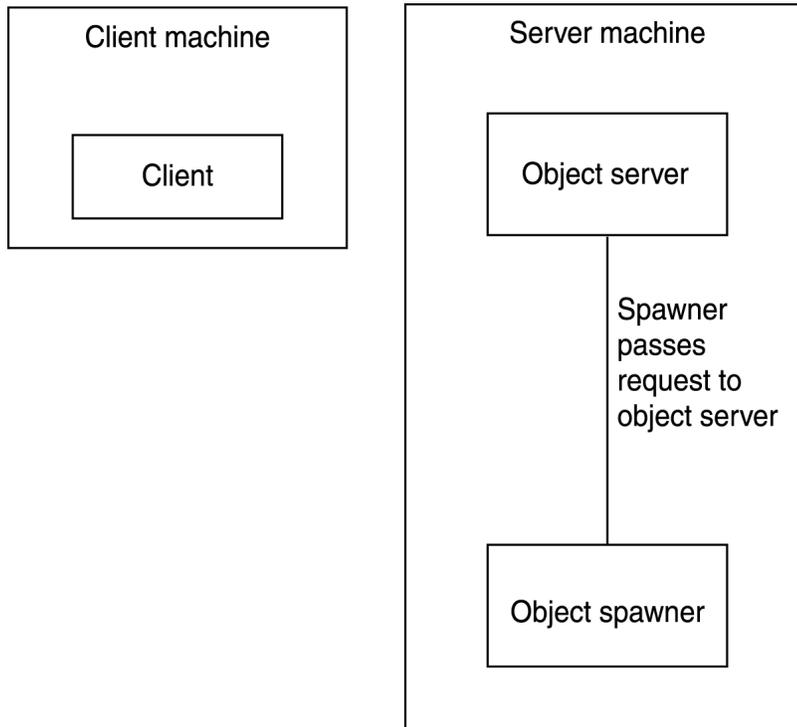
One way to communicate between Enterprise Guide and a server is to use an IOM Bridge connection. The server uses an object spawner and one or more object servers to communicate with the client. The object spawner is a service that listens for requests for SAS sessions that come from the client, then passes those requests to the object server. The object server accepts the requests from the object spawner, initiates a SAS session, and then communicates with the client directly. An object spawner can run on Windows, UNIX, or OS/390 machines.

How an IOM Bridge Connection Works

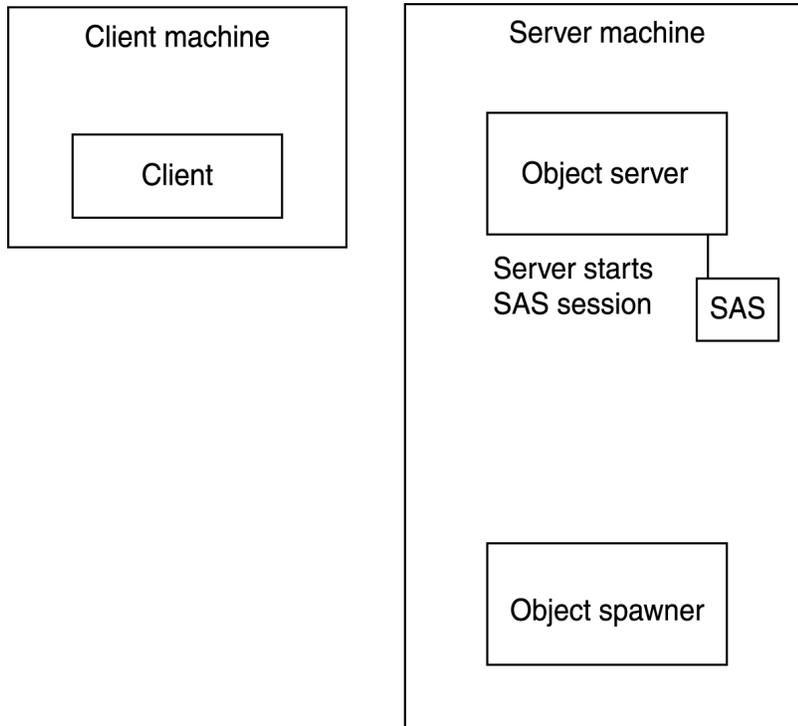
When a client wants to start a SAS session on a server by using an IOM Bridge connection, the request goes to the object spawner by way of a “listen port,” a port that is designated to listen for requests from clients.



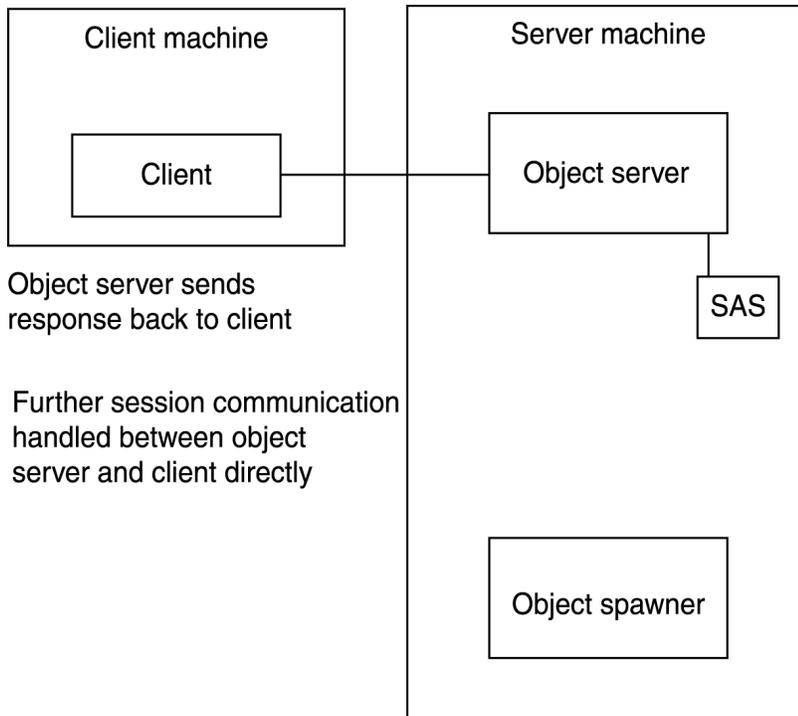
The object spawner then passes the request to the object server.



The object server then initiates a SAS session.



After the SAS session has been established, the object server sends a response back to the client and begins communicating with the client directly. All further communications between the client and the object server are direct, rather than through the server.



Setting Up an Object Spawner

The main task that you must perform in setting up an object spawner is to define the object spawner and object server, which is accomplished through the use of a configuration file. However, you must also perform several tasks to prepare your system for running the object spawner and to set up the Enterprise Guide Administrator to communicate with the spawner.

Setting up an object spawner consists of the following basic steps:

- 1 Set up a login for the spawner. This is the user ID under which the spawner runs.
- 2 Identify which users can run the spawner and give access rights to those users.
- 3 Choose and define the spawner and server ports.
- 4 Create the spawner configuration file.
- 5 Install and start the spawner.
- 6 Test the server definition.
- 7 Set up the production environment.

Details about each step in this procedure are presented in the following sections. See Appendix 1, “Object Spawner Details,” on page 125 for information about setting up object spawners on UNIX, OS/390, and Windows.

Step 1 — Setting Up a Spawner Login

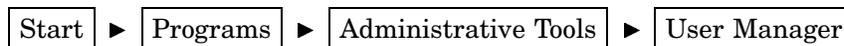
You must first set up a login for the spawner on the server machine. Creating a dedicated login for the server, rather than using an existing login, simplifies maintenance tasks. With a dedicated account, administrators always know which account to use to run the spawner and which account to check if the spawner has problems. The spawner login must have administrator privileges.

After you set up the spawner login, you must give the necessary rights to the account. For a Windows NT login, you must give these rights to the account:

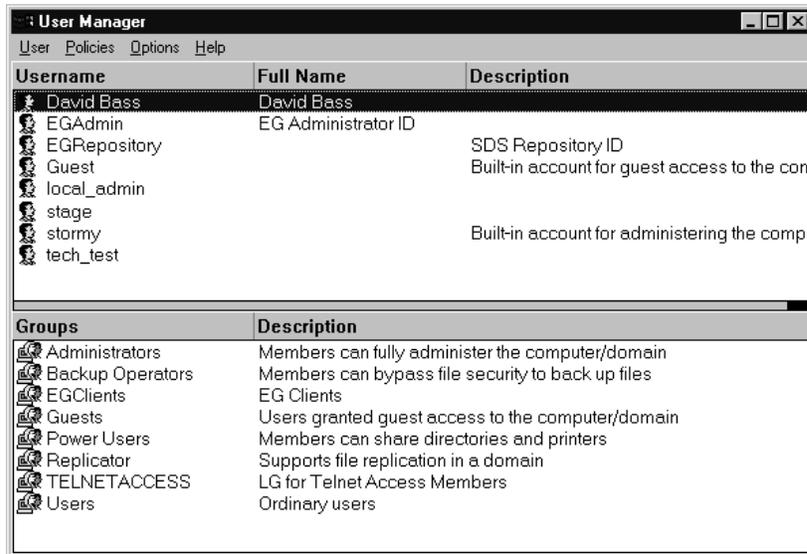
- Act as part of the operating system
- Increase quotas
- Replace a process level token.

To set these rights:

- 1 On the server machine, select



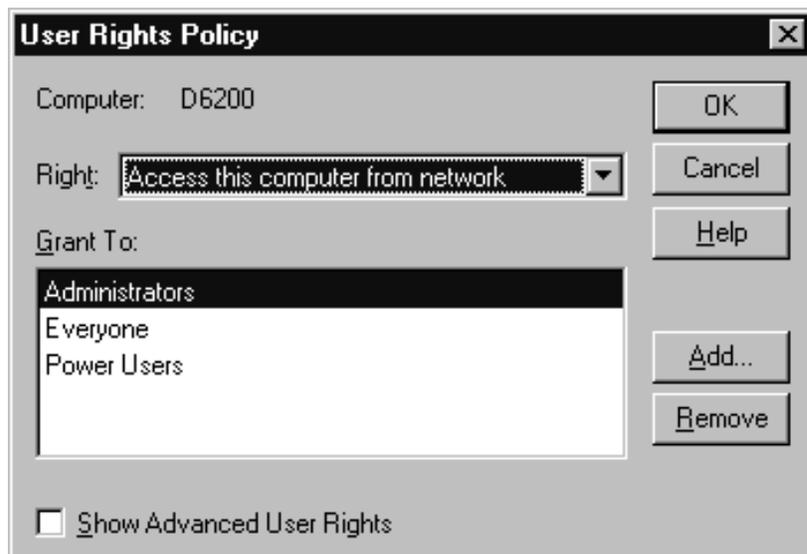
The User Manager window opens.



2 Select



The User Rights Policy window opens.



- 3 Select the **Show Advanced User Rights** check box.
- 4 From pull-down menu in the **Right** field, select **Act as part of the operating system**, and click **Add**. The Add Users and Groups window opens.
- 5 Select the login that you created to run the spawner, and click **Add**.
- 6 From the User Rights Policy window, repeat the procedure to add these rights:
 - Increase quotas
 - Replace a process level token.
- 7 Close all windows.

Step 2 — Identifying Client Users

After you set up the login that will run the spawner, you must identify the user IDs and groups that can launch SAS by using the spawner. One approach is to create a Windows user group that contains the IDs of all users who need to access the spawner. With the user group in place, adding a new user requires only adding the user's ID to the group, rather than explicitly specifying the required permissions.

To create a Windows user group:

- 1 Select

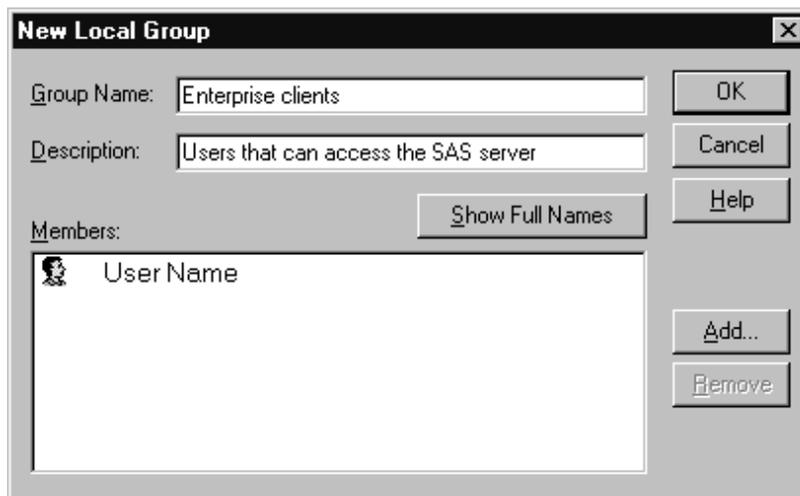


The User Manager window opens.

- 2 Select



The New Local Group window opens.



- 3 Select **Add** to open the Add Users and Groups window. Select the ID for each user in the group and select **Add**.

When you finish, select **OK** in the New Local Group window to return to the User Manager window.

- 4 From the User Manager window, select



The User Rights Policy window opens.

- 5 Click the **Show Advanced User Rights** check box, then select **Log on as a batch job** from the pull-down menu in the **Right** field.
- 6 Click **Add** to open the Add Users and Groups window. Select the client group that you just created and click **Add**. Click **OK** to close the window.
- 7 Click **OK** to close the User Rights Policy window, then close the User Manager window.

Step 3 — Choosing and Defining Ports

After you create the spawner login and the user group, you must choose and define the spawner ports. You use the spawner operator port to issue commands to the spawner. The spawner listen port receives requests from client machines.

To configure the ports:

- 1 Choose the spawner and server ports. You must find and reserve an operator port for the spawner, as well as one listen port for each server. Check the services file for available port numbers greater than 1024.
- 2 Add the following line to the services file to define the operator port:

```
sasobjoper    xxxx/tcp    #SAS/Object spawner operator
```

sasobjoper is a unique name for the operator port, and *xxxx* is the port number that you chose for the operator port.

- 3 Add the following line to the services file for each server port:

```
sasobjspawn xxxx/tcp    #SAS/Object spawner listen
```

sasobjspawn is a unique name for each spawner listen port, and *xxxx* is the port number that you chose.

- 4 Save and close the services file.

Step 4 — Creating the Spawner Configuration File

After you assign the port numbers for the spawner operator port and the listen ports, you can create the object spawner configuration file. This file provides the commands to define the server and spawner. The entries in the configuration file are in the form

```
attribute: value
```

See Appendix 1, “Object Spawner Details,” on page 125 for detailed information about configuration file entries.

To create a simple configuration file:

- 1 Start the spawner definition. After a commented header (which is optional; use # to identify comments), enter a value for the dn: attribute. This attribute identifies the start of the spawner definition and must be the first attribute that is listed. Use any unique name as the value for dn:.

```
#
# Object Spawner definition
#
dn: sasSpawnercn=myspawner
```

- 2 Use the objectClass attribute to identify the definition as a spawner definition. The value for objectClass must be sasSpawner.

```
#
# Object Spawner definition
#
dn: sasSpawnercn=myspawner
objectClass: sasSpawner
```

- 3 Name the spawner. Use the sasSpawnercn attribute to assign a name to the spawner.

```
#
# Object Spawner definition
```

```
#
dn: sasSpawnercn=myspawner
objectClass: sasSpawner
sasSpawnercn: myspawner
```

- 4 Identify the machine on which the spawner runs. Use the `sasMachineDNSName` attribute to specify where the spawner runs. The value for this attribute is the DNS name or the IP address of the machine.

```
#
# Object Spawner definition
#
dn: sasSpawnercn=myspawner
objectClass: sasSpawner
sasSpawnercn: myspawner
sasMachineDNSName: M5678
```

- 5 Specify the domain name of the spawner machine. The DNS name and the domain name combine to form the complete IP address of the machine.

```
#
# Object Spawner definition
#
dn: sasSpawnercn=myspawner
objectClass: sasSpawner
sasSpawnercn: myspawner
sasMachineDNSName: M5678
sasDomainName: pc.mycorp.com
```

- 6 Turn on logging. The `sasLogFile` attribute specifies the pathname and filename of the spawner activity log.

```
#
# Object Spawner definition
#
dn: sasSpawnercn=myspawner
objectClass: sasSpawner
sasSpawnercn: myspawner
sasMachineDNSName: M5678
sasDomainName: pc.mycorp.com
sasLogFile: C:\spawner\spawner.log
```

- 7 Turn on verbose logging.

```
#
# Object Spawner definition
#
dn sasSpawnercn=myspawner
objectClass: sasSpawner
sasSpawnecn: myspawner
sasMachineDNSName: M5678
sasLogFile: C:\spawner\spawner.log
sasVerbose: True
```

- 8 Specify the operator port. Use the `sasOperatorPort` attribute to specify the port that is used to issue commands to the spawner during operation.

```
#
# Object Spawner definition
```

```
#
dn sasSpawnercn=myspawner
objectClass: sasSpawner
sasSpawnecn: myspawner
sasMachineDNSName: M5678
sasLogFile: C:\spawner\spawner.log
sasVerbose: True
sasOperatorPort: 5307
```

- 9** Include a description. Use the description attribute to describe the spawner.

```
#
# Object Spawner definition
#
dn sasSpawnercn=myspawner
objectClass: sasSpawner
sasSpawnecn: myspawner
sasMachineDNSName: M5678
sasLogFile: C:\spawner\spawner.log
sasVerbose: True
sasOperatorPort: 5307
description: SAS Object Spawner
```

- 10** Start the server definition. As with the spawner definition, include a header if you want, then enter a unique value for the dn attribute to identify the start of the server definition. You must include a blank line before the start of the server definition.

```
#
# Object Spawner definition
#
dn sasSpawnercn=myspawner
objectClass: sasSpawner
sasSpawnecn: myspawner
sasMachineDNSName: M5678
sasLogFile: C:\spawner\spawner.log
sasVerbose: True
sasOperatorPort: 5307
description: SAS Object Spawner
```

```
#
# Object Server definition
#
dn: sasServercn=myserver_1
```

- 11** Use the objectClass attribute to identify the definition as a server definition. The value for objectClass must be sasServer.

```
#
# Object Spawner definition
#
dn sasSpawnercn=myspawner
objectClass: sasSpawner
sasSpawnecn: myspawner
sasMachineDNSName: M5678
sasLogFile: C:\spawner\spawner.log
sasVerbose: True
```

```

sasOperatorPort: 5307
description: SAS Object Spawner

#
# Object Server definition
#
dn: sasServercn=myserver_1
sasServercn: myserver_1
objectClass: sasServer

```

12 Name the server. Use the `sasServercn` attribute to assign a name to the server.

```

#
# Object Spawner definition
#
dn sasSpawnercn=myspawner
objectClass: sasSpawner
sasSpawnecn: myspawner
sasMachineDNSName: M5678
sasLogFile: C:\spawner\spawner.log
sasVerbose: True
sasOperatorPort: 5307
description: SAS Object Spawner

#
# Object Server definition
#
dn: sasServercn=myserver_1
sasServercn: myserver_1
objectClass: sasServer

```

13 Specify the server port. Use the `sasPort` attribute to specify the listen port that is used to communicate with the client.

```

#
# Object Spawner definition
#
dn sasSpawnercn=myspawner
objectClass: sasSpawner
sasSpawnecn: myspawner
sasMachineDNSName: M5678
sasLogFile: C:\spawner\spawner.log
sasVerbose: True
sasOperatorPort: 5307
description: SAS Object Spawner

#
# Object Server definition
#
dn: sasServercn=myserver_1
sasServercn: myserver_1
objectClass: sasServer
sasPort: 5308

```

14 Use the `sasMachineDNSName` attribute to identify the machine on which the server runs. The value for this attribute is the DNS name or the IP address of the machine.

```

#
# Object Spawner definition
#
dn sasSpawnercn=myspawner
objectClass: sasSpawner
sasSpawnecn: myspawner
sasMachineDNSName: M5678
sasLogFile: C:\spawner\spawner.log
sasVerbose: True
sasOperatorPort: 5307
description: SAS Object Spawner

#
# Object Server definition
#
dn: sasServercn=myserver_1
sasServercn: myserver_1
objectClass: sasServer
sasPort: 5308
sasMachineDNSName: M5678

```

- 15** Identify the type of connection to the client by using the `sasProtocol` attribute. Specify a value of `bridge` for an IOM Bridge connection.

```

#
# Object Spawner definition
#
dn sasSpawnercn=myspawner
objectClass: sasSpawner
sasSpawnecn: myspawner
sasMachineDNSName: M5678
sasLogFile: C:\spawner\spawner.log
sasVerbose: True
sasOperatorPort: 5307
description: SAS Object Spawner

#
# Object Server definition
#
dn: sasServercn=myserver_1
sasServercn: myserver_1
objectClass: sasServer
sasPort: 5308
sasMachineDNSName: M5678
sasProtocol: bridge

```

- 16** Specify the command that was used to start a SAS session on the server with the `sasCommand` attribute. The value should specify the complete path, rather than a relative path, and it must be enclosed in quotation marks if the path contains blanks.

```

#
# Object Spawner definition
#
dn sasSpawnercn=myspawner
objectClass: sasSpawner

```

```

sasSpawnecn: myspawner
sasMachineDNSName: M5678
sasLogFile: C:\spawner\spawner.log
sasVerbose: True
sasOperatorPort: 5307
description: SAS Object Spawner

#
# Object Server definition
#
dn: sasServercn=myserver_1
sasServercn: myserver_1
objectClass: sasServer
sasPort: 5308
sasMachineDNSName: M5678
sasProtocol: bridge
sasCommand: 'C:\Program Files\SAS\SAS System\9.0\sas.exe

```

17 Include a description by using the description attribute.

```

#
# Object Spawner definition
#
dn sasSpawnercn=myspawner
objectClass: sasSpawner
sasSpawnecn: myspawner
sasMachineDNSName: M5678
sasLogFile: C:\spawner\spawner.log
sasVerbose: True
sasOperatorPort: 5307
description: SAS Object Spawner

#
# Object Server definition
#
dn: sasServercn=myserver_1
sasServercn: myserver_1
objectClass: sasServer
sasPort: 5308
sasMachineDNSName: M5678
sasProtocol: bridge
sasCommand: 'C:\Program Files\SAS\SAS System\9.0\sas.exe
sasDomainName: pc.mycorp.com

```

Step 5 — Installing and Starting the Spawner

Follow the procedure for the platform on which the object spawner runs.

Installing and Starting the Spawner on Windows

- 1 Determine the location of objspawn.exe. This file is part of the SAS installation and is typically located in the `\inttech\sasexe\` directory under the SAS installation. A typical complete path is

```
c:\Program Files\SAS Institute\SAS\V8\inttech\sasexe\objspawn
```

- 2 Open a Command Prompt window.
- 3 Change to the root directory.
- 4 Issue the following command to install the spawner:

```
"objspawn_path\objspawn" -configFile "config_path\config.cfg"
-install
```

where *objspawn_path* is the path to objspawn.exe, and *config_path* is the path to the configuration file that you created earlier. Replace config.cfg with the name that you chose for your configuration file. Here is an example:

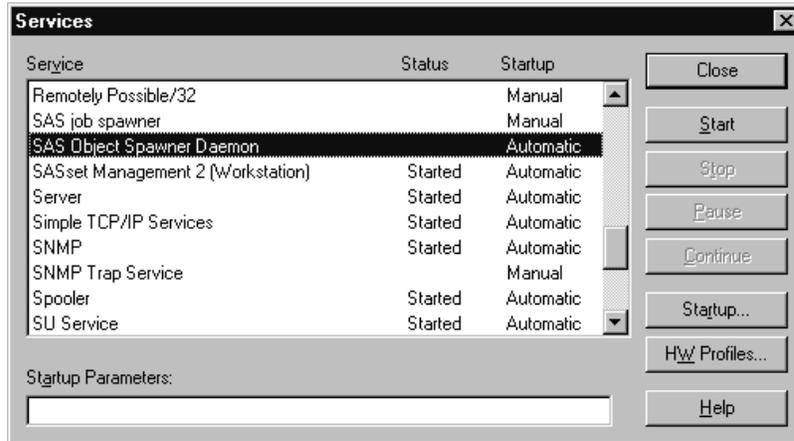
```
"c:\Program Files\SAS Institute\SAS\V8\inttech\sasexe\objspawn"
-configFile "c:\Program Files\SAS Institute\SAS\V8\inttech\sasexe\
myspawner.cfg" -install
```

Be sure to specify the entire path for objspawn and the configuration files, and be sure that the paths are enclosed in quotation marks.

- 5 After installing the spawner, you must start it. From the Windows desktop, select



The Services window opens.



- 6 In the Services window, scroll through the **Services** box until you see SAS Object Spawner Daemon. Select this item, and click **Start**.
- 7 If the spawner starts successfully, then the **Status** column will show a value of "Started." If you see any error messages, then check the log file and make corrections to the configuration file before attempting to start the spawner again.

Installing and Starting the Spawner on UNIX

- 1 Log on to a user ID that has root privileges.
- 2 Determine the directory where the object spawner resides, and change to that directory.
- 3 Issue the following command to start the spawner:

```
./objspawn -configFile config_path\config.cfg
```

where *config_path* is the path to the configuration file that you created earlier. Replace config.cfg with the name that you chose for your configuration file. Here is an example:

```
./objspawn -configFile ~\sapwner\myspawner.cfg
```

- 4 The message “Objspawn initialization” appears if the spawner started successfully. If it did not, then check the log file, make any needed corrections to the configuration file, and attempt to start the spawner again.

Step 6 — Testing the Server Definition

To check whether the object spawner receives requests and starts object server sessions properly, you must use the Enterprise Guide Administrator application. Follow these steps:

- 1 Start Enterprise Guide Administrator.
- 2 Select

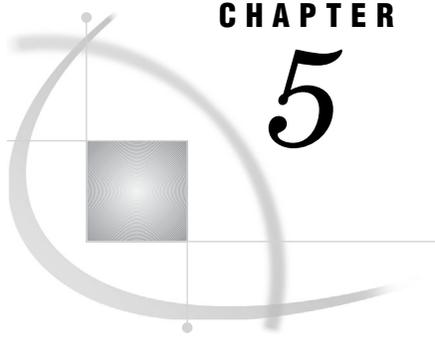


Then select **Server** from the New window.

- 3 The Server - General window opens. Give the server a name (and description, if you want), and click **Next**.
- 4 Accept the default value for the Server - Type window and click **Next**.
- 5 The Server - Connection window opens. For **Connection Protocol**, specify IOM. In the **Hostname** field, enter the value that you used for the spawner's sasMachineDNSName attribute in the configuration file. In the **Port** field, specify the value for sasPort from the configuration file.
- 6 For purposes of testing, you can accept the default values for the remainder of the windows through the Server - Options window.
- 7 In the Server - Test window, select the **Test** button. If the object spawner and object server are specified correctly in the configuration file, and if Enterprise Guide Administrator is correctly referring to the spawner, then the window will display a “Test Successful” message. If you receive an error message, then select **Show Details** from the SDS Server Wizard error message window to see details about the message, and then check the object spawner log.

Step 7 — Setting Up the Production Environment

After you verify the configuration and connections, you can create a permanent connection to the object spawner and make the defined server available to client users.



CHAPTER

5

Configuring a DCOM Connection

<i>Overview</i>	47
<i>Requirements</i>	47
<i>DCOM Servers</i>	47
<i>SAS Clients</i>	48
<i>Step 1 — Enabling DCOM on the Client and the Server</i>	48
<i>Step 2 — Creating Client Groups</i>	49
<i>Step 3 — Setting SAS Launch Policies</i>	50
<i>Setting Global Launch Policies</i>	50
<i>Setting Launch Policies for SAS</i>	54

Overview

You can configure a Distributed Component Object Model (DCOM) server that Enterprise Client applications can use to access a remote SAS session (a SAS object server). (See Enterprise Client Applications and SAS Object Servers.)

Before you can use DCOM to connect to a remote object server, you must enable DCOM on the SAS server and on the client machine. Then you must define properties that affect how and where SAS is launched on the SAS server. You configure the DCOM properties in a Windows environment by using the administration utility `dcomcnfg`. The `dcomcnfg` utility enables you to perform one of two actions: to set security policies for the server globally so that they affect all COM-enabled applications, or to set specific policy settings to be applied on a per-application component basis. In general, SAS servers use the default settings in the DCOM configuration. You should modify security settings based on the needs of your users and your site.

Note: After you configure for a DCOM connection to a remote SAS session, you must test the connection from the remote machine that you are connecting to. If you are trying to set up a connection to SAS by using your local machine as the default SAS server, then use a COM connection (see “Managing Servers” on page 72). You cannot use DCOM to communicate with SAS that is running on your local machine. △

Requirements

DCOM Servers

- NT server or NT Workstation is supported as DCOM servers.

- SAS software, SAS Integration Technologies, and other SAS products that are installed on the remote SAS server. For complete server setup information, see “Installing SAS Servers” on page 10.

SAS Clients

- Windows 95 or a later version.

Note: Windows 95 and Windows 98 are supported as DCOM clients only; they cannot be DCOM servers. However, you can run SAS locally for Windows 95 and Windows 98 users. △

Note: If you want to use a Windows 95 machine as a client, then you must verify that Microsoft’s DCOM95 Version 1.2 upgrade to enable DCOM has been installed on the machine. If the upgrade has not been installed, then you must install it yourself. DCOM95 Version 1.2 is available from Microsoft. △

Step 1 — Enabling DCOM on the Client and the Server

In order to establish a DCOM session, you must ensure that DCOM is enabled on both the client machine and the server machine. Perform the following steps both on the client machine that is running Enterprise Client software and on the SAS object server.

- 1 From the Windows taskbar on the server machine, select

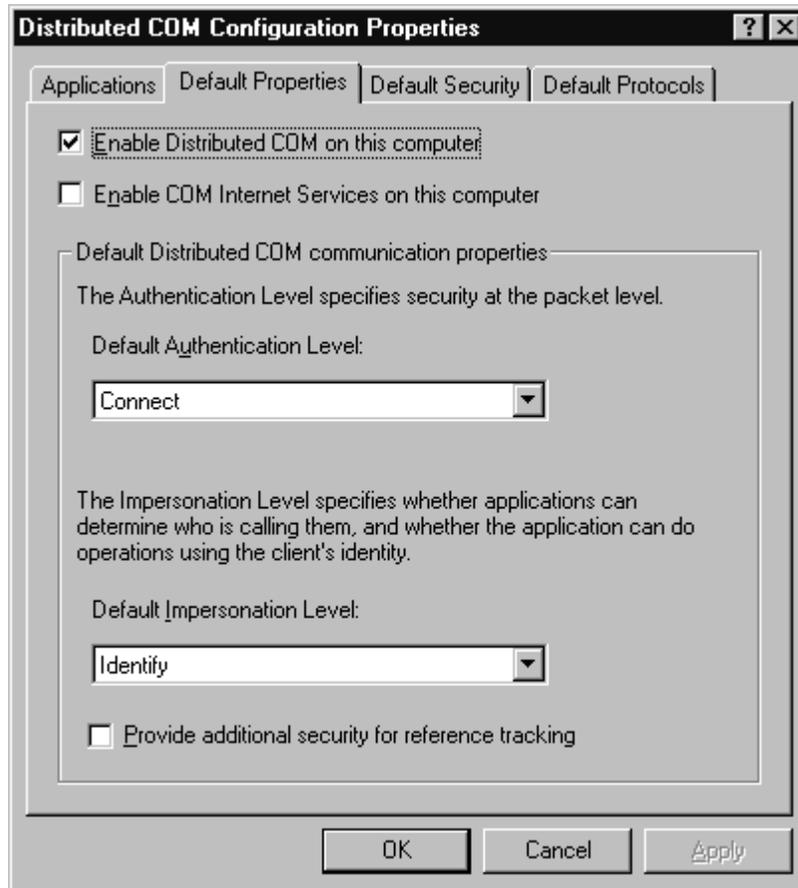


- 2 Type **dcomcnfg**, and select **OK**.



- 3 From the Distributed COM Configuration Properties window, select the **Default Properties** tab. Select **Enable Distributed COM on this computer**.

Note: The dcomcnfg dialog boxes might look slightly different based on which version of Windows you are running and which service pack you have applied. △



The authentication level is negotiated by NT, using values that are provided by both the client and the server. The authentication level **Connect** and the impersonation level **Identify** are the default values and provide a good balance between security and system performance. These values are global; that is, they apply to all COM-enabled applications. More restrictive security levels might be required based on your users' and site's needs. Some Enterprise Client applications might also set authentication and impersonation values that override default settings. For a description of additional levels, consult Windows NT Help.

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

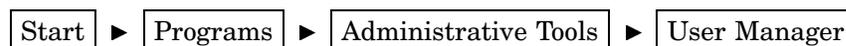
- 4 Repeat steps 1 through 3 on the client machine.

Step 2 — Creating Client Groups

After you enable DCOM on the SAS object server, you must identify what user IDs and groups can launch SAS. One approach is to create a Windows user group that contains the IDs of all users who need to access the server. With the user group in place, adding a new user requires only adding the user's ID to the group, rather than explicitly specifying the required permissions.

To create a Windows user group:

- 1 Select



The User Manager window opens.

- 2 Select

►

The New Local Group window opens.

- 3 Select **Add** to open the Add Users and Groups window. Select the ID for each user in the group and select **Add**.

When you finish, select **OK**.

Step 3 — Setting SAS Launch Policies

After you create the user group on the SAS object server, you must identify the group as having permission to launch SAS, and you must specify any other user IDs (such as those for administrators) that also have permission. Windows users or groups must have launch access in order to request services from SAS on the server. There are two ways to identify users and groups that have launch access: one is globally, and the other is to associate them with the SAS application. Granting users and groups global access to launch applications means that they can launch any DCOM-enabled application, unless they are otherwise restricted by the application's specific permissions. Associating users and groups with an application allows specific users and groups to launch that application.

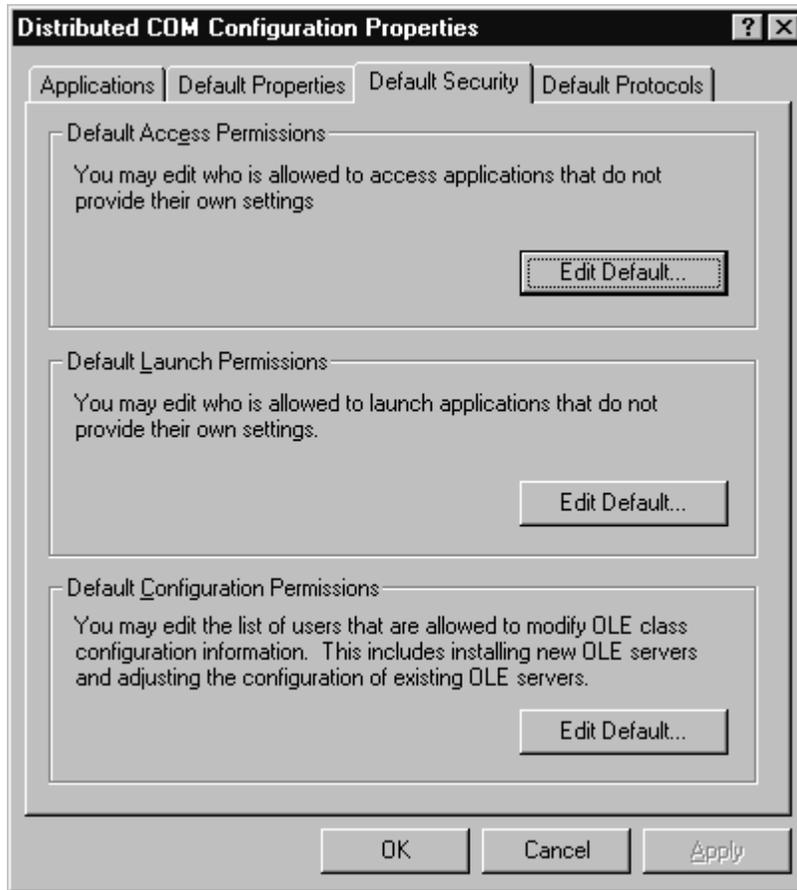
Setting Global Launch Policies

Global launch policies affect all COM-enabled applications. To set global launch policies for selected users and groups:

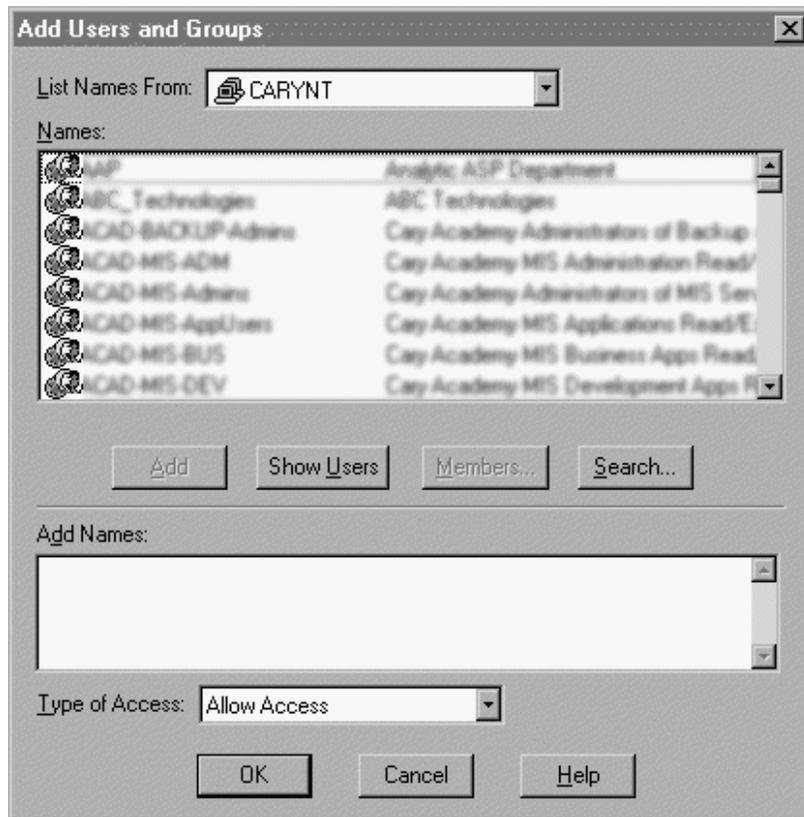
- 1 If the Distributed COM Configuration Properties window is not open, then open the window by selecting

►

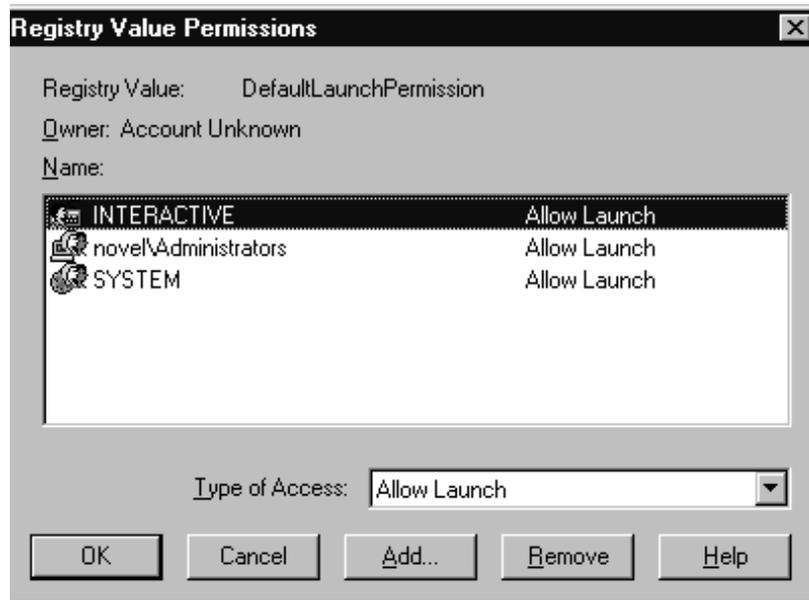
Then type **dcomcnfg** in the Run window and select **OK**. In the Distributed COM Configuration Properties window, select the **Default Security** tab.



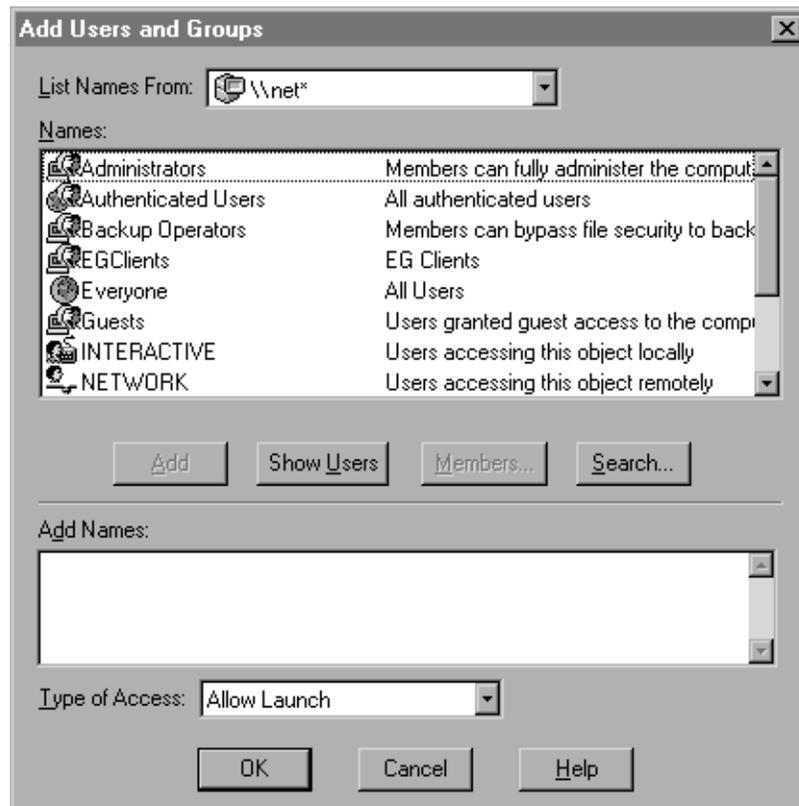
- 2 You are required to edit both default access permissions and default launch permissions. (Default configuration permissions do not require any modifications from the default values for SAS applications.) In the **Default Access Permissions** frame, click **Edit Default**, and then click **Add** in the Registry Value Permissions window. The Add Users and Groups window opens.



- 3 Select the users and groups at your site that should have default access to DCOM applications. You can also identify users and groups that are denied access permissions to an application by changing the selection in the **Type of Access** field to **Deny Access**. When you finish, select **OK** to close the Add Users and Groups window, and then select **OK** to close the Registry Value Permissions window.
- 4 In the **Default Launch Permissions** frame in the Distributed COM Configuration Properties window, click **Edit Default**. The Registry Value Permissions window opens.
- 5 Click **Add** to open the Add User and Groups window.



- 6 Select the users and groups at your site that should have default launch permissions for DCOM applications. You can also identify users and groups that are denied launch permissions by changing the selection in the **Type of Access** field to **Deny Access**. When you finish, select **OK** to close the Add Users and Groups window, and then select **OK** to close the Registry Value Permissions window.



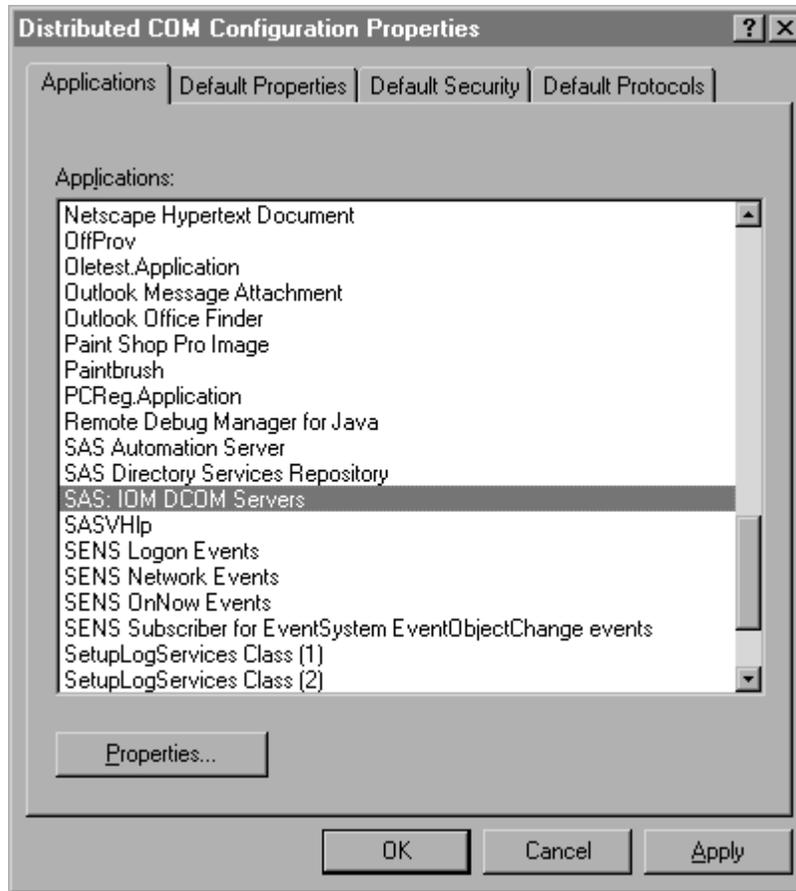
- 7 To save your selections and exit, click **OK** in each of the open Distributed COM windows.

To identify more restrictive configuration permissions based on site and user needs, consult Windows NT Help.

Setting Launch Policies for SAS

To set launch policies for SAS:

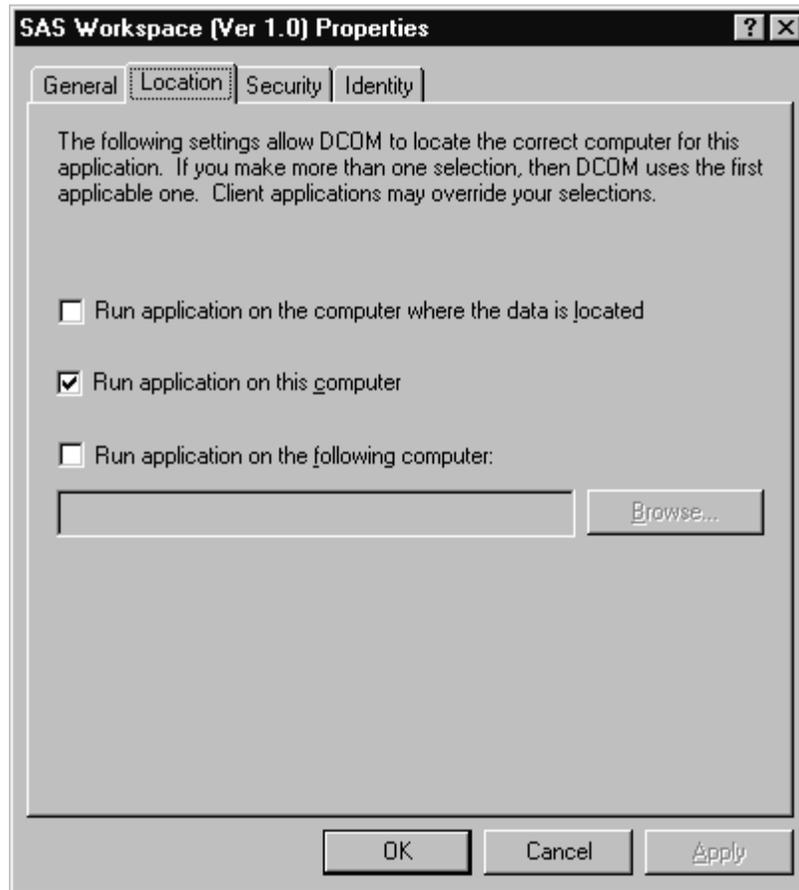
- 1 From the Distributed COM Configuration Properties window, select the **Applications** tab.



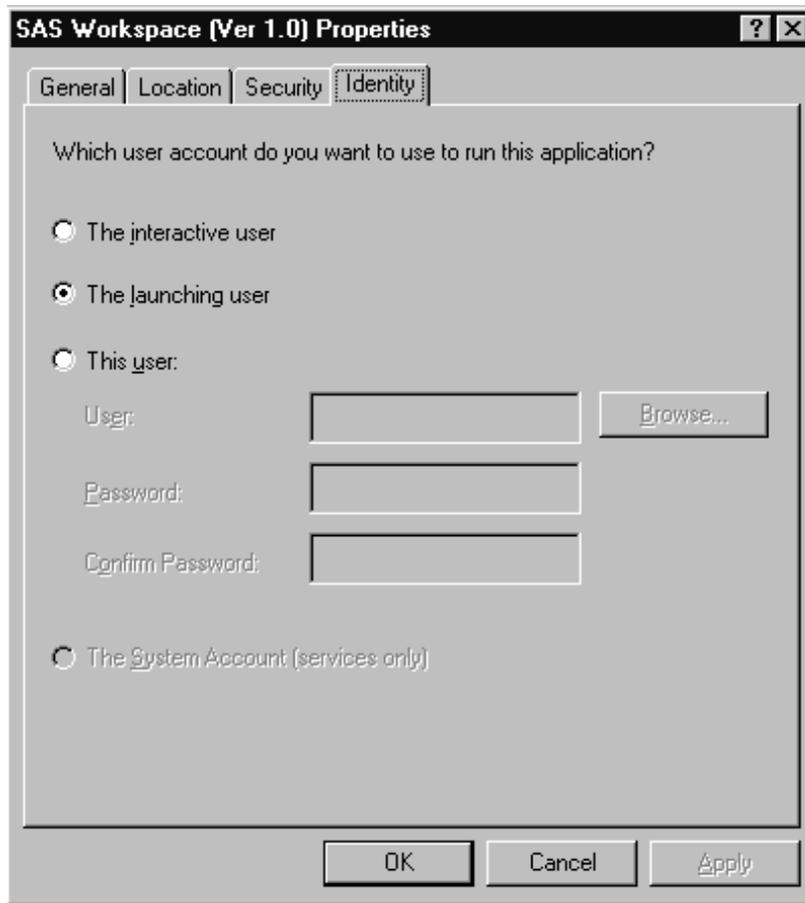
- 2 Select SAS: IOM DCOM Servers in the **Applications** list and click **Properties**. Note that the name in the list varies depending on the version of SAS that you have installed, as listed below:

SAS Version	Choice
8.0	SAS Workspace (Ver. 1.0)
8.1	SAS: Integrated Object Model (IOM) Server 1.0
8.2	SAS: IOM DCOM Servers

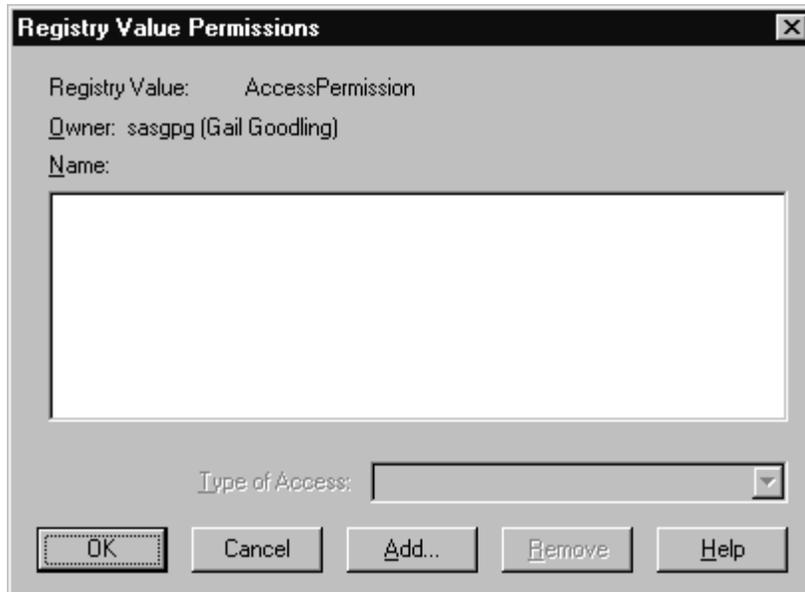
- 3 Select the **Location** tab to check the default setting. By default, the only option that is enabled is **Run application on this computer**. No other options are required for SAS applications.



- 4 Select the **Identity** tab to check the default setting. Select the option **The launching user** for maximum security.



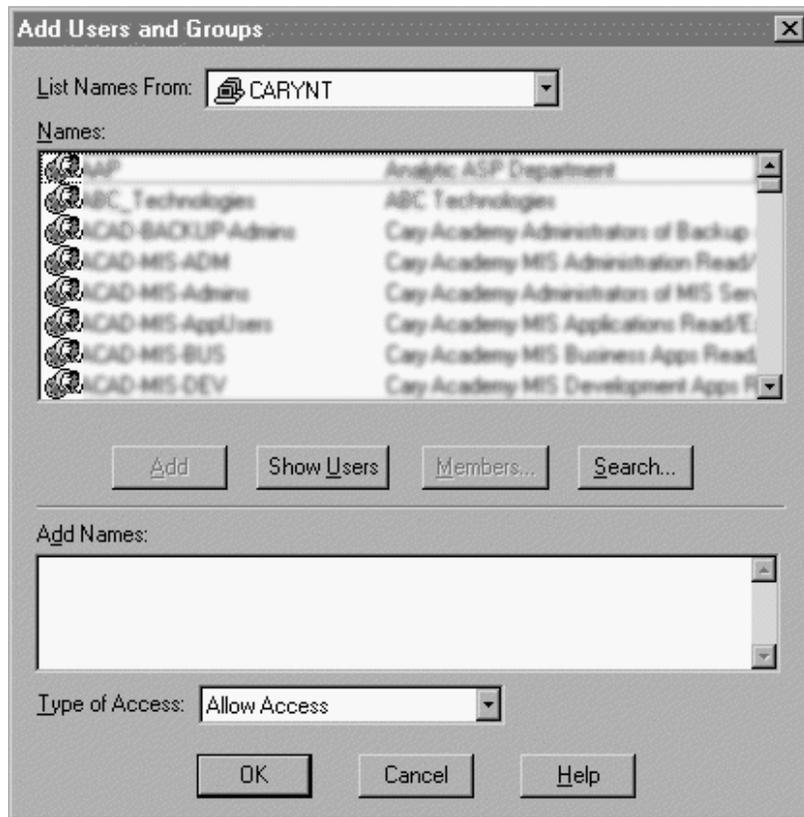
- 5 Select the **Security** tab. If you are using the default access permissions that you specified in "Setting Global Launch Policies" on page 50, then select **Use default access permissions** and continue with the next step. If you want to grant access to users who are not in the list of users with default access permissions, then select **Use custom access permissions** and click **Edit** to open the Registry Value Permissions window.



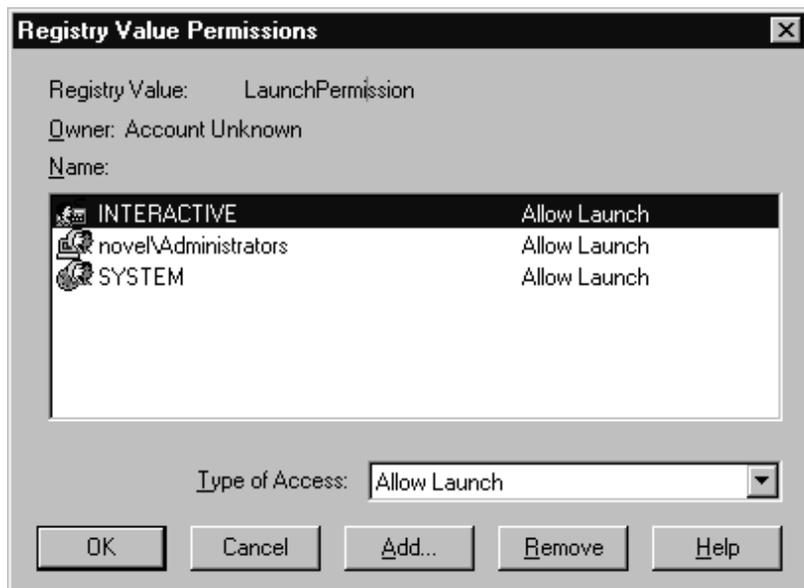
Select **Add** in the Registry Value Permissions window to open the Add Users and Groups window.

Use the Add Users and Groups window to grant users and groups (who are not in the list of users with default access permissions) access to SAS through DCOM. You can also identify users and groups that are denied access to SAS by changing the selection in **Type of Access**.

When you finish, select **OK** in the Add Users and Groups window and then select **OK** in the Registry Value Permissions window.

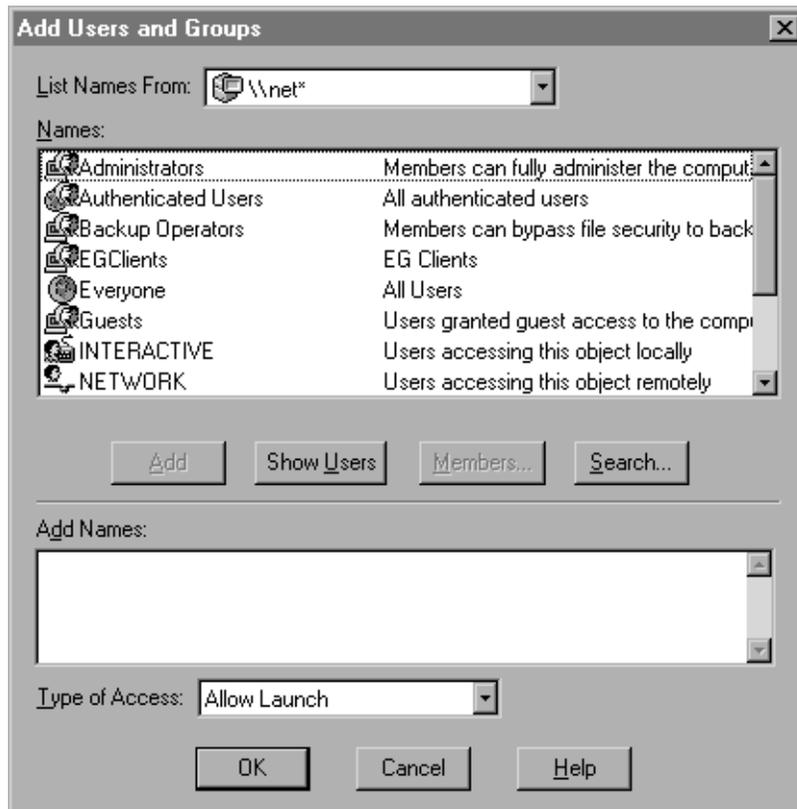


- 6 If you want to use launch permissions other than the default, then select **Use custom launch permissions**. Click **Edit** in the **Use custom launch permissions** frame to edit the registry values.
- 7 Click **Add** in the Registry Value Permissions window.



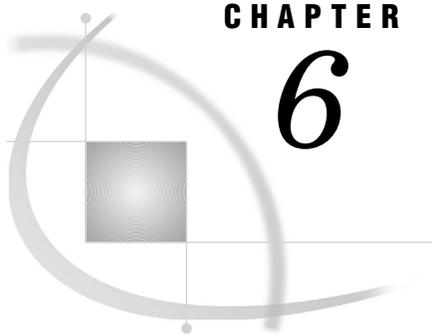
- 8 Use the Add Users and Groups window to identify users and groups at your site and the type of access (allow or deny launch access). It is recommended that you

enter the same values as in **Custom Access Permissions**. (For field descriptions, refer to Windows NT Help.) When you finish, 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 that application through default permissions. Δ

- 9 To save your selections and exit, click **OK** in each of the open Distributed COM windows.



CHAPTER

6

Preparing a Setup Image for Users

Preparing a Setup Image for Users in a Networked Environment 61

Preparing a Setup Image for Users in a Networked Environment

Setting up the Enterprise Clients includes giving users access to a *setup image* of the Enterprise Clients. Users must access the setup image in order to install the Enterprise Clients software on their local machines. In a networked environment, you create a setup image on a server. Note that the setup procedure includes pointing users to the correct metadata repository server process.

In a networked environment you can create *several* repositories for different groups of users. This is the most complex way to use repositories, and it is also the most powerful. If you create several repositories, then you will need to configure a different setup image for each group of users so that each group will access the correct repository at startup. This is appropriate if the groups do not have access to the same Windows servers.

To create a setup image for users in a networked environment:

- 1 If you have not already done so, choose a location for the setup image. The location should be on a server that your users can access. Using Windows Explorer or another file manager, copy the installation files (from the installation CD or from another location) to the chosen location.
- 2 Create a customized set of defaults for Enterprise Guide installations, which are automatically applied when users install Enterprise Guide. The defaults are read from the options.ini file when the installation is performed. The default values appear in the Enterprise Guide Options window, which you access by selecting

Tools ► Options

The following table contains the name, values, and description for each of the defaults in the options.ini file. The divisions in the table and in the options.ini file correspond to the tabs in the Options window.

Name	Values	Description
Advanced		
DisplayLogSubmitFails	0=False, 1=True	If true, display error log if errors occur when modifying data in the grid.
OpenExternFilesInEG	0=False, 1=True	If true, open external files in Enterprise Guide. If false, open files in external viewer.

Name	Values	Description
UseLocalProvider	0=False, 1=True	If true, automatically use the SAS OLE DB local provider to access SAS data sets if no local SAS server is available.
AutomaticallySaveTempData	0=False, 1=True	If true, automatically save data nodes that reference temporary data.
EncodeODS	0=False, 1=True	If true, encode the character set in HTML and RTF results.
EnableAccessibilityFeatures	0=False, 1=True	If true, enable accessibility features.
Code		
CodeDisplaySASCode	0=False, 1=True	If true, display detailed SAS code (for code entered in the Code Editor) for items such as the current OPTIONS statements, ODS specifications, and LIBNAME statements.
CodeExportCustomCode	0=False, 1=True	If true, include custom code when code is exported.
CodeGenerateSASCode	0=False, 1=True	If true, automatically generate additional SAS code so that user code looks like task code.
InsertSASCodeAfterCode	0=False, 1=True	If true, insert custom code after code written in the Code Editor.
InsertSASCodeAfterTask	0=False, 1=True	If true, insert custom code after automatically generated task code.
InsertSASCodeBeforeCode	0=False, 1=True	If true, insert custom code before code written in the Code Editor.
InsertSASCodeBeforeTask	0=False, 1=True	If true, insert custom code before automatically generated task code.
TaskExportCustomCode	0=False, 1=True	If true, include custom code together with automatically generated task code when code is exported.
TaskDisplaySASCode	0=False, 1=True	If true, display detailed SAS code (for automatically generated task code) for items such as the current OPTIONS statements, ODS specifications, and LIBNAME statements.
General		
AllowDoubleClickEdit	0=False, 1=True	If true, allow users to rename nodes by clicking twice on the node name.
DisplayDataPath	0=False, 1=True	If true, display the full path for data nodes.
ShowAgent	0=False, 1=True	If true, show the agent.

Name	Values	Description
ShowCode	0=False, 1=True	If true, display an icon for code generated by Enterprise Guide tasks.
ShowLog	0=False, 1=True	If true, display an icon for the log file generated when tasks are run.
Graph		
ActivexURL	URL	Specifies the URL for the ActiveX codebase.
EmbedData	0=False, 1=True	If true, embed data as part of the ActiveX control.
FootnotesInGIF	0=False, 1=True	If true, footnotes are embedded in the GIF file.
GraphFormat	0=ActiveX, 1=Java Applet, 2=GIF	Specifies graphic output format.
JavaURL	URL	Specifies the URL for the generated Java applet.
TextureMapLoc		Specifies the location of texture maps.
TitlesInGIF	0=False, 1=True	If true, titles are embedded in the GIF file.
UseJavaArchive	0=Local, 1=Remote	If local, use local Java JAR files for Java control; if remote, use JAR files found at the location specified in the JavaURL parameter.
Data		
AutoDisplayData	0=False, 1=True	If true, automatically display contents of data table in a data grid when data is added to project.
AutoDisplayQuery	0=False, 1=True	If true, automatically display query results when added to project.
AutoFormatColumns	0=False, 1=True	If true, use cell values to determine whether column is character or numeric.
CacheSASFormat	0=False, 1=True	If true, cache data format and informat information (might improve performance when processing large files).
DataSetSliceExpanded	0=False, 1=True	If true, cache data format and informat information (might improve performance when processing large files).
DataSetSliceExpanded	0=False, 1=True	If true, exclude observations that contain missing values from the data set generated when a task is run on an MDDB slice.

Name	Values	Description
DisplayColumnAlphaOrder	0=False, 1=True	If true, data table columns are displayed in alphabetical order in the task dialog boxes and the Query Builder.
DisplayPerfWarnings	0=False, 1=True	If true, display a performance warning when the Query Builder attempts to copy a table with more than the number of rows specified in PerfWarningRows.
DistinctValueCount	number	If DistinctValuePerfWarnings is true, threshold of distinct values will trigger distinct value warning.
DistinctValuePerfWarnings	0=False, 1=True	If true, display a performance warning if a query produces a table with more than the number of distinct values specified in DistinctValueCount.
FirstRowHasNames	0=False, 1=True	If true, use the first row of data as column headings instead of data (only when importing an Excel file).
QueryCodeFormat	0=Std SQL, 1=SAS (Proc SQL)	Specifies the type of SQL generated from the Query Builder.
IncludeSchemaInfo	0=False, 1=True	If true, when saving data in text format, use the Microsoft Jet database engine (which saves a schema file) instead of writing to a text file.
OpenExistingDataUnprotected	0=False, 1=True	If true, data tables are opened in unprotected mode (data can be changed).
PerfWarningRows	number	If DisplayPerfWarnings is true, threshold of row numbers will trigger a row count warning.
RDBMSRecCount	0=False, 1=True	If true, finds the total record count for RDBMS table (such as Oracle or DB2) when the table is opened.
ScanColumnForWidth	0=False, 1=True	If true, use longest value as column width (only when an Excel file is saved from Enterprise Guide).
ShowColAsLabels	0=False, 1=True	If true, use the values in the first row as column names when importing Excel and text data.
AliasLength	length	Do not generate an alias length greater than the specified value.
Results		

Name	Values	Description
AutoDisplayHTML	0=False, 1=True	If true, and if GenerateHTML is true, automatically display results in HTML.
AutoDisplayPDF	0=False, 1=True	If true, and if GeneratePDF is true, automatically display results in PDF.
AutoDisplayRTF	0=False, 1=True	If true, and if GenerateRTF is true, automatically display results in RTF.
DisplayLog	0=False, 1=True	If true, open a SAS log window if errors occur while code is running.
DisplayOutputDatasets	0=False, 1=True	If true, display the last data table created when task or code is run.
GenerateHTML	0=False, 1=True	If true, generate results in HTML format.
GeneratePDF	0=False, 1=True	If true, generate results in PDF format.
GenerateRTF	0=False, 1=True	If true, generate results in RTF format.
HTMLBrowser	HTML browser path	Specifies full path to preferred HTML browser.
ODSStyleSheet	ODS style sheet	Specifies the ODS style sheet.
OverwriteResults	0=False, 1=True	If true, replaces existing results with new results for the same task.
PDFBrowser	PDF browser path	Specifies full path to preferred PDF browser.
ReplaceResults	0=prompt, 1=no prompt	If OverwriteResults is true, specifies whether a prompt is displayed before results are overwritten.
ResultsSizeThreshold	File size	Prompt before opening results larger than the specified file size.
RTFBrowser	RTF browser path	Specifies full path to preferred RTF browser.
RTFUseWebBrowser	0=False, 1=True	If true, display RTF results using the Web browser.
ShowOutput	0=False, 1=True	If true, generate results in text format.
StreamHTML	0=False, 1=True	If true, generate results in streaming HTML format.
ViewHTMLResults	0=Enterprise Guide, 1=Internet Explorer, 2=Preferred	Specifies the viewer to use for HTML results (preferred viewer path set by HTMLBrowser).

Name	Values	Description
ViewPDFResults	0=Enterprise Guide, 1=Acrobat, 2=Preferred	Specifies the viewer to use for PDF results (preferred viewer path set by PDFBrowser).
ViewRTFResults	0=Enterprise Guide, 1=Microsoft Word, 2=Preferred	Specifies the viewer to use for RTF results (preferred viewer path set by RTFBrowser).

- 3 If needed, modify the following files to customize the text that appears in code that Enterprise Guide generates:

PreCode.sas	Text that appears at the beginning of generated code
PostCode.sas	Text that appears at the end of generated code
PreTask.sas	Text that appears at the beginning of generated task code
PostTask.sas	Text that appears at the end of generated task code

- 4 Edit the SITENUM.TXT file. This enables you to suppress the prompt that asks users for site information. To suppress the prompt, edit SITENUM.TXT to include a site number or name on the first line. The site information that is contained in this file is stored in the registry. If you delete SITENUM.TXT, or if the first line is blank, then the dialog box that prompts for the site number is displayed during setup.

The current version of EGAUTO.SAS

- defines a temporary location in order to store GIF files that are created during a session by graphics tasks and code. This logic is different for OS/390 servers than for other servers.
- sets the FORMCHAR option for character-based plots.
- submits a PROC TEMPLATE step that defines a custom ODS style that Enterprise Guide uses by default.
- contains comments throughout the file in order to help you understand what is included.

An administrator can add other SAS statements to the file in order to set particular options if desired. Note that this file is a run-time SAS file (unlike a .CFG file), so you cannot specify invocation-only options here. Note that if EGAUTO.SAS is removed, then a new file is created that contains the original values that were included on the CD.

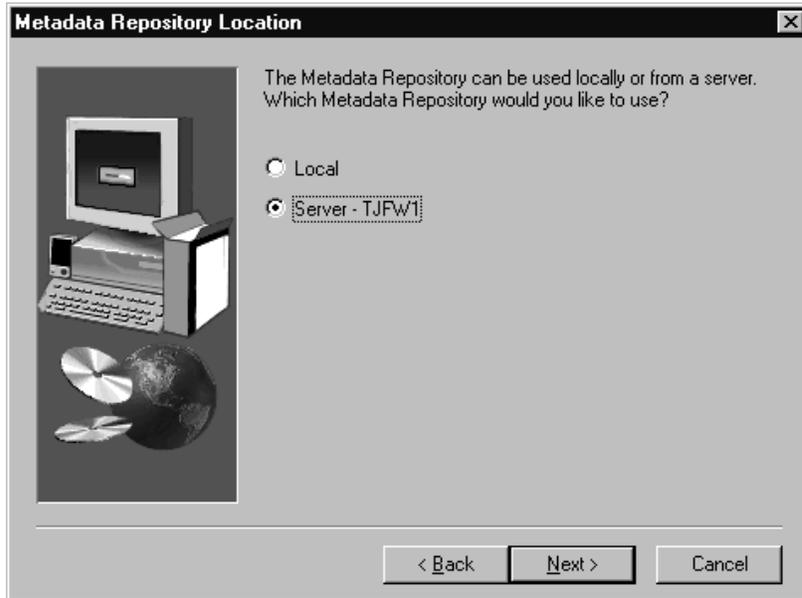
- 5 Edit the EGAUTO.SAS file, which is a SAS program that is submitted to a server the first time that an Enterprise Guide connects to it during a SAS session. It is used in much the same way as an AUTOEXEC.SAS file, except that it is sent from the client.
- 6 Make sure that the repository information for this set of users is in a location where users can access it. (It does not need to be in the same location as the installation files.)
- 7 Edit the SDS.TXT file that is included in the setup image so that when a user launches an Enterprise Client application, it points to the correct repository server. Using a text editor, type the name of the repository server on the first line of SDS.TXT.

You can force users to skip the dialog box that enables them to create a copy of the database locally (see the dialog box in step 4). To do this, type the keyword **force** on the second line of the SDS.TXT file. **Force** is not case sensitive.

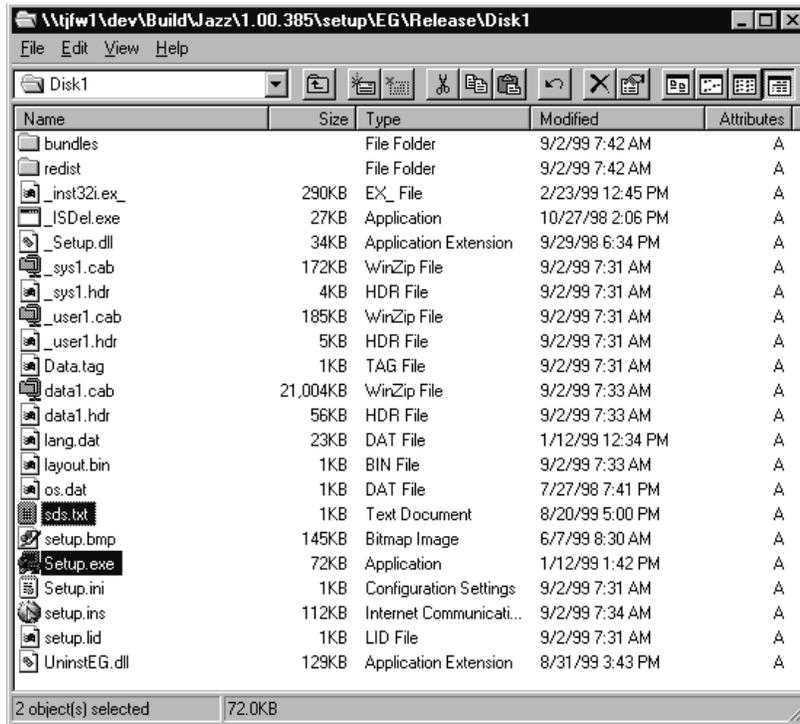
For example, to force the users who are running the setup program to use the repository server on a server named SERVER2, edit the SDS.TXT file so that it reads

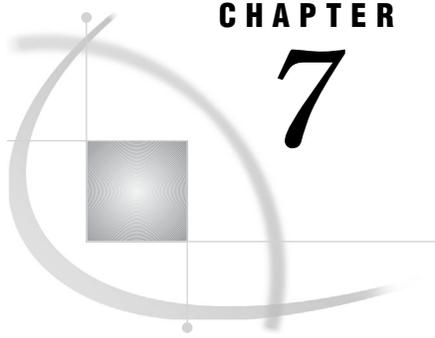
```
SERVER2  
force
```

- 8 If you enable users to select from the Metadata Repository Location dialog box, then tell them which repository to choose during setup—either **Local** or **Server** (the edited SDS.TXT file provides the server name).



- 9 Copy the Enterprise Clients application setup directory to a network drive that users can access in order to run SETUP.EXE, which is the program that installs a copy of the setup image on their local machines. Add the edited SDS.TXT file to the directory where SETUP.EXE resides. You can provide a link from a Web site to the directory for easy access.





CHAPTER

7

Tasks That You Perform in Enterprise Guide Administrator

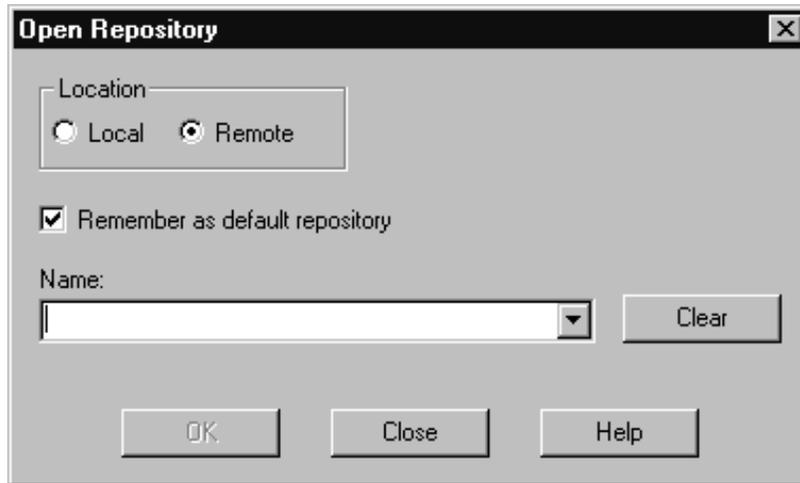
<i>Changing Repositories</i>	69
<i>Viewing the Objects</i>	70
<i>Managing Servers</i>	72
<i>About SAS Servers</i>	72
<i>Adding SAS Servers</i>	73
<i>Updating Server Properties</i>	81
<i>Managing Libraries</i>	81
<i>About Libraries</i>	81
<i>Adding a Library</i>	82
<i>Creating the EGTASK Library</i>	88
<i>Updating Library Properties</i>	88
<i>Unassigning Libraries</i>	89
<i>Managing Binders</i>	90
<i>About Binders</i>	90
<i>Binders That Map to Server Locations</i>	90
<i>Binders That Map to Native Files</i>	90
<i>Adding a Binder</i>	90
<i>Adding Objects to a Binder</i>	94
<i>Deleting a Binder</i>	94
<i>Managing Groups</i>	95
<i>About Groups</i>	95
<i>Adding a Group</i>	95
<i>Deleting a Group</i>	98
<i>Updating Group Properties</i>	98
<i>Managing Users</i>	99
<i>About Users</i>	99
<i>Adding a User</i>	100
<i>Deleting a User</i>	103
<i>Updating User Properties</i>	103
<i>Adding a Format</i>	104
<i>Importing Server Definitions</i>	108

Changing Repositories

You can change the metadata repository that you are connected to by selecting

File ► **Open Repository**

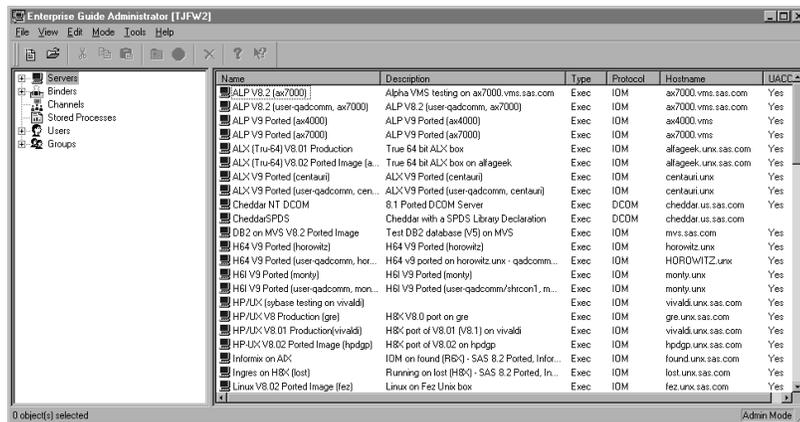
The Open Repository window opens.



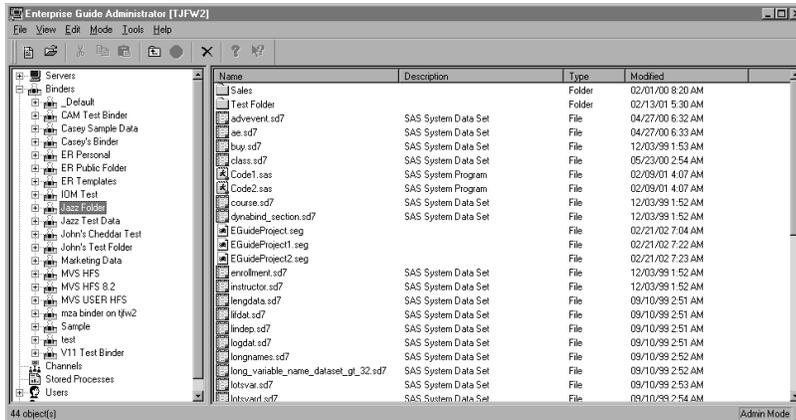
Enter the repository name, or use the pull-down menu to select from a list of recently accessed repositories. The **Clear** button erases the list of recently accessed repositories. This prevents users from trying to access a repository that has been deactivated.

Viewing the Objects

You view objects in Enterprise Guide Administrator in much the same way as you view objects in the Windows Explorer. Click once on a category of objects in the left side of the window (Servers, Binders, Users, and so on) to display its contents on the right side.

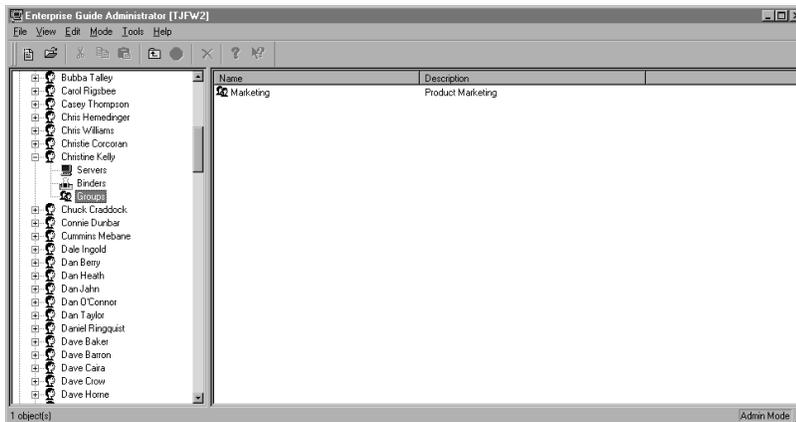


Double-click a category name to expand that category. In the following example, the user double-clicked Binders and then clicked on the Jazz Folder binder to reveal a list of SAS data sets.



If you are accessing objects that reside on SAS servers, then you might be prompted to enter a user ID and password. In Enterprise Guide Administrator, when you create a server object, you can configure it to require users to enter a user ID and password.

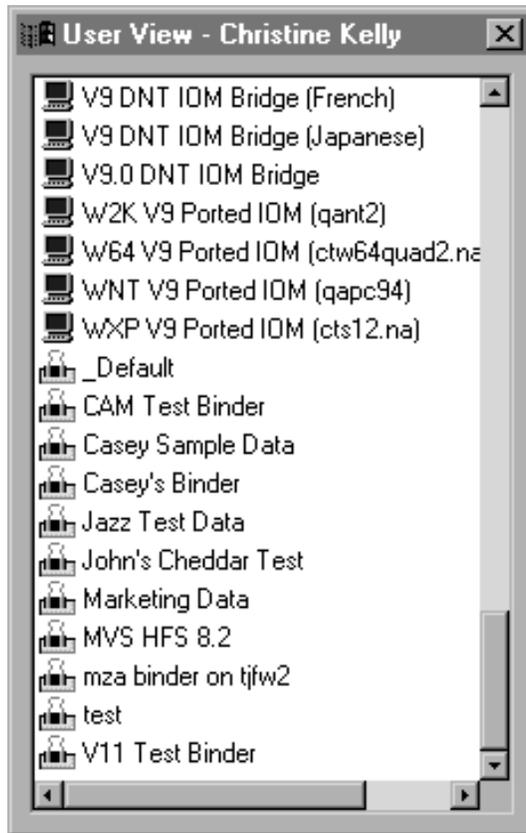
The objects in the viewing window are nested to show the relationships between the objects. The following example explores a user named Christine Kelly. Expanding the Groups category below her name shows that she belongs to the Marketing group.



You can customize your session further by selecting **View** from the pull-down menu. From the menu you can activate options for displaying different views of the icons, sizes of the icons, and information about the objects.

If you have selected a user in the viewer, then you can choose **User** from the View menu. The User View window shows the user's servers, binder, library, group, and privileges in a separate floating window. Default values for this user are shown in bold and appear red on your screen.

The information in the User View window is updated dynamically as you make changes to that user's default settings.



Managing Servers

About SAS Servers

SAS servers are networked computers on which SAS is installed. SAS servers enable multiple clients to access and use SAS data libraries (and members of the libraries) concurrently. A SAS server for Enterprise Clients can be running in a Windows, OS/390, or UNIX operating environment. Enterprise Guide Administrator defines which servers to use with the client platform.

Clients can be Enterprise Client sessions on networked Windows workstations that request services from SAS servers. The services might include SAS processing, data access, and data manipulation. The client is used to submit SAS processing either locally or to a remote SAS server.

You add SAS servers in Enterprise Guide Administrator so that networked Enterprise Client users can execute SAS and access their data. If you add a SAS server that uses the TCP/IP protocol, then you will need the following information:

- server type
- IP address/host name
- protocol
- port address
- user IDs and passwords (optional).

Adding SAS Servers

Note: The following information includes steps for adding existing SAS servers in Enterprise Guide Administrator. △

To add a SAS server:

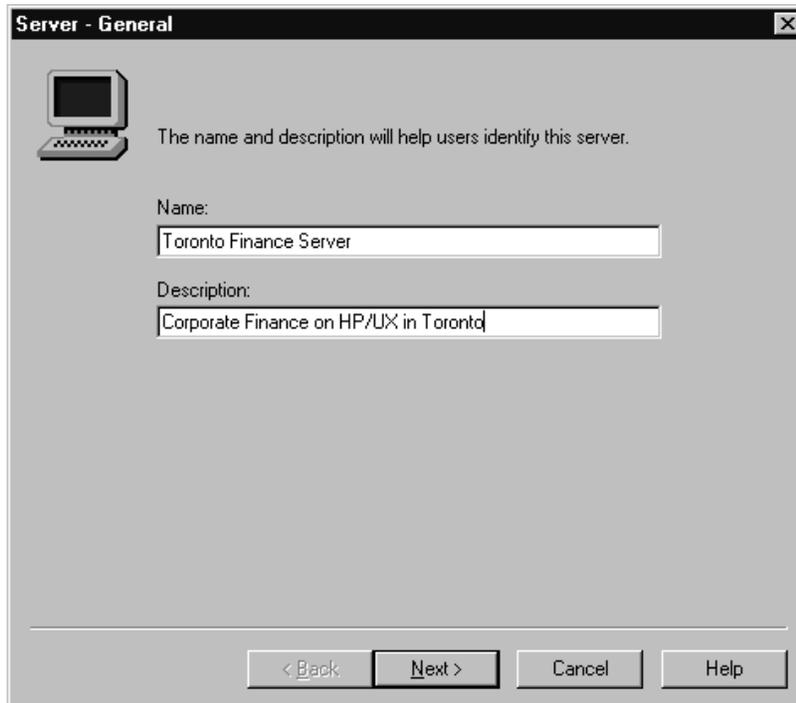
- 1 Open the New window. Either select



from the menu or select the **New** icon  from the toolbar.



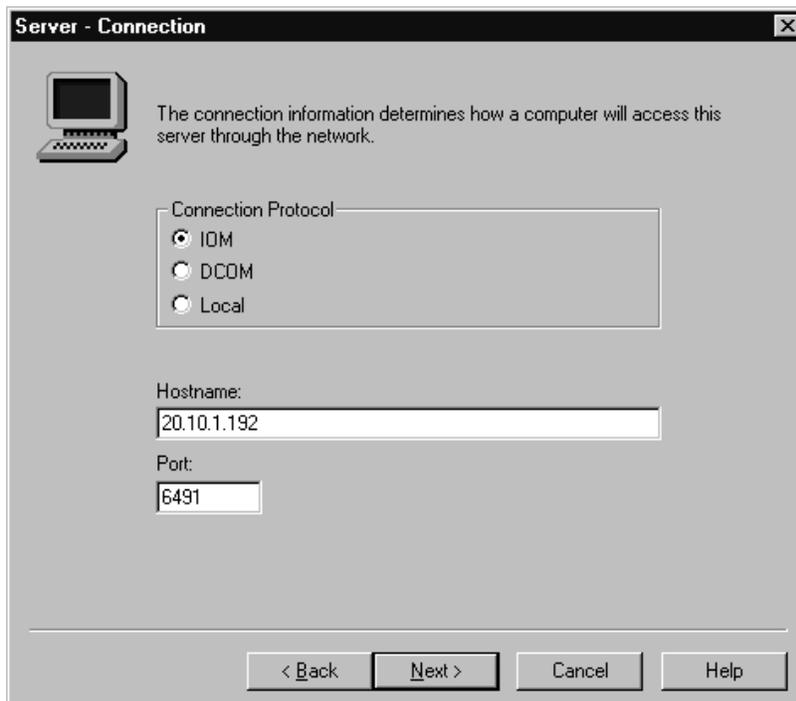
- 2 Select **server** and click **OK**. The Server - General window opens.



- 3 Type the name of the server plus any description. The name cannot contain any of these characters:

\ / : * ? " < > |

The description that you enter will appear in the viewer window. Click **Next** when you are ready to continue. The Server - Connection window opens.

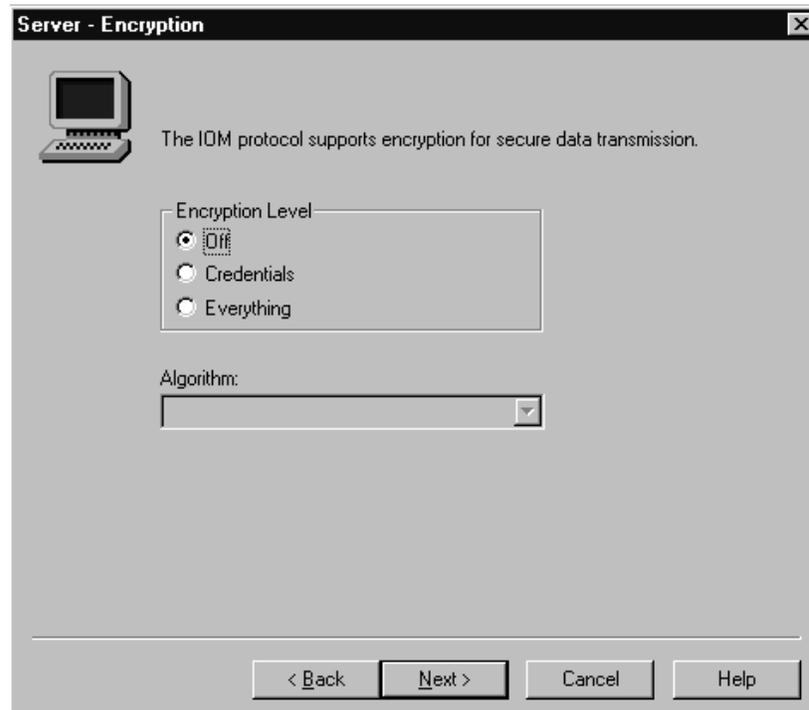


- 4 In the Server - Connection window, enter a protocol and, if required, a valid host name and port address (see the following table).

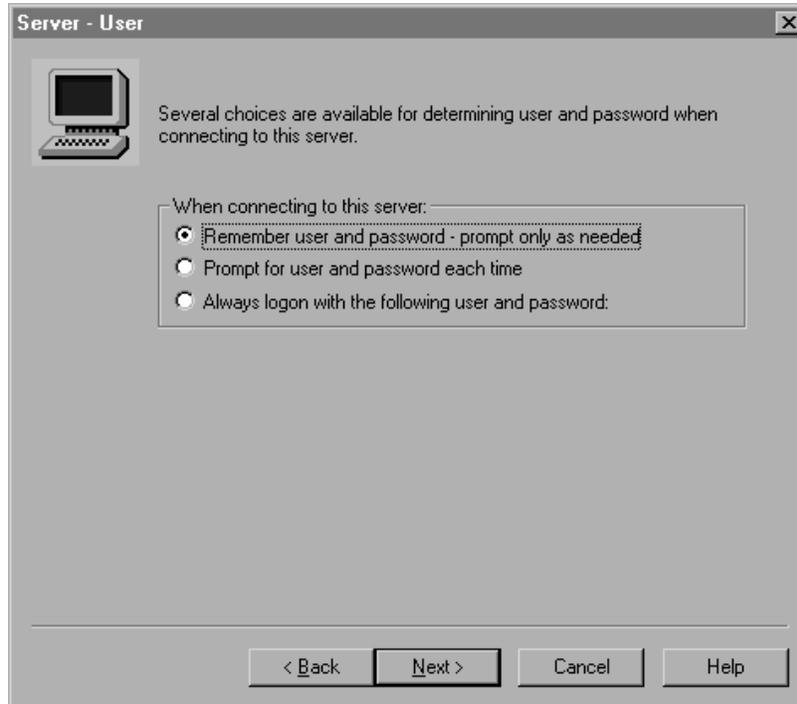
If the server protocol is...	Then enter host name...	Add port address?
IOM	DNS Name (example: NEVADA.PC.ABC.COM), IP Address (example: 12.34.5.678), or a template using the wildcard <userid> (Example: \\nevada\users\ <userid>)*.	Yes. Example: 4362
DCOM	Computer Name** (example: NEVADA)	N/A
Local	Defaults to your machine name.	N/A

- * Use the wildcard **<userid>** to indicate that the real host name or address is found in a file that is named by substituting the user's Windows user ID into the template. For example, \\samba**<userid>**\.sasserver becomes \\samba\ebeacon\.sasserver for the user who is logged on as ebeacon. This is useful when file systems on UNIX servers are shared with Windows files systems by using third-party software such as Samba.
- ** To find the computer name, go to the computer on which you are running dcomcnfg. From the Windows NT Explorer, select **Network Neighborhood** so that it remains highlighted. With the mouse pointer on **Network Neighborhood**, click the right mouse button. Select **Properties** from the menu. Computer Name is shown in the first dialog box (the **Identification** tab).

After you enter the host information, click **Next**. If you selected IOM as the server protocol, then the Server - Encryption window opens. If you selected DCOM or Local as the protocol, then the Server - Files window opens (go to step 8).



- 5 Specify whether you want to use encryption for communications to the server. You can select **Off** (no encryption), **Credentials** (only login information is encrypted), or **Everything** (all communications are encrypted). If you specify **Credentials** or **Everything**, then you must also specify an encryption method in the **Algorithm** field. Click **Next** to open the Server - User window.



- 6 In the Server - User window, specify how the user ID and password will be supplied when logging on to an IOM server.

Remember user and password

stores the user ID and password (encrypted) after a successful connection is established. The password prompt is displayed again only if the password is invalid. The user ID and password are stored in the current user section of the local machine registry, not in the repository database. To reset the user ID and password for a SAS server, select the server in the Enterprise Guide Administrator window, and then access the shortcut menu (click the right mouse button). From the menu, select **Reset Login**. After you reset the login, the user will be prompted to enter the user ID and password at login.

Prompt for user and password each time

always prompts for a user ID and password. This user ID and password (encrypted) are stored in the repository database.

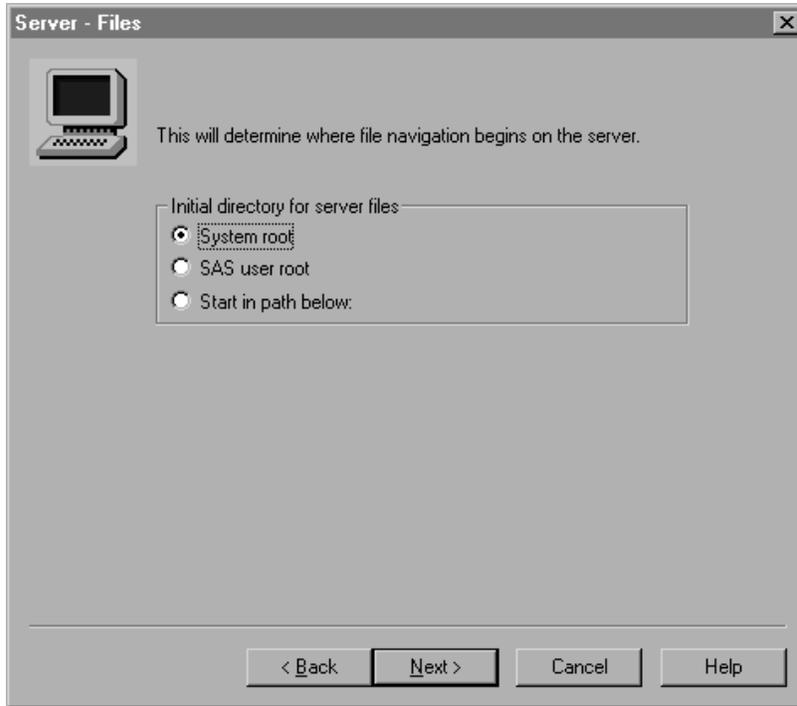
Always logon with the following user and password

forces the user to log on to this server with a specific user ID and password.

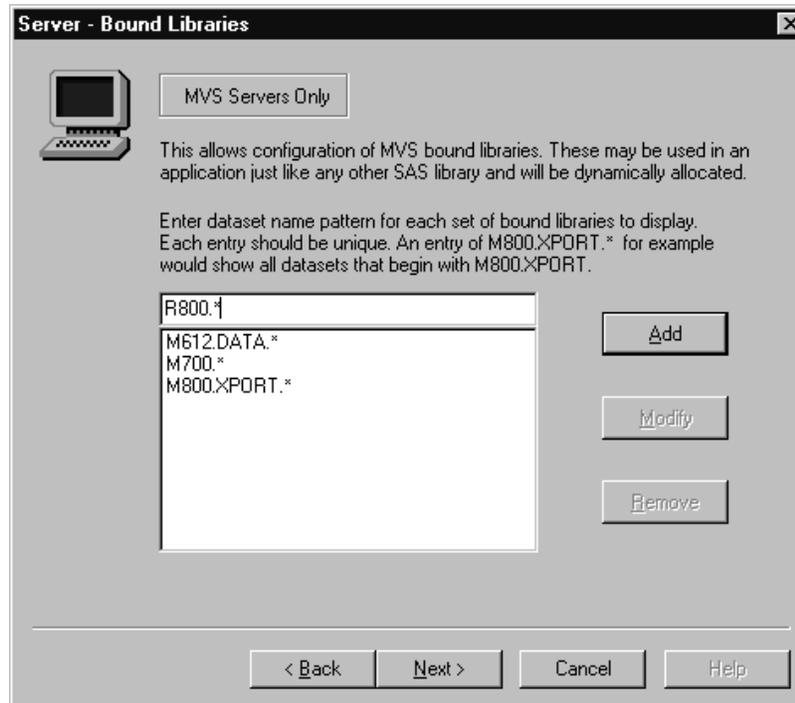
Note: If you are using a DCOM server, then you do not need to supply a user ID or password. The reason is that user authentication for DCOM servers is based on either the identity of the Windows user who is launching the DCOM server or the user or system authentication that is specified in dcomcfg. running on the remote machine. For more information about dcomcnfg, see Appendix 1, “Object Spawner Details,” on page 125. △

The launching user specified in `dcomcnfg`, however, does need the appropriate privileges on the DCOM server in order to create SAS data sets and other files on the server. Typically, the required SAS directory (for example, `SASWORK`) grants read/write access to a Windows group to which the requesting user belongs.

- 7 Click **Next** to open the Server - Files window.



- 8 Specify where users will begin navigation when logged on to the server. You can start navigation from the system root, the SAS user root, or a specified path. Click **Next** to open the Server - Bound Libraries window.



- 9 If you are defining a connection to an MVS server, then specify the search patterns that will help users locate MVS libraries. Enterprise Guide Administrator creates a filing cabinet for each pattern that you enter and creates folders for each data set that is a part of that pattern. If your site has a large number of MVS libraries, then users can use the bound library filing cabinets to quickly navigate to the specific data set that they are interested in. Because you are creating a navigational aid for users, you should choose patterns that will help users navigate to data sets quickly.

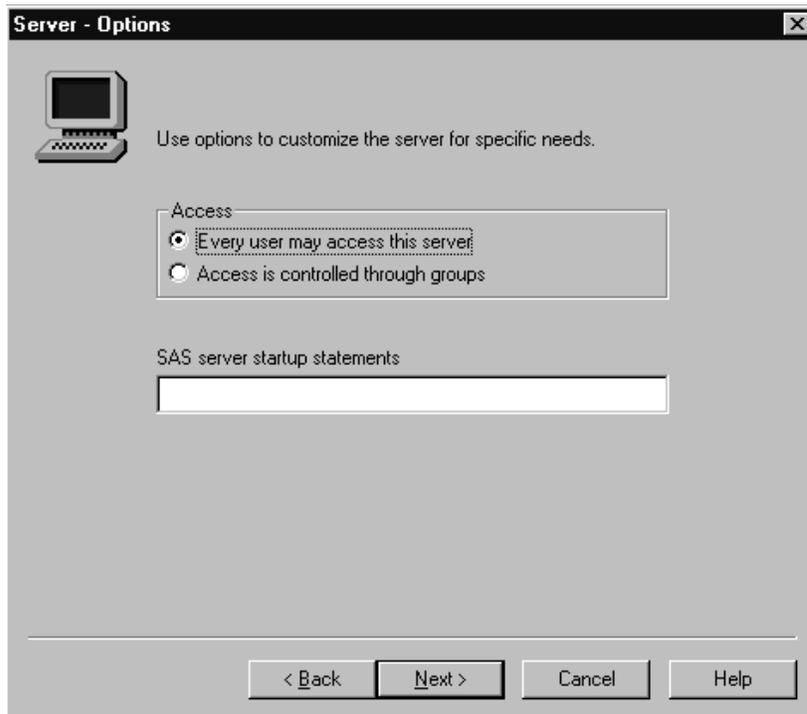
For example, assume that you need access to the following data sets on an MVS server:

- SRV.MFG.PROD1 ... SRV.MFG.PROD20
- SRV.DEV.INV1 ... SRV.DEV.INV5
- SRV.TEST.DEV1

Specifying a pattern of SRV.*.* would result in a bound library that was too broad and contained too many sublevels to allow for rapid navigation. A more efficient solution would be to specify three bound library patterns:

- SRV.MFG.*
- SRV.DEV.*
- SRV.TEST.*

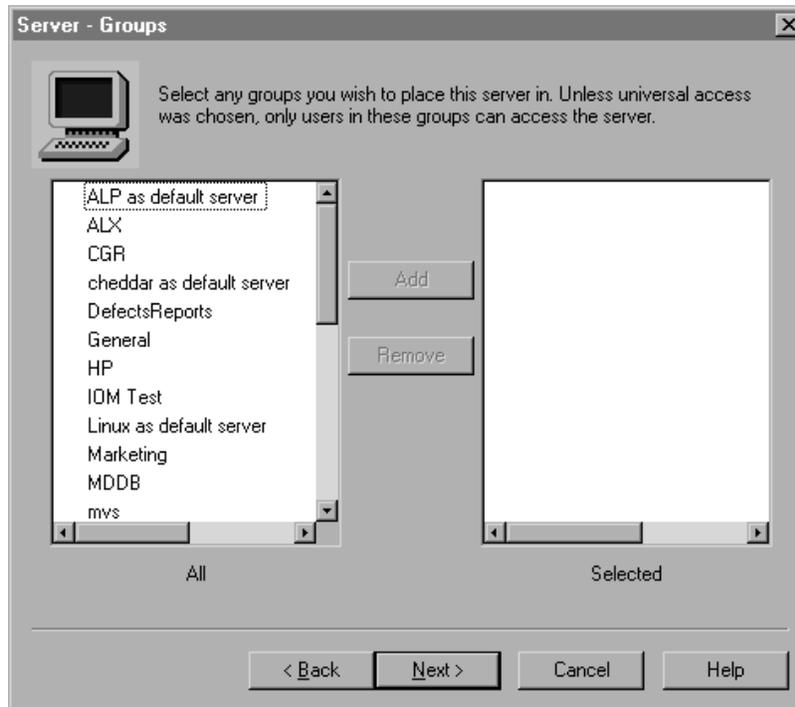
Bound libraries must be specified as a fully qualified name, with the last member specified as a wildcard character (*). When you have specified all bound libraries, click **Next** to open the Server - Options window.



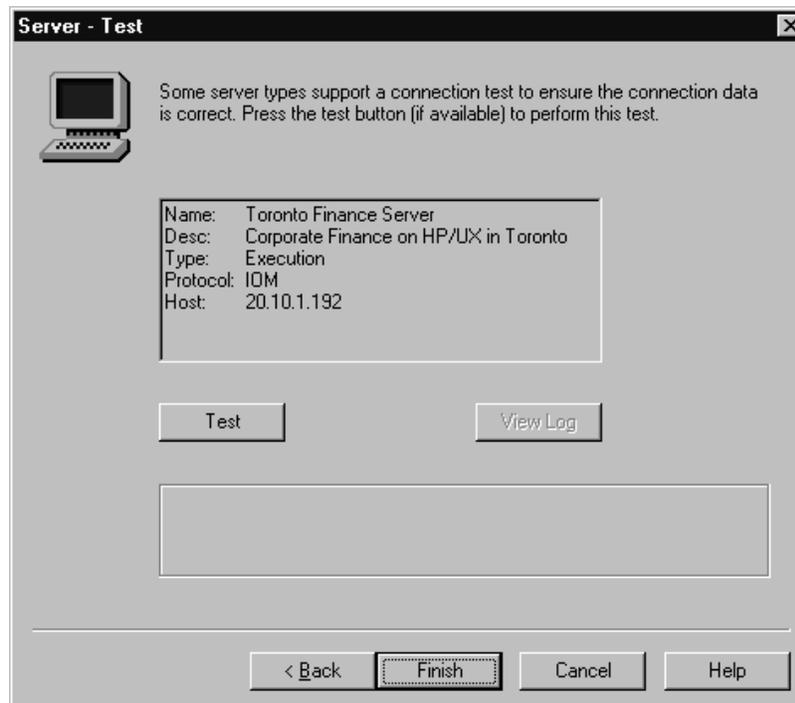
- 10** In the Server - Options window, specify how users can access this server. Select **Every user may access this server** if you want everyone to have access, regardless of group membership. Select **Access is controlled through groups** if you want to specify which groups will be allowed to access this server. If you choose this second option, then you must modify the properties of the groups that will be allowed to access this server.

Under **Additional options**, enter the SAS system option(s) that you want submitted each time SAS is invoked on this server. For more information about SAS system options, see *SAS Language Dictionary: Reference* or the SAS documentation for the server's operating environment.

- 11** Click **Next** to continue.
- 12** If you specified that access to the server should be controlled through groups, then the Server - Groups window opens.



Select the desired group from the **All** box, and click **Add**. If you have not yet added groups, then you can select the General group now and update the server's properties later. Select **Next** to open the Server - Test window.



13 You can test the server connection by clicking **Test**. The result of the test is shown in the **Result** area of the window. Select **View Log** to display the contents of the SAS log.

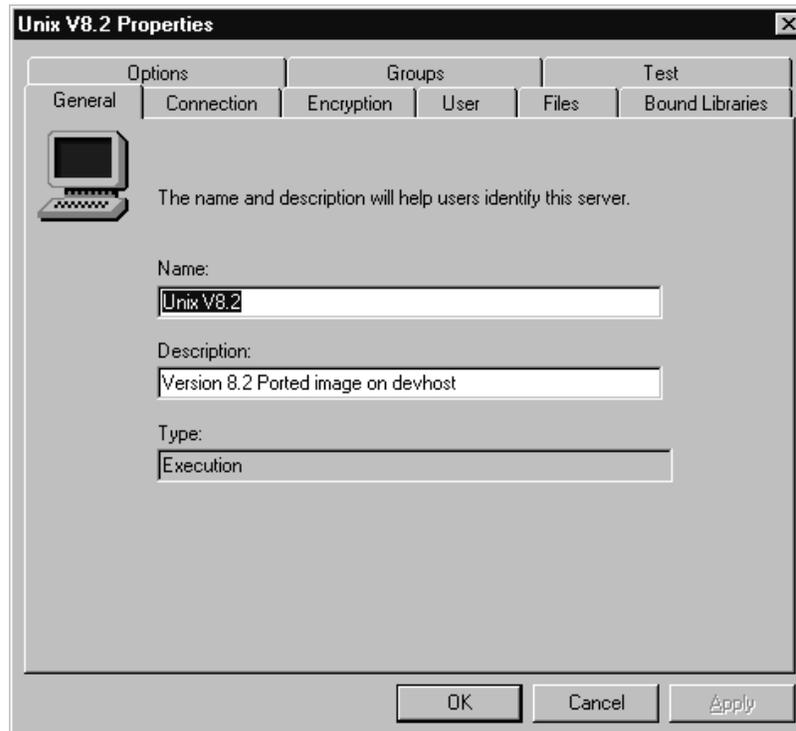
14 Click **Finish** to add the SAS server.

The new server should now appear in the list of servers in the viewer.

Updating Server Properties

Server properties include information such as names, protocols, and preferences. To update this information:

- 1 Select a server in the Enterprise Guide Administrator viewer so that it is highlighted.
- 2 With the mouse pointer on the selected server, click the right mouse button to open the shortcut menu.
- 3 Select **Properties**.



- 4 Select the tab that corresponds to the information that you want to update. See “Adding SAS Servers” on page 73 for information about each field.

Managing Libraries

About Libraries

Libraries are collections of files that are stored on a server and accessed from SAS Enterprise Client applications. Libraries give Enterprise Clients convenient access to their data on servers. When you add a library through Enterprise Guide Administrator, you make it available to all the networked Enterprise Client applications that have access to a specific metadata repository.

Note: Before you add a library, you must first add the server on which the library resides. △

To define a library:

- 1 Define the library's path on the server platform.
- 2 Specify the SAS engine that is identified with the library.
- 3 Enter any associated options that you can specify in a LIBNAME statement. LIBNAME options work like the commands in an AUTOEXEC.BAT file. The LIBNAME options are stored on the client and then executed when an Enterprise Client application connects to the specified SAS server. See *SAS Language Reference: Dictionary* for more information about SAS libraries, LIBNAME statements, and LIBNAME options.

Adding a Library

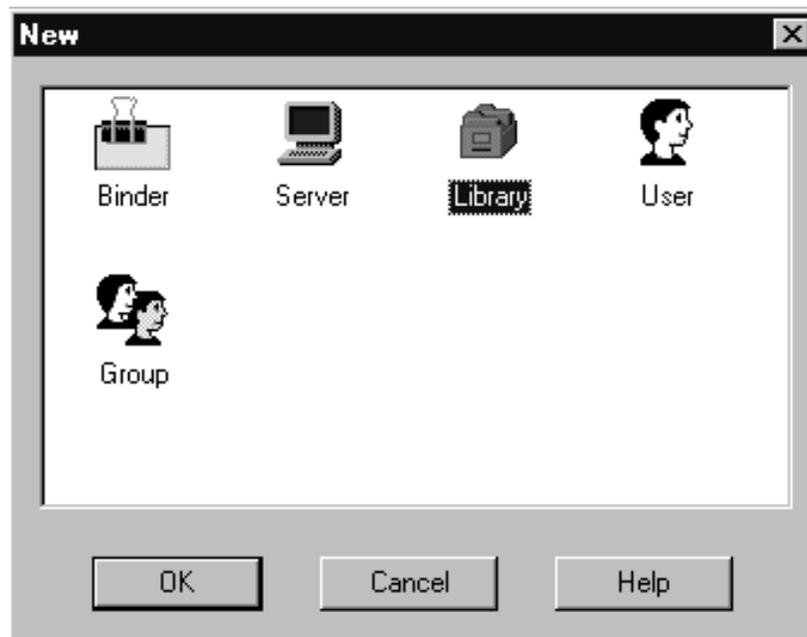
To add a library:

- 1 Access the New window. Either select

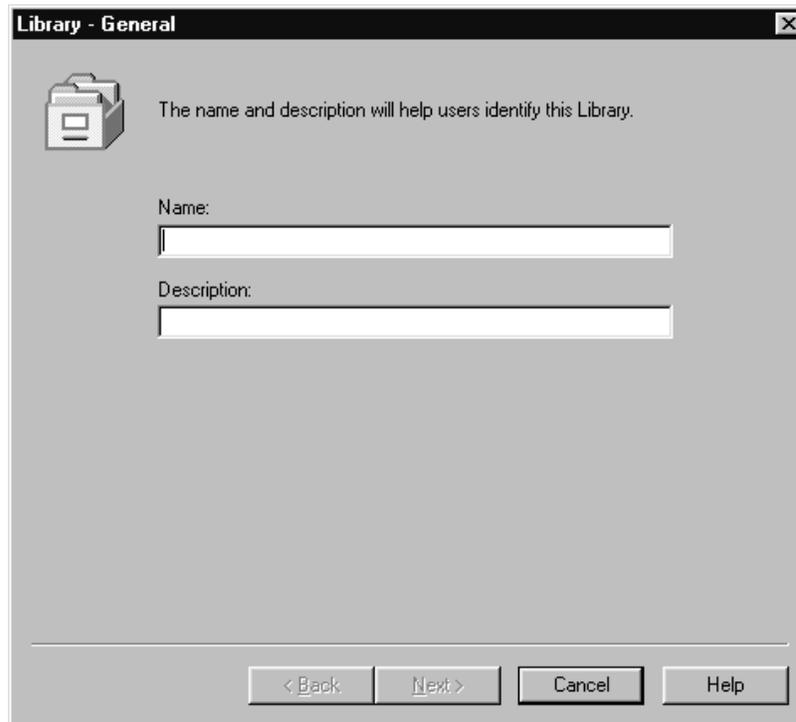


from the menu or select the **New** icon  from the toolbar.

- 2 Select **Library** and click **OK**.



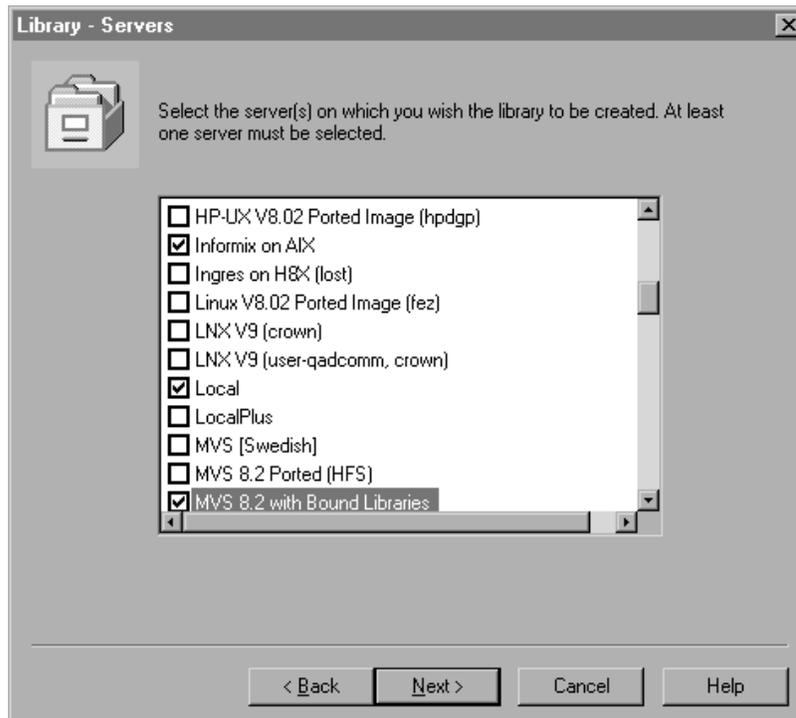
The Library - General window opens.



- 3 Enter a name for the library and a description of the contents. The name can be up to 8 characters long, and must be in all uppercase.

The name must not be duplicated on any server on which you are creating the library.

Click **Next** to open the Library - Servers window.



- 4 Select the servers on which you want to create this library. A separate library definition will be created for each server.

Click **Next** to open the Library - Details window.

- 5 Enter the specific information about the library.

Generate Libref Automatically
is not a selectable field.

Libref

specifies the libref for the library. The libref must be unique for each server on which you are creating the library definition. This field is not selectable.

Assign at Server Startup

specifies that the library is automatically assigned whenever a server starts on which the library exists.

Engine

specifies the SAS engine with which the data was created. Leave this blank if you want the system to automatically determine the appropriate engine. For more information about engines, refer to the SAS documentation for your operating environment. See also the chapter that corresponds to your database management system in *SAS/ACCESS Software for Relational Databases: Reference, Version 8*.

DBMS Connection

specifies that the library that you are defining is a database library.

Path

is the physical path where the library resides on the server.

You can use the following wildcards in the **Path** field:

<userid>	substitutes a Windows user ID. For example, if a user logs on as ebeacon, then <code>\\librarypath<userid></code> becomes <code>\\librarypath\ebeacon</code> .
----------	--

- <sample> substitutes the Enterprise Client sample directory. For example, if you installed by using the default directories, <sample>\data becomes **C:\Program Files\SAS Institute\Enterprise Clients\Sample\data**. If, however, another user installed in **D:\EC**, then <sample>\data becomes **D:\EC\Sample\data**.
- <serveruser> substitutes the server login user ID.
- <serverpassword> substitutes the server login password.
- <libraryuser> substitutes the library login user ID.
- <librarypassword> substitutes the library login password.

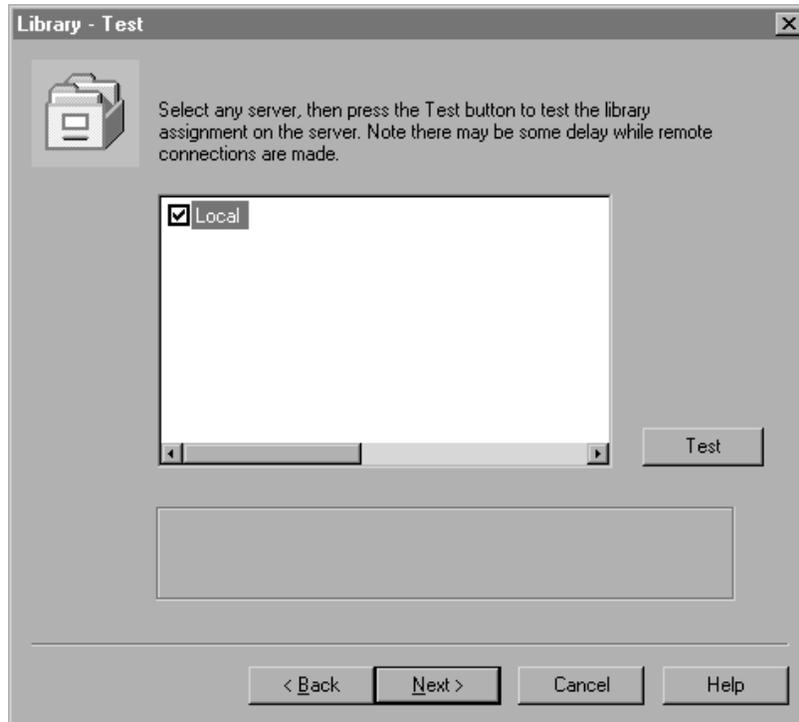
Options

Refer to the documentation for your operating environment for the appropriate options. If wildcards are valid in the **Path** field, then you can also use them in the **Options** field. Avoid using prompting options such as **DBPROMPT=YES**. For additional options, refer to the chapter that corresponds to your database management system in *SAS/ACCESS Software for Relational Databases: Reference, Version 8*.

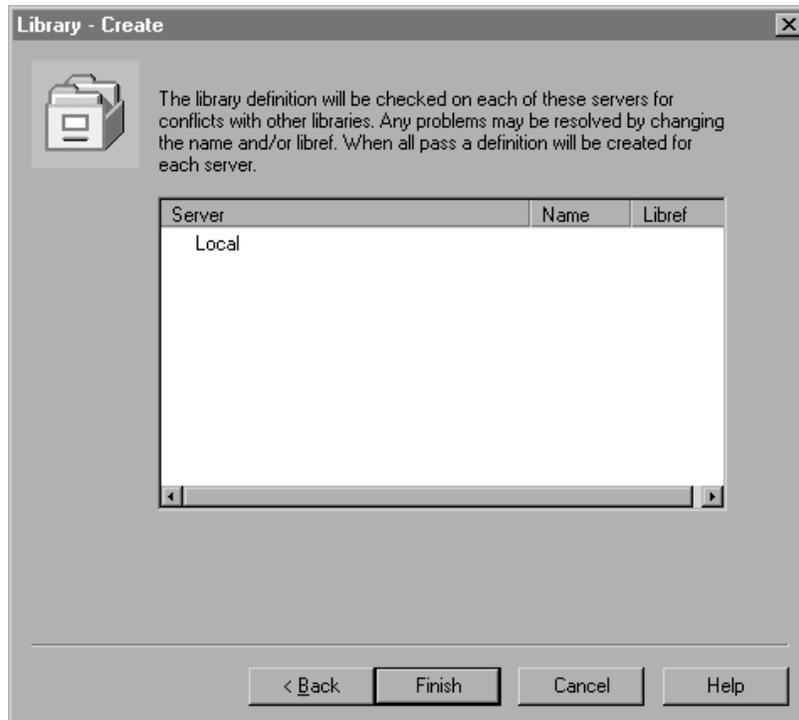
Click **Next** to open the Library - Credentials window.

- 6 Specify how access to the library is controlled, either by not requiring credentials, by prompting for credentials as needed, by prompting every time, or by using a specified user ID and password.

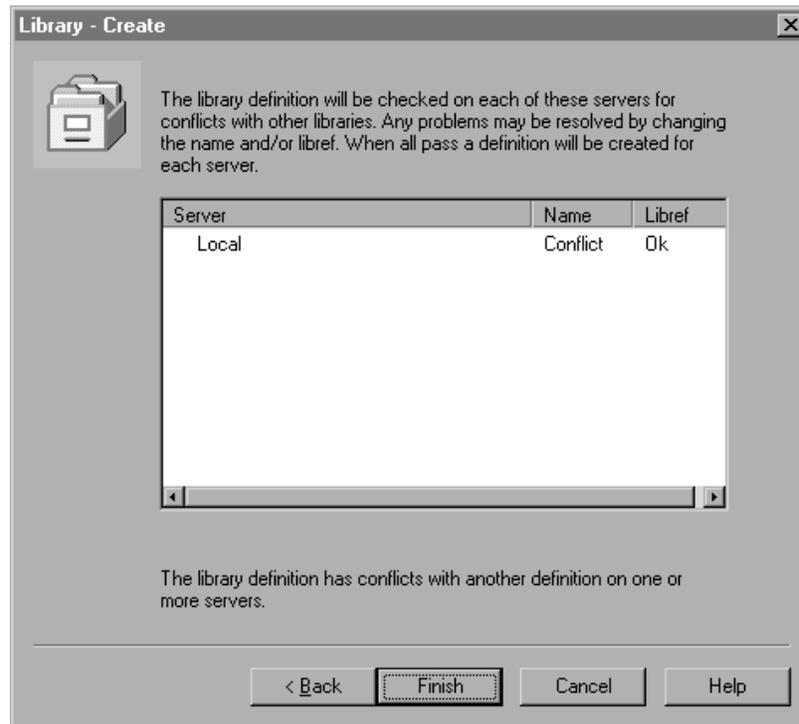
Click **Next** to open the Library - Test window.



- 7 Select a server on which you want to test the library definition and click **Test**. The results of the test appear at the bottom of the window. Click **Next** to open the Library - Create window.

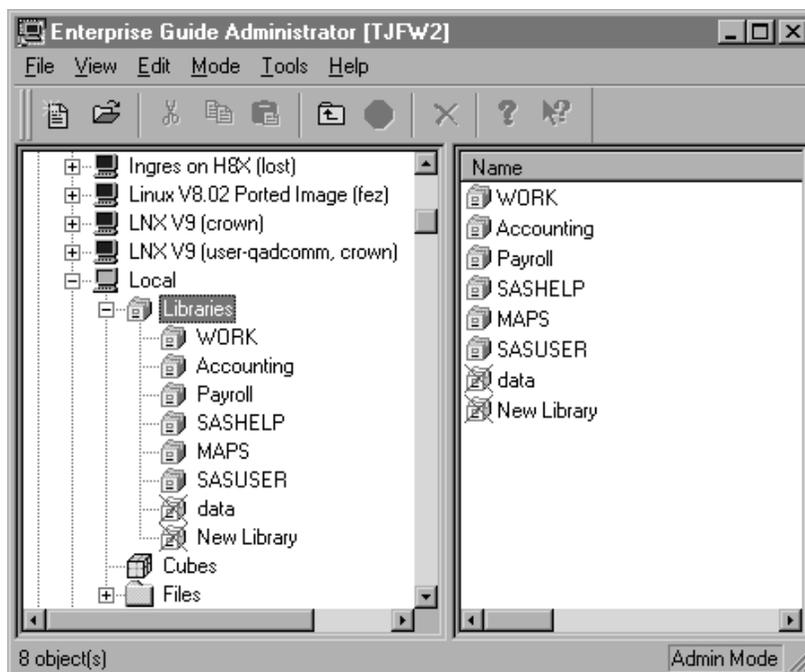


- 8 Click **Finish** to assign the library definitions on the specified servers. If the library name or libref is not unique on all servers, then the appropriate column indicates the conflict.



Click **Back** to return to the appropriate window and change the name or libref.

The library is created and then is displayed in the viewer under the client libraries for the specified servers.



Creating the EGTASK Library

By default, Enterprise Guide stores output data in the SASUSER library on the SAS server. However, there are situations in which you need to change the output location. For example, you might want all output to go to a central library that all users can access. In addition, SASUSER is a temporary location on OS/390 and some UNIX systems, so output needs to be directed to a different location in order to be saved permanently. Enterprise Guide now supports the EGTASK library as an alternative default output location. If the EGTASK library is defined, then Enterprise Guide will send output data to EGTASK instead of to SASUSER.

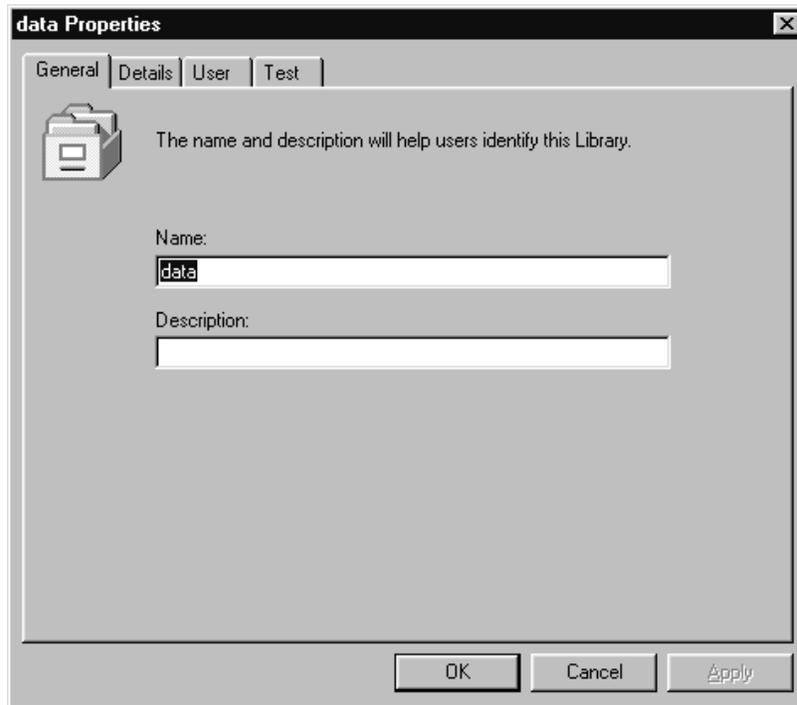
To define the EGTASK library:

- 1 Select **File New** and select **Library** from the New window. The Library - Server window opens.
- 2 Select a server on which users run tasks. If more than one server is used to run tasks, then you must go through the library definition process separately for each server that needs to have EGTASK available. Click **Next** to open the Library - General window.
- 3 Type **EGTASK** in the **Name** field, and select the **Assign at Server Startup** check box. This value ensures that EGTASK is created and available every time the server is started. Click **Next** to open the Library - Details window.
- 4 Select **DEFAULT** in the **Engine** field. In the **Path** field, type or select a location that is not protected and is available to all users. Click **Next** to open the Library - Create window.
- 5 Click **Test** to confirm the library definition. Click **Finish** to create the definition.

Updating Library Properties

Properties for libraries include information such as servers, pathnames, and LIBNAME options. To view or update this information:

- 1 Select a server, and then select a library so that it is highlighted. You can change properties for unassigned libraries only. If the library that you want to change is assigned, then click the right mouse button and select **Unassign** from the shortcut menu.
- 2 With the mouse pointer on the selected library, click the right mouse button to open the shortcut menu.
- 3 Select **Properties**.



- 4 Select the tab that corresponds to the information that you want to update. See “Adding a Library” on page 82 for information about each field.

Unassigning Libraries

You can assign and unassign libraries dynamically during an Enterprise Guide Administrator 1.2 session. To unassign an assigned library, right-click the library name and select **Unassign** from the shortcut menu. The color of the library icon changes from yellow to white to indicate that the library is not assigned. While the library is disconnected, its icon is also white in any open dialog boxes to indicate that the library is selectable but currently unassigned.

To assign the library, right-click the library icon and select **Assign** from the shortcut menu. The library icon changes back to yellow.

To enable you to open servers without making a connection, select

View ► Server ► No Connection

Use this option to modify libraries without connecting to the server. To require a connection to the server when you modify libraries, select

View ► Server ► Normal

Note: Do not manually assign and unassign libraries on OS/390 servers. When a library on an OS/390 server is assigned, it locks out all other potential users. △

Managing Binders

About Binders

Binders are a type of virtual folder that provide users with transparent access to data and, at the same time, enable users and work groups to easily share information across an enterprisewide network of servers on multiple platforms. As you create binders in Enterprise Guide Administrator, you can map them to

- folders or directories on a *server* that you can access through Enterprise Guide Administrator. The server locations can contain SAS data files or data files not created in SAS.
- folders in your Windows *native file system* that can contain SAS data files or data files not created in SAS.

Here is some additional information about binders:

- Users access binders through client applications such as Enterprise Guide.
- You can control user access to binders through groups, or you can grant all Enterprise Client users access to binders.
- In binders, you can insert shortcuts to data so that users can easily access information without having to know its exact location. See “Adding Objects to a Binder” on page 94 for more information.

Binders That Map to Server Locations

When an Enterprise Client user opens a file from a binder that maps to a location on a server, a *copy* of the file is sent to the client machine so that other users can still access the file. After the Enterprise Client user closes the file, the file is updated on the server.

Binders That Map to Native Files

Binders that map to native files contain files and directories that users can access by using Windows native file protocols. The files and directories can reside on local drives or on any file server in the Windows network; they do not reside on servers that you administer.

Note: Because others can share native files on Window servers, be aware that when users open files in native binders, they are granted direct access to those files. Therefore, they might lock out other users. The files in native folders are not copied to client machines as they are in files on a server location. △

Adding a Binder

To add a binder (either native or shared):

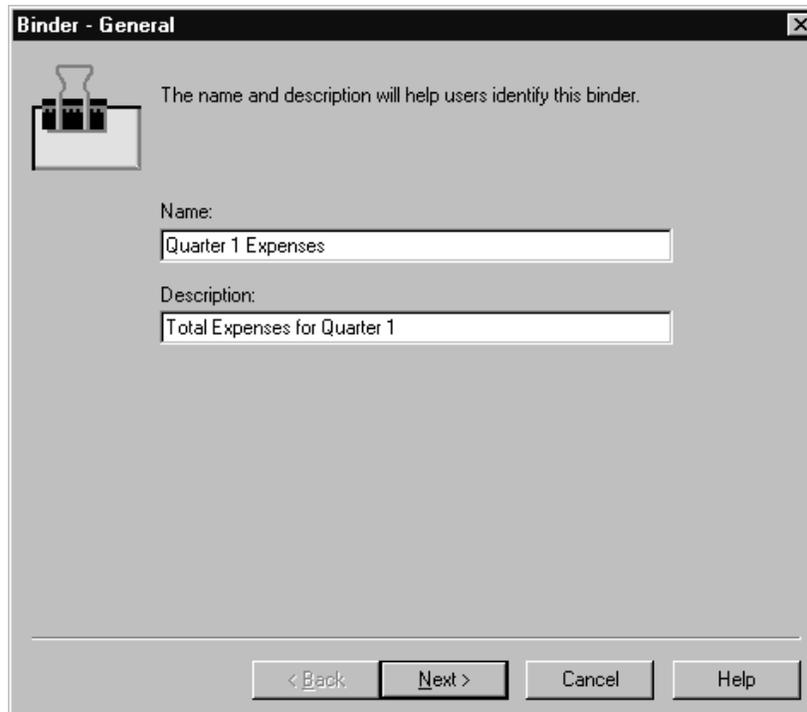
- 1 Access the New window. Either select



from the menu or select the **New** icon  from the toolbar.



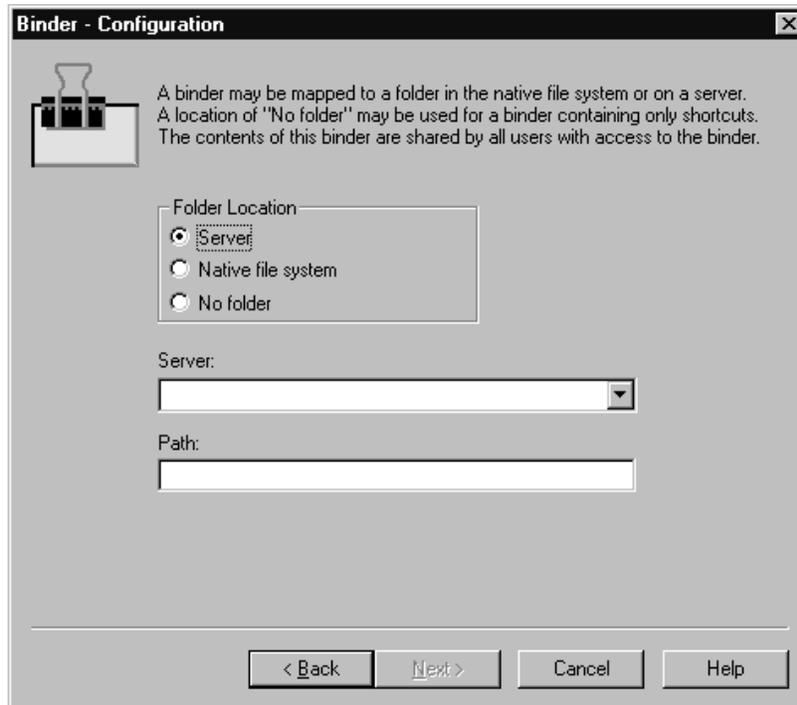
- 2 Select **Binder** and click **OK** to open the Binder - General window.



- 3 Enter the name of the binder and a description of its contents. This name and description will appear in the Enterprise Guide Administrator viewer under Binders and in other Enterprise Client applications, such as Enterprise Guide. The name cannot contain any of these characters:

\ / : * ? " < > |

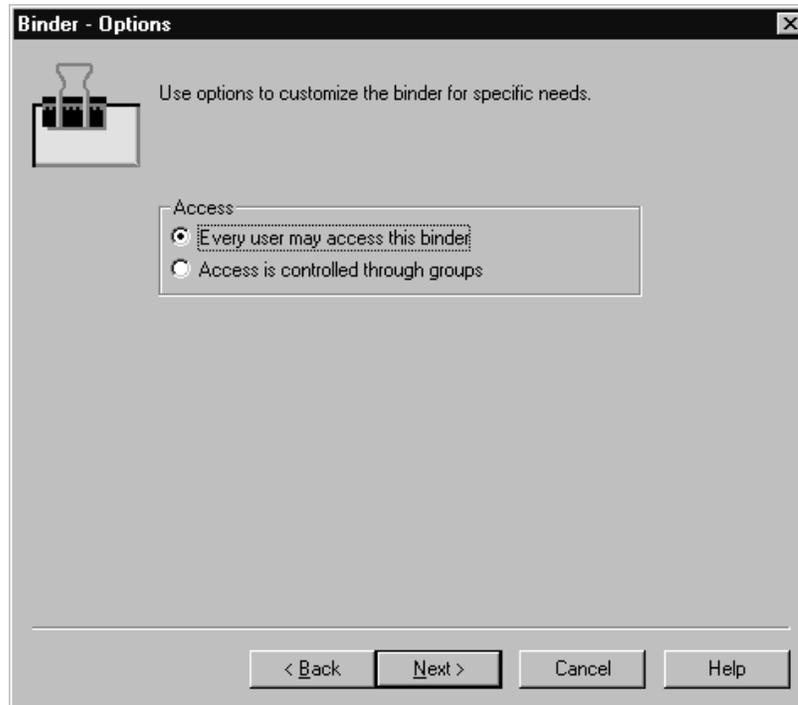
Click **Next** to display the Binder - Configuration window.



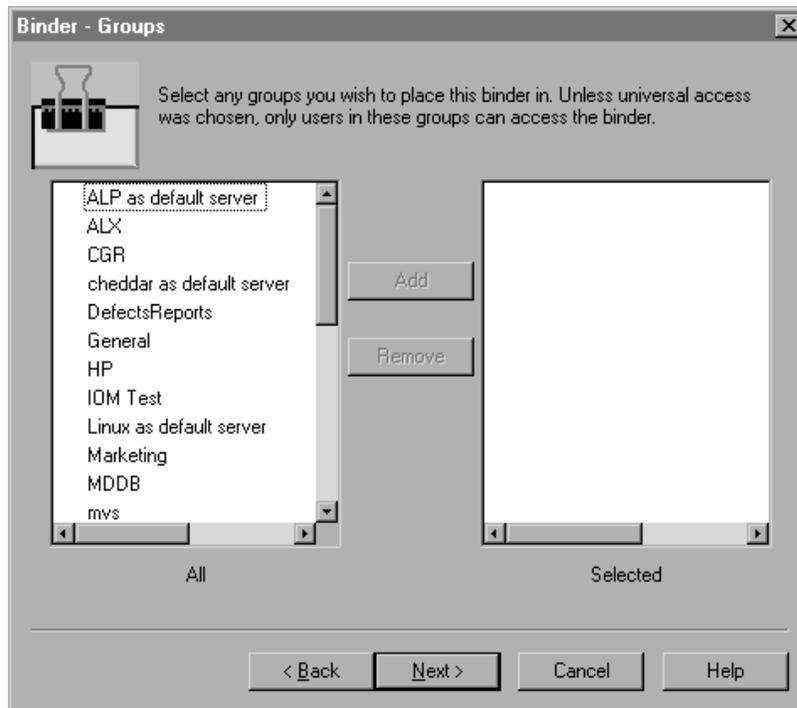
4 Use this window to select the location that you are mapping to. In the **Folder Location** area,

- **Server** maps to a location on a server that you administer through Enterprise Guide Administrator. Select the server, and then enter the location in the server’s native format.
- **Native File System** maps to a folder on any server in your Windows network or to a location on your local machine. Enter a full Windows pathname.
- **No Folder** creates an “empty” binder that you can later populate with shortcuts, files, and other binders.

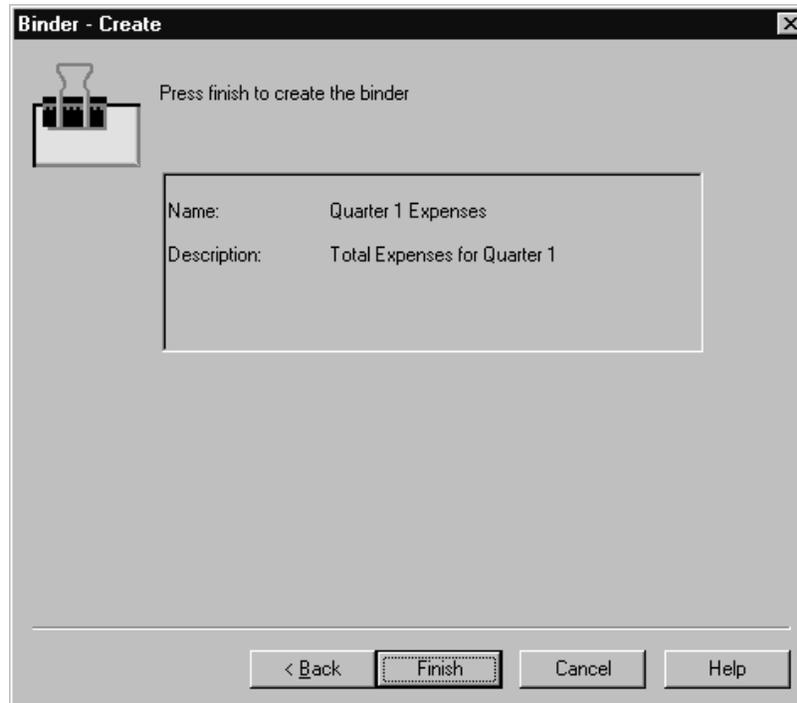
Note: If the server has not been added in Enterprise Guide Administrator, then it will not appear in the list box. For details about adding servers, see “Adding SAS Servers” on page 73. Click **Next** to open the Binder - Options window. △



- 5 Specify whether you will allow all users to access this binder or whether you will control access to this binder through groups. Click **Next**.
- 6 If you chose to control access through groups, then the Binder - Groups window opens.



Select a group or groups from the **All** box, and click **Add**. To remove group access, select a group or groups from the **Selected** box, and click **Remove**. Click **Next** to open the Binder - Create window.



7 Click **Finish** to create the binder.

Adding Objects to a Binder

After you create the binder, you can add shortcuts to data that might be scattered throughout the enterprise on multiple platforms. For example, you can create a binder for a department, and then in that binder add shortcuts to that group's relevant data files on OS/390, UNIX, and Windows NT. A binder can contain shortcuts to data objects that are accessible through SAS servers that you have added in Enterprise Guide Administrator. You can add shortcuts for

- data files
- data sets
- folders
- libraries.

To add a shortcut to a binder, navigate to the object to which you want to add a shortcut, and then either drag and drop or copy and paste to add the shortcut. To drag and drop, select the object and drag it to the binder. To copy and paste, click on the object with the right mouse button and select **Copy** from the shortcut menu. Then navigate to the binder and select **Paste** from the shortcut menu.

Deleting a Binder

Note: A binder merely points to items that reside on servers. Therefore, the original items are **not** deleted when you delete the binder. However, any shortcuts in the binder are deleted when you delete a binder. △

To delete a binder:

- 1 Select the binder in the Enterprise Guide Administrator viewer.

- 2 Press the DELETE key. Or, with the mouse pointer on the selected group, click the right mouse button to open the shortcut menu. Select **Delete**.
- 3 Confirm your choice.

Managing Groups

About Groups

A *group* is a collection of Enterprise Client users who are granted access to the same set of resources, such as SAS servers, libraries, and binders. In a networked environment, you can create groups of users to control access to servers, libraries, and binders. If you will be adding groups, it is best to add at least one group first, and then add the users that belong to that group. Then, as you add each user, you assign the user to one or more existing groups.

Compared with user profiles, groups provide a lower maintenance approach to resource access management, because updating information for a group is easier than updating information for individual users. Groups and users are useful for high-security environments where you need strict control of user access to resources. If, however, you are granting your users universal access to servers and binders, then you do not need to add groups.

Adding a Group

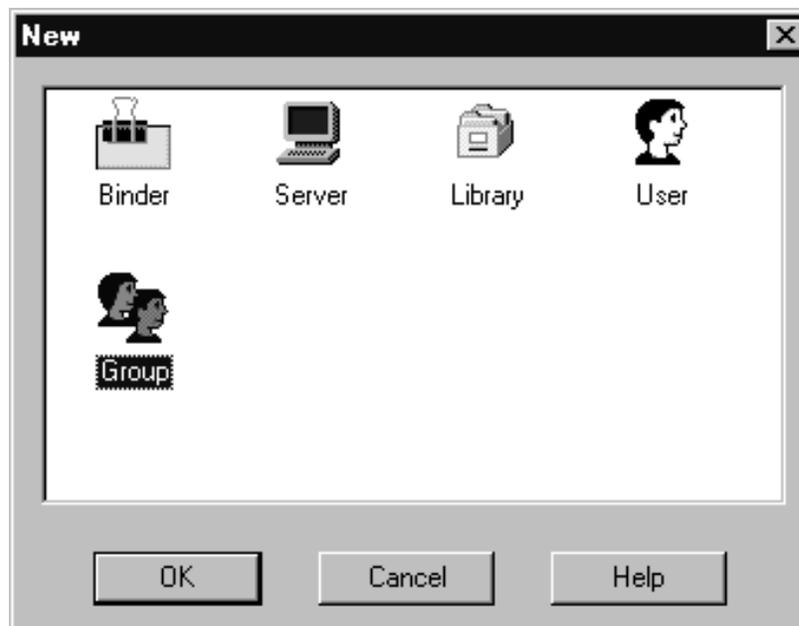
To add a group:

- 1 Access the New window. Either select



from the menu or select the **New** icon  from the toolbar.

- 2 Select **Group** and click **OK**.



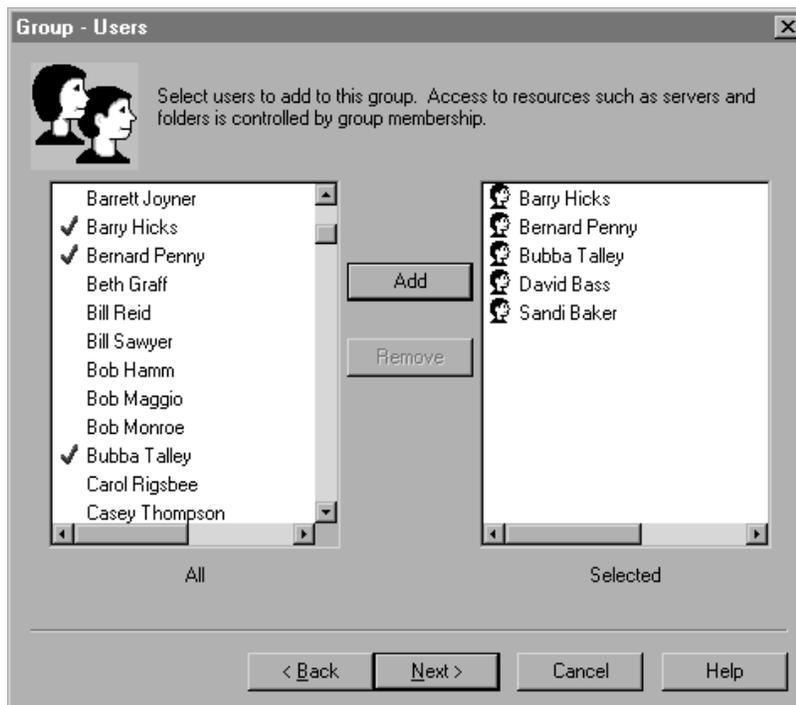
- 3 Enter the new group's name and description. The name cannot contain any of these characters:

\ / : * ? " < > |

Click **Next** to continue.

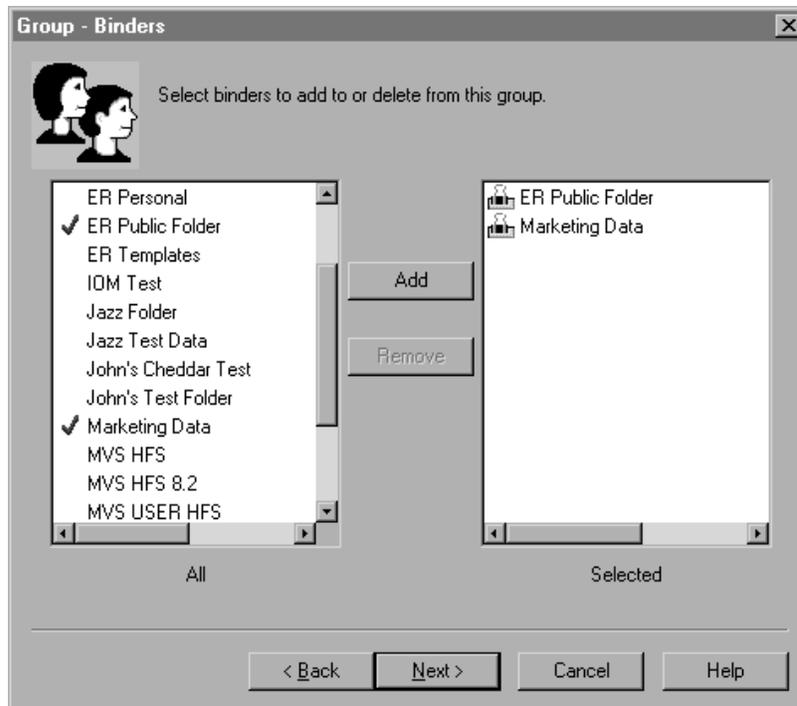
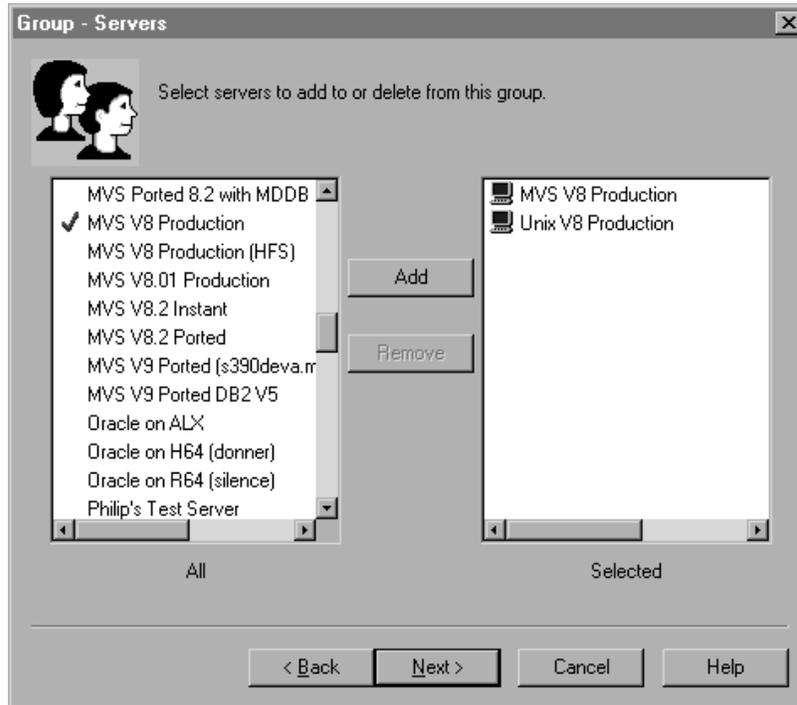


- 4 To add users to the group, select users from the **All Users** box, and click **Add**. The change is reflected in the **Selected** box. To remove users from the group, select a user from the **Selected** box and click **Remove**. Click **Next** to continue.



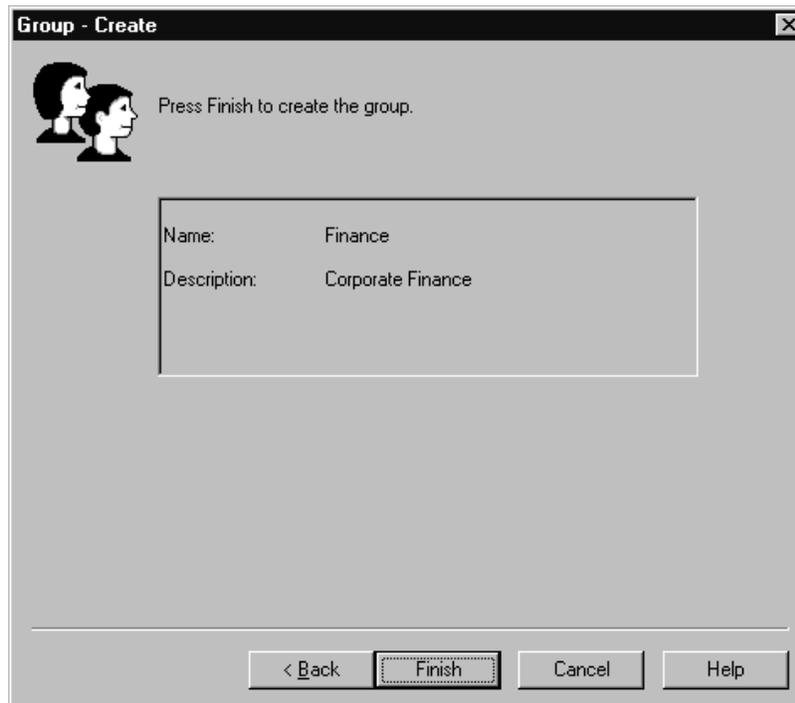
- 5 To specify which servers the group can access, select a server or servers from the **All** box, and click **Add**. To remove access to specific servers, select a server or servers from the **Selected** box, and click **Remove**.

Click **Next** to continue.



- 6 Specify the binders that you want the group to access. Select one or more binders from the **All** box, and click **Add**. To remove access to specific binders, select one or more binders from the **Selected** box, and click **Remove**.

Click **Next** to continue.



- 7 Add the group by clicking **Finish**.

Deleting a Group

To delete a group:

- 1 Select the desired group in the Enterprise Guide Administrator viewer.
- 2 With the mouse pointer on the selected group, click the right mouse button to open the shortcut menu. Select **Delete**.
- 3 Confirm your choice.

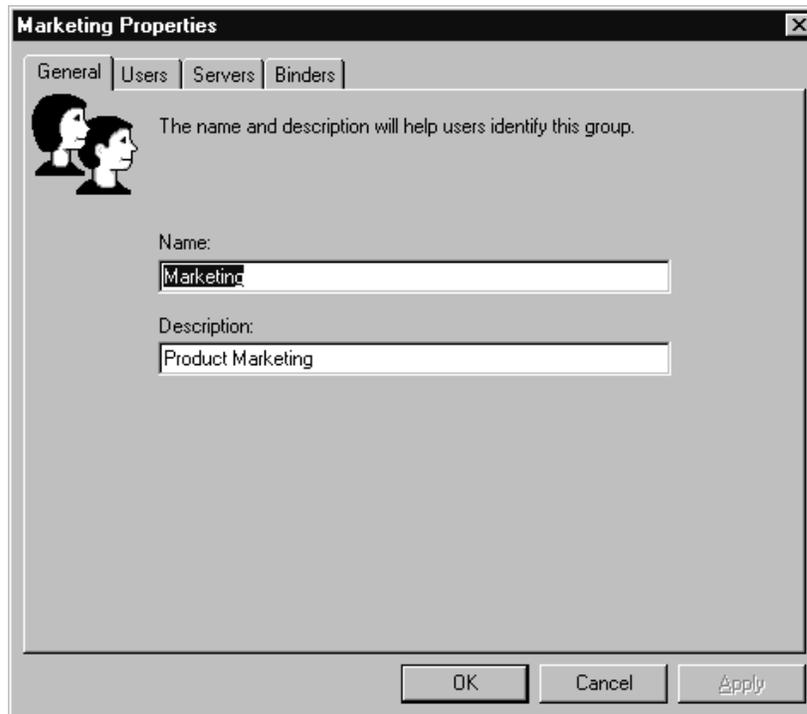
Note: Deleting the group does not delete the users who are its members. To delete specific users, see "Deleting a User" on page 103. Δ

Updating Group Properties

Access group properties to update information for groups and set defaults.

To change the properties for a group:

- 1 Select the desired group in the Enterprise Guide Administrator viewer.
- 2 With the mouse pointer on the selected group, click the right mouse button to open the shortcut menu.
- 3 Select **Properties**.



- 4 Select the tab that corresponds to the information that you want to update. For information about the fields in each tab, see “Managing Groups” on page 95.

Managing Users

About Users

Users are individuals with unique ID and password combinations. In a networked environment, you can create *groups* of users to control access to servers, libraries, and binders. If you will be adding groups, it is best to add at least one group first, and then add the users who belong to that group. For information about groups, see “Managing Groups” on page 95.

Note: If you are an administrator in a networked environment, you must first add yourself as a user with administrator privileges, and then assign yourself to a group. △

For example, suppose that you have a new user who just installed the Enterprise Client software on his local machine. Suppose that he wants to access an administered library on a UNIX machine that you added earlier by using Enterprise Guide Administrator. As you add the new user, you can assign him to a group that already has access to the UNIX server and the administered library.

You will need to gather this information before adding each user:

- name
- user ID (for the PC)
- phone number (optional)
- e-mail address (optional)
- office location (optional).

Adding a User

To add a user:

- 1 Access the New window. Either select



from the menu or select the **New** icon  from the toolbar.

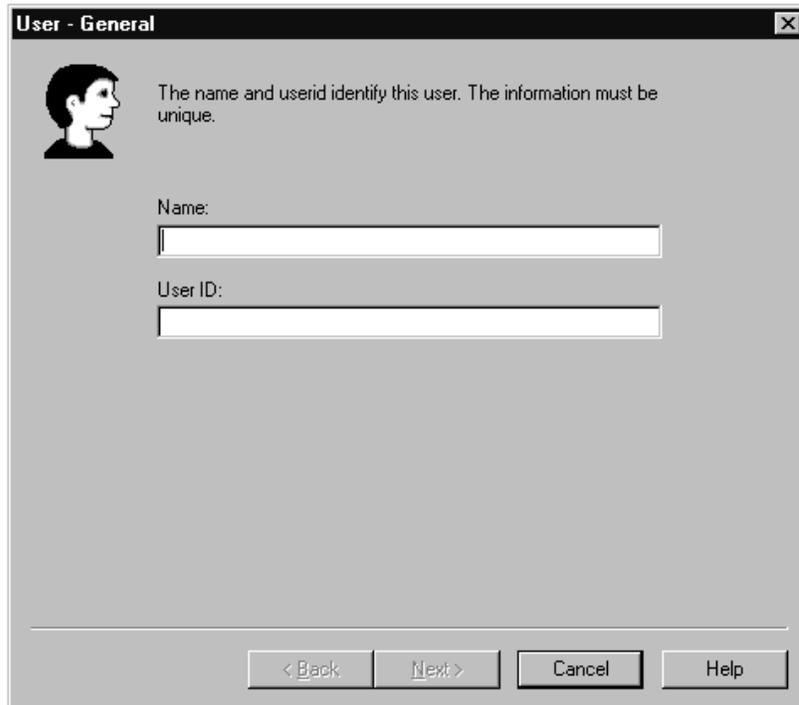
- 2 Select **User** and click **OK**.



- 3 Enter the user's name and user ID. The name cannot contain any of these characters:

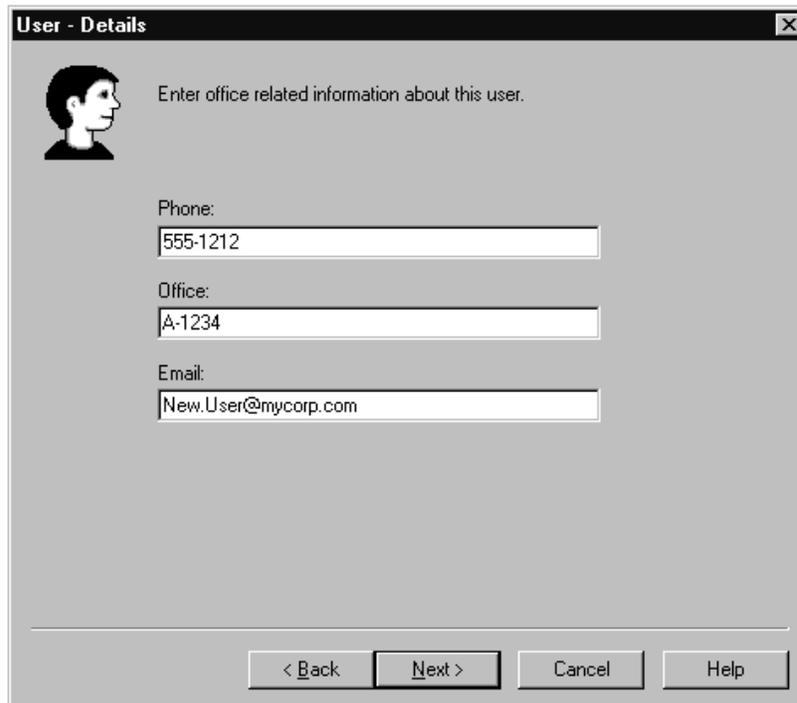
\ / : * ? " < > |

Click **Next** to continue.



The dialog box is titled "User - General" and contains a small icon of a person's head and shoulders. To the right of the icon, the text reads: "The name and userid identify this user. The information must be unique." Below this text are two text input fields. The first is labeled "Name:" and the second is labeled "User ID:". At the bottom of the dialog box, there are four buttons: "< Back", "Next >", "Cancel", and "Help".

- 4 Add more information about this user. Click **Next** to continue.

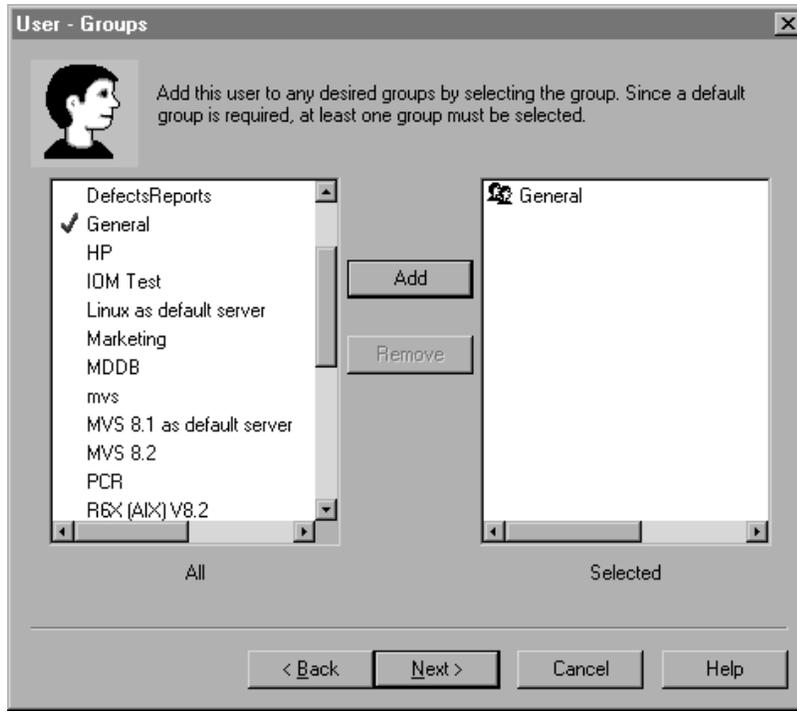


The dialog box is titled "User - Details" and contains a small icon of a person's head and shoulders. To the right of the icon, the text reads: "Enter office related information about this user." Below this text are three text input fields. The first is labeled "Phone:" and contains the text "555-1212". The second is labeled "Office:" and contains the text "A-1234". The third is labeled "Email:" and contains the text "New.User@mycorp.com". At the bottom of the dialog box, there are four buttons: "< Back", "Next >", "Cancel", and "Help".

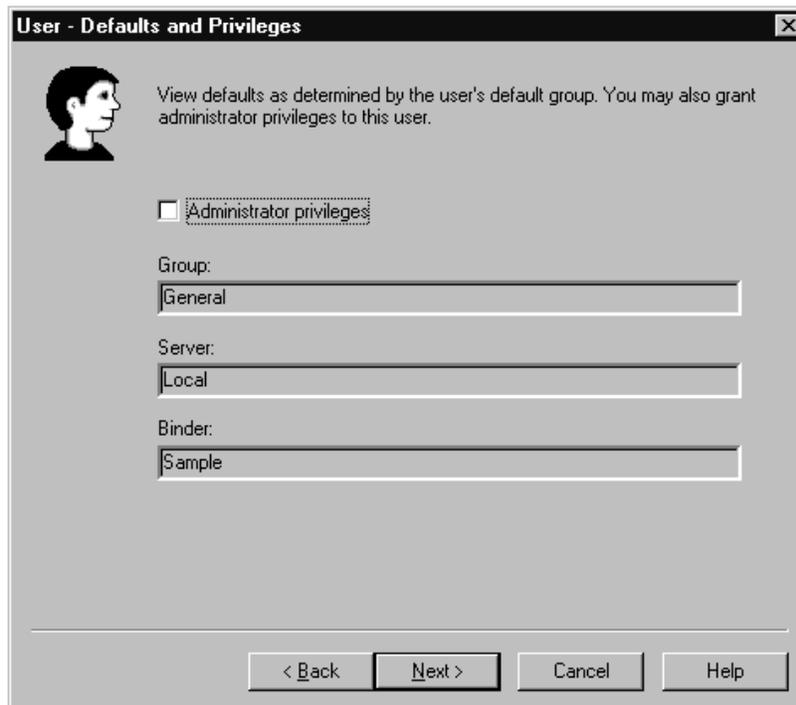
- 5 To add this user to one or more groups, select a group from the **All** box and click **Add**. To remove the user from a group, select the group from the **Selected** box and click **Remove**. If you assign the user to only one group, then that group automatically becomes the user's default group, and it appears in red. If you add the user to more than one group, then you must designate one group as the user's

default group. To do this, select the group from the **Selected** box, and click **Make Default**. Click **Next** to continue.

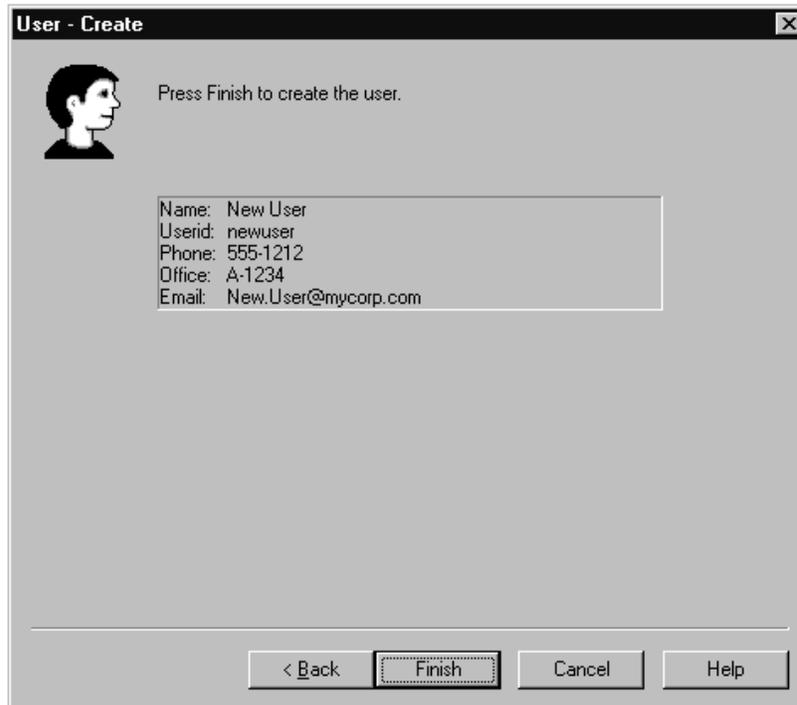
Note: You can designate General as the user's default group, add a new group later (see "Adding a Group" on page 95), and then include the user in that group. △



6 Review the user's default values and privileges.



Check the **Administrator privileges** box to grant the user administrator privileges, thereby granting permission for write access to the metadata repository. Click **Next** to continue.



7 Click **Finish** to add the user.

Deleting a User

To delete a user:

- 1 Select the user that you want to delete.
- 2 With the mouse pointer on the selected user, click the right mouse button, and select **Delete** from the shortcut menu (or use the DELETE key). You cannot delete objects whose names are in brackets [].
- 3 Confirm the deletion. The user is automatically deleted from any groups that included the user.

Updating User Properties

User properties include user IDs, group memberships, defaults, and privileges.

Note: It is helpful to be in View User mode when updating user properties so that you can see the user's defaults. To switch to View User mode, select a user so that it is highlighted. Then from the menu select

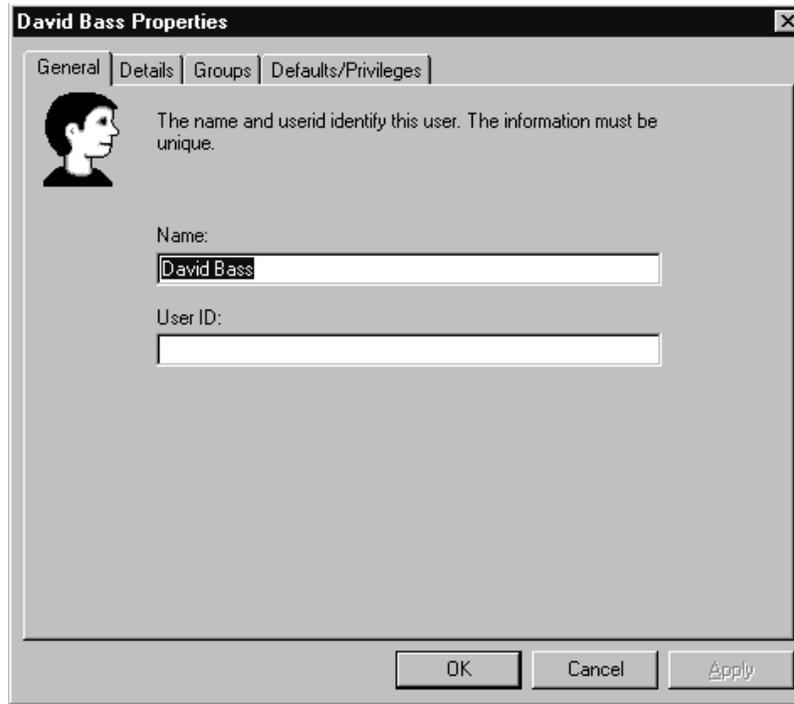


△

To update properties for a user:

- 1 Select the desired group in the Enterprise Guide Administrator viewer.

- 2 With the mouse pointer on the selected user, click your right mouse button to open the shortcut menu.
- 3 Select **Properties**.



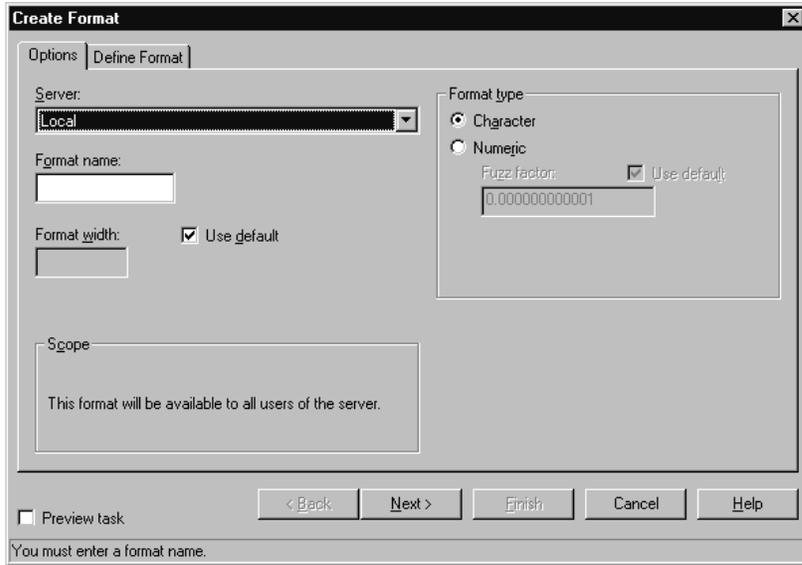
- 4 Select the tab that corresponds to the information that you want to update. For information about the fields in each tab, see “Adding a User” on page 100.

Adding a Format

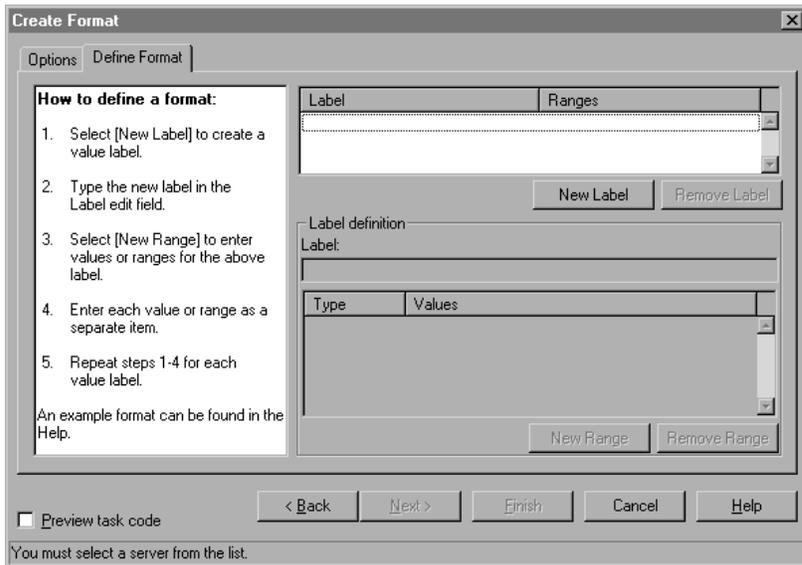
You can create SAS formats that will be available for Enterprise Guide users. The formats are available in the Enterprise Guide data grid and query builder and from most Enterprise Guide tasks. The formats must be stored in the LIBRARY library. Because this library is referenced in the search path, the formats are available to users.

To add a format:

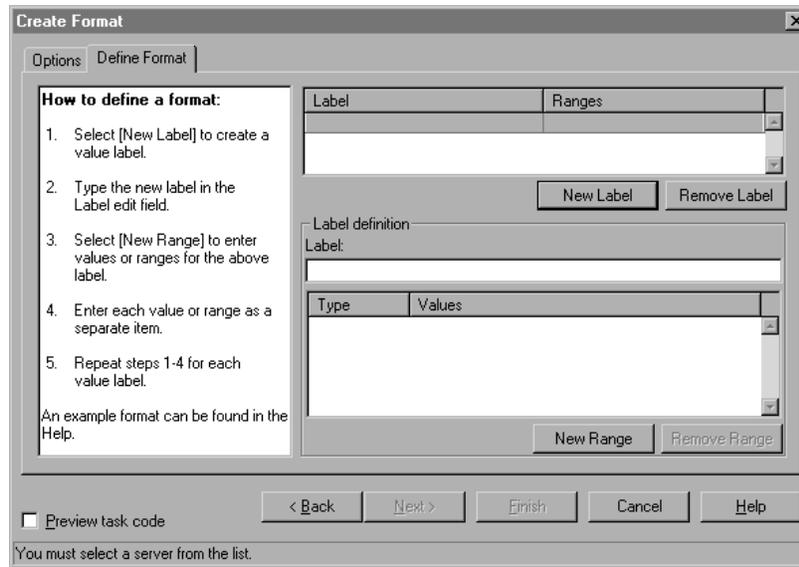
- 1 Ensure that the LIBRARY library is defined on the server that will contain the formats. If it is not defined, then you must add the definition. See “Adding a Library” on page 82 for information about defining libraries.
- 2 Select **Tools Create Format**. The Create Format Task window opens, with the **Options** tab selected.



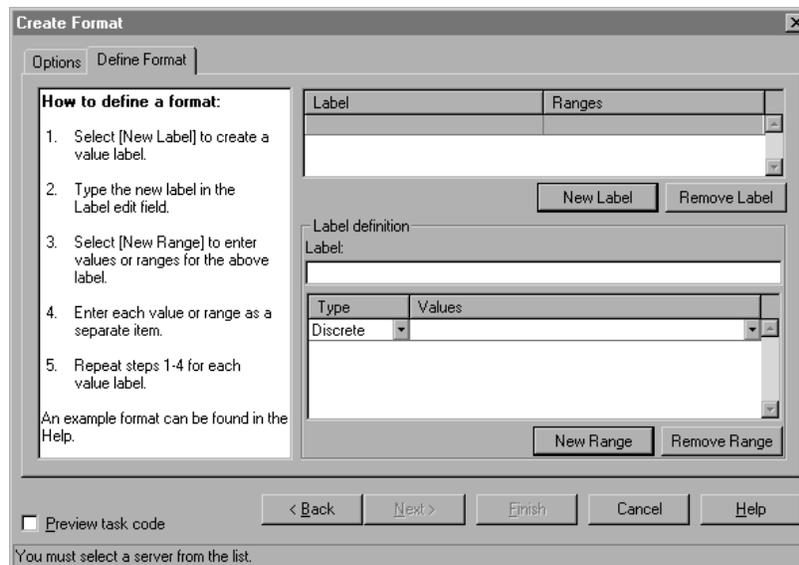
- 3 Type a name for the format and specify the server where the format will be stored. In addition, select whether the format type will be character or numeric. All formats that are created in Enterprise Guide Administrator are available to all users. When you finish, click **Next** or select the **Define Format** tab. The Create Format Task window displays the **Define Format** tab.



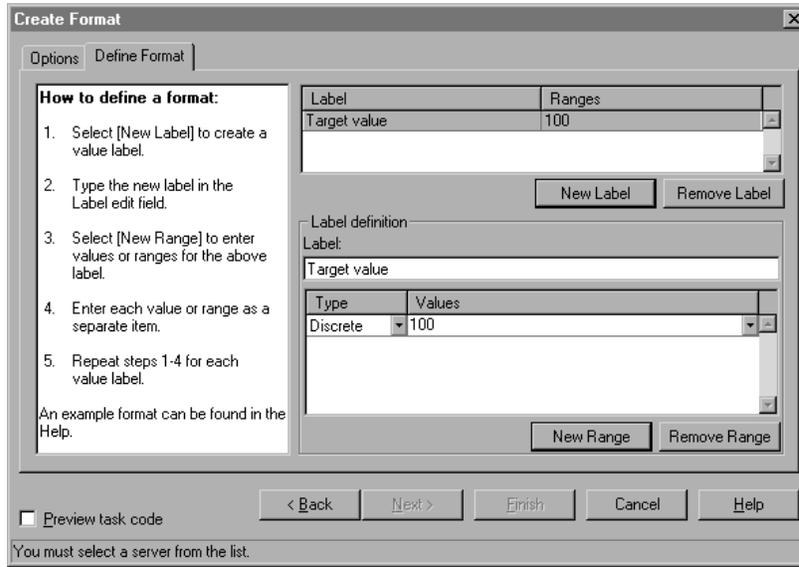
- 4 To define a new format, click the **New Label** button. The buttons and fields in the **Label definition** frame then become active.



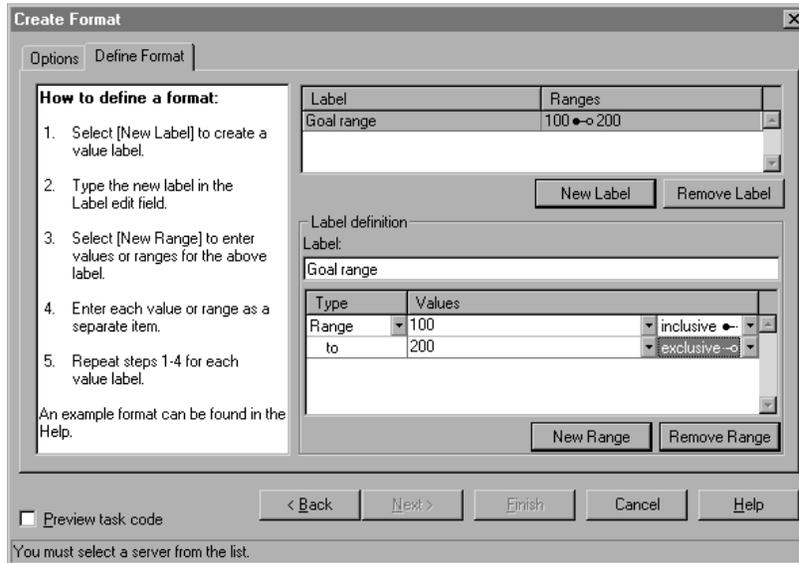
- 5 Click the **New Range** button. The **Type** field displays a pull-down menu that enables you to select whether the format should be Discrete (applied to a single value) or Range (applied to a range of values).



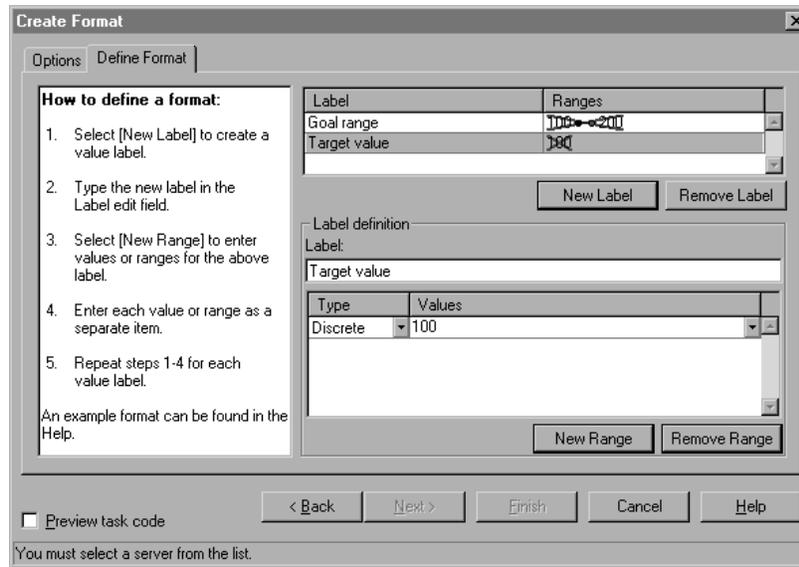
- 6 If you select **Discrete** as the format type, then type the value to which the format should be applied in the **Values** field. In the **Label** field, type the text that you want to be displayed when the specified value is encountered.



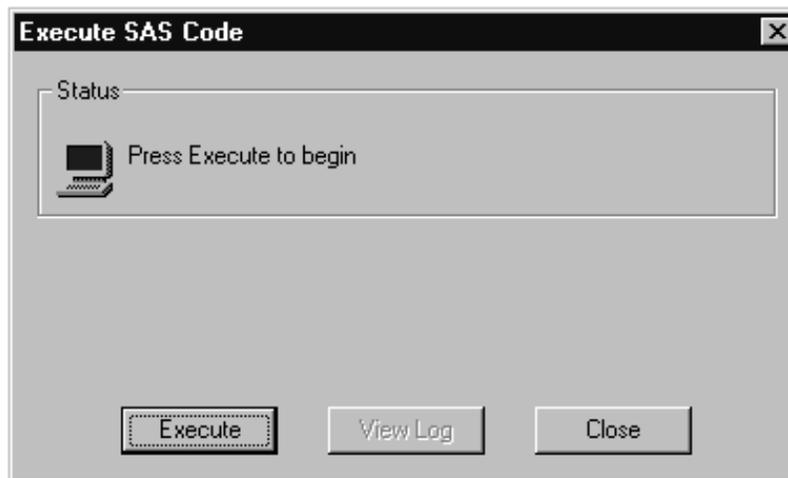
- 7 In the example that is illustrated here, the label **Target value** is displayed when the discrete value 100 is encountered.
- 8 If you select **Range** in the **Type** field, then type the values for the upper and lower range boundaries in the **Values** field. Specify whether the range should include or exclude the boundary values by selecting **inclusive** or **exclusive** from the pull-down menu. Type the label for the format in the **Label** field.



- 9 If you type a value for a format that causes it to be invalid, then the area with the error is marked in the top portion of the window. In this example, the user is trying to define a discrete format for the value 100, but that value is already contained in a previously defined range format.



- 10 When you finish defining the format, click **Finish**. The Execute SAS Code dialog box opens.



- 11 Click **Execute** to run the SAS code and create the format. A message is displayed when the code finishes running.

Importing Server Definitions

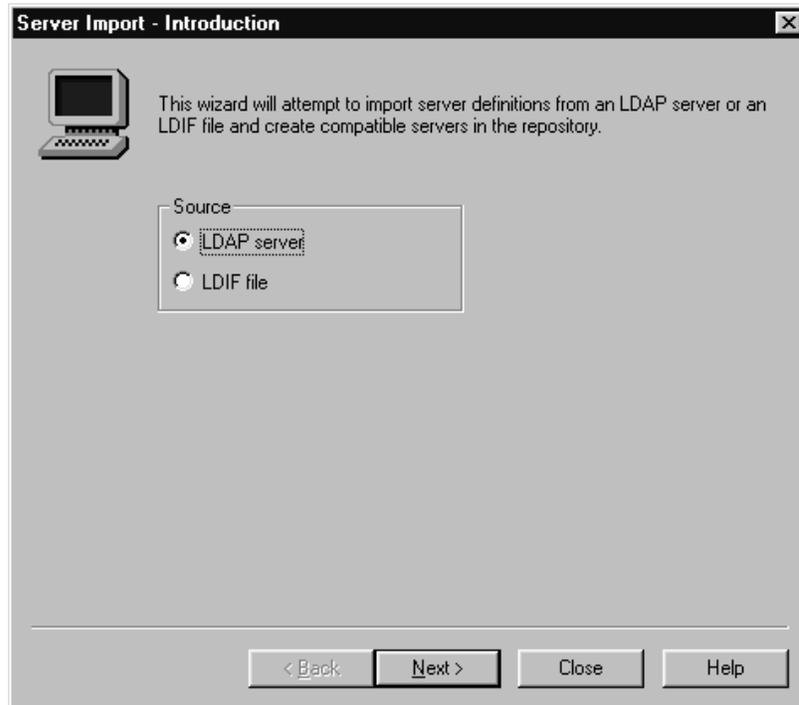
If you have server definitions that are contained in LDAP directories or LDIF files, then you can import those server definitions to the SDS directory and view them in Enterprise Guide Administrator.

To import server definitions:

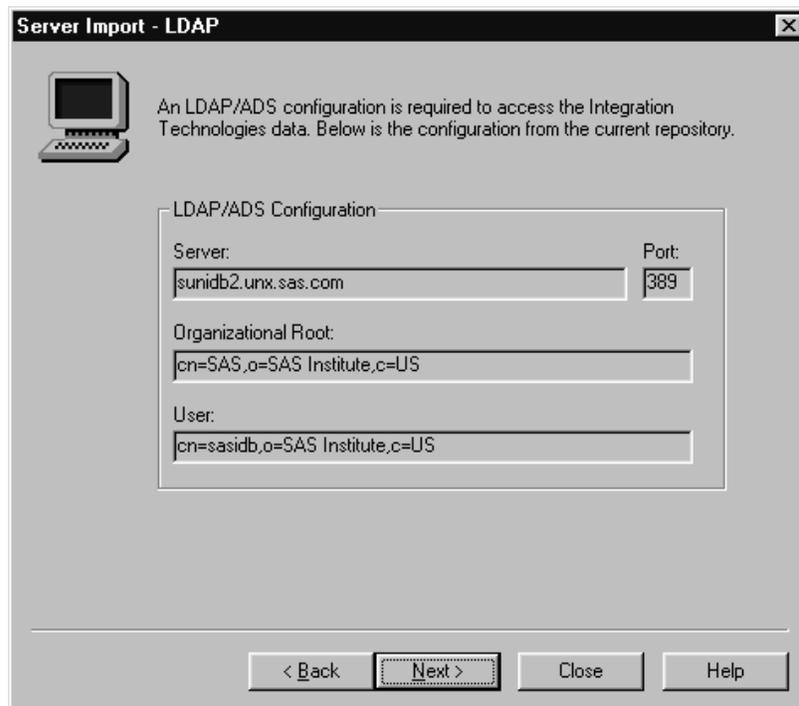
- 1 Select

Tools \blacktriangleright Server Import Wizard

The Server Import - Introduction window opens.



- 2 Select **LDAP server** or **LDIF file**, depending on where the server definitions are located, and click **Next**. If you select **LDAP server**, then the **Server Import - LDAP** window opens.

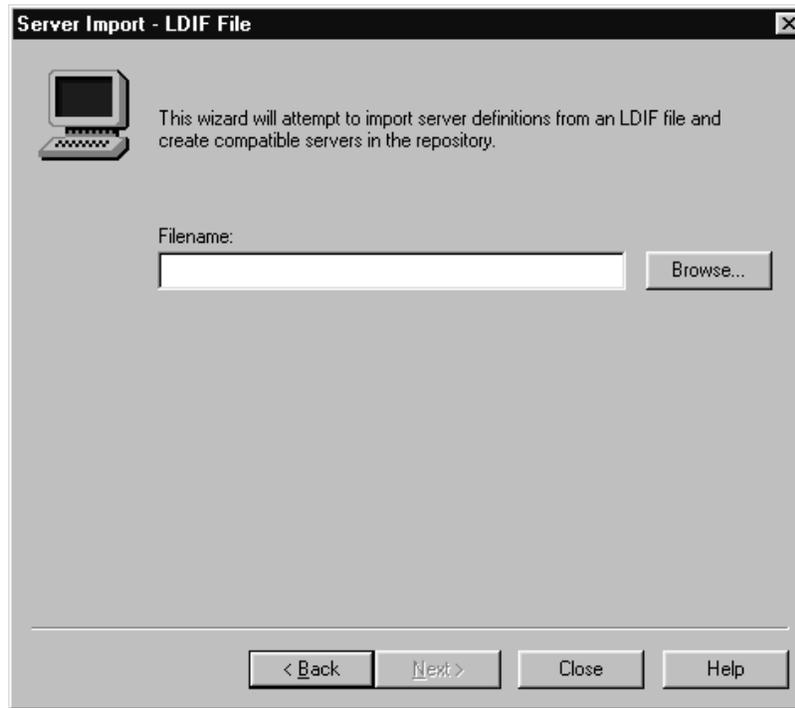


This window displays the LDAP server information as it was entered in the **Configure LDAP/ADS Server** window. To change this information, close the wizard and select

File ► LDAP/ADS Configuration

If the server information is correct, then click **Next**.

- 3 If you selected **LDIF file** from the Introduction window, then the Server Import - LDIF File window opens.

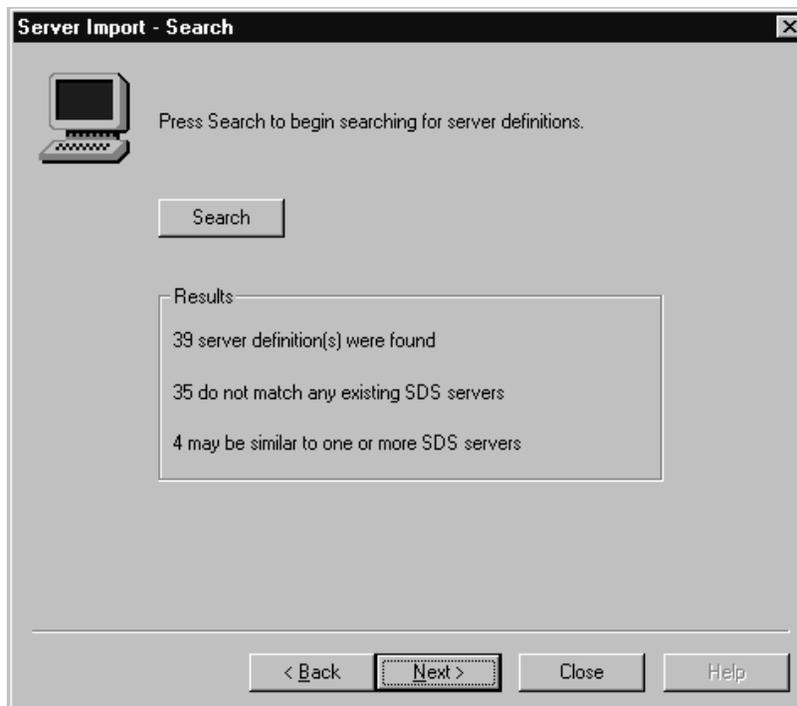


Type the complete pathname and filename of the LDIF file, and click **Next**.

- 4 After you verify the LDAP server information or enter the LDAP filename and click **Next**, the Server Import - Search window opens.



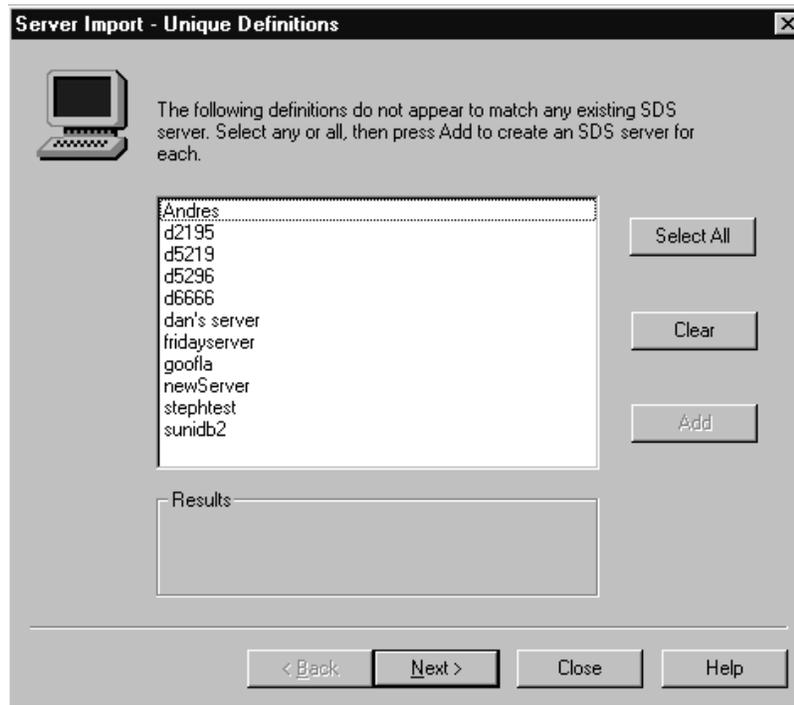
- 5 Click **Search** to begin checking the LDAP directory or LDIF file for server definitions.



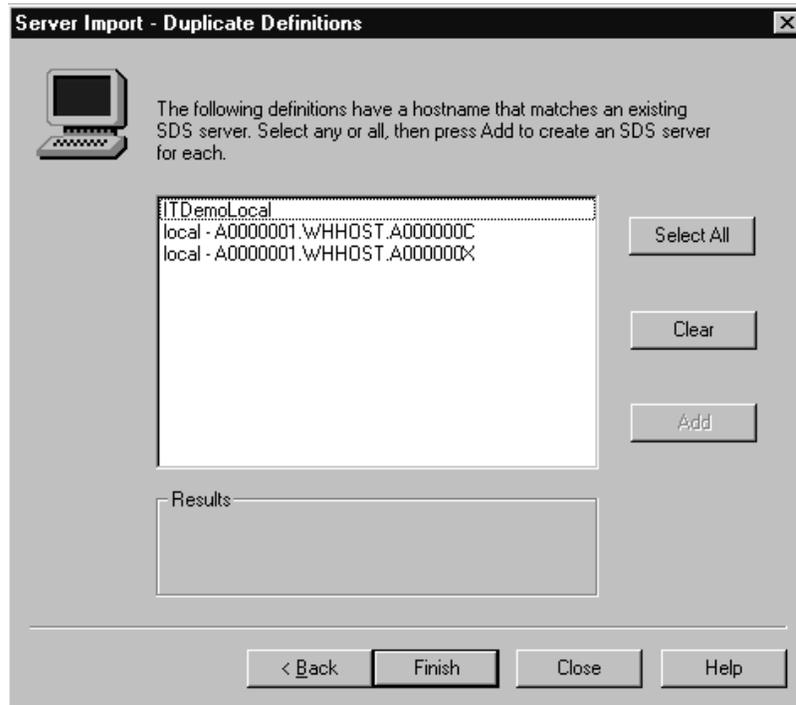
- 6 Enterprise Guide Administrator finds all of the server definitions in the LDAP or LDIF definitions and then compares the host names with the server definitions already in the SDS directory. The **Results** frame lists the number of servers that

do not duplicate existing server definitions as well as the number of duplicate definitions. Click **Next** to continue.

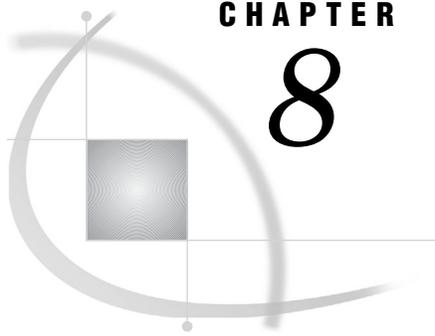
- 7 The Server Import - Unique Definitions window lists all of the server definitions whose host name does not match a definition in the SDS directory.



- 8 Select the servers whose definitions you want to add to the SDS directory by clicking on their names in the list. Then click **Add**. The **Results** frame confirms the success or failure of the process. Click **Next** to continue.
- 9 The Server Import - Duplicate Definitions window lists the server definitions whose host name matches a host that already exists in the SDS directory.



- 10 If you need to have different views of a particular server, then you can create a duplicate definition for the server. Select the servers for which you want to create duplicate definitions, and select **Add**. The **Results** frame confirms the success or failure of the process.
- 11 Click **Finish** to complete the import procedure.



CHAPTER

8

Examples of Creating Database Libraries

<i>Creating an Oracle Library</i>	115
<i>Configuring the Object Server</i>	115
<i>Adding the Library</i>	116
<i>Creating a DB2 Library</i>	117
<i>Configuring the Object Server</i>	117
<i>Adding the Library</i>	118

Creating an Oracle Library

To create an Oracle library on a UNIX server, you must provide additional information both when you configure the object server and when you define the library.

- When you configure the object server, first you must run a shell script that sets up your local environment variables for access to database libraries.
- When you create the administered library, you must add database connection options.

For additional information about accessing Oracle, refer to *SAS/ACCESS Software for Relational Databases: Reference, Version 8* (Oracle chapter) and to the SAS/ACCESS installation instructions for your operating environment.

Configuring the Object Server

Configure the object server as described in Chapter 4, “Configuring an IOM Bridge Connection,” on page 33 and Appendix 1, “Object Spawner Details,” on page 125. When you configure the object spawner (objspawn), you must add a shell script to set up your local environment variables and invoke SAS. The following sample UNIX shell script, OBJSPAWN.SETUP, defines some sample Oracle environment variables. You can name the script anything that you like. Note that if you use the following sample file, you will need to modify the pathnames in the file in order for the file to work at your site.

Note: The shell script must be defined as an executable file. Use the UNIX command **chmod +x** to do so. An example is **chmod +x OBJSPAWN.SETUP**. △

```
#
# Define environment variables required by Oracle
#
export Oracle_HOME=/usr/local/Oracle/8.0.4.0.0
export Oracle_SID=o8040
export TWO_TASK=node1_o8040
export PATH=$Oracle_HOME/bin:$PATH
```

```

export SHLIB_PATH=$SHLIB_PATH:$Oracle_HOME/lib:/usr/lib
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/usr/openwin/lib
#
# Execute SAS
#
exec $*

```

Ask your Oracle database administrator or your SAS Installation Representative to specify environment variables and values that are correct for your site. For example, these variables might already reside in a startup file called `.PROFILE`.

You can incorporate this shell script into the object spawner in different ways, depending on how you configure it:

- When you create an object spawner by using a configuration file, you create a file like the following sample file called `Oracle.CFG`. You will need to modify the pathnames in the file in order for the file to work at your site.

```

## Objspawn Definitions
dn: sasSpawnercn=testh8x,sascomponent=sasServer,cn=SAS
Institute,
    o=SAS Inst.,c=US
objectclass:sasSpawner
sasDomainName: unx.sas.com
sasMachineDNSName: taylor.unx.sas.com
sasOperatorPort: 6961
sasLogFile: /users/sascnn/sascnn1/mylog.log
sasVerbose:

## Server Definitions
dn: sasServercn=beta,sascomponent=sasServer,
    cn=SAS Institute,o=SAS Inst.,c=US
objectclass: sasServer
sasCommand: /users/sascnn/sasnn1/objspawn.setup
    /sas -config /sasv8.cfg
sasMachineDNSName: taylor.unx.sas.com
sasPort: 6960
sasProtocol: bridge
sasServercn: beta

```

Note: The `sasCommand` attribute has been modified to call the `OBJSPAWN.SETUP` shell script and to launch the SAS object server. \triangle

- The sample command to invoke the spawner with `Oracle.CFG` would appear as follows:

```

Prompt> <SASROOT>/utilities/bin/objspawn %sascommand
Oracle.cfg

```

Adding the Library

Create an administered library as described in “Managing Libraries” on page 81. To specify that you want to create an Oracle library, fill in the information in the Details dialog box as follows:

- Engine – Oracle
- DBMS connection – selected
- Path – blank

- Options – any Oracle engine connection and LIBNAME options that you want to specify for this library. For example, you could specify the following connection options in the **Options** box:
 - USER = *your Oracle user ID* (optional)
 - PASSWORD = *your Oracle password* (optional)
 - PATH = *the Oracle driver, node, and database* (optional)
 - REREAD_EXPOSURE = YES (required)

For additional information about Oracle LIBNAME statement options, refer to “SAS/ACCESS LIBNAME Statement” in *SAS/ACCESS Software for Relational Databases: Reference, Version 8*.

Creating a DB2 Library

To create a DB2 administered library on an OS/390 server, you must provide additional information both when you configure the object server and when you define the library.

- When you configure the object server, you must include the DB2 load libraries and specify the DB2SSID= option.
- When you create the administered library, you must specify the DB2 engine.

For additional information about accessing DB2, refer to *SAS/ACCESS Software for Relational Databases: Reference, Version 8* (DB2 chapter), and to the SAS/ACCESS installation instructions for your operating environment.

Configuring the Object Server

Configure the object server with the object spawner as described in Chapter 4, “Configuring an IOM Bridge Connection,” on page 33 and Appendix 1, “Object Spawner Details,” on page 125. Run the ATBSDFMU utility to define the transaction program profile. (An example of this is shown in “Object Spawner on OS/390” on page 130.) You must include the following steps:

- Be sure that the DB2 load libraries are included in the steplib concatenation if they are not included in the link list at your site.
- Specify the DB2SSID= option on the SAS server step to set up a default subsystem ID for all administered libraries. This option can be overridden by the SSID= connection option.
- If your site meets the requirements for RRSAF, then specify the DB2RRS invocation option on the SAS server step. For additional information about the DB2RRS option, refer to “Information for the Database Administrator” in *SAS/ACCESS Software for Relational Databases: Reference, Version 8* (DB2 chapter).

The following sample SAS step includes the two DB2 load libraries and specifies the DB2SSID= and DB2RRS options.

```
//SAS      EXEC SAS8,
//          LOADL=DBI.DB2V5R1.SDSNEXIT,
//          LOADT=DBI.DB2V5R1.SDSNLOAD,
//          OPTIONS='OBJECTSERVER, DB2SSID=DBV5, DB2RRS'
```

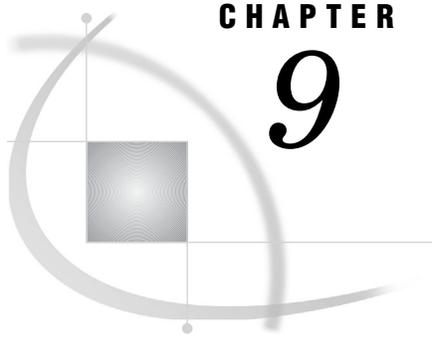
Adding the Library

Create an administered library as described in “Adding a Library” on page 82. To specify that you want to create a DB2 library, fill in the information in the Details dialog box as follows:

- Engine – DB2
- DBMS connection – selected
- Path – blank
- Options – any DB2 engine connection and LIBNAME options that you want to specify for this library. For example, you could specify the following connection options in the **Options** box:
 - SSID = your DB2 subsystem ID (optional)
 - SERVER = DRDA server (optional)
 - AUTHID= authid (optional)
 - CONNECTION= connection type (optional)

There are no required options for DB2.

For additional information about DB2 LIBNAME statement options, refer to “SAS/ACCESS LIBNAME Statement” in *SAS/ACCESS Software for Relational Databases: Reference, Version 8*.



CHAPTER

9

Administering Enterprise Guide Tasks

<i>Administering Tasks</i>	119
<i>Importing Task Definitions</i>	119
<i>Enabling Tasks for User Groups</i>	122
<i>Enabling Task Administration</i>	124

Administering Tasks

Beginning in Version 2.0, Enterprise Guide Administrator enables you to specify which tasks are available to each Enterprise Guide user group. You can use this capability to disable certain tasks for user groups that do not need to run those tasks. You can also increase users' efficiency by enabling only those tasks that they need to use. Task administration applies only to tasks that are supplied with Enterprise Guide.

The task administration process consists of three steps:

- 1 Importing tasks definitions
- 2 Selecting tasks to enable for each user group
- 3 Turning on task administration.

Importing Task Definitions

Task definitions are entries in an XML file that contain the information that Enterprise Guide Administrator needs in order to work with each task. You can administer only tasks whose definitions you have imported into Enterprise Guide Administrator.

To import task definitions:

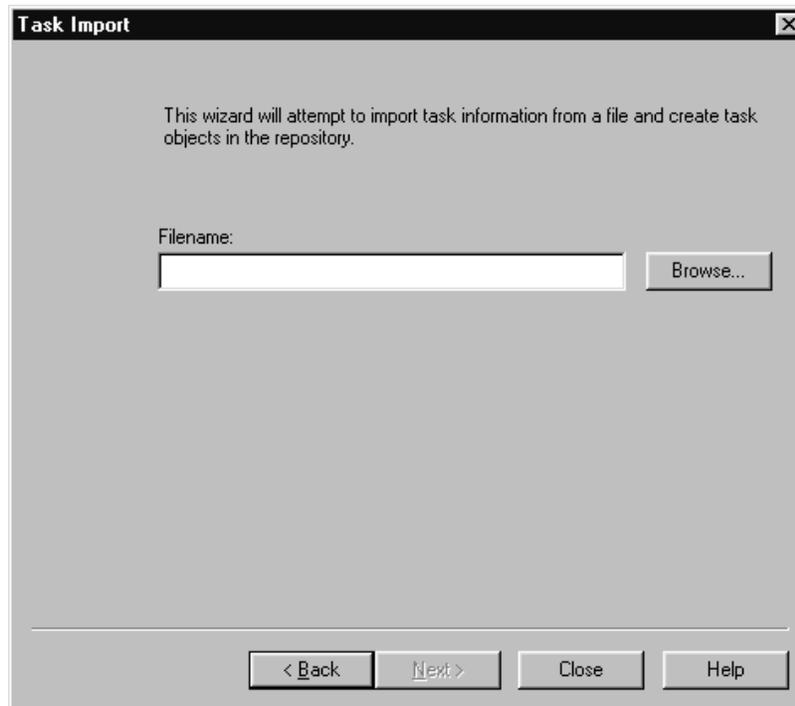
- 1 From a DOS command line or the Windows Run box, run the program `segtask.exe` to create the administration task information file. The program is located in the directory where Enterprise Guide is installed. The parameters for the program are

<code>/adm</code>	specifies that the program should create an administration task information file
<code>/f</code>	specifies the name of the administration task information file (default is <code>segtaskinfo.dat</code>)
<code>/l</code>	specifies the locale (default is 1033, which specifies English)

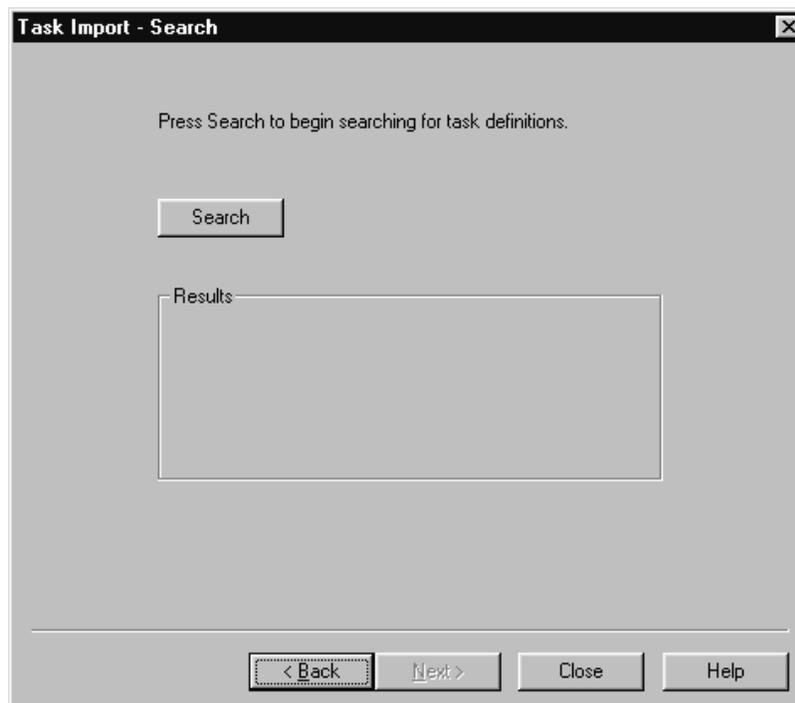
- 2 Select

Tools ► Task Import Wizard

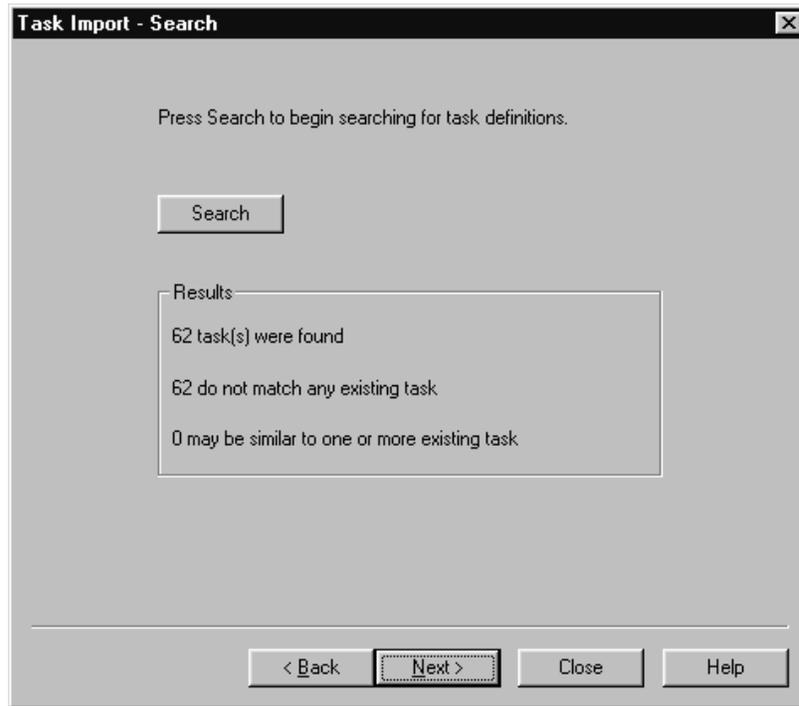
The Task Import window opens.



- 3 Type the filename of the XML file that contains the task definitions, or select **Browse** to choose the file interactively. Click **Next** to continue to the Task Import - Search window.
- 4 Click **Search** to examine the specified XML file for valid task definitions.

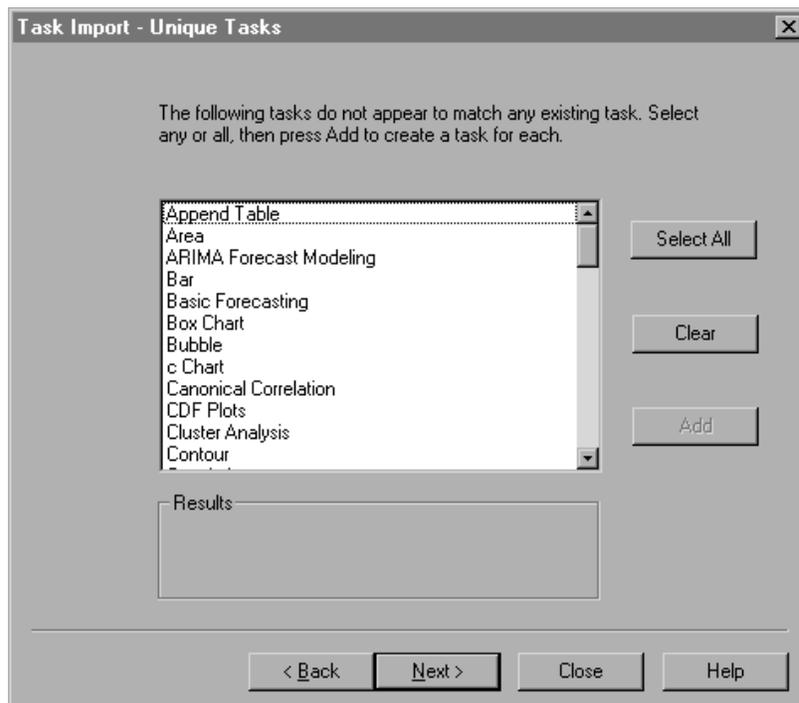


The window displays the number of task definitions that were found and the number that are identical or similar to task definitions that have already been imported.



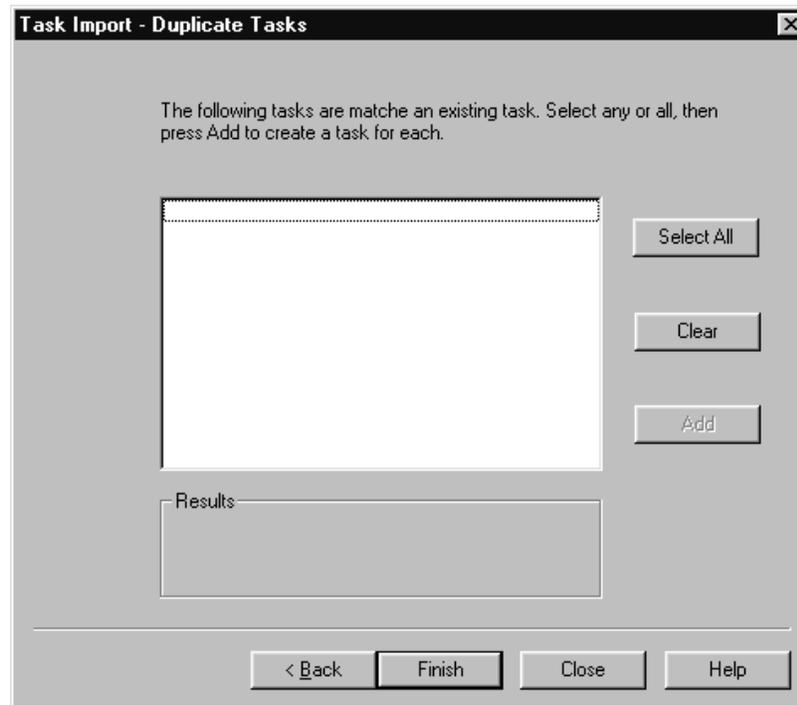
Click **Next** to continue to the Task Import - Unique Tasks window.

- The window lists the tasks in the file that you are importing that do not match previously imported tasks.



Select the tasks whose definitions you want to import (or click **Select All** to select all of the listed tasks), and click **Add**. The **Results** area lists the number of task definitions that were successfully created and the number of task definitions that failed. Click **Next** to continue to the Task Import - Duplicate Tasks window.

- 6 This window lists the tasks in the file that you are importing that match a previously imported task definition.



Select the tasks whose definitions you want to import. Importing a new task definition overwrites the existing definition. Click **Finish** to complete the process of importing task definitions.

Enabling Tasks for User Groups

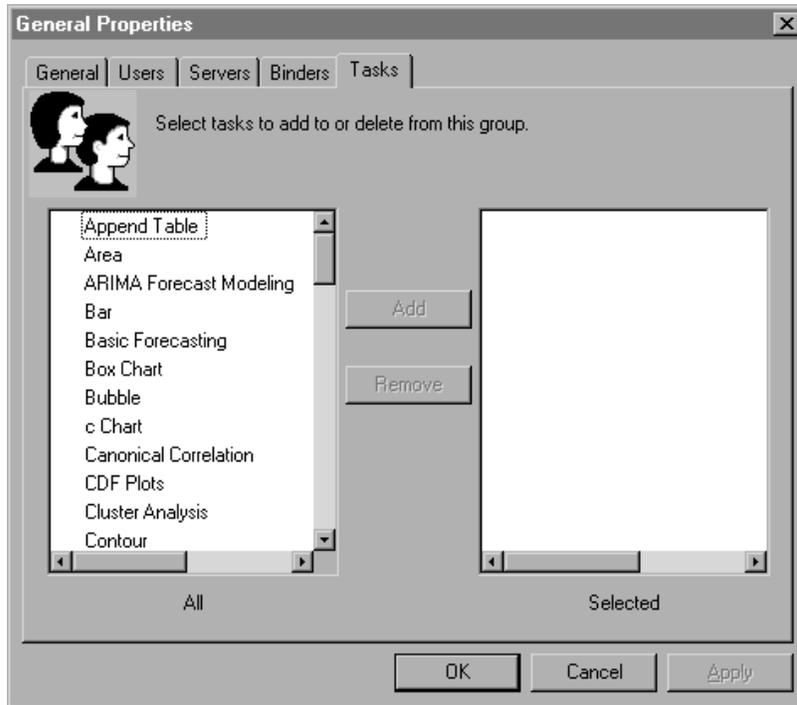
After you import the task definitions, you must determine which tasks each user group should have access to. To successfully implement task administration, every Enterprise Guide user must be assigned to a group, and you must enable a set of tasks for every group. A user who is not assigned to a group will not have access to any Enterprise Guide tasks.

To enable tasks for user groups:

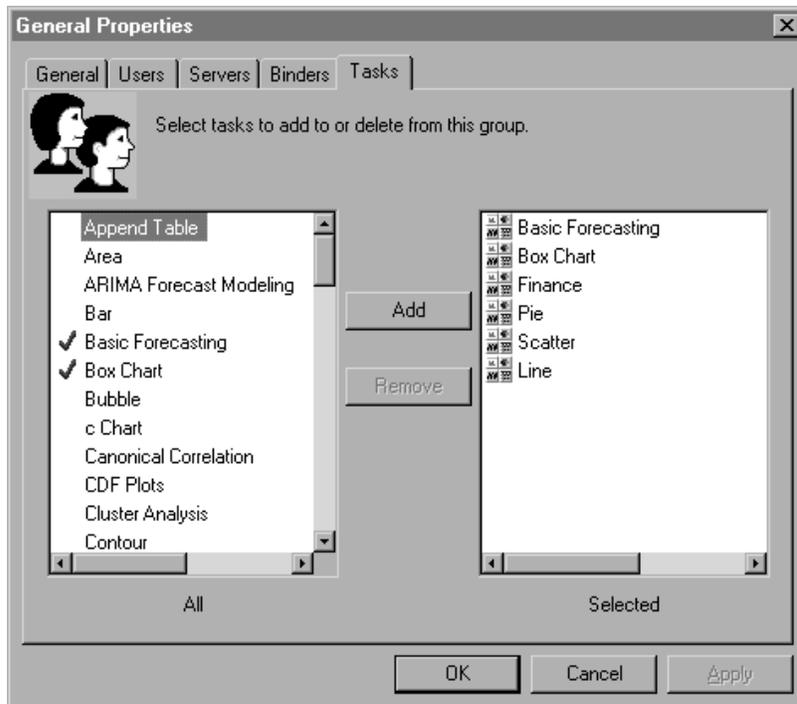
- 1 Ensure that all users have been assigned to a user group. For information about assigning users to a group, see “Managing Groups” on page 95.
- 2 Select a group whose tasks you want to enable. To open the Properties window for the group, select

File ► **Properties**

- 3 In the Properties window, select the **Tasks** tab.



- 4 The **All** box lists the Enterprise Guide tasks whose definitions have been imported into Enterprise Guide Administrator. Select the tasks that you want the members of the user group to have access to, and click **Add**. The tasks that you select are labeled with a check mark in the **All** box and are also listed in the **Selected** box.



- 5 Click **OK** to close the Properties window.

- 6 Repeat this selection process for all user groups. If you do not enable a set of tasks for a user group, then by default all tasks will be turned off for the members of that group.

Enabling Task Administration

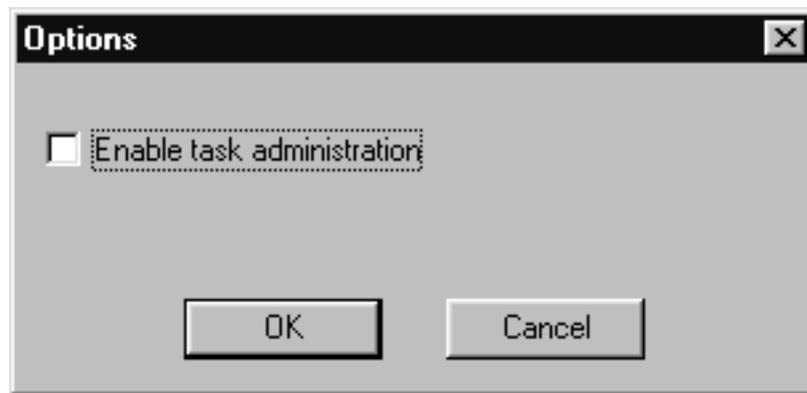
After you import the task definitions and enable tasks for all user groups, you must enable task administration in order for the settings to take effect.

To enable task administration:

- 1 Select

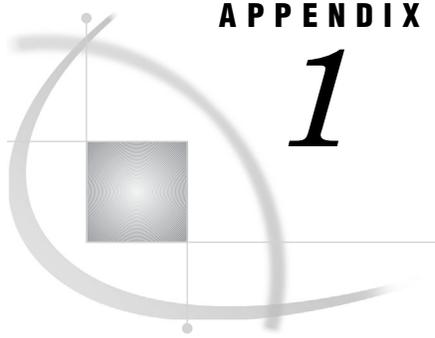


The Options window opens.



- 2 Select **Enable task administration** and click **OK** to close the window.

The task settings for each user group will take effect whenever a user starts an Enterprise Guide session. If users have already started an Enterprise Guide session when task administration is enabled, then the task settings will not take effect until they start a new session.



APPENDIX

1

Object Spawner Details

<i>Object Spawner Overview</i>	125
<i>Requirements</i>	126
<i>Hardware</i>	126
<i>Software</i>	126
<i>Information You Need before Configuring</i>	126
<i>Modifying the Services File</i>	126
<i>Example of Services File Entries</i>	127
<i>Creating the Spawner Configuration File</i>	127
<i>SasSpawner Object Classes</i>	128
<i>SasServer Object Classes</i>	129
<i>Setting Up and Invoking the Object Spawner</i>	130
<i>Object Spawner on OS/390</i>	130
<i>Setup</i>	130
<i>Invoking the Spawner on OS/390</i>	133
<i>Sample Configuration File for OS/390</i>	134
<i>Object Spawner on UNIX</i>	134
<i>Spawner Setup on UNIX</i>	134
<i>Invoking the Spawner on UNIX</i>	135
<i>Sample Configuration File for UNIX</i>	135
<i>Object Spawner on Windows NT</i>	136
<i>Spawner Setup on Windows NT</i>	136
<i>Invoking the Spawner on Windows NT</i>	136
<i>Sample Configuration File for Windows NT</i>	137
<i>Administering a Running Spawner</i>	138
<i>Error Messages</i>	139

Object Spawner Overview

The Enterprise Client suite employs SAS Integration Technologies software to provide the protocol for seamless application communication to data and servers. SAS Integration Technologies provides an object spawner that resides on the SAS server. Enterprise Client applications communicate with SAS software through an object spawner, which launches the SAS object server. The SAS object server is also called an IOM (Integrated Object Model) server. Setting up the object spawner requires that you create an object server definition and configure the object spawner to launch the server. This appendix documents how to create a simple object server definition and how to create an object spawner configuration file. Refer to the SAS Integration Technologies documentation for information about advanced features and updates to the configuration process. The documentation is provided in the Integration Technologies

installation directory (`<SASROOT>/inttech/sasmisc`), where `<SASROOT>` is the directory where SAS software is installed on the server machine.

Note: Enterprise Guide Administrator does not support configuration or administration of the object spawner, both of which are performed outside of Enterprise Guide Administrator. \triangle

Requirements

Hardware

The object spawner is installed on a server host that is running on several operating systems, including OS/390, Windows NT, and UNIX. See the installation documentation that comes with SAS Integration Technologies for specific operating systems.

Software

The following software must be installed on the SAS server:

- SAS Version 8 (or later)
- SAS Integration Technologies.

Information You Need before Configuring

SAS Integration Technologies was included as part of your SAS Version 8 installation. If the object server and spawner were configured at installation, then you must obtain the following information from the spawner configuration file in order to add the server in Enterprise Guide Administrator:

- server name or IP address
- number of the port on which the spawner will be listening for requests.

If you must create the server definition and configuration file, then follow these steps:

- 1 Determine the server name or IP address.
- 2 Determine the port number for your SAS server. (You might want to enter this information in your services file as described in “Modifying the Services File” on page 126.)
- 3 Determine the SAS invocation command that is used on the SAS server.
- 4 Create the spawner configuration file. Example configuration files are shown in “Setting Up and Invoking the Object Spawner” on page 130.
- 5 Invoke the spawner.

Modifying the Services File

The services file associates service names with their respective port numbers and protocols. The port numbers must be unique and will be used when you configure the object spawner.

Service Name(s) The TCP/IP service name identifies the port where the spawner listens for requests from users. The default service name is

`sasobjspawn`. Service names are defined in your installation's services file.

Port Number(s) The port number identifies the location where the spawner listens for requests from users. For a spawner that is supporting multiple object server definitions, a range of port numbers should be specified. Typical locations for the port numbers that are defined in the TCP/IP services file are shown in the following table.

TCP/IP Service File Locations

Operating System	TCP/IP Services location
UNIX	/etc/services
Windows NT	C:\WINNT\system32\drivers\etc\services
Windows 95/98	c:\windows\services
OS/390	sitespecific

For each service, a single line should be present with the following information:

```
<official service name> <port number/protocol name> <aliases>
```

Example of Services File Entries

In this example, the following lines allocate four ports — one operator port and three ports that the spawner will be monitoring for client requests. Each port number must be unique and must be greater than 5000. The name of the operator port is fixed; however, you must supply unique names for the listening ports.

```
sasOperatorPort 5306/tcp # SAS/Objspawn operator port
<sasobjspawn> 5307/tcp # SAS/Objspawn listen port
<sasobjspawn2> 5308/tcp # SAS/Objspawn listen port 2
<sasobjspawn3> 5309/tcp # SAS/Objspawn listen port 3
```

Creating the Spawner Configuration File

The object spawner configuration file uses the LDAP Data Interchange Format (LDIF). Here are the formatting guidelines for the configuration file.

- All comment lines begin with a # in column one.
- Each entry must start in column one.
- Entries are in the form attribute:value.
- Entries can span multiple lines by leaving column one blank in each subsequent line.
- A distinguished name (DN name) must be preceded by a blank line (excluding comment lines and the first distinguished name in the file). In LDIF, the DN is required to identify the beginning of the next object class definition. The LDIF parser in the spawner relies on this requirement in order to separate object class definitions. The DN name can be any value, unless it is a `sasMultiUserObject` attribute that points back to a `sasLogin` object class instance. In this case the specific DN name must be used.

- Two consecutive blank lines indicate the end of the configuration file definitions.

A configuration file must include the `sasSpawner` and `sasServer` object classes. Example configuration files are shown for each operating system in “Setting Up and Invoking the Object Spawner” on page 130.

SasSpawner Object Classes

The following table lists attributes (options) that you can include in a `sasSpawner` object definition.

SasSpawner Object Classes

SasSpawner Object Class	Description
ObjectClass (Required)	The object class identifier. For <code>sasSpawner</code> objects, this is always <code>sasSpawner</code> .
<code>sasSpawnercn</code> (Required)	The unique name of a particular <code>sasSpawner</code> object. When specified at object spawner invocation, its value identifies which <code>sasSpawner</code> definition to use.
<code>sasDomainName</code> (Optional)	The security domain in which the object spawner participates. <code>sasDomainName</code> and <code>sasMachineDNSName</code> are used to acquire the <code>sasServer</code> definitions that the object spawner should service. When specified, only <code>sasServer</code> definitions with the same <code>sasDomainName</code> are serviced by the object spawner instance. When not specified, only <code>sasServer</code> definitions without a <code>sasDomainName</code> are serviced by the object spawner instance.
<code>sasLogFile</code> (Optional)	A fully qualified path to the file in which to log object spawner activity.
<code>sasLUName</code> (Required)	<i>Only on OS/390.</i> Logical Unit Name for the object spawner instance.
<code>sasMachineDNSName</code> (Optional)	The DNS name(s) and IP address(es) for the machine(s) in which this object spawner definition can execute. This definition can be specified multiple times with different values. In addition, the string <code>localhost</code> can be used to signify the host in which the object spawner is executing.
<code>sasOperatorPassword</code> (Optional)	The password that must be entered when connecting to administer an object spawner. The default password is <code>sasobjspawn</code> .
<code>sasOperatorPort</code> (Optional)	The port in which to connect to administer an object spawner. If neither <code>sasOperatorPort</code> nor <code>sasOperatorService</code> is specified, the service name <code>sasobjoper</code> is used as the <code>sasOperatorService</code> . An object spawner will not start without an Administrator listen port/service.

SasSpawner Object Class	Description
sasOperatorService (Optional)	The service in which to connect in order to administer an object spawner. If neither sasOperatorPort nor sasOperatorService is specified, the service name sasobjoper is used as the sasOperatorService. An object spawner will not start without an Administrator listen port/service.
sasVerbose (Optional)	The attribute that, when present, causes the object spawner to record more detail in the log file (sasLogFile).
Description (Optional)	Text to summarize why this definition exists. This attribute is not used by the object spawner, but it is helpful to administrators who read the configuration file.

SasServer Object Classes

The sasServer object class identifies the startup and connection information for a SAS object server instance. The following table lists the attributes (options) that are used by the object spawner that exists in a sasServer object definition.

SasServer Object Classes

SasServer Object Class	Description
objectClass (Required)	The object class identifier. For sasServer objects, this is always sasServer.
sasCommand (Required)	The command that is used to launch the SAS object server. It is not used on OS/390.
sasMachineDNSName (Required)	The DNS name(s) for the machine(s) in which this SAS object server definition can execute. This definition can be specified multiple times with different values.
sasPluName (Required)	<i>Only on OS/390.</i> The Partner Logical Unit name that is used, in conjunction with the sasTPName, by the object spawner to launch the SAS object server.
sasProtocol (Required)	The protocol (one of bridge, com, corba) that clients can use for connection.
sasTpName (Required)	<i>Only on OS/390.</i> The Transaction Program name that is used, in conjunction with the sasPLUName, by the object spawner to launch the SAS object server.
sasServercn (Required)	The unique name of a particular sasServer object.
sasDomainName (Optional)	The security domain in which the sasServer definition participates. sasDomainName and sasMachineDNSName are used to acquire the sasServer definitions that the object spawner should service.

SasServer Object Class	Description
sasPort (Optional)	The port in which to launch/connect to this SAS object server. If neither sasPort nor sasService is specified, then the object spawner will attempt to use the service name sasobjspawn as the sasService. If sasobjspawn has already been used, then the object spawner will remove this sasService definition from its list.
sasService (Optional)	The service in which to launch/connect to this SAS object server. If neither sasPort nor sasService is specified, then the object spawner will attempt to use the service name sasobjspawn as the sasService. If sasobjspawn has already been used, then the object spawner will remove this sasService definition from its list.

Setting Up and Invoking the Object Spawner

You must start the object spawner on the server. Which command you use to invoke the object spawner depends on the host.

Note: If an object spawner is running and changes are made to the configuration file, then the object spawner must be stopped and restarted in order for the changes to take effect. \triangle

Refer to the appropriate topic for information about invoking the spawner on the desired host and for a list of options available at invocation.

OS/390 (see “Object Spawner on OS/390” on page 130)

UNIX (see “Setting Up and Invoking the Object Spawner” on page 130)

Windows NT (see “Object Spawner on Windows NT” on page 136)

Refer to “Administering a Running Spawner” on page 138 for information about how to verify that your object spawner is working properly.

Object Spawner on OS/390

Follow the steps below to invoke the object spawner on OS/390.

Setup

The spawner runs as a started task on OS/390. A started task procedure is used by the operator to start the spawner. An example of a started task procedure follows.

Note: The values that are displayed in the following example and throughout this section are for illustration only. You will need to change the values in order for the code to work at your site. \triangle

```
//*****
//***** JCL FOR OBJECT SERVER SPAWNER *****
//*****
//OBJSPAWN PROC ENTRY=OBJSPAWN,
// OPTIONS='CONFIGFILE //DSN:USER50.OS390.CFG'
```

```
// OPTIONS2=' '
//*****
//OBJSPAWN EXEC PGM=&ENTRY,REGION=4096K,
//      PARM='&OPTIONS&OPT2'
//STEPLIB DD DISP=SHR,DSN=MYID.SASOBSV.LOAD
//SYSUDUMP DD SYSOUT=D
//SYSOUT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSTEM DD SYSOUT=*
//SYSIN DD DUMMY
```

Any of the invocation options can be specified in the started task procedure (see “Invoking the Spawner on OS/390” on page 133). Also see “Sample Configuration File for OS/390” on page 134 for an example of how an OS/390 configuration file might look.

You need to perform the following additional steps to enable the APPC/MVS subsystem capabilities that are used by the OS/390 object spawner. Refer to *MVS/ESA Planning: APPC Management* (GC281110) for a detailed discussion of these procedures.

- 1 Define the VSAM transaction program profile data set to be run when an inbound connection is requested.

Run IDCAMS by using the following example to create a KSDS VSAM cluster to contain the transaction program profile that defines the object spawner transaction program.

```
DEFINE CLUSTER ( NAME ( vsam.profile.dataset )
                VOLUME ( volser )
                KEYS ( 112 0 )
                RECORDSIZE ( 248 4000 )
                RECORDS ( 100 100 )
                SHAREOPTIONS ( 3 3 )
                INDEXED REUSE )
DATA ( NAME ( vsam.profile.data ) )
INDEX ( NAME ( vsam.profile.index ) )
```

The member **ATBTPVSM** in the **SYS1.SAMPLIB** data set contains other examples for creating this type of data set.

- 2 Run the ATBSDFMU utility to define the transaction program profile.

Run the APPC/MVS administrative utility **ATBSDFMU** by using the following example to define the transaction program profile:

```
//UPDTPSI EXEC PGM=ATBSDFMU
//SYSPRINT DD SYSOUT=*
//SYSSDLIB DD DSN=vsam.profile.dataset,DISP=SHR
//SYSSDOUT DD SYSOUT=*
//SYSIN DD DATA,DLM=$$
TPDELETE
TPNAME(SASOBSJ)
SYSTEM
TPADD
TPNAME(SASOBSJ)
SYSTEM
ACTIVE(YES)
TPSCHED_DELIMITER(SCHEND)
TAILOR_SYSOUT(YES)
TAILOR_ACCOUNT(YES)
CLASS(XXX)
JCL_DELIMITER(JCLEND)
```

```

//SASOBS JOB (),'APPC/MVS SAS V8',MSGLEVEL=(1,1),MSGCLASS=X
//*
//PROCLIB JCLLIB ORDER=(SYS1.PROCLIB)
//*
//SAS      EXEC SAS8,

Object Spawner Details ###Appendix 2
82
//          LOADL=DBI.DB2V5R1.SDSNEXIT,
//          LOADT=DBI.DB2V5R1.SDSNLOAD,
//          OPTIONS='OBJECTSERVER'
//CONFIG   DD DISP=SHR,DSN=SAS8.SAS.CONFIG(TSO)
//          DD DISP=SHR,DSN=SAS8.SASOBSV.CONFIG
//SASCLOG  DD SYSOUT=*
//SASLOG   DD SYSOUT=*
//SASLIST  DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSIN    DD DUMMY
//*
JCLEND
      KEEP_MESSAGE_LOG(ERROR)
      MESSAGE_DATA_SET(&SYSUID.&SYSUID.&TPDATE.&TPTIME.JOBLOG)
      DATASET_STATUS(NEW)
SCHEND
$$
//
//

```

Note: The utility SYSIN input includes embedded JCL. The APPC/MVS transaction scheduler uses this JCL to invoke the transaction program SASOBS, which starts a SAS object server on OS/390 by executing a SAS8 cataloged procedure. You need to specify the DSN of the cataloged procedure library where your SAS8 procedure resides in the JCLLIB statement of this JCL. Δ

The concatenated configuration file, **SAS8.SASOBSV.CONFIG**, must contain the following statements:

```

OBJECTSERVER
OBJECTSERVERPARMS="PROTOCOL=BRIDGE SPAWNED"
NOTERMIAL

```

3 Define and activate a target logical unit name.

Create an **APPCPMxx** member in **SYS1.PARMLIB**, where **xx** is a two-character identifier that distinguishes the member that defines the target LU name in order to service object server spawner requests.

```

LUADD ACBNAME(C02IOMSV) TPDATA(vsam.profile.dataset)
      TPLEVEL(SYSTEM)

```

You also need to define a VTAM application node for this LU that is dedicated to APPC/MVS, as well as the LU that is used by the object spawner-started task. The following VTAM APPL definitions satisfy this requirement.

```

VBUILD TYPE=APPL
C02IOMSV APPL APPC=YES, +
          MODETAB=MODEAPPC, +
          DLOGMOD=MAPPC32, +
          DMINWNL=0, +

```

```

        DMINWNR=0, +
        DSESLIM=32767, +
        SECACPT=CONV
C02IOMSD APPL APPC=YES, +
        AUTOSES=0, +
        DDRAINL=NALLOW, +
        MODETAB=MODEAPPC, +
        DLOGMOD=MAPPC32, +
        DMINWNL=5, +
        DMINWNR=5, +
        DRESPL=NALLOW, +
        DSESLIM=10, +
        LMDENT=19, +
        PARSESS=YES, +
        SECACPT=CONV

```

Finally, you need to define the VTAM mode table and entries that are used in the VTAM APPL definitions. The following VTAM MODE definitions satisfy this requirement:

```

MODEAPPC TITLE 'MODE TABLE FOR APPC TESTING'
MODEAPPC MODETAB
MAPPC32 MODEENT LOGMODE=MAPPC32, APPC SESSIONS
        FMPROF=X'13',
        TSPROF=X'07',
        PRIPROT=X'B0',
        SECPROT=X'B0',
        COMPROT=X'50B1',
        RUSIZES=X'F8F8',
        PSERVIC=X'06020000000000000000102F00',
        TYPE=0
SNASVCMG MODEENT LOGMODE=SNASVCMG, APPC SESSIONS
        FMPROF=X'13',
        TSPROF=X'07',
        PRIPROT=X'B0',
        SECPROT=X'B0',
        COMPROT=X'50B1',
        RUSIZES=X'F8F8',
        PSERVIC=X'0602000000000000000000300',
        TYPE=0
        MODEEND
END

```

Invoking the Spawner on OS/390

To start the object spawner, enter the following operator command:

```
START OBJSPAWN
```

The following options are available when you invoke the spawner:

-configfile <pathname>

points to a fully qualified path to the object spawner configuration file. Example:

```
//dsn:myid.objspawn.log
```

-sasLogFile <pathname>

points to a fully qualified path to the file in which to log object spawner activity.

-sasSpawnercn <spawner definition name>

specifies which sasSpawnercn of the sasSpawner object to use for this object spawner invocation. When sasSpawnercn is not specified, the first sasSpawner definition with a sasMachineDNSName of the current host is used. The spawner definition can be any unique text string.

---sasVerbose

causes the object spawner to record more detail in the log file that is specified in

- sasLogFile

Sample Configuration File for OS/390

The following is a sample configuration file for OS/390, named OS390.CFG:

```
## Define MyApplication Object Spawner
#
dn: sasSpawnercn=production,sascomponent=sasServer,cn=ABC,
   o=ABC Inc,c=US
objectClass: sasSpawner
sasSpawnercn: production
sasDomainName: mvs.abc.com
sasLuName: C03ABCSD
sasMachineDNSName: bigiron.mvs.abc.com
sasOperatorPort: 6340
sasOperatorPassword: myPassword
description: Production MyApplication Object Spawner Daemon
#
## Define MyApplication SAS Object Server
#
dn: sasServercn=MyApplication,sascomponent=sasServer,cn=ABC,
   o=ABC Inc,c=US
objectClass: sasServer
sasServercn: MyApplication

Appendix 2 ###Object Spawner Details 85
sasDomainName: mvs.abc.com
sasMachineDNSName: bigiron.mvs.abc.com
sasPluName: C03ABCSD
sasPort: 6341
sasProtocol: bridge
sasTpName: SASOBS
```

Object Spawner on UNIX

Follow the instructions below to invoke the object spawner on UNIX.

Spawner Setup on UNIX

If the object spawner is to service *one client*, then the spawner can be launched under that client's user ID. If the object spawner is to service *more than one client* user ID, then the spawner must be launched with root privileges. In addition, the SAS process is launched in the client's home directory (as specified in the client's password entry).

Note: If the object spawner is to service more than one client user ID, then root privileges are required in order for the object spawner to fork a process with the connecting client as the owner. Δ

Note: If you are printing or using SAS/GRAPH procedures, then you must set the DISPLAY environment variable to a running X server. Here is an example:

```
export DISPLAY=[machine name]:0.0
```

Δ

Invoking the Spawner on UNIX

You invoke spawner from a directory in which SAS software is installed.

The following sample command invokes the spawner by using <SASROOT> as the directory in which SAS software is installed:

```
Prompt><SASROOT>/utilities/bin/objspawn
```

The following options are available when you invoke the spawner:

-configFile <pathname>

points to a fully qualified path to the object spawner configuration file. Paths that contain blanks must be enclosed in quotation marks. The following example command invokes the spawner by using myspawner.cfg as the configuration filename:

```
Prompt> <SASROOT>/utilities/bin/objspawn
--configFile myspawner.cfg
```

See “Sample Configuration File for UNIX” on page 135 for an example of how a UNIX configuration file might look.

-authProg <pathname>

points to a fully qualified path to the executable file for use in authentication. Paths that contain blanks must be enclosed in quotation marks.

sasLogFile <pathname>

points to a fully qualified path to the file in which to log object spawner activity. Paths that contain blanks must be enclosed in quotation marks.

sasSpawnercn <spawner definition name>

specifies which sasSpawnercn of the sasSpawner object to use for this object spawner invocation. When sasSpawnercn is not specified, the first sasSpawner definition with a sasMachineDNSName of the current host is used. The spawner definition can be any unique text string.

sasVerbose

causes the object spawner to record more detail in the log file that is specified in sasLogFile.

Sample Configuration File for UNIX

This is a sample configuration file for UNIX called MYSPAWNER.CFG:

```
## Objspawn Definitions
dn: sasSpawnercn=testh8x,sascomponent=sasServer,cn=SAS
Institute,
  o=SAS Inst.,c=US
objectclass:sasSpawner
sasDomainName: unx.sas.com
```

```

sasMachineDNSName: grommit.unx.sas.com
sasOperatorPort: 6961
sasLogFile: /users/sascnn/sascnn1/mylog.log
sasVerbose:
## Server Definitions
dn: sasServercn=beta,sascomponent=sasServer,
   cn=SAS Institute,o=SAS Inst.,c=US
objectclass: sasServer
sasCommand: /users/sascnn/sasnn1/oracle.setup
           /sas --config /sasv8.cfg
sasMachineDNSName: grommit.unx.sas.com
sasPort: 6960
sasProtocol: bridge
sasServercn: X39J

```

Object Spawner on Windows NT

Follow the steps below to invoke the object spawner on Windows NT.

Spawner Setup on Windows NT

The user who invokes the spawner, in addition to being an administrator, must have the proper user rights. To set the user rights on a Windows NT server machine, follow these steps:

- 1 Select



- 2 Select



- 3 Select **Show Advanced User Rights**.

- 4 Add rights by using the **Right** pull-down menu.

- 5 Select these user rights:

- act as part of the operating system**
- increase quotas**
- replace the process level token.**

- 6 The proper user rights also need to be set for each client that connects to the object spawner. You must include the domain name for clients that connect to the object spawner (DOMAIN\userid). Follow the previous steps to add the user right “logon as batch machine” for each client machine.

Invoking the Spawner on Windows NT

The object spawner is installed in the **inttech\sasexe** folder that is found in your installed SAS folder. The following examples use **C:\<SASROOT>** as the installed SAS folder.

```
C:\><SASROOT>\inttech\sasexe\objspawn.exe
```

The following options are available when you invoke the spawner:

```
-configFile <pathname>
```

points to a fully qualified path to the object spawner configuration file. Paths that contain blanks must be enclosed in quotation marks. The following sample command invokes the spawner by using the invocation and server options that are specified in the configuration file:

```
c:\objspawn.cfg.
c:\><SASROOT>\inttech\sasexe\objspawn.exe
--configFile spawnnt.cfg
```

See “Sample Configuration File for Windows NT” on page 137 for an example of how a Windows NT configuration file might look.

-install

instructs the object spawner to install as a Windows NT service. When asked to install as an NT service, the object spawner records all options that are specified at install time in the registry under the following key:

```
System\CurrentControlSet\Services\
SAS Object Spawner Daemon\Parameters
```

The following sample command installs the object spawner as a Windows NT service and updates the registry to hold the options that are specified (in this case configFile). The command should be typed on one line:

```
c:\><SASROOT>inttech\sasexe\objspawn.exe
configFile spawnnt.cfg -install
```

When the object spawner is started as an NT Service, it will configure itself by using the options that are placed in the registry at install time.

deinstall

instructs the object spawner to deinstall as a Windows NT service.

The following command deinstalls the object spawner as an NT service:

```
c:\><SASROOT>\inttech\sasexe\objspawn.exe --deinstall
```

-sasLogFile <pathname>

points to a fully qualified path to the file in which to log object spawner activity. Paths that contain blanks must be enclosed in quotation marks.

sasSpawnercn <spawner definition name>

specifies which sasSpawnercn of the sasSpawner object to use for this object spawner invocation. When sasSpawnercn is not specified, the first sasSpawner definition with a sasMachineDNSName of the current host is used. The spawner definition can be any unique text string.

-sasVerbose

causes the object spawner to record more detail in the log file that is specified in sasLogFile.

Sample Configuration File for Windows NT

Following is a sample configuration file for Windows NT called SPAWNNT.CFG:

```
dn: sasSpawnercn=testpc
objectclass: sasSpawner
sasspawnercn: testpc
sasdomainname: carynt
#saslogfile: c:\temp\obj.log
sasmachinednsname: bigpc.wnt.abc.com
sasmachinednsname: mediumpc.wnt.abc.com
```

```

sasoperatorport: 6340
#sasverbose: true
description: Joe's PC spawner
dn: sasServercn=PCs
objectclass: sasServer
sasservercn: PCs
sascommand: "c:\program files\sas institute\sas\sas"

config "c:\program files\sas institute\sas\sasv8.cfg"
description: SAS PC servers
sasmachinednsname: bigpc.wnt.abc.com
sasmachinednsname: mediumpc.wnt.abc.com
sasprotocol: bridge
sasport: 6338
sasdomainname: wilder

```

Administering a Running Spawner

To use a telnet client to control and monitor the object spawner daemon:

- 1 Start a telnet session to connect to the administrator interface port/service (**sasOperatorPort/sasOperatorService**) that is specified in the sasSpawner definition. (For information about service names, see “Example of Services File Entries” on page 127.)

The following example on UNIX assumes that 6337 was specified as the **sasOperatorPort**.

```

myhost> telnet serverhost 6337
Trying...
Connected to serverhost.
Escape character is '^]'.

```

If neither **sasOperatorPort** nor **sasOperatorService** is specified in the configuration file, then use the default operator service name (sasobjoper) in the following command:

```

myhost> telnet serverhost sasobjoper

```

- 2 After the telnet conversation is active, enter the password that is specified in sasOperatorPassword. Note that when you use the telnet utility on Windows, you might not be prompted for a password, and your input might not be displayed on the screen. Here is an example of your session:

```

sasobjspawn
Operator conversation established

```

- 3 You can now interact with the executing daemon by issuing any of these commands:

Object Spawner Commands

Command	Description
bye	Terminate the object spawner.
btrace <filename>	Begin trace. <i>Filename</i> is a fully qualified path to the file in which to log object spawner activity. On OS/390, <filename> must be specified as a data set name (for example, //dsn:myid.objspawn.log).
etrace	End trace.
help	List available administrator commands.
kill <pid> or all	<i>Only on Windows NT.</i> You must specify whether you want to kill the process that is launched by this object spawner daemon (identified by the pid), or kill all processes that are launched by this object spawner daemon.
lista	List sasServer definitions in which the object spawner daemon is currently listening for requests.
listi	List sasServer definitions in which the object spawner daemon is <i>not</i> currently listening for requests.
list	List all known sasServers that are supported by this object spawner daemon. This is a shortcut for lista , followed by listi .
Listl	<i>Only on Windows NT.</i> List the pid for active SAS object servers that are launched by this object spawner daemon.
Quit	Exit object spawner administrator session.
Start	Attempt to start idle services.

Error Messages

Following are error messages that clients might receive and explanations of how to correct the errors. If after reading the explanation and following the suggestions you are still unable to correct an error, then you might want to have the object spawner daemon begin tracing its activity (see “Administering a Running Spawner” on page 138) or use the `sasVerbose` and `sasLogFile` options (see “SasSpawner Object Classes” on page 128).

ERROR: Unable to allocate cmd memory

The object spawner daemon has exhausted its resources. Review the list of started object servers (see the **listl** and **kill** administrator commands) to determine if there are any orphaned SAS object servers. If so, kill the orphaned SAS object servers.

ERROR: Unable to CreateThread

The object spawner daemon has exhausted its resources. Review the list of started object servers (see the **listl** and **kill** administrator commands) to determine if there are any orphaned SAS object servers. If so, kill the orphaned SAS object servers.

ERROR: Unable to launch SAS ObjectServer (AddToDefault failure)
The object spawner daemon has exhausted its resources. Review the list of started object servers (see the **list1** and **kill** administrator commands) to determine if there are any orphaned SAS object servers. If so, kill the orphaned SAS object servers.

ERROR: Unable to launch SAS ObjectServer (AddToDesktop failure)
The object spawner daemon has exhausted its resources. Review the list of started object servers (see the **list1** and **kill** administrator commands) to determine if there are any orphaned SAS object servers. If so, kill the orphaned SAS object servers.

ERROR: Unable to launch SAS ObjectServer (Allocation of Environment block failure)
The object spawner daemon has exhausted its resources. Review the list of started object servers (see the **list1** and **kill** administrator commands) to determine if there are any orphaned SAS object servers. If so, kill the orphaned SAS object servers.

ERROR: Unable to launch SAS ObjectServer (Child allocation failure)
The object spawner daemon has exhausted its resources. Review the list of started object servers (see the **list1** and **kill** administrator commands) to determine if there are any orphaned SAS object servers. If so, kill the orphaned SAS object servers.

ERROR: Unable to launch SAS ObjectServer (Child CreateEvent failure)
The object spawner daemon has exhausted its resources. Review the list of started object servers (see the **list1** and **kill** administrator commands) to determine if there are any orphaned SAS object servers. If so, kill the orphaned SAS object servers.

ERROR: Unable to launch SAS ObjectServer (Child DuplicateHandle failure)
The object spawner daemon has exhausted its resources. Review the list of started object servers (see the **list1** and **kill** administrator commands) to determine if there are any orphaned SAS object servers. If so, kill the orphaned SAS object servers.

ERROR: Unable to launch SAS ObjectServer (Child username allocation failure)
The object spawner daemon has exhausted its resources. Review the list of started object servers (see the **list1** and **kill** administrator commands) to determine if there are any orphaned SAS object servers. If so, kill the orphaned SAS object servers.

ERROR: Unable to launch SAS ObjectServer (Conversion of environment block failure)
The object spawner daemon has exhausted its resources. Review the list of started object servers (see the **list1** and **kill** administrator commands) to determine if there are any orphaned SAS object servers. If so, kill the orphaned SAS object servers.

ERROR: Unable to launch SAS ObjectServer (CreateEnvironmentBlock failure)
The object spawner daemon has exhausted its resources. Review the list of started object servers (see the **list1** and **kill** administrator commands) to determine if there are any orphaned SAS object servers. If so, kill the orphaned SAS object servers.

ERROR: Unable to launch SAS ObjectServer (CreateProcess failure)

The object spawner daemon has exhausted its resources. Review the list of started object servers (see the **list1** and **kill** administrator commands) to determine if there are any orphaned SAS object servers. If so, kill the orphaned SAS object servers.

ERROR: Unable to launch SAS ObjectServer (CreateProcessAsUser failure)
The object spawner daemon has exhausted its resources. Review the list of started object servers (see the **list1** and **kill** administrator commands) to determine if there are any orphaned SAS object servers. If so, kill the orphaned SAS object servers.

ERROR: Unable to launch SAS ObjectServer (exec failure)
Ensure that the sasCommand value specified is correct. Log on to the host in which the object spawner daemon is executing. Issue the sasCommand to determine if it is correct.

ERROR: Unable to launch SAS ObjectServer (fork failure)
Ensure that the host in which the object spawner daemon is executing has not overflowed the process table. If so, you might want to stop the object spawner daemon and restart it to clean up possible orphaned processes.

ERROR: Unable to launch SAS ObjectServer (GetUserProfilePathDirectory failure)
The object spawner daemon has exhausted its resources. Review the list of started object servers (see the **list1** and **kill** administrator commands) to determine if there are any orphaned SAS object servers. If so, kill the orphaned SAS object servers.

ERROR: Unable to launch SAS ObjectServer (LoadUserProfileA failure)
The object spawner daemon has exhausted its resources. Review the list of started object servers (see the **list1** and **kill** administrator commands) to determine if there are any orphaned SAS object servers. If so, kill the orphaned SAS object servers.

ERROR: Unable to launch SAS ObjectServer (putenv conv)
Ensure that the object spawner daemon is executing in a process that is large enough to support the number of servers that are defined and executing.

ERROR: Unable to launch SAS ObjectServer (putenv sid)
Ensure that the object spawner daemon is executing in a process that is large enough to support the number of servers that are defined and executing.

ERROR: Unable to launch SAS ObjectServer (putenv sock)
Ensure that the object spawner daemon is executing in a process that is large enough to support the number of servers that are defined and executing.

ERROR: Unable to launch SAS ObjectServer (too many tokens in command)
Place your sasCommand in an executable shell. Then update your sasCommand to invoke the shell. Note: The shell must accept arguments and pass the arguments that are received on to the SAS object server.

ERROR: Unable to launch SAS ObjectServer (unable to allocate command token)
Ensure that the object spawner daemon is executing in a process that is large enough to support the number of servers that are defined and executing.

ERROR: User does not have authority
Ensure that the user name and password that are specified are correct and have not expired. If the user name and password are correct, then ensure that the object spawner daemon is executing with the privileges that are specified in the appropriate host setup section in this document.

OBJSPAWN(E): ATBALC2 returned %d, ATBEES3 (%*.*s)

Ensure that the sasPLUName and sasTPName that are provided in the sasServer definition are correct.

OBJSPAWN(E): ATBRCVW returned %d, ATBEES3 (%*.*s)

The object spawner daemon encountered an APPC error. Ask your systems administrator to verify that APPC is operating correctly. Also, ensure that the object server configuration/definition is correct.

OBJSPAWN(E): ATBRCVW statRcvd not correct %d

The object spawner daemon encountered an APPC error. Ask your systems administrator to verify that APPC is operating correctly. Also, ensure that the object server configuration/definition is correct.

OBJSPAWN(E): ATBRCVW(2) returned %d, ATBEES3 (%*.*s)

The object spawner daemon encountered an APPC error. Ask your systems administrator to verify that APPC is operating correctly. Also, ensure that the object server configuration/definition is correct.

OBJSPAWN(E): ATBRCVW(2) statRcvd not correct %d

The object spawner daemon encountered an APPC error. Ask your systems administrator to verify that APPC is operating correctly. Also, ensure that the object server configuration/definition is correct.

OBJSPAWN(E): ATBRTS returned %d, ATBEES3 (%*.*s)

The object spawner daemon encountered an APPC error. Ask your systems administrator to verify that APPC is operating correctly. Also, ensure that the object server configuration/definition is correct.

OBJSPAWN(E): ATBRTS(2) returned %d, ATBEES3 (%*.*s)

The object spawner daemon encountered an APPC error. Ask your systems administrator to verify that APPC is operating correctly. Also, ensure that the object server configuration/definition is correct.

OBJSPAWN(E): ATBSEND returned %d, ATBEES3 (%*.*s)

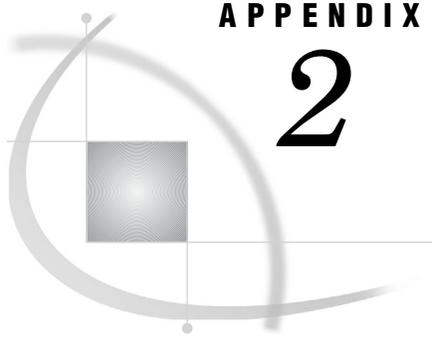
The object spawner daemon encountered an APPC error. Ask your systems administrator to verify that APPC is operating correctly. Also, ensure that the object server configuration/definition is correct.

OBJSPAWN(E): ATBSEND(2) returned %d, ATBEES3 (%*.*s)

The object spawner daemon encountered an APPC error. Ask your systems administrator to verify that APPC is operating correctly. Also, ensure that the object server configuration/definition is correct.

OBJSPAWN(E): givesocket() returned %d

The object spawner daemon failed to hand off the client connection socket. Ensure that TCP/IP is operating correctly.



APPENDIX

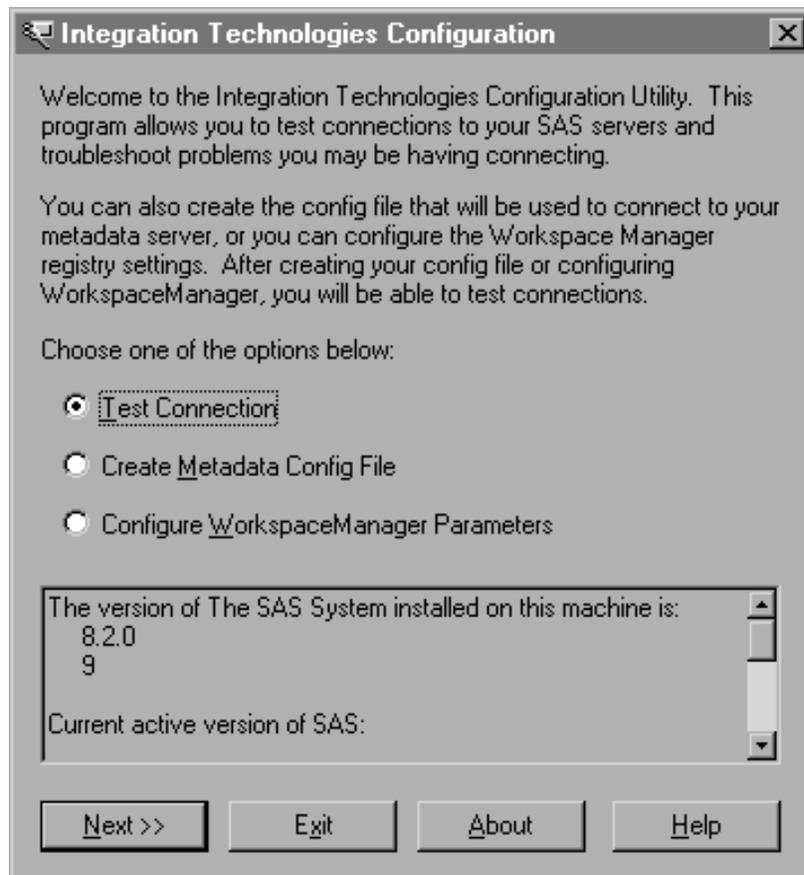
2

Testing Server Connections with Integration Technologies Configuration

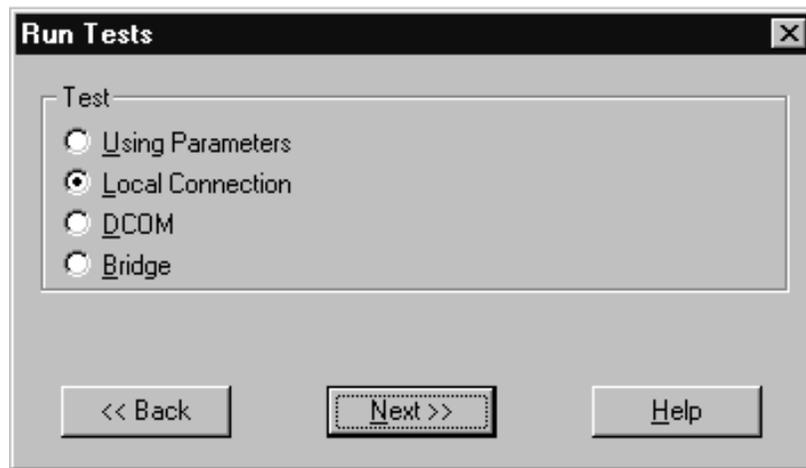
Integration Technologies Configuration is a stand-alone application that is delivered with SAS Integration Technologies. You can use this application to test connections to a SAS server without having to use the Integration Technologies Administrator application.

To test a server connection:

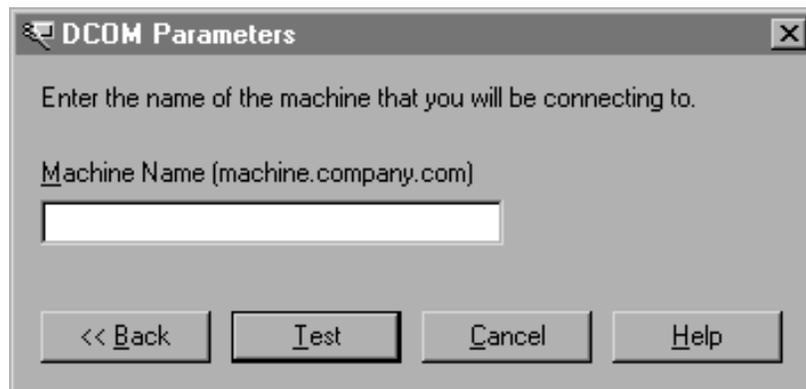
- 1 Locate and execute the file `itconfig.exe`. The default location for the file is **C:\Program Files\SAS Institute\Shared Files\Integration Technologies**.
- 2 When the program starts, it checks the Windows program registry for unused Integration Technologies entries. If any unused entries are found, then the application gives you the option of removing the entries. After the entries are checked, the Integration Technologies Configuration window opens.



- 3 Select **Test Connection** and click **Next**. The Run Tests window opens.



- 4 If you are connecting to a local server, then select **Local Connection** and click **Next**. If you are using the DCOM protocol for the connection to the server, then select **DCOM**, click **Next**, and go to step 6. If you are using the IOM Bridge protocol for the connection to the server, then click **Next** and go to step 9.
- 5 If you selected **Local Connection**, then the program tests the connection and displays a window that indicates whether or not the test was successful.
- 6 If you selected DCOM, then the DCOM Parameters window opens.



- 7 Type the computer name of the SAS server in the **Machine Name** field and click **Test**. The value that you type in this field is the same value that you type in the Server - Login window in Enterprise Guide Administrator. See “Adding SAS Servers” on page 73.
- 8 When the test is completed, a window opens and indicates whether or not the test was successful.
- 9 If you selected **Bridge** in the Run Tests window, then the Bridge Parameters window opens.

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)
[Text Box]

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 [Text Box]

Login Information
Enter a valid username and password for your server.

Username [Text Box]
Password [Text Box]

<< Back Test Cancel Help

- 10 Type the DNS name (for example, NEVADA.PC.ABC.COM) or IP address (for example, 12.34.5.678) in the **Machine Name** field. Select the **Port** radio button, and then type the port number in the **Port** field. Type a valid user ID and password for the machine in the appropriate fields, and click **Test**.
- 11 The application tests the connection and displays a window that indicates whether or not the test was successful.

Index

A

administering tasks
See task administration
 administrator account 7

B

binders 90
 adding 90
 adding objects to 94
 deleting 94
 managing 90
 mapped to native files 90
 mapped to server locations 90
 shortcuts in 94

C

Citrix server 7
 creating administrator account 7
 editing the registry 9
 setting DCOM permissions 8
 setting Sdslocal file permissions 8
 user for repository process 9

client groups 49
 client users 38
 clients 72
 cubes
 registering 27

D

database libraries
 DB2 117
 Oracle 115
 DB2 library
 adding the library 118
 configuring the object server 117
 creating 117
 DCOM connections, configuring 47
 client groups 49
 enabling DCOM on client and server 48
 launch policies 50
 requirements 47
 DCOM permissions 8

DCOM servers
 configuring DCOM connections 47

E

EGTASK library 88
 engines 84
 Enterprise Clients 2
 installing 5
 setting up 13
 Enterprise Guide 2
 installing on Citrix server 7
 silent setup program 5
 Enterprise Guide Administrator 1
 setup 3
 uninstalling 9
 Enterprise Reporter 2
 error messages
 object spawner 139

F

formats
 adding 104
 defining new formats 105

G

groups 95
 adding 95
 deleting 98
 managing 95
 updating properties 98
 user profiles compared with 95

H

Help 3

I

importing
 server definitions 108
 task definitions 119

installation
 customized defaults 61
 Enterprise Clients 5
 Enterprise Guide, on Citrix server 7
 Enterprise Guide, with silent setup 5
 object spawner 44
 recorded installations 6
 SAS servers 10
 troubleshooting 12
 verifying 12
 Integration Technologies Configuration
 testing server connections 143
 IOM Bridge connection 33
 how it works 33
 object spawner setup 36

L

launch policies 50
 for SAS 54
 global 50
 LDAP directories 108
 LDIF files 108
 libraries 81
 adding 82
 DB2 database library 117
 defining 82
 EGTASK library, creating 88
 managing 81
 Oracle database library 115
 unassigning 89
 updating properties 88
 librefs 84
 listen port 33

M

MDDBs
 registering 27
 metadata repository
See repositories
 MVS servers 78

N

native files
 binders mapped to 90

networked environment
 repositories in 61
 setup image for users 61

O

object definitions 13
 object server
 configuring 10
 configuring, DB2 117
 configuring, Oracle 115
 object spawner 33
 administering a running spawner 138
 client users 38
 configuration file 39, 126, 127
 error messages 139
 hardware requirements 126
 installing 44
 invoking 130
 login setup 36
 OS/390 130
 overview 125
 ports 39, 126
 production environment setup 46
 sasServer object classes 129
 sasSpawner object classes 128
 services file 126
 setting up 36, 130
 software requirements 126
 starting 44
 testing the server definition 46
 UNIX 45, 134
 user rights 36
 Windows 44
 Windows NT 136
 objects
 adding to binders 94
 viewing 70
 options.ini file 61
 Oracle library
 adding the library 116
 configuring the object server 115
 creating 115
 OS/390
 creating DB2 library 117
 manually assigning/unassigning libraries 89
 object spawner 130
 object spawner, configuration file 134
 object spawner, invoking 133
 object spawner, setup 130

P

permissions
 DCOM permissions 8
 Sdslocal file permissions 8
 playing back installations 6
 ports 39, 126
 properties
 group properties 98
 library properties 88
 SAS server properties 81
 user properties 103

R

recording installations 6
 registry
 editing for Citrix server 9
 repositories 13
 backing up 9
 changing 69
 compacting 27
 description 13
 in networked environment 61
 local process 14
 maintaining 25
 path changes 26
 populating 18
 preserving during upgrades 9
 registering MDDBs 27
 remote process 15
 security 18
 setting up 13
 taking shared repositories offline 26
 upgrading the schema 10
 working with 17
 repository process
 running locally 14
 running on remote server 15
 shared access to 18
 user specification for 9

S

SAS clients
 configuring DCOM connections 48
 SAS Directory Services (SDS) 15
 SAS Integration Technologies 125, 143
 SAS servers 72
 adding 73
 installing 10
 updating properties 81
 sasServer object classes 129
 sasSpawner object classes 128
 schema
 manually upgrading 10
 SDS (SAS Directory Services) 15
 Sdslocal file permissions 8
 security
 for repositories 18
 server connections
 testing 143
 server definitions
 importing 108
 testing 46
 servers
 binders mapped to server locations 90
 Citrix server 7
 importing server definitions 108
 managing 72
 MVS 78
 OS/390 117
 SAS servers 10, 72
 testing connections 143
 testing definitions 46
 UNIX 115

services file 126
 example of entries 127
 setup
 Enterprise Clients 13
 Enterprise Guide Administrator 3
 object spawner 36, 130
 repositories 13
 setup image 61
 silent setup program 5
 playing back installations 6
 recording installations 6
 troubleshooting 6
 spawner
 See object spawner

T

task administration 119
 enabling 124
 enabling tasks for user groups 122
 importing task definitions 119
 task definitions
 importing 119
 troubleshooting
 installation 12
 object spawner 139
 silent setup program 6

U

uninstalling
 preserving the repository 9
 UNIX
 creating Oracle library 115
 object spawner 45, 134
 object spawner, configuration file 135
 object spawner, invoking 135
 object spawner, setup 134
 upgrades
 preserving the repository 9
 user groups
 enabling tasks for 122
 user profiles
 groups compared with 95
 user rights 36
 users 99
 adding 100
 client users 38
 deleting 103
 managing 99
 updating properties 103

W

Windows
 object spawner 44
 Windows NT
 object spawner 136
 object spawner, configuration file 137
 object spawner, invoking 136
 object spawner, setup 136

Your Turn

If you have comments or suggestions about *Administrator for Enterprise Clients: User's Guide, Second Edition*, please send them to us on a photocopy of this page, or send us electronic mail.

For comments about this book, please return the photocopy to

SAS Publishing
SAS Campus Drive
Cary, NC 27513
email: yourturn@sas.com

For suggestions about the software, please return the photocopy to

SAS Institute Inc.
Technical Support Division
SAS Campus Drive
Cary, NC 27513
email: suggest@sas.com