SAS® 9.1 Integration Technologies

Administrator's Guide
(LDAP Version)

The Power to Know®

The correct bibliographic citation for this manual is as follows: SAS Institute Inc. 2004. *SAS® 9.1 Integration Technologies: Administrator's Guide (LDAP Version)*. Cary, NC: SAS Institute Inc.

SAS 9.1 Integration Technologies: Administrator's Guide (LDAP Version)

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

April 2004

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

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

SAS® 9.1 Integration Technologies: Administrator's Guide (LDAP Version)

This is the LDAP version of the Administrator's Guide for SAS Integration Technologies. It is provided for Integration Technologies customers who store metadata for SAS applications on a directory server that is compliant with the Lightweight Directory Access Protocol (LDAP), as well as customers who use flat configuration files in the LDAP Data Interchange Format (LDIF) format.

Note: Some new features of SAS 9.1 require the use of the SAS Open Metadata Architecture instead of LDAP. For details, refer to the main SAS Integration Technologies Administrator's Guide.

This guide provides detailed instructions for all of the administrative tasks that are required for an Integration Technologies implementation. If you are using an LDAP server to store your metadata, you can perform most of these tasks using the Integration Technologies (IT) Administrator application. IT Administrator is a graphical user interface that allows you to easily enter and modify metadata on your LDAP server for SAS applications.

Before you begin performing Integration Technologies administration tasks, refer to the Getting Started section for important introductory information and guidelines. The Getting Started section provides

- a summary of the administrative steps involved in an Integration Technologies implementation
- basic instructions for using IT Administrator
- a summary of the steps for setting up an LDAP server
- guidelines for determining which communications protocol your Integrated Object Model (IOM) server should use: the Windows Component Object Model (COM) or the Integration Technologies IOM Bridge protocol
- a description of the use of logical names to group server resources
- a description of the use of connection pooling to conserve server resources.

Then refer to the other sections in the Administrator's Guide for detailed documentation of each administrative task, including

- setting up and starting an IOM server using COM/DCOM
- setting up and starting an IOM Bridge server and spawner
- administering the metadata that is needed to implement the Integration Technologies Publishing Framework and stored processes
- administering SAS data sources that are to be accessed by IOM servers
- administering security settings for resources associated with IOM servers.

Use this Administrator's Guide in conjunction with the Developer's Guide, which provides details about using Integration Technologies to develop and integrate applications.

Getting Started

Getting Started

This chapter describes the processes for administering a SAS Integration Technologies (IT) implementation. In general, IT administration involves configuring the application resources for your site and defining the properties for each resource. The specifics of the process depend on the requirements for your implementation. The basic administration steps are as follows:

1. **Plan your implementation:**

   - Define how your organization intends to use the features of Integration Technologies, such as publish/subscribe and distributed applications.
   - Determine the hardware and software elements that will be involved in your Integration Technologies implementation. For example, if you are administering a distributed application implementation, you will need to know the communication requirements for connecting your client and server platforms.
   - Determine security roles and access control policies for data and other resources.
2. **Set up the LDAP directory server.** Integration Technologies uses an enterprise directory provided through the Lightweight Directory Access Protocol (LDAP). This directory provides a common repository from which user, resource, and security−policy information can be centrally managed. Because all of the IT administration tasks involve working with LDAP information, the first administration task is to install and set up an LDAP server. If you already have an LDAP server, you must install the Integration Technologies schema. For details about installation and setup, see Setting Up an LDAP Directory Server.

   Note: For a limited implementation of Integration Technologies, you can use flat configuration files instead of LDAP. However, if your configuration requires more than one or two SAS object servers, or if multiple clients will be using the servers, we strongly recommend the use of LDAP as a central metadata repository. The use of LDAP also gives you the ability to use access control lists to control access to the servers in your enterprise.

   Note: Alternatively, for a limited implementation of Integration Technologies, you can supply the server parameters for the configuration directly in the application program.
3. **Define LDAP objects.** After you have set up the LDAP server, begin adding definitions for the LDAP objects needed for your implementation. Examples include SAS server and spawner definitions, archive paths, and channel definitions. Use the Integration Technologies (IT) Administrator application to perform these tasks. For general instructions, see Using IT Administrator. For instructions on setting up specific resources, refer to the appropriate topic in the index at left (Administering SAS Servers, Administering Stored Processes, Administering the Publishing Framework, or Administering SAS Data Sources).
4. **Implement Security.** Integration Technologies uses Access Control Information (ACI) rules in the LDAP directory to determine which resources can be accessed by the user. For a given object or group of objects, an ACI rule can either allow or disallow access to individual registered users or groups of users. For instructions on setting up users and access rules, refer to the Administering Security topic in the index at left. Further security is available through SAS Login objects, which control access to the SAS data and processes which reside on servers.
5. **Perform ongoing maintenance tasks.** After you set up your implementation and roll it out to the user community, you may occasionally need to change your Integration Technologies configuration. Therefore, you should establish a maintenance procedure for making changes to the LDAP information.

Getting Started

Setting up an LDAP Directory Server

In order to use LDAP entries with Integration Technologies software, you must install and configure an LDAP directory server. This server stores the LDAP entries and makes them accessible to SAS (and other) applications that need the information. See LDAP Structure in the *SAS Integration Technologies Developer's Guide* for details of how the entries are structured in the LDAP directory.

The three basic steps involved in setting up the server are

1. Installing the server. If an LDAP directory server exists already, you may omit this step.
2. Installing LDAP Schema for IPlanet, Netscape, and SecureWay. This step configures the server to recognize the entry types that SAS uses and to create the entries that SAS expects to see.
3. Adding person entries to the directory. Person entries provide information about specific users within the LDAP directory. If an LDAP server exists, person entries may be defined already.

If you are using the Microsoft Active Directory, you must install an LDAP schema for the Active Directory. This schema uses a different relative distinguished name (RDN) than the standard schema and is required if you are using the Active Directory.

Getting Started

Installing the Server

In order for Integration Technologies software to use the LDAP directory, you must set up a directory server. These instructions provide the procedure for setting up a directory server using iPlanet (previously known as Netscape), which is the option distributed with SAS software. These instructions assume that you are using a Windows/NT server.

1. Verify that the TCP protocol is configured and that at least a host name and domain name are defined. To verify, follow these steps:

    a. Right−click on the Network Neighborhood icon on the Windows/NT desktop and select Properties from the pop−up menu. The Network window appears.
    b. If it is not selected already, select TCP/IP Protocol in the Network Protocols list.
    c. Select the Protocols tab, and then select the Properties button. The TCP/IP Properties dialog box appears.
    d. Select the DNS tab in the dialog box.
    e. Verify that values are entered in the Host Name and Domain fields. If the fields are blank, you must enter valid values before you can continue with directory installation.
    f. Close all dialog boxes.
2. Start the iPlanet Directory Server (previously known as Netscape Directory Server) installation program. After you agree to the software license, you are asked what type of installation to perform. Select Custom Install.
3. Select which components to install. In general, the default selections should be acceptable. You do not need to install Synch Services unless you plan to use replication.
4. The next dialog box asks whether this directory service instance is the configuration directory server. Accept this selection unless you fully understand the consequences of using a different server to store the configuration information.
5. The installation procedure requests the server identifier, port, and suffix. In most cases, the default identifier and port should be acceptable without change. However, you can modify the suffix to match your installation.

    For example, a for company named Alphalite Airways, you might enter the suffix `o=Alphalite Airways,c=US`. Alternatively, you could use the domain component format: `dc=alpair,dc=com`.

    If you are not sure what value to use, enter `o=`*`CompanyName`*`,c=US` for a US company. For other countries, use the appropriate two−character ITU abbreviation.
6. You are prompted for the Configuration Directory Administrator user ID and password. The Configuration Directory Administrator has access to the configuration data stored in the directory server. Enter secure values for these fields and remember the values, because they are difficult to recover if they are lost. This user ID is automatically added as the Configuration Administrator user in the directory's configuration data tree.
7. When prompted for the Administration Domain, accept the default value.
8. You are prompted for the Directory Manager DN. This value is different from the Configuration Directory Administrator user ID and password. The Directory Manager has access to all user−added data in the server. Any user with the specified DN and password can freely access all data in the directory regardless of the access control settings.

    A common manger DN is `cn=root`, although you can use any valid DN that is formatted as a comma−separated series of name/value pairs. Select a secure password.
9. Next, you are asked whether you want to configure the directory as a supplier or consumer of replication data. Unless you have read the instructions and know you want to replicate your directory, accept the defaults for this prompt.

10. You are asked whether you want to install sample data into the directory. If you want to test the directory installation but do not have any data immediately available, select Yes for this prompt.
11. You are asked whether you want to disable schema checking. SAS recommends that you NOT disable schema checking, which ensures that all of the entries in the directory conform to the schema definition. The schema is a list of attribute types consisting of name, object identifier (OID), matching rule (case−insensitive string, case−exact string, etc.), and a list of object classes that defines which attributes are required and allowed for that class. With schema checking enabled, new entries are compared against the schema before those entries are added to the directory. Entries that do not conform to the schema are not added to the directory.
12. You are prompted for the Administration Server Access user ID and password. This is the ID and password that are required when you start the console application. It is convenient, but not necessary, to select the same ID and password as the Configuration Directory Administrator.
13. You are prompted for the administration port. Accept the default value.
14. Delete the installation cache.
15. Reboot, if requested to do so.
16. The installation procedure is finished.

Getting Started

Installing LDAP Schema for Sun One, Netscape, and SecureWay

After you install the LDAP directory server, you must change the configuration so that SAS software can use the server correctly. The steps for performing this configuration are as follows:

1. Locate the LDAP configuration files in your IT Administrator directory.
2. Copy the appropriate LDAP configuration file(s) into your server configuration directory.
3. Take the necessary steps to identify the configuration files to the server.
4. Restart the server.
5. Locate and edit the file named containers.ldif.
6. Make sure that the directory contains an entry representing your suffix.
7. Add or import the containers to the directory.
8. Check the success of the import or add procedure.
9. Set the access control on the directory.
10. Set up indexes on the LDAP server.
11. Set the server limits to improve search performance.

The detailed procedures for performing these steps are as follows:

1. Locate the LDAP configuration files in the directory where Integration Technologies (IT) Administrator was installed. You will find the files in *admin_loc*\ldap, where *admin_loc* is the drive and directory where IT Administrator is installed. The default location is C:\itadmin\ldap.

   The LDAP configuration files define the attributes and object classes that are used by SAS Integration Technologies and other related SAS software.

   The files are as follows:

   75sas.ldif
   contains the schema data for Sun ONE Directory Server 5.1
   nsslapd.sas_at.conf
   contains the attribute schema data for Netscape Directory Server 4
   nsslapd.sas_oc.conf
   contains the object class schema data for Netscape Directory Server 4
   slapd.sas_at.conf
   contains the attribute schema data for an OpenLDAP directory server.
   slapd.sas_oc.conf
   contains the object class schema data for an OpenLDAP directory server.
   V3.sas.oc
   contains the schema data for an IBM SecureWay V3 server.
   msadClassesAttrs.ldif
   contains the schema data for a Microsoft Active Directory server.
   containers.ldif
   creates the containers for SAS application data.
2. Depending on which server software you are using, copy the appropriate LDAP configuration file(s) into your server configuration directory.

   - For Sun ONE Directory Server 5.1, copy **75sas.ldif** to the server's schema directory. As a default, the

schema directory is in the following path: slapd−localhost\config\schema.

- For Netscape Directory Server 4, copy **nsslapd.sas_at.conf** and **nsslapd.sas_oc.conf** into the server's configuration directory. As a default, the configuration directory is in the following path: *drive*:\netscape\server4\slapd−*instance*\config
- For an OpenLDAP directory server, copy **slapd.sas_at.conf** and **slapd.sas_oc.conf** into the server's configuration directory.
- For an IBM SecureWay V3 server, copy **V3.sas.oc** into the server's configuration directory.
- If you are using Microsoft Active Directory, refer to Installing the LDAP Schema for Microsoft Active Directory for instructions on loading the msadClassesAttrs.ldif schema file.

3. Take the necessary steps to identify the configuration files to the server. Generally, this is performed by placing `include` statements in the server's configuration file. Check the documentation for your server to verify the procedure.

   For Netscape Directory Server 4, the procedure is as follows:

   a. Use a text editor to open the slapd.conf file.
   b. Search for an `include` directive at the beginning of a line.
   c. After the last existing `include`, add a new `include` directive that contains the full path of the new nsslapd.sas_at.conf file. The new line should have the same syntax as the line above it.
   d. Add another `include` directive for the file nsslapd.sas_oc.conf.

      The new lines should be similar to the following examples:

      ```
      include "c:/netscape/suitespot/slapd−D1354/config/nsslapd.sas_at.conf"
      include "c:/netscape/suitespot/slapd−D1354/config/nsslapd.sas_oc.conf"
      ```

   Note: This procedure is not necessary for Sun ONE Directory Server 5.1.

4. Restart the server so that the server reads the new configuration information.

   For a Sun ONE Directory Server or a Netscape Directory Server, the procedure is as follows:

   a. Start the directory console. To start the console from a Windows/NT desktop, select Start ➧ Programs ➧ Netscape Server Products ➧ Netscape Console.
   b. Restart the server from the console.

5. Locate and edit the file named containers.ldif. This file contains the entries that SAS expects to find when it starts using the directory server.

   Edit containers.ldif to replace each instance of $SAS_CONTEXT$ with the correct LDAP suffix for your installation. Place this suffix everywhere that $SAS_CONTEXT$ appears. For example, if your suffix is `o=ACE Industries, c=US`, you would edit the first line of containers.ldif to read as follows: `dn: cn=SAS,o=ACE Industries,c=US`.

   Alternatively, you can put the $SAS_CONTEXT$ entry lower in the directory tree. However, if you put it below the root, you must be sure that all entries between the root and the suffix are in place in the directory tree. For example, if the SAS tree starts at `ou=Finance,o=Alphalite Airways,c=US` and the database suffix is `o=Alphalite Airways,c=US`, then the organizational unit entry for `ou=Finance,o=Alphalite Airways,c=US` must be in the directory before you import the SAS containers.

6. Make sure that the directory contains an entry representing the suffix that you specified in the containers.ldif file. For example, if your suffix is o=ACE Industries, c=US, make sure the directory includes the entry `dn: o=ACE Industries,c=US.`

If your database is completely empty, then you must create the root object, which is usually an organization object class. An example of a simple organization entry is

```
dn: o=Alphalite Airways,c=US
objectclass: organization
o: Ace Industries
```

Either use the command

```
ldapmodify –a –D manager DN –w manager password
```

to insert the root object entry into the directory, or add the command to the containers.ldif file.

7. After you edit the containers.ldif file, use the `ldapadd` command to add the containers to the directory. Use a bind DN that has the appropriate permissions.

   If you are using Sun ONE Directory Server or Netscape Directory Server, you can import the containers.ldif file using the following procedure:

   a. Start the console.
   b. Open the Directory Server.
   c. Select the **Configuration** tab in the Directory Server window.
   d. Select the database icon.
   e. Select **Import** from the console menu.
   f. Enter the path for the containers.ldif file.
   g. Select **Append to Database** in order to import the file.
8. Check the success of the import or ldapadd procedure by noting the number of rejected entries. If more than one or two entries are rejected, check the two most likely reasons:

   - The schema was not updated correctly.
   - The parent entry of the first container was not created.

   See the previous step for information about creating the parent entry.
9. Set the access control on the directory. The installation process may have created some default access control lists (ACLs). Normally, the installation process will create an ACL called "anonymous access" that allows anonymous users to search the data in the directory. Until you understand access control, modify this value to allow all access.

   Although this is not a permanent solution, it lets you operate until you can create users and groups and can define ACLs that give those groups appropriate access to the data.

   For more information about LDAP access control, refer to Adding Person Entries to the Directory and LDAP Configuration Access Control Overview.
10. Set up indexes on the LDAP server. These indexes will improve the performance of SAS with the server. Consult the documentation for your server for information on creating the indexes.

    Create these indexes:

Attribute	Index Type
sasInterface	eq, pres
sasKeyword	eq, pres
sasSubscriberName	eq, pres
sasSubscriberGroupDn	eq, pres

sasDomainName	eq, pres
sasLogicalName	eq, pres
sasReferenceDn	eq, pres
sasPersonDn	eq, pres
sasPortalSubwindows	Sub
sasSubscriberCn	eq, pres

11. Set the server limits to improve search performance. Using the directory console software, set the look−through limit, size limit, and time limit to −1 (minus 1). This value disables all three limits, and permits searches against the LDAP directory to return accurate results.

The server is now ready for use by SAS software.

Getting Started

Installing the LDAP Schema for Microsoft Active Directory

If your LDAP server is Microsoft Active Directory, you must use Release 1.2 or later of the Integration Technologies Administrator, and you must install the LDAP schema for the Active Directory. The schema uses a different format for the relative distinguished name (RDN) that the Active Directory can recognize. The procedures in this section assume you have already installed the Active Directory on a Windows 2000 Domain Controller (DC).

To install the schema, follow these steps:

1. Enable schema updates. To be able to modify the schema, you must modify the registry key located at

   HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\NTDS\Parameters

   Insert `Schema Update Allowed` as a REG_DWORD value into the registry, and set the value to `1` (or any other value greater than `0`).
2. Edit the msadClassesAttrs.ldif file, provided with Integration Technologies. In the file, replace the string $SAS_CONTEXT$ with your active directory domain suffix. An example suffix is `dc=mydomain,dc=mycompany,dc=com`.
3. Import the classes and attributes. To perform the import and to create the log file in the current directory, run the following command on the Windows 2000 server from the MS−DOS command prompt:

   ```
   ldifde −i −f msadClassesAttrs.ldif
   ```
4. Determine where in the directory hierarchy you want to put the SAS entries. The SAS containers create a top level container named SAS. If you do not have a container for applications, then create a container (typically named Applications, although you can use any name) at the root level of the active directory. The top−level SAS container is installed in this container.
5. Edit the containers.ldif file. In the file, replace the string $SAS_CONTEXT$ with the container into which you want the SAS containers installed. Using the example values from Step 2, an example container name is `cn=Applications, dc=mydomain,dc=mycompany,dc=com`.
6. Create the containers. To create the SAS containers, run the following command on the Windows 2000 server:

   ```
   ldifde −i −f containers.ldif
   ```
7. Disable schema updates. Modify the registry key located at

   HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\NTDS\Parameters

   (the same key modified in Step 1). Set the value for `Schema Update Allowed` to `0`.

After you install the schema updates, you must always provide the relative distinguished name when logging into the server through the Integration Technologies Administrator. In the User field of the Administrator's Login window, you must specify the distinguished name relative to the user base name that you specified when you installed the Administrator. Example logins include `cn=username` and `c=us, cn=users, dn=mydomain`. If you did not specify a user base name, you must specify the entire distinguished name, for example `cn=username,cn=users,dc=mydomain,dc=mycompany,dc=com`.

Getting Started

Adding Person Entries to the Directory

After you start the directory server, update the schema, and set the access control, the directory is ready to use. However, you must add person entries to the directory in order to make the directory useful to SAS applications. For example, when you update access control, access decisions are based on the DN that the person binds to the directory.

SAS software also uses person entries to identify users and to obtain information such as user ID and e−mail address. Some of the options for user data are object class, directory structure, and DN.

To add person entries to the directory, follow these steps:

1. Select an object class to use for the entries. A common choice is the `inetOrgPerson` class, which accepts many useful attributes. If you need to add attributes to your person entries and the attributes are not allowed by `inetOrgPerson`, you can create your own object class using `inetOrgPerson` as a parent class.
2. Enter the person entries in the directory. Follow these guidelines to help your person data work better with SAS software:

   - Keep common names unique. Some SAS applications use the common name when associating a person entry with other entries in the SAS application entries.
   - Include the user ID and e−mail address in the person entry. Applications need to look up the user ID.
   - When you load the directory with person entries for the first time, add a default `userpassword` attribute. This attribute allows users to bind to this DN when they use the directory.
3. Decide how the person data is laid out in the directory. The two most popular options are as follows:

   Flat structure
   : puts all of the data in one place in the directory. The benefit is that you do not have to move the entries if users change organizations within the company.

   Organizational unit structure
   : places the entries in a subtree according to the organizational unit within the company. This structure can resemble the company's organization, which allows you to visualize the relationships between entries.
4. Decide on the structure of the distinguished names for your person entries. Although your selection of the attribute for the relative distinguished name is not critical, you must be consistent. Two acceptable choices are common name and user ID. If you use a flat structure for the person data, then use user ID for the DN, because common names are duplicated more often than user IDs.

LDAP Configuration Overview

This file provides instructions on how to configure and run the LDAP Server with the required server, archive, and publishing framework metadata.

To configure the LDAP server, follow these steps:

1. Locate the LDAP configuration files in the directory /itadmin/ldap. The files are as follows:

   containers.ldif
   : is an LDIF that creates the containers for SAS application data.

   nsslapd.sas_at.conf
   : is the attribute schema data for an iPlanet Directory Server (previously known as Netscape Directory Server).

   nsslapd.sas_oc.conf
   : is the object class schema data for an iPlanet Directory Server.

   V3.sas.oc
   : SecureWay V3

   slapd.sas_at.conf
   : is the attribute schema data for an OpenLDAP directory server.

   slapd.sas_oc.conf
   : is the object class schema data for an OpenLDAP directory server.

2. In order to prepare your LDAP server to receive SAS application data, add the appropriate files to your server configuration. Use the table below to determine which files to add for your installation.

iPlanet (previously Netscape)	nsslapd.sas_at.conf nsslapd.sas_oc.conf
SecureWay V3	V3.sas.oc
OpenLDAP	slapd.sas_at.conf slapd.sas_oc.conf

   Normally, adding files to a server configuration involves placing `include` statements in the slapd.conf file. Check the documentation for your server to verify the procedure.

   For example:

```
include slapd.sas_at.conf
include slapd.sas_oc.conf
```

3. Restart the server so that the server reads the new schema information.
4. Edit the containers.ldif file in order to include the correct LDAP suffix for your directory. The entry representing the suffix must be in the directory before you add the SAS containers. For example, if your suffix is `o=ACE Industries, c=US`, make sure the directory includes the entry `dn: o=ACE Industries,c=US` already.
5. Use the `ldapadd` command to add the containers. Use a bind DN that has the appropriate permissions.

Getting Started

Using the Integration Technologies (IT) Administrator

The Integration Technologies (IT) Administrator is a Java application for creating and modifying the LDAP definitions for objects that are used by SAS Integration Technologies.

This section tells you how to install IT Administrator on your machine. It also provides general instructions for using IT Administrator to create, modify, and search for objects.

The objects that you can maintain by using IT Administrator are as follows:

- **SAS Configuration** objects, including definitions for your enterprise's SAS servers, spawners, and logins. For detailed instructions on defining SAS configuration objects, refer to the Getting Started chapter.
- **Applications**, including definitions for SAS stored processes and for the paths in which they are stored. For detailed instructions on defining stored processes, refer to the Stored Processes chapter.
- **Publishing Framework** objects, including definitions for publication channels, subscribers, and subscriber groups. For detailed instructions on defining Publishing Framework objects, refer to the Publishing chapter.
- **SAS Archiving**, inlcuding definitions for archived reports and published packages and for the archive paths in which they are stored. For detailed instructions, refer to Creating Archive Paths.
- **SAS Data Sources**, including definitions for SAS libraries, tables, and columns that can be accessed by client applications. For detailed instructions on defining SAS data sources, refer to the Data Sources chapter.

Getting Started

IT Administrator Requirements

Client Requirements

Integration Technologies Administrator is a Java program. In order to run it, you must have one of the following installed:

- Java 2 Runtime Environment, Standard Edition, Version 1.4.1 (J2RE) or later
- jview, release 5.00.3167 or later

You can download jview from Microsoft's Web site.

IT Administrator runs only on Windows platforms.

Server Requirements

To run the administrator application, you must have the following on the server machine:

- an LDAP V2 or V3 compliant server.
- the schema definitions file.
- the initial LDIF file for populating the required metadata, which is used to create the container structure that SAS expects. After the directory is set up, this file should not be needed.

Metadata Requirements

Before the SAS Integration Technologies Administrator can manage the SAS metadata, you must apply the provided schema and LDIF file to an LDAP server. The administration program requires the location and port number of the LDAP server and the location of the base of the SAS metadata. Both of these values are specified in the site.cfg file. For details on configuring the site.cfg file, see IT Administrator Installation and Startup.

The following is an example of the defined hierarchy and the location of the base directory. This structure is set up when you apply the supplied schema and LDIF file.

```
c=US
   o=SAS Institute (the base)
      cn=SAS
         sascomponent=sasPublishSubscribe
            cn=saschannels
            cn=sassubscribers
         sascomponent=sasServer
            cn=sasservers
            cn=sasspawners
            cn=saslogins
         sascomponent=Archiving
            cn=sasarchivepaths
```

In the previous example, the distinguished name of the base entry is

```
   o=SAS Institute,c=US
```

The following structure must exist below the base entry:

```
saschannels
sassubscribers
```

Hardware Requirements

There are no hardware requirements for running the administrator application.

Getting Started

SAS Integration Technologies Administrator Installation and Startup

A Windows wizard is provided to assist you in installing SAS Integration Technologies Administrator. The wizard will prompt you for the following information:

Destination location
specifies the path where the SAS Integration Technologies Administrator files will be installed. By default, the files are installed in C:\Program Files\SAS\ITAdmin

Java runtime environment (JRE)
specifies the pathname of the Sun Java 2 Runtime Environment required by the SAS Integration Technologies Administrator. If the JRE that is installed on your machine is older than the required version, then you will need to install the required version on your machine. If you install a new version, you might need to reboot your machine before you use the JRE and run the SAS Integration Technologies Administrator.

LDAP server type
defines the type of server that you are using for your LDAP directory. You can choose one of the following:

`Netscape`
iPlanet Directory Server (previously known as Netscape Directory Server)

`IBM`
IBM Secureway server

`Other`
A server type other than Netscape or IBM

LDAP server
specifies the fully qualified domain name of the machine that runs the LDAP server. For example, if your LDAP server is running on a machine that is called topgun, the value might read

```
topgun.pc.x.com
```

LDAP server port
specifies the port number on which the LDAP server listens for connections. For example, if your LDAP server is running on a machine called topgun.com and listening on a port number of 389, enter the value `389`.

DN for SAS/Integration Technologies metadata
specifies the distinguished name for the context in LDAP that contains the SAS Integration Technologies metadata. Typically, this is in the form o=COMPANY_NAME,c=COUNTRY. For more information, see Setting Up an LDAP Server.

This value is the distinguished name for accessing the LDAP information from the specified LDAP server. The distinguished name is the starting point of the publish/subscribe information that is maintained in the server.

For example, if the distinguished name in the LDAP server is o=Company X,c=US, you would enter

```
o=Company X,c=US
```

in this field. This value is dependent on how your LDAP directory server is configured. For more information, see Configuring the Server.

Authentication value
specifies the type of authentication to use, either **`simple`** or **`none`**.

If the value is set to **none**, the LDAP server performs no authentication, which enables a user to log in as an anonymous login.

If the value is set to **simple**, the LDAP server performs authentication and requires the user to enter a username and password in order to access the information that is stored in the LDAP server. The server will apply any access control information to the user and set their privileges based on access level.

User DN

specifies the distinguished name (DN) for the context in LDAP that contains user metadata. The administrator uses this value to determine the distinguished name of the user logging in as well as finding information on objects that define people to the LDAP server.

For example, if the user's person object password attribute is defined in the distinguished name cn=John Smith,ou=People,o=ABCToysCompany,c=US, enter

```
ou=People,o=ABCToysCompany,c=US
```

Group DN

specifies the distinguished name (DN) for the context in LDAP that contains group metadata. The administrator uses this value to determine objects that define groups of people to the LDAP server.

For example, if a group object entity is defined in the distinguished name cn=Accounting,ou=Groups,o=ABCToysCompany,c=US, enter

```
ou=Groups,o=ABCToysCompany,c=US
```

Application DN

specifies the distinguished name (DN) for the context in LDAP that contains application metadata. The administrator uses this value to find information on stored process paths and stored processes in the LDAP server.

For example, if a group object entity is defined in the distinguished name sascomponent=sasApplications,cn=SAS,o=ABCToysCompany,c=US, enter

```
cn=SAS,o=ABCToysCompany,c=US
```

User objectclass

specifies the attribute that is used in the LDAP directory for identifying person entries. The default value is **person**. See Adding Person Entries to the Directory for information on person entries in the LDAP directory.

After you enter all the required information in the installation wizard, the SAS Integration Technologies Administrator program is installed in the directory you specified and a shortcut is added to the Windows Start menu.

To start the Administrator, select Start ➧ Programs ➧ SAS ➧ SAS Integration Technologies Administrator. The Login dialog box appears and prompts you for a user and password for logging into the LDAP directory server. To log in, you must use either a full distinguished name (for example, cn=Andrew Williams,ou=People,o=Alphalite Airways,c=US) or a user ID (for example "awill").

Getting Started

IT Administrator: How it Works

The Integration Technologies Administrator allows a user with administration privileges to create, modify, and delete objects on an LDAP server. The objects that the administrator can manage are grouped into five categories:

Publish Framework
: Publish channels and subscribers, including subscriber groups

SAS Configuration
: SAS servers, spawners, and SAS logins

SAS Archiving
: Archives and archive paths

Applications
: Stored SAS processes and stored process paths

SAS Data Sources
: Identifiers for SAS libraries, tables, and columns, which can then be used by clients

To better understand the objects that the Administrator creates as well as the relationships between the objects, let's look at the structure of the LDAP directory.

LDAP Overview

The Lightweight Directory Access Protocol (LDAP) was created to help manage network data such as users, resources, and security from a central location.

Conceptually, an LDAP server maintains a hierarchy of objects. An object is made up of name/value pairs called attributes. An object is based on a class, which defines which attributes are required for the object, and which are optional. The set of defined classes and their attributes is called the Directory Schema.

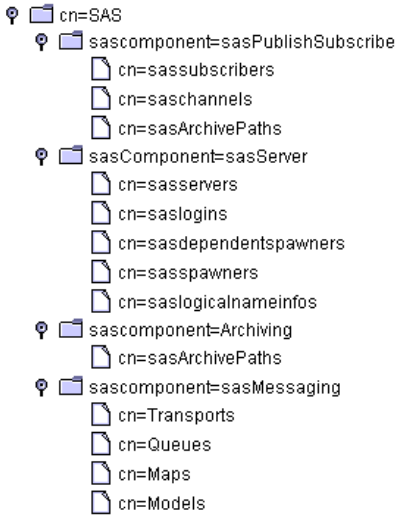
The layout of the classes is called the Directory Information Tree (DIT). Beginning from the top of the tree, the path to each class in the tree is called its distinguished name. Each distinguished name in the tree is unique. Distinguished names are defined from the lower element up to the root, which is typically a country (c), followed by an organization (o).

In the tree below, the root of the tree is *o=SAS Institute, c=US*.

The distinguished name for the ABC Toys entry is *cn=ABC Toys, o=SAS Institute, c=US*.

```
c=US
  o=SAS Institute (the base)
    cn=ABC Toys
      cn=SAS
         sascomponent=sasPublishSubscribe
           cn=saschannels
           cn=sassubscribers
         sascomponent=sasServer
           cn=sasservers
           cn=sasspawners
           cn=saslogins
         sascomponent=Archiving
           cn=sasarchivepaths
```

The `cn=` entries under the `sascomponent=` entries are the objects that the Administrator defines. Under the `cn=sassubscribers` entry will be a number of sassubscribercn objects, each of which defines a specific SAS subscriber or subscriber group. Likewise, a number of sasservercn objects will be defined under the `cn=sasservers` entry, each defining a specific SAS server. For example:



Publish Framework objects

Selecting Publish Framework in the Integration Technologies Administrator gives you access to the LDAP definitions for

- channels (saschannelcn objects)
- subscribers (sassubscribercn objects)

Channels

The channel definition lets you specify a channel, which is a conduit for sending information from a publisher to all users subscribed to the channel. In addition to specifying descriptive information (such as name, description, and subject), you can also add subscribers to the channel. The subscribers you add must have entries in the LDAP directory.

You can create archive paths underneath a channel definition for use by that channel. The definitions for archive path and any archives contained in the path are displayed under the channel in the tree. See SAS Archiving objects for more information.

Subscribers

The subscriber definition lets you specify information for a subscriber, which is any recipient of information published to a channel. Each subscriber definition must point back to a person reference in the LDAP directory. In addition to identifying the person reference for the subscriber, you can also specify a number of options for the subscriber, including where the user receives information, what format to use when sending information, and what filters to apply to the information.

SAS Configuration objects

Selecting SAS Configuration in the Administrator lets you create and modify LDAP definitions for

- servers
- spawners
- SAS logins

Creation of server and spawner definitions is automated through the Integration Technologies Server Wizard. The Wizard guides you through the process of defining a COM/DCOM or IOM Bridge server and spawner, if one is required.

Server definitions are grouped according to their purpose under a series of logical names. For example, you could use a logical name of Payroll to group all the servers that could be used for payroll operations. Because a single server can be used for more than one purpose, each server definition can be associated with several logical names.

The spawner definitions also use logical names, but their function is to determine which servers the spawner can connect to. When a request is sent to a spawner to start a SAS session on a server, the spawner checks the server definitions to find one that has logical names that are a subset of the spawner's.

For example, if the definition for `spawnerABC` includes the logical names `Payroll`, `Accounting` and `Finance`, that spawner could start any of these servers:

- `server1` − logical name `Payroll`
- `server2` − logical name `Accounting`
- `server3` − logical name `Payroll`, `Finance`

However, it could not start this server:

- `server4` − logical name `Inventory`

Servers

The Administrator creates the server definitions through the Wizard, then lets you modify the definitions as needed through the properties panel. The server definition includes:

Domain
: The server's domain. In order for a spawner to work with the server, the spawner must be defined for the same domain (in addition to having matching logical names).

Protocol
: IOM Bridge or COM/DCOM protocols, as well applicable service or port IDs.

Logical name
: All logical names under which this server can operate.

Machines
: All machines on which the server can run.

Encryption
: Client and server algorithms, what content to encrypt

SAS logins
: The SAS logins available to start a SAS session.

Commands
: The command to start the SAS session on the server.

Maximum workspaces per pool
: The maximum number of workspaces that will be available for any workspace pool that is established with the server.

Spawners

When a server definition requires that a spawner also be defined, the Wizard automatically goes through the spawner definition process. As with the server definitions, you can then modify the definitions as needed through the properties panel. The spawner definition includes:

Domain
: The spawner's domain. In order for a spawner to work with the server, the spawner must be defined for the same domain (in addition to having matching logical names).

Protocol
: IOM Bridge or COM/DCOM protocols, as well applicable service or port IDs.

Logical name
: All logical names under which this spawner can operate.

Connection information
: Service, port and passwords for master, operator, and UUID connections

Machines
: All machines on which the spawner can run.

Encryption
: The modules path and key length for encryption

Logging
: The path to the log file and whether to use verbose logging

OS/390
: The z/OS logical unit name

SAS Logins

A SAS login may need to be available in order to start a SAS session on a server or to connect to a client. Each SAS login definition contains a user name, password, and domain, as well as a pointer to the user's person reference entry in the LDAP directory.

SAS logins may be used to provide credentials when creating a client connection. Whether or not SAS logins are required depends on the method calls used to start the server or create the connection. If the method calls request a logical name, SAS logins are required. Otherwise, SAS logins are not required, but if you do not use them, you must track and specify the user credentials manually.

The SAS login definition includes:

Person reference
: The person reference entry in the LDAP directory for the user. The person reference entry is created outside of the Administrator application

User
: The user ID

Password
: The user's password

Domain
: The domain on which the user ID is valid.

Logical name
: The logical name of the SAS server with which this login is associated (used only with workspace pooling).

Min workspace size
: The number of workspaces currently serving or waiting to service a request (used only with workspace pooling).

Min available workspaces
: The number of workspaces waiting to service a request (used only with workspace pooling).

SAS Archiving objects

Selecting SAS Archiving in the Administrator lets you create and modify LDAP definitions for archive paths. Archives are stored copies of packages that have been published using SAS Publish and Subscribe. Archive paths and archives are also present in Publish Framework under individual channels. Archive paths created under individual channels are for use by that channel exclusively.

Archive Paths

The archive path contains the location where a server can publish an archive package. The archive path definition includes:

Archive path
: The full path name for the location to which archives are to be published.

Logical name
The logical name for the path, used to identify valid publishing paths.

Archives

Because the archives are created outside of the Administrator, their LDAP information cannot be modified. However, you can view this information:

Creation date
The date the archive package was created
Channel
The channel to which the package was published. The channel is only present if the archive object is under an individual channel.

Stored Process objects

Selecting Applications in the Administrator lets you create and modify LDAP definitions for stored processes and stored process paths. A stored process is a SAS program that is saved (in a stored process path) and can be executed at a later time by an Integration Technologies user or application.

Stored Process Paths

The stored process path defines the location where stored processes are kept. The stored process path definition includes:

Stored process path
The full path name for the location where stored processes are kept.
Logical names
All logical names associated with the path.

Stored Processes

The stored process definitions provide information about saved SAS programs. The stored process definition includes:

Description
An identifying description of the process
Stored process value
The name of the stored SAS program
Portal JSP
The Java stored page (JSP) from which a user or application can access the program.
Parameters
Sets of parameters that are passed to the SAS program upon execution

Data Source Objects

Selecting SAS Data Sources in the Manager Bar lets you create and modify LDAP definitions for library, table, and column data source definitions. A data source is a SAS library, table, or column that is identified by an LDAP entry. Client applications can use the LDAP entry to locate the data source and access the information in the source.

Libraries

The library definition contains information to create a SAS LIBREF statement for the library, including the name, libref, path, and options. You can associate one or more logical names with the data source to identify the server on which the library resides.

Tables

The table definition contains information required to identify a SAS table, including the name, the distinguished name of the library containing the table, and any password protections needed for the table. As with the library definition, you can associate a table definition with a logical name to identify the location of the table.

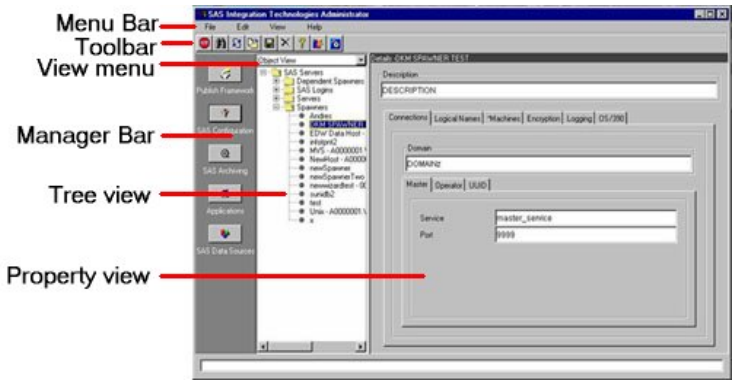
Columns

The table definition contains the information required to identify a column in a table. You must have already identified the parent table as a data source before you can identify a column. The information required for a column definition includes the column name, type, and length, as well as any formats or informats applied to the column.

Getting Started

About the IT Administrator Interface

The Integration Technologies Administrator interface is comprised of six areas:



Menu bar

The menu bar, located across the top of the window, contains commands that operate on the current selection in the tree view. The items in the menu correspond to tools on the toolbar. The menu selections are as follows:

- File menu
  - ♦ New
  - ♦ Save
  - ♦ Delete
  - ♦ Exit
- Edit menu
  - ♦ Search
- View menu
  - ♦ Manager bar
  - ♦ Refresh
  - ♦ Connection/Directory Configuration
- Help
  - ♦ Contents
  - ♦ About IT Administrator

Toolbar

The toolbar, located across the top of the administrator application, contains tools which operate on the current selection in the tree view. The tools include:

- Exit Application
- Search
- Refresh
- New
- Save

- Delete
- Help
- Server Wizard
- Set Access Permissions

The buttons in the toolbar have tooltips which are displayed when the mouse pointer is paused over a button. Although all the tool buttons are always visible, they can be applied only to certain objects, and are greyed out when an inappropriate object is selected in the tree view.

Tree view menu

The tree view menu, located directly beneath the toolbar on the left side, is used to select between different views of server, subscriber, or archive information. The selections available depend on which button you select from the Manager Bar:

- Publish Framework

  Select between main view of all channels, subscribers, and groups (Publish Subscribe), objects that have been found through the search tool (Search Results), and channels, subscribers and groups organized according to logical names (Logical View).
- SAS Configuration

  Select between views of defined servers, spawners, logical names and user logins (Object View); users and servers organized according to logical names (Logical View); defined machines (Machine View); and objects found through searches (Search Results).
- SAS Archiving

  Select between main view of archives and archive paths (Archive View); archive paths created under a channel organized according to logical names (Logical View); and results of using the search tool (Search Results).
- Applications

  Select between main view of stored processes and stored process paths (Application View); processes and paths organized according to logical name (Logical View); and results of using the search tool (Search Results).
- SAS Data Sources
  Select between main view of libraries and tables (Data Sources View); data sources organized according to logical name (Logical View); and results of using the search tool (Search Results).

Manager Bar

The Manager Bar, located on the left side of the application, lets you select which area of administration you want to work with. When you make a selection from the Manager Bar, the information displayed in the tree view and the options available on the View menu changes to match your selection. The selections on the Manager Bar are

Publish Framework
: Administration for publish recipients. The tree view displays channels and subscribers.

SAS Configuration

Administration for servers. The tree view displays logical names, SAS logins, servers, and spawners.

SAS Archiving
: Administration for publish archives. The tree view displays archive paths and archives.

Applications
: Administration for stored processes and stored process paths. The tree view displays stored process paths and stored processes.

SAS Data Sources
: Administration of directory definitions for libraries, tables, and columns. These definitions are used by an application such as the Information Delivery Portal to link to sources of specific SAS data on your servers.

You can turn off the Manager Bar by using the **View ➧ Manager Bar** selection on the menu bar.

Tree view

The tree view, located on the left side of the administrator application, displays objects that correspond to the type of administration selected in the Manager Bar. If you select Publish Framework in the Manager Bar, the tree view displays all channels and subscribers. If you select SAS Configuration in the Manager Bar, the tree view displays SAS logins, servers, and spawners. If you select SAS Archiving, the tree view displays archives and archive paths. If you select Applications, the tree view displays stored processes and stored process paths.

Objects selected in the tree view have their properties displayed in the property view, on the right side of the administrator application. The division between the tree view and property view can be adjusted by grabbing the dividing line between the two areas and dragging.

Property view

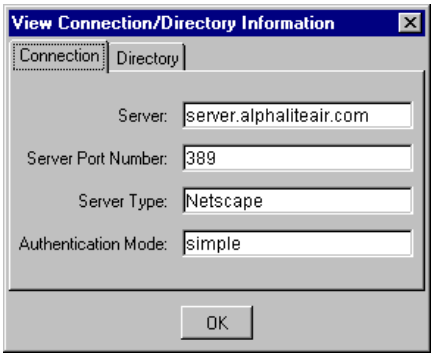
The property view, located on the right side of the administrator application, displays the properties of the object selected in the tree view. The properties displayed depend on the type of object selected, and represent the current state of the object.

Getting Started

Verifying IT Administrator Connection and Directory Information

When you install Integration Technologies Administrator, the LDAP server and base directory information that you specify is saved in the site configuration file (site.cfg). If you want to review this information without having to open the site configuration file, select **View → Connection/Directory Configuration** on the menu bar.

The View Connection/Directory Information window appears.



The Connection tab displays the server, port number, server type, and authentication mode. The Directory tab displays the root base, user base, group base, application base, and person objectclass.

All the fields displayed in this window are read−only. To change the connection or directory information, you must change the site configuration file (site.cfg).

Getting Started

Adding Objects with IT Administrator

To add an object to the LDAP directory:

1. Open IT Administrator.
2. In the manager bar, select the administration area in which you would like to add an object: **Publish Framework**, **SAS Configuration**, **SAS Archiving**, **Applications**, or **SAS Data Sources**.
3. In the tree view, select the folder for the type of object you would like to add. (For example, to add a new table object, you would select the **Tables** folder under SAS Data Sources.) Then select the **New** button () on the toolbar.

   An alternative method is to select **File** ➔ **New** on the menu bar, and select the object type from the menu that appears.
4. In the window that appears, enter the necessary properties. The property fields that are marked with an asterisk (*) are required. For a description of the properties for the object type you are adding, select the **Help** button; or select the appropriate topic on the Integration Technologies Administration index page.
5. When you are finished, select **OK**. The new object appears in the tree view.

Getting Started

Modifying Objects with IT Administrator

To modify an object in the LDAP directory:

1. Open IT Administrator.
2. In the manager bar, select the administration area in which you would like to modify an object: **Publish Framework**, **SAS Configuration**, **SAS Archiving**, **Applications**, or **SAS Data Sources**.
3. In the tree view, find the folder for the type of object you would like to modify. Then click the plus sign to open the folder. (For example, to modify a table object, you would open the **Tables** folder under SAS Data Sources.)
4. Select the object that you wish to modify. The object's current properties will be displayed in the property view in the right portion of the window.
5. Select the appropriate tabs, and enter the necessary changes. For a description of the properties, select the **Help** button; or select the appropriate topic on the Integration Technologies Administration index page.
6. When you are finished, select the **Save** icon ( ) on the toolbar; or select **File → Save** from the menu bar. (If you skip this step, IT Administrator will prompt you to save your changes when you attempt to navigate to another object.)

Getting Started

Deleting Objects with IT Administrator

Note: Version 9 does not support dependent spawners. If you are using Version 9, do not configure a dependent spawner.

Deleting an object (for example, a table) removes it from the LDAP server. Once an object has been deleted from the server, there is no way to restore it.

You cannot delete any of the following nodes:

- Publish Framework
  - ♦ Publish Subscribe root
  - ♦ Channels
  - ♦ Subscribers
- SAS Servers
  - ♦ Dependent Spawners
  - ♦ SAS Logins
  - ♦ Servers
  - ♦ Spawners
- SAS Archiving
  - ♦ Archive Paths
- Applications
  - ♦ Stored Process Paths
- SAS Data Sources
  - ♦ Libraries
  - ♦ Tables

Some channels may appear as folders because they contain overrides. To delete channels that contain subscription overrides, you must first delete all the overrides. Subscription overrides may be deleted by selecting them and clicking the **Delete** button. Deleting a subscriber's override does not delete the subscriber. See Overrides for more information.

To delete an object from the LDAP directory:

1. Open IT Administrator.
2. In the manager bar, select the administration area in which you would like to delete an object: **Publish Framework**, **SAS Configuration**, **SAS Archiving**, **Applications**, or **SAS Data Sources**.
3. In the tree view, find the folder for the type of object you would like to delete. Then click the plus sign to open the folder. (For example, to delete a table object, you would open the **Tables** folder under SAS Data Sources.)
4. Select the object that you wish to delete. The object's current properties will be displayed in the property view in the right portion of the window.
5. Select the **Delete** icon (✕) on the toolbar; or select **File ➔ Delete** from the menu bar.
6. Select **OK** in the confirmation dialog box. The item is deleted.

Getting Started

Searching for Objects Using IT Administrator

You can use the search tool in Integration Technologies Administrator to locate specific objects in the LDAP directory that meet your search criteria. You can search for the following types of objects:

Note: Version 9 does not support dependent spawners. If you are using Version 9, do not configure a dependent spawner.

- Channels
- Subscribers
- Servers
- Spawners
- Dependent spawners
- SAS logins
- Logical names
- Archives
- Archive paths
- Stored processes
- Stored process paths
- Tables
- Columns
- Libraries

The search tool consists of a text field in which you enter words to search for, and three tabs to specify the conditions of the search.

To perform a search:

1. Select the **Search** button ( ) on the toolbar; or select **Edit → Search** from the menu bar.
2. In the Search For field, enter one or more words. By default, multiple words are interpreted as a collection of separate words. You can use the Advanced tab, as described below, to specify that multiple words are to be interpreted as a phrase. Searching is not case sensitive.
3. On the Search In tab, select the administration area in which you want to search (Publish, Server, Archive, Applications, or Data Sources). You can search in only one administration area at a time.

   Then select the types of objects that you want to search for. If you search across multiple object types (for example, servers and spawners), the search results will be displayed together in a single list.
4. On the For Attributes tab, select the properties for the search tool to examine. By default, the search tool searches in the Name field.
5. On the Advanced tab, select the match criteria. The default match criterion is "Contains one of the words". The match criteria are explained in the following table.

Match criteria	Explanation
Contains one of the words	Finds objects in which at least one of the fields specified in the For Attributes tab ***contains at least one of the words*** specified in the Search For field.
Contains the phrase	Finds objects in which at least one of the fields specified in the For Attributes tab ***contains the entire phrase*** specified in the Search For field.

Exactly contains the phrase	Finds objects in which at least one of the fields specified in the For Attributes tab ***exactly matches the entire phrase*** specified in the Search For field.

6. Click **OK**.

The results of a search are displayed in a tree view labeled Search Results, with a folder for each type of object returned from the search. For example, if you were searching through both servers and spawners, and the search found only two spawners, only a spawners folder would be displayed.

To view the objects returned by the search, expand the folder or folders. Except for Refresh, New, and Delete, you can perform any operation in the Search Results tree view that you can perform in the main tree view. All operations in the property view are also available.

You can switch between the results of the last search and the main listing by selecting your choice from the tree view menu.

Each search matches against all the available objects, regardless of what is currently displayed. You cannot search the results of a prior search.

Getting Started

Reloading (Refreshing) IT Administrator Information

In certain situations, the information displayed on the Integration Technologies Admiinstrator window may become "out of sync" with the information stored on the LDAP server. For example, a user may use the Subscription Manager application to update a channel subscription while you have the IT Administrator application open; or another administrator may use IT Administrator or the directory console to update SAS server information while IT Administrator is open on your machine. To ensure that your IT Administrator window is displaying accurate information, use the **Refresh** tool periodically to load the most current information from the LDAP server.

To reload LDAP server data for all object types within an administration area:

1. In the manager bar, select the administration area whose information you want to reload (**Publish Framework**, **SAS Configuration**, **Archives**, **Application**s, or **SAS Data Sources**).
2. In the tree view, select the administration area's root node (**Publish Framework**, **SAS Configuration**, **Archives**, **Applications**, or **SAS Data Sources**).
3. Click the **Refresh** button ( ); or select **View** → **Refresh** from the menu bar. The LDAP information will be reloaded for all nodes in the tree view.

To reload LDAP server data for only a portion of an administration area:

1. In the manager bar, select the administration area that contains the information you want to reload (**Publish Framework**, **SAS Configuration**, **Archives**, **Applications**, or **SAS Data Sources**).
2. In the tree view, select the node that contains the objects or object types you want to reload.
3. Click the **Refresh** button ( ); or select **View** → **Refresh** from the menu bar. The LDAP information will be reloaded for all nodes and objects beneath the node you selected.

Getting Started

Administering SAS Servers

Using Integration Technologies, you can implement SAS object servers which use the Integrated Object Model (IOM) to deliver SAS functionality to clients. The IOM provides distributed object interfaces that are based on industry−standard technologies, including Microsoft's Distributed Component Object Model (DCOM) and the Object Management Group's (OMG) Common Object Request Broker Architecture (CORBA).

This section provides the information you need to set up, start, and administer an object server. The following tasks are documented:

- Choosing the appropriate server configuration for your installation:
    - A COM/DCOM server configuration, which enables client access using COM/DCOM; or
    - An IOM Bridge server configuration, which enables client access using the Integration Technologies IOM Bridge for COM or IOM Bridge for Java.
- Assigning logical names to groups of related object server resources.
- Setting up workspace pooling to improve the efficiency of connections between clients and servers.
- Setting up an object server using either a COM/DCOM configuration or an IOM Bridge configuration. These sections provide detailed instructions for creating the metadata to define your server configuration. Instructions are also provided for enabling and launching the server on the host machine, and for performing server administration and troubleshooting tasks.

Getting Started

Choosing a Server Configuration

Integration Technologies provides two types of server configurations:

- **COM/DCOM Server Configuration.** A COM/DCOM server configuration enables client access using the native Windows Component Object Model (COM) or Distributed Component Object Model (DCOM). In a client−server environment, DCOM must be enabled on both the client machine and the machine where the object server runs.
- **IOM Bridge Server Configuration.** An IOM Bridge server configuration enables client access using the Integration Technologies IOM Bridge for COM or IOM Bridge for Java. The IOM Bridge for COM allows you to develop native COM/DCOM applications that access server data on non−Windows platforms such as UNIX or z/OS. The IOM Bridge for Java allows you to develop applications using Java that access server data on either Windows or non−Windows platforms.

When to Use a COM/DCOM Server Configuration

You can use a COM/DCOM server configuration if:

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

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

For more information about COM/DCOM distributed clients, refer to Connecting Clients to IOM Servers in the *Integration Technologies Technical Overview*. For information on using Integration Technologies to set up a COM/DCOM server configuration, see Setting Up a COM/DCOM Server.

When to Use an IOM Bridge Server Configuration

You must use an IOM Bridge server configuration if:

- The object server will run on a non−Windows machine (for example, a UNIX−based machine); or if
- The object server will be accessed by Java client applications

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

For more information about the IOM Bridge for COM and the IOM Bridge for Java, refer to Connecting Clients to IOM Servers in the *Integration Technologies Technical Overview*. For information on using Integration Technologies to set up an IOM Bridge server configuration, see Setting Up an IOM Bridge Server.

Getting Started

Assigning Logical Names

Note: Version 9 does not support dependent spawners. If you are using Version 9, do not configure dependent spawners.

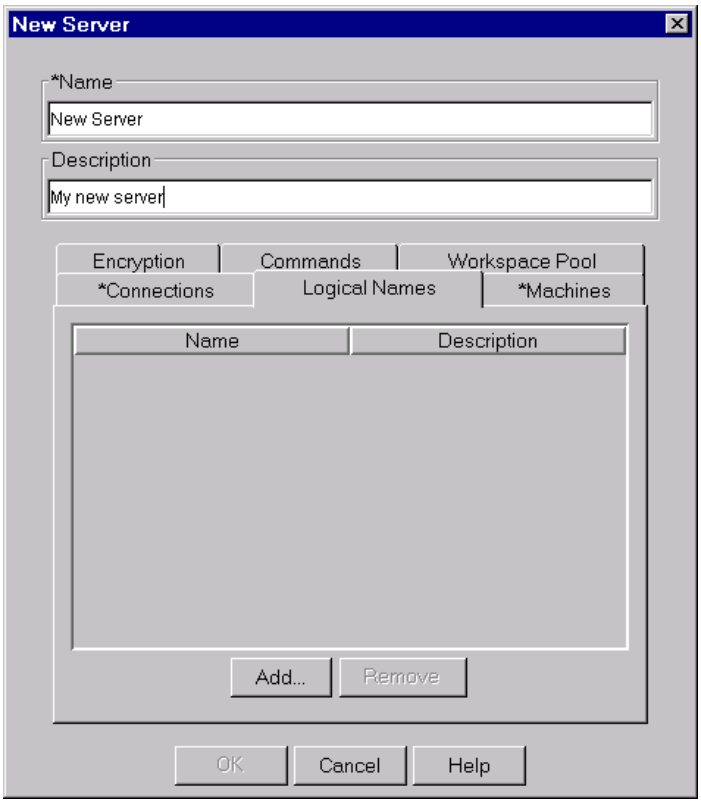
In Integration Technologies, a logical name is a unique name that you can assign to a group of related resources that are defined on the LDAP enterprise directory. Integration Technologies uses logical names to associate SAS object servers to related objects. You can use logical names to associate a server with:

- Spawners and SAS logins that support the server (these resources are used by IOM Bridge servers only)
- Paths for stored processes that reside on the server
- Archive paths for packages that are published to the server using the Publishing Framework
- SAS libraries, tables, and multi−dimensional databases (MDDBs) that reside on the server

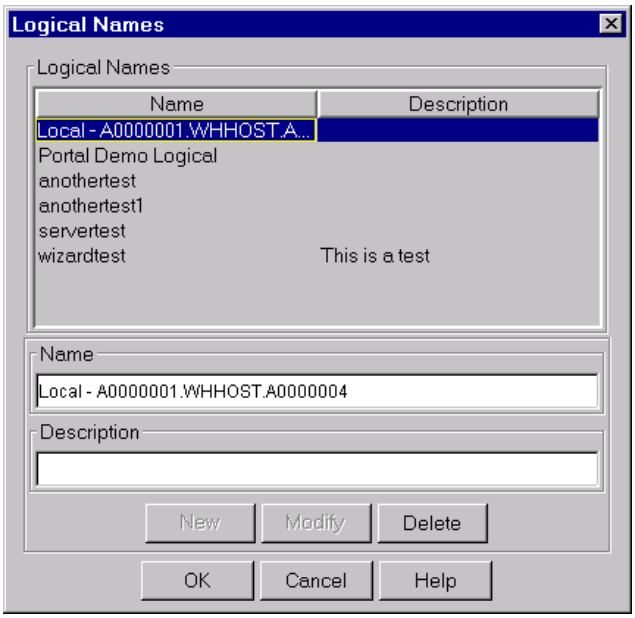
Any object can belong to more than one logical name grouping.

How to Assign Logical Names

When you create or modify an object using the IT Administrator interface, you can use the **Logical Names** tab to enter logical name assignments:



On this tab, click **Add** to assign a logical name to the object. The following window appears:



On this window, you can:

- **Assign an existing logical name to the object.** To assign an existing logical name, highlight the logical name you wish to assign, and click **OK**. The previous window will be displayed, with the logical name showing on the list. If you want to assign additional logical names, repeat this procedure.
- **Create a new logical name.** To create a new logical name, place your cursor in the Name field and type the new name. (The **New** button will become active.) Then type a description, and click **New**. The new logical name will appear in the list. To assign the new logical name to the object, highlight it and click **OK**.
- **Modify the description of an existing logical name.** To do so, highlight the logical name, place your cursor in the Description field, and type or edit the description. (The **Modify** button will become active.) To save the changes, click **Modify**.
- **Delete an existing logical name.** To do so, highlight the logical name and click **Delete**. The association will be removed from any objects that currently have this logical name.

SAS Logical Name Objects

When you create a new logical name in IT Administrator, a sasLogicalName object is created in LDAP. If you delete an object that a logical name is associated with, the sasLogicalNameInfo object will still exist

If you are not using an LDAP directory, you can use a configuration file to define a sasLogicalNameInfo object. For example configuration files, see Configuration File Example: Using Logical Names

Using the Logical View in IT Administrator

To see the logical name groupings that have been set up for SAS resources in your LDAP directory:

1. Open IT Administrator.
2. In the manager bar, click the types of objects whose logical names you would like to see (for example, click **SAS Configuration** to see SAS servers, spawners, and logins).
3. Select **Logical View** from the tree view menu.
4. Click the plus sign (+) to open the Logical View folder.

5. Click the plus sign (+) to open the Names folder. The tree expands to show all of the logical names that would appear in a dialog box if you were assigning a logical name to an object.

If a logical name appears next to a folder icon, this means that the logical name has been assigned to one or more resources in the management area you selected. (For example, if you are in the Configuration management area, folder icons will appear next to each logical name that has been assigned to SAS servers or spawners.) To see the resources that have been assigned to a particular logical name, click on the plus sign next to the logical name folder.

Logical names that are currently not assigned to any of the resources in the management area you selected will appear in the tree as terminated nodes rather than as folders. (For example, if you are in the SAS Archiving management area, and none of the SAS archives has been assigned to a logical name, all of the logical names will appear as terminated nodes.)

Getting Started

Overview of Pooling

A workspace pool is a group of workspaces that are created and ready for use on one or more servers. Workspace pooling improves the efficiency of connections between clients and servers because SAS processes remain active between uses of SAS and can be reused by multiple clients within the same client process. For both the pooling and non−pooling cases, the client uses the workspace as long as required. With pooling, when the client is finished with the workspace, the process that supports the workspace stays active and can be used by another client. Without pooling, a new SAS process must be created for each client connection.

An authorized application (such as the SAS Information Delivery Portal), which will be referred to here as the pool administrator, runs and controls access to the pool. The pool administrator uses its credentials to connect and authenticate itself to the LDAP directory, giving it authorization to create workspaces. The method that the pool administrator uses to authorize clients to use workspaces depends upon whether the pool is set up for exclusive use by a single client or shared access by many clients.

If a pool is set up for use by a single client, the client uses its own credentials to access the SAS server and SAS login definitions the client is allowed to see. The client can then use attributes on these objects to create a workspace or a pool of workspaces. This scenario is useful for development environments.

If the pool is to be used by many different clients, two main problems arise:

- Workspaces may need to be created before the clients are known
- Workspaces created with one client's login may be accessible by other clients

To ensure security, the pool administrator must have some method of verifying authorization. The method provided with the Integration Technologies Administrator is the use of the Client DN attribute on the SAS login. When the pool administrator issues the request for the workspace, it includes the client's distinguished name as part of the request. The request is accepted only if the client's distinguished name either exactly matches the Client DN attribute or is a member of a group specified in the Client DN attribute. Regardless of the method used, when the client has been authorized to the pool administrator, the administrator uses the pool manager to allocate a workspace from the pool to the client. When the client is finished with the workspace, it releases the workspace to return to the pool, where it is then available for other clients.

A workspace pool consists of a SAS login and one or more SAS servers that have a common

- Base distinguished name
- Logical name
- Domain name

In addition, the servers must all be accessible under the same access credentials.

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

For information about where to specify the parameters that are needed to set up Integration Technologies pooling, see Locations for Specifying Pooling Parameters.

Getting Started

Locations for Specifying Pooling Parameters

Java Clients

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

- LDAP Server. For instructions, see Using the Integration Technologies (IT) Administrator.
- source code. For information about providing parameters information in the Java client source code, see Using Connection Pooling with Java in the *Developer's Guide*.

Windows Clients

For Windows clients using a COM/DCOM server connection, you should specify pool parameters in the source code.

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

- LDAP Server. For instructions, see Using the Integration Technologies (IT) Administrator.
- source code. For information about providing pooling parameters in the Windows client source code, see Using Connection Pooling with Windows in the *Developer's Guide*.

Getting Started

Setting up Workspace Pooling

You can use the Integration Technologies Administrator to create a workspace pool. See Using the Integration Technologies (IT) Administrator. To create a workspace pool, perform the following steps:

1. Display or create the definitions for the SAS servers that will host the workspaces in the pool.
2. Select the Workspace Pool tab. In the Maximum Workspaces per Workspace Pool field, enter the maximum number of workspaces that you want to allocate to each workspace pool on that server. Factors you should consider when determining a value for this field include the number and type of processors on the machine, the amount of memory present, the type of clients that will be requesting workspaces, and the number of different pools the server participates in.

   In the Recycle Activation Limit field, specify the number of times a server is used before the process is disposed of and a new process is used in pooling. A value of 0 indicates that the process will have no limit. The default value is 0.

   In the Server Process Shutdown pane, specify whether an idle server should be shut down or remain running (including the number of minutes before the server shuts down).
3. Display or create a definition for a SAS login to access the server. This login will supply the credentials necessary to connect to the workspace pool. On the SAS login definition, specify the following:

   - **Domain name:** The domain name must be the same as that specified on the definition for the SAS server that contains the workspace pool.
   - **Logical name:** The logical name on the SAS login definition must be the same as the logical name on the definition for the SAS server.
   - **Client DN:** The pool administrator may supply the client's distinguished name as part of its request for a workspace. The request will be granted only if the client's distinguished name either
     - ◊ Matches exactly the Client DN for the pool, specified in this field
     - ◊ Is a member of the group whose distinguished name is specified in this field

     This field is optional.
   - **Min Workspace Size:** The minimum number of workspaces (total of active and idle) that should be present in the pool at any time. Note that the workspaces may be distributed among several SAS servers. This value cannot exceed the total of the Maximum Workspaces per Pool for all the servers with which this login can connect. This field is optional.
   - **Min Available Workspaces:** The minimum number of workspaces that should be idle and available in the pool at any time. As with the Min Workspace Size field, this value cannot exceed the total of the Maximum Workspaces per Pool for all the servers with which this login can connect. This field is optional.

COM/DCOM

Setting Up a COM/DCOM Server: Introduction

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

Since COM launches the SAS object server, spawners are not used in the COM/DCOM server environment. However, you can (and should) use the Version 9 Object Manager (or Version 8 Workspace Manager) to obtain a DCOM server. The server definitions can be administered through LDAP or provided in a configuration file. Alternatively, you can imbed the server information in your application program.

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

When to Use a COM/DCOM Server Configuration

You can configure an object server as a COM/DCOM server if:

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

If you use a Java client, you must use an IOM Bridge server configuration instead.

COM/DCOM

Server and Client Requirements

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

Server Requirements

Install the following software on the server machine:

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

COM/DCOM

Summary of Setup Steps (COM/DCOM Server)

Standalone Windows Development Machine

To set up a standalone Windows development machine, just install SAS Version 9 (including Integration Technologies) on the machine. On Windows, the SAS Integration Technologies Client is installed with base SAS software.

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

For more information about developing Windows client applications, see Windows Clients in the *Developer's Guide*.

Separate Client and Server machine

To set up a the server machine:

1. Install SAS Version 9 (including Integration Technologies) on the server machine. Refer to the SAS documentation for the details of this procedure.

   Note: If you are using the Integration Technologies client with 64−bit SAS, extra setup steps are required for IOM COM servers on 64−bit Windows. For details, see the SAS installation documentation.
2. Enable DCOM on the server machine. For details, see Enabling DCOM on the Server and on the Client.
3. Edit your SAS CONFIG file (SASV9.CFG) for use with DCOM. For details, see Configuring SAS for DCOM.
4. Set SAS launch policies on the server. You can set global policies that affect all COM−enabled applications, or set application policies for individual to grant permissions to users and groups specifically for accessing and launching the object server. For details, see Setting SAS Permissions.
5. Before attempting to run a COM/DCOM application, test the client−server connection by using the tips provided in Troubleshooting the DCOM Connection.
6. If your applications need to access metadata that describes your COM/DCOM server configuration, you must create the necessary definitions for server objects and (optionally) logical name objects. For details, see Creating the Metadata for a COM/DCOM Server. The method for creating the metadata depends on whether you are using an LDAP server for your metadata repository:

   - If you are using an LDAP server, you can use IT Administrator to create the necessary metadata. For details, see Using the IT Administrator Wizard to Define a COM/DCOM Server or Using IT Administrator to Define a COM/DCOM Server.
   - If you are not using an LDAP server, you must create a configuration file that contains the necessary metadata, and then install the configuration file on the server machine. For details, see Using a Configuration File to Define the Metadata (COM/DCOM).

To set up the client machines:

1. On each client machine, enable DCOM. For details, see Enabling DCOM on the Server and on the Client.
2. On each client machine, install the client software, either as part of the installation of a pre−written application or as a separate installation of a custom application. For details about installing custom applications, refer to Developing Windows Clients in the *Developer's Guide*. If you are not using an LDAP

server and need metadata definitions, you may also need to copy the configuration file (created in Step 5) to the client machine.

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

COM/DCOM

Metadata Overview (COM/DCOM)

The metadata that supports Integration Technologies consists of objects, each defined by a collection of attributes that define the object. If you require metadata, the metadata for a COM/DCOM server configuration must include a sasServer object, which contains startup and connection information for a particular instance of a SAS object server.

For each instance of an object server, the following information is defined in the metadata:

- Server name
- Machine name
- Logical name
- Connection information
- Encryption and pooling information, if required.

For detailed information about the attributes included in the metadata for a server, see the sasServer Attributes List.

The **sasLogicalNameInfo** object is created automatically if you use IT Administrator to assign a logical name to a server. Logical names are used to create resource groupings.

COM/DCOM

Creating the Metadata for a COM/DCOM Server

If your applications need to access metadata from the LDAP server or from a configuration file, you must create the metadata that describes your COM/DCOM server configuration. If you require metadata, use the appropriate method depending on whether you use an LDAP server or a configuration file to store your metadata:

- **If you are using an LDAP server:**
    - ♦ You can use the IT Administrator Wizard to create definitions for server and logical name objects. For instructions, see Using the IT Administrator Wizard to Define a Server (COM/DCOM).
    - ♦ You can use the IT Administrator interface to create and modify the object definitions. For instructions, see Using IT Administrator to Define an Object Server (COM/DCOM).
- **If you are *not* using an LDAP server,** you can create and install configuration files that contain the object definitions. For instructions, see Using a Configuration File to Define the Metadata.

  Note: If your configuration requires more than one or two servers, or if multiple clients will be using the servers, we strongly recommend the use of LDAP as a central metadata repository. The use of LDAP also gives you the ability to use access control lists to control access to the servers in your enterprise.

COM/DCOM

Using the IT Administrator Wizard to Define a Server (COM/DCOM)

The SAS Integration Technologies Administrator provides a wizard to guide you through the process of creating LDAP−based metadata for a COM/DCOM server. (Alternatively, you can create this metadata using the regular IT Administrator interface.) For general information about IT Administrator, refer to Using the Integration Technologies (IT) Administrator.

Before beginning this procedure, be sure that you have:

- Installed an LDAP directory server and configured it for use with SAS software. For detailed instructions, refer to Setting up an LDAP Directory Server.
- Installed the Integration Technologies (IT) Administrator application. For details, see IT Administrator Installation and Startup.

To define a COM/DCOM server object using the wizard:

1. Start IT Administrator.
2. Select the **SAS Configuration** button in the Manager Bar.
3. Click the **Wizard** button (). The welcome screen for the wizard appears.
4. On each screen of the wizard, follow the instructions given; then select the **Next** button to move to the next screen. The **Next** button remains grayed out until you enter the required information. If you need to change information you have already entered, select the **Back** button. Select the **Help** button on any screen of the Wizard to receive instructions for the screen currently displayed. Select the **Cancel** button to exit the wizard at any time; if you select **Cancel**, no server object will be created.

   Each wizard screen is described below. For detailed information about a field, click on the corresponding attribute name, which is shown in parentheses. The appropriate definition on the sasServer Attributes List will be displayed. You can then use your browser's **Back** button to return to this page. The wizard screens are as follows:

   - **Select a Logical Grouping**. On this screen, select a defined logical name (sasLogicalName) from the list. If you want to create a new logical name, select the **Create** button; the wizard will help you create a new sasLogicalNameInfo object. For more information about logical names, refer to Assigning Logical Names.
   - **Create a New Server**. On this screen, enter a server name (sasServercn), (optionally) a description (description), and a domain (sasDomainName).
   - **Specify a Protocol**. On this screen, select the **COM/DCOM** protocol (sasProtocol).
   - **Specify COM Host Information**. On this screen, specify a fully−qualified host name (sasMachineDNSName).
   - **Specify COM Encryption Settings**. On this screen, select the Authentication (sasRequiredEncryptionLevel) and DCOM Security Service from the drop−down menus.
   - **Done**. On this screen, you can:

     ◊ Select the **Do Another** button if you want to define another server object.
     ◊ Select the **Add Host** button if you want to specify another host machine for this server object.
     ◊ Select the **Finish** button if you want to exit the wizard and save your work.

     If you select the **Cancel** button, the server object will not be created.

When you complete all of the steps in the above procedure, your LDAP directory will contain fully defined sasServer and sasLogicalNameInfo objects. You can now begin using the server.

COM/DCOM

Using IT Administrator to Define a Server (COM/DCOM)

The SAS Integration Technologies Administrator provides a graphical user interface that allows you to create or modify a definition for a COM/DCOM server. (Alternatively, you can use the IT Administrator wizard to perform the initial creation. For details, see Using the IT Administrator Wizard to Define a COM/DCOM Server.)

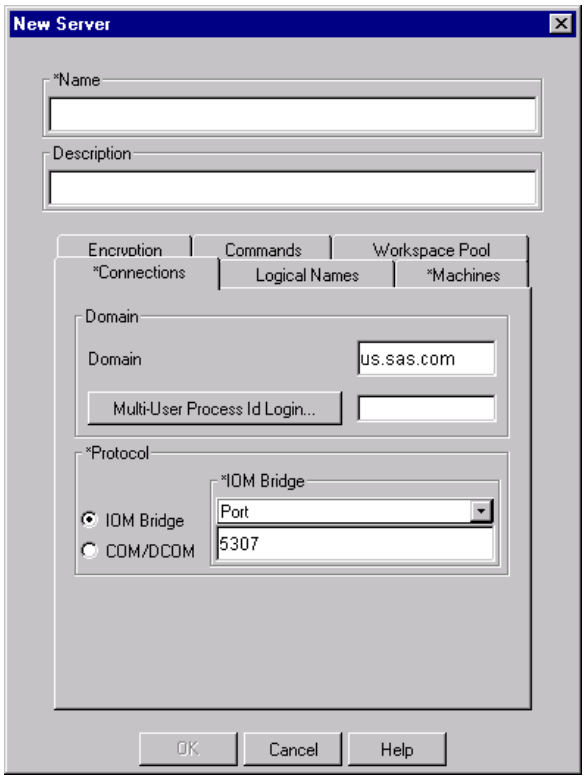
Note: The following attributes which appear in the Commands tab of the IT Administrator interface are not used in Version 9 of SAS Integration Technologies

- Transaction Program Name
- Partner Logical Unit Name

If you are using Version 9, do not use these attributes for your configuration.

To define a COM/DCOM server object using IT Administrator:

1. Open IT Administrator.
2. In the manager bar, select **SAS Configuration**.
3. In the tree view, select the **Servers** folder (found under the **SAS Servers** folder); then select the **New** button ( ) on the toolbar. Alternatively, you can select **File** → **New** → **Server** from the menu bar. The following window appears:



4. Enter the necessary attributes. The attribute fields that are marked with an asterisk (*) are required. The **OK** button will remain greyed out until you have entered all of the required fields. Select the **Help** button on any tab to display entry instructions. Brief entry instructions are provided below. For detailed information about a field, click on the corresponding attribute name, which is shown in parentheses. The appropriate definition on the sasServer Attributes List will be displayed. You can then use your browser's **Back** button to return to this page.

- Enter a unique name (sasServercn) for the server. Optionally, enter a description (description).
- On the **Connections** tab:

    a. Enter the Domain (sasDomainName).
    b. Select the **COM/DCOM** protocol (sasProtocol).
    c. From the pull−down menu that appears, select either **Port** or **Service** and enter either the service (sasService) or the port number (sasPort) in the field below the menu. The port number is required if the server will have Java clients.
- On the **Logical Names** tab, select one more logical names (sasLogicalName) that this server is to be associated with. If you want to create a new logical name, select the **Add** button to create a new sasLogicalNameInfo object. For more information about logical names, refer to Assigning Logical Names.
- On the **Machines** tab, select the **Add** button, and then enter the fully qualified host name (sasMachineDNSName) for the machine on which the server is to run. Repeat for each additional host machine. To change an entry, highlight the machine name and select **Edit**. To remove an entry, highlight the machine name and select **Remove**.
- On the **Encryption** tab, select the Authentication (sasRequiredEncryptionLevel) and DCOM Security Service from the drop−down menus.
- You can use the **Workspace Pool** tab to set up workspace pooling for the server. Specify the Maximum Workspaces per Workspace Pool (sasMaxPerWorkspacePool) and the Recycle Activation Limit (sasMaxPerWorkspacePool) in the fields provided. (Select the **Default** button if you want to reset the Recycle Activation Limit to its default value of 10.) Then select **Leave running when idle** (sas−ServerRunForever); or select **Minutes until idle shutdown** and enter the number of minutes (sas−ServerShutdownAfter) after which an idle server should be shut down.

5. When you are finished entering information in the fields, select **OK**. The new server object appears in the tree view.

To modify a COM/DCOM server object using IT Administrator:

1. Open IT Administrator.
2. In the manager bar, select **SAS Configuration**.
3. In the tree view, find the **Servers** folder (found under the **SAS Servers** folder) and click the plus sign to open it.
4. Select the server object that you wish to modify. The server's current attributes will be displayed in the property view in the right portion of the window.
5. Select the appropriate tabs, and enter the necessary changes. For a description of the fields, refer to the sasServer Attributes List.
6. When you are finished, select the **Save** icon ( ) on the toolbar; or select **File → Save** from the menu bar. (If you skip this step, IT Administrator will prompt you to save your changes when you attempt to navigate to another object.)

COM/DCOM

Using a Configuration File to Define the Metadata (COM/DCOM)

For LDAP, if you do not use a metadata server as the metadata repository, you can create a flat configuration file that contains the object definitions for an COM/DCOM server configuration. The configuration file must then be installed on the server and on each client machine.

Note: If your configuration requires more than one or two servers, or if multiple clients will be using the servers, we strongly recommend the use of LDAP as a central metadata repository. The use of LDAP also gives you the ability to use access control lists to control access to the servers in your enterprise.

To define a COM/DCOM server configuration using a configuration file:

1. Use a text editor to code the configuration file. At a minimum, the file must define a server object. You can also define one more SAS logical name objects. To create the file:
    - Refer to the attribute descriptions for each object type:
        - Attributes for sasServer
        - Attributes for sasLogicalName
    - Refer to the following examples:
        - Example Minimal Configuration
        - Example Using Logical Names
    - Use the LDAP Data Interchange Format (LDIF), format, which has the following syntax rules:
        - Start each entry in column one.
        - To indicate a comment line, place '#' in column one.
        - Use the following general format for each entry: "attribute: value.".
        - If an entry spans multiple lines, insert a blank in the first column of each continuation line. The blank in column one is a continuation character and is consumed by the LDIF file parser. Therefore, it should not be considered part of the entry.
        - A blank line must precede a distinguished name (exclude 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 spawner's LDIF parser relies on this requirement in order to separate object class definitions. The DN name can be any value.
        - Two consecutive blank lines indicate the end of the configuration file definitions.
2. Save the file with a name of your choice.
3. Install the file on the server machine and on each client machine.

You can now access the server using the Object Manager (or Version 8 Workspace Manager). For instructions, refer to Using the Object Manager or Using the Workspace Manager in the *Developer's Guide*.

COM/DCOM

Configuration File Example: Minimal Configuration

The following LDIF file can be used as a minimal configuration file for your IOM COM server. It contains a definition for the object server.

```
#
# Object Server Definition
#
dn: cn=Finance,o=AlphaliteAirways,c=US
objectClass: sasServer
sasServercn: mySASObjectServer
sasLogicalName: myLogicalName
sasMachineDNSName: localhost
sasProtocol: com
```

COM/DCOM

Configuration File Example: Using Logical Names

Logical names provide a mechanism to identify similar functionality. They are specified via the sasLogicalName attribute.

For instance, your installation may want to stage a new application without altering its production applications. To do this, the SAS Object Server definitions specify the same logical name. Here is a configuration file that illustrates how this is accomplished:

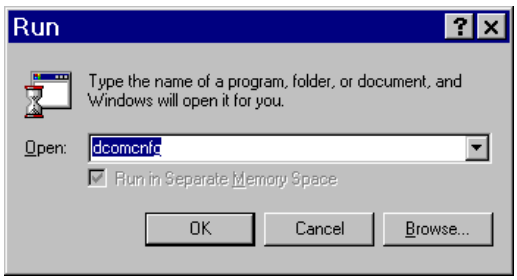
```
#
##  Define our production MyApplication SAS Object Server
#
dn: sasServercn=MyApplication,sascomponent=sasServer,cn=SAS,o=ABC Inc,c=US
objectClass: sasServer
sasServercn: MyApplication
sasDomainName: mvs.abc.com
sasLogicalName: stage
sasMachineDNSName: bigiron.mvs.abc.com
sasProtocol: com

#
##  Define our test MyApplication SAS Object Server
#
dn: sasServercn=testApplication,sascomponent=sasServer,cn=SAS,o=ABC Inc,c=US
objectClass: sasServer
sasServercn: testApplication
sasDomainName: mvs.abc.com
sasLogicalName: stage
sasMachineDNSName: bigiron2.mvs.abc.com
sasProtocol: com
```

COM/DCOM

Enabling DCOM on the Server and the Client

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

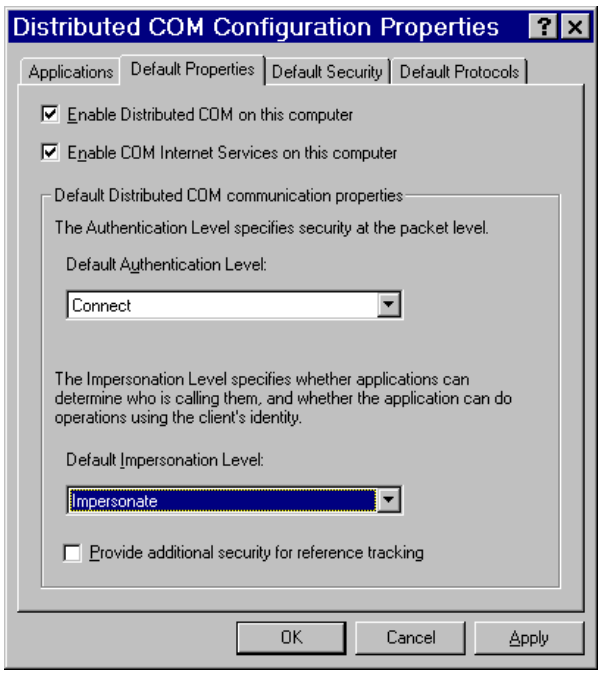
1. From the Windows Taskbar, click **Start** ➔ **Run**.
2. Type `dcomcnfg`, as shown in the illustration.



3. Click **OK**. The dialog box that appears depends on the Windows operating system you are using:

   - If you are using Windows NT/2000, the Distributed COM Configuration Properties dialog box appears.
   - If you are using Windows XP, the Component Services dialog box appears. Expand the Component Services folder, expand the Computers folder, then right−click on My Computer and select Properties.

4. Select the **Default Properties** tab.



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

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

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

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

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

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

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

COM/DCOM

Configuring SAS for DCOM

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

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

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

Default Lines from SASV9.CFG:

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

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

Recommended Change:

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

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

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

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

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

COM/DCOM

Setting SAS Permissions on the Server (COM/DCOM)

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

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

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

These methods are discussed in the following sections:

- Setting Default COM Security on Windows NT/2000
- Setting Permissions per Application on Windows NT/2000
- Setting Default COM Security on Windows XP
- Setting Permissions per Application on Windows XP

COM/DCOM

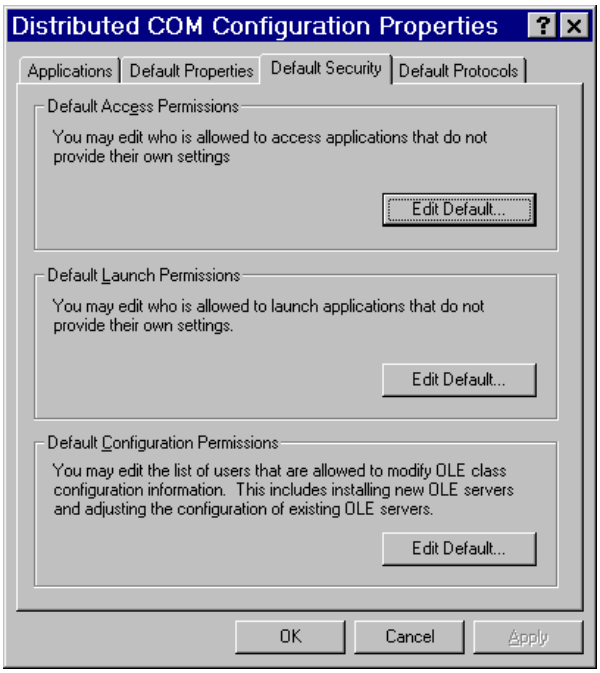
Setting Default COM Security on Windows NT/2000

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

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

To set default COM security on Windows NT/2000:

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

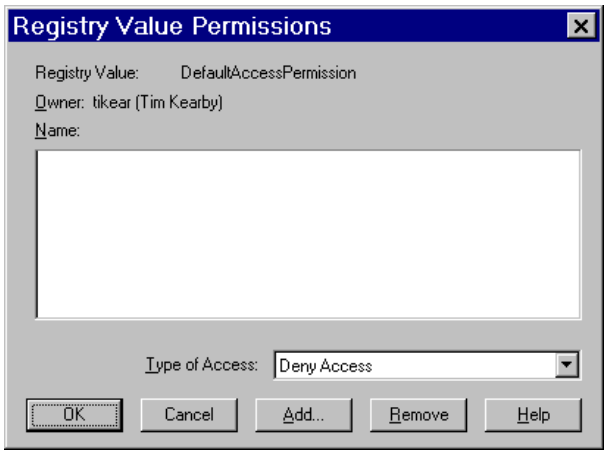


4. From the Default Security tab, you must edit the Default Access Permissions and the Default Launch Permissions. (The Default Configuration Permissions are adequate for a development environment). For details, see
   - ♦ Global Access Permissions
   - ♦ Global Launch Permissions
   - ♦ Global Configuration Permissions

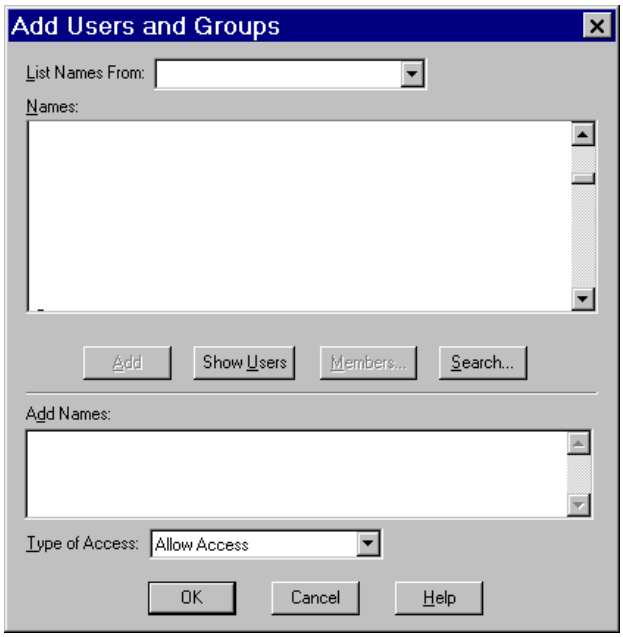
Global Access Permissions

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

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



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

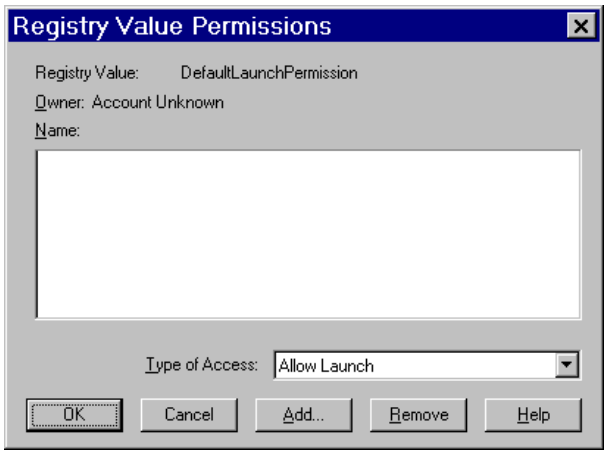


3. Use the Add Users and Groups dialog box to identify the users and groups at your site and the type of access (allow or deny access). You should also give access permission to System. For field descriptions, refer to the Windows NT or Windows 2000 Help. When you are finished, click **OK** and then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

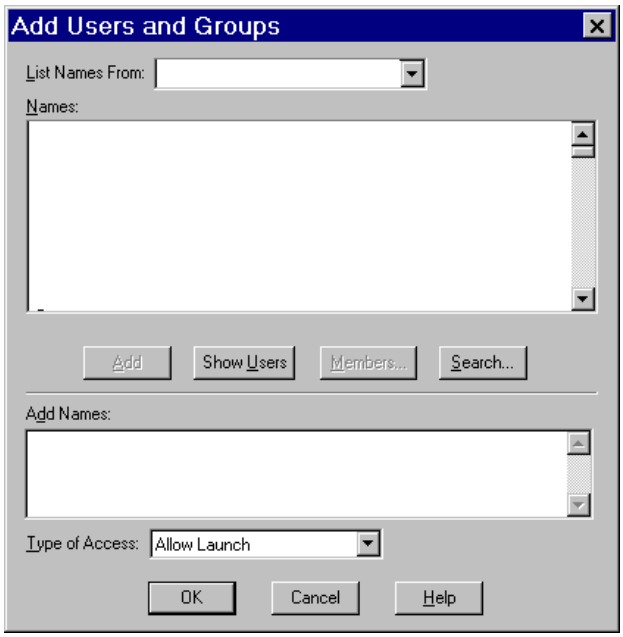
Global Launch Permissions

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

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



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



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

Global Configuration Permissions

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

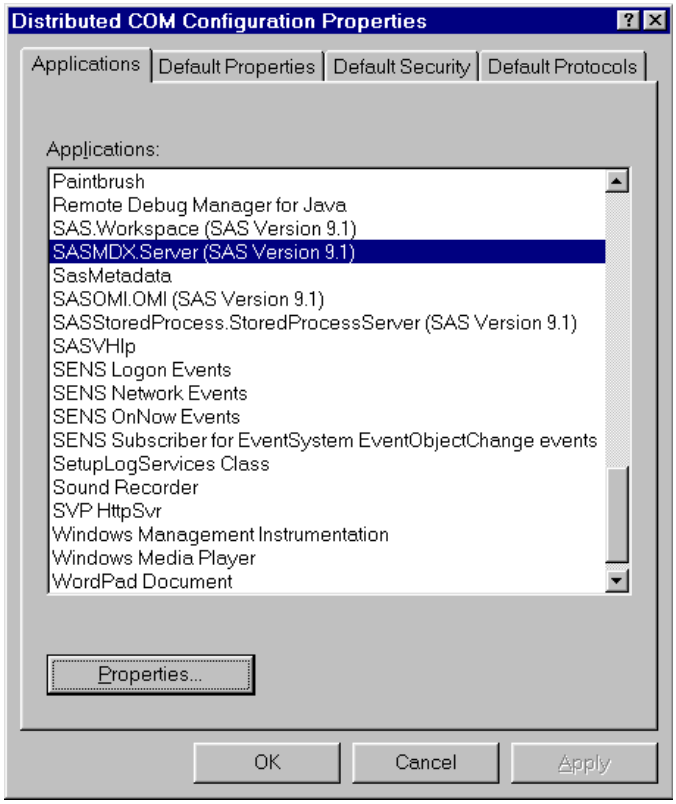
1. If you need to specify more restrictive configuration permissions, from the Default Security tab of dcomcnfg, click **Edit Default** in the Default Configuration Permissions box. Consult the Windows NT or Windows 2000 Help for further information.
2. When you are finished, click **OK** to save the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

COM/DCOM

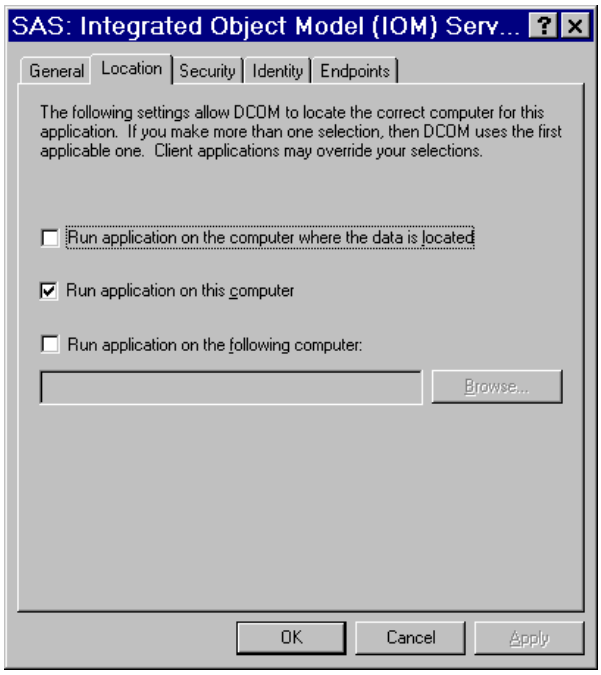
Setting Permissions per Application on Windows NT/2000

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

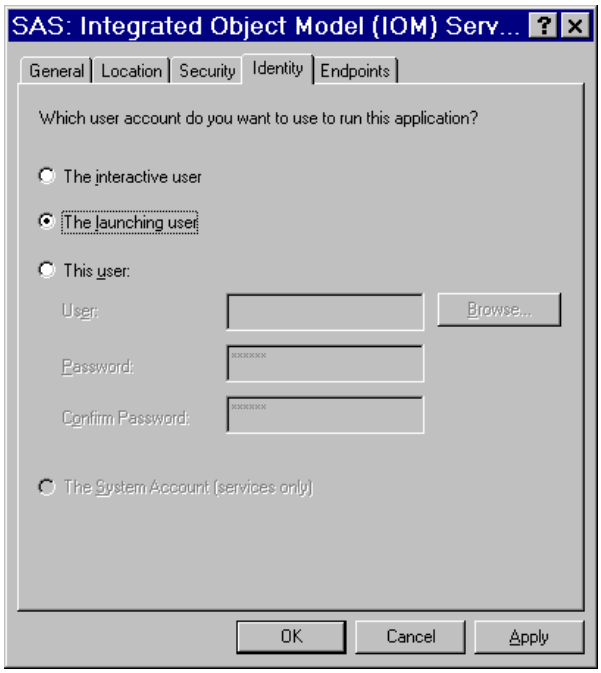
1. From the Windows taskbar, click **Start** ➔ **Run**.
2. Type `dcomcnfg` and click **OK**. The Distributed COM Configuration Properties window appears.
3. Select the Applications tab:



   This tab shows the AppID description for each DCOM server that can be run on your machine. (The AppID GUID is shown for servers that register without a description.)
4. Locate the IOM server that you are configuring and select it. For example, if you want to set policies for the Workspace, select **SAS.Workspace (SAS Version 9.1**. The application listing differs depending on which version of SAS is installed. See AppIDs for Configuring DCOM to determine which AppID to look for.
5. After you highlight the selection, click on the **Properties** button. The Properties dialog box for the server object appears.
6. Select the Location tab.



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



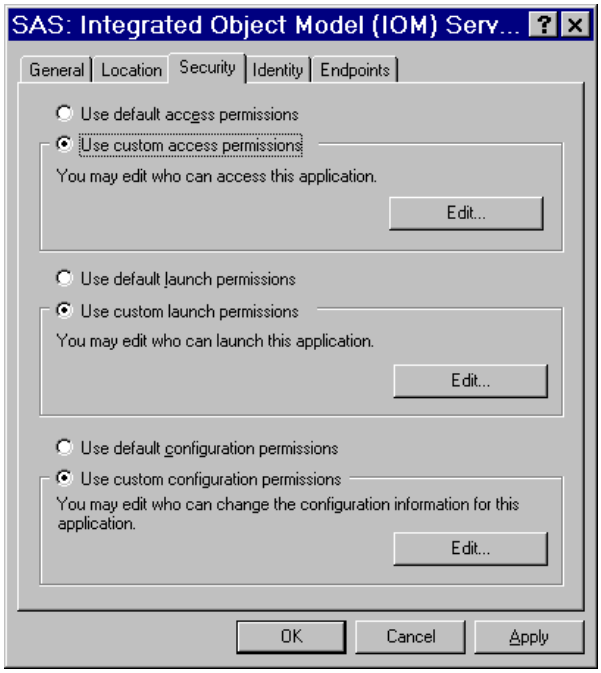
9. Select the identity based on the type of server:

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

     CAUTION: Support for the use of COM in the SAS Metadata Server is experimental in SAS 9.1. Do not use COM in the SAS Metadata Server in production jobs.
   - For SAS Workspace Servers, check the desired default identity setting. For maximum security, select the option **The launching user**.

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

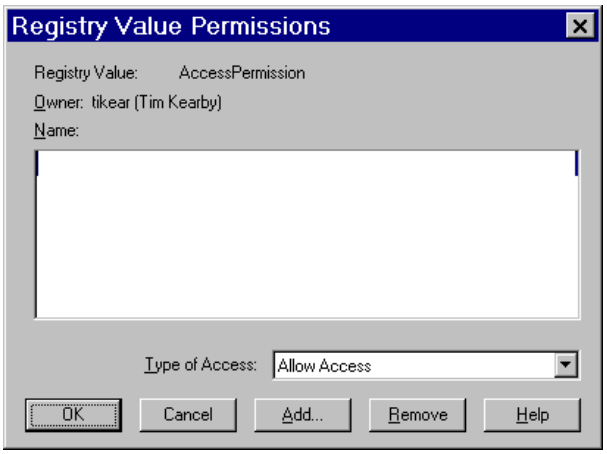
10. Select the Security tab.



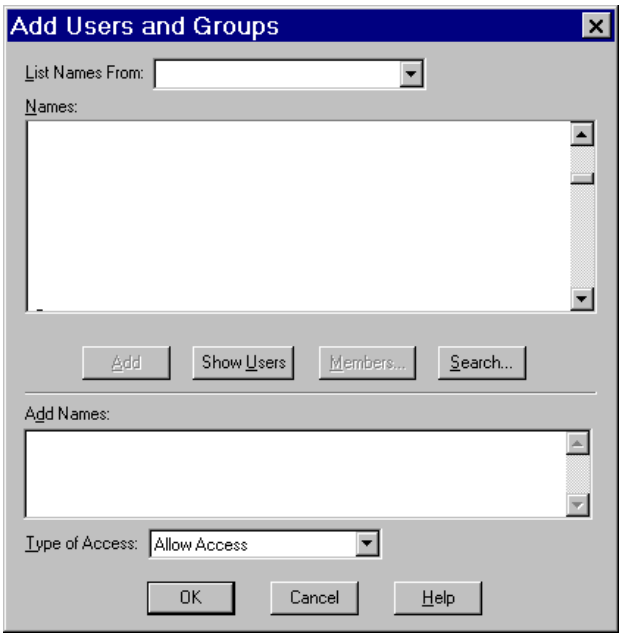
11. If you want to rely on the system−wide default access permissions, select **Use default access permissions**, click **Apply**, and then continue with Step 12.

    If you want your IOM server application to have its own set of access permissions:

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



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



c. Use this dialog box to grant users and groups access to SAS through DCOM. You should also give access permission to System. (For field descriptions, refer to the Windows NT Help.) You can also identify users and groups that are denied access to SAS by changing the selection in Type of Access.

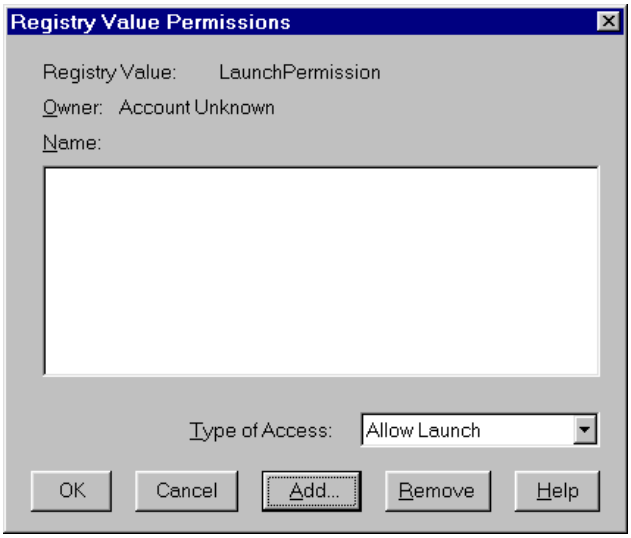
d. When you are finished, click **OK** in the Add Users and Groups dialog box, and then click **OK** in the Registry Value Permissions dialog box.

12. If you are configuring a Workspace server, which is launched by COM, you will also need to choose your launch permissions. It is recommended that they be the same as the access permissions; additionally, ensure that the **System** account has launch permissions.

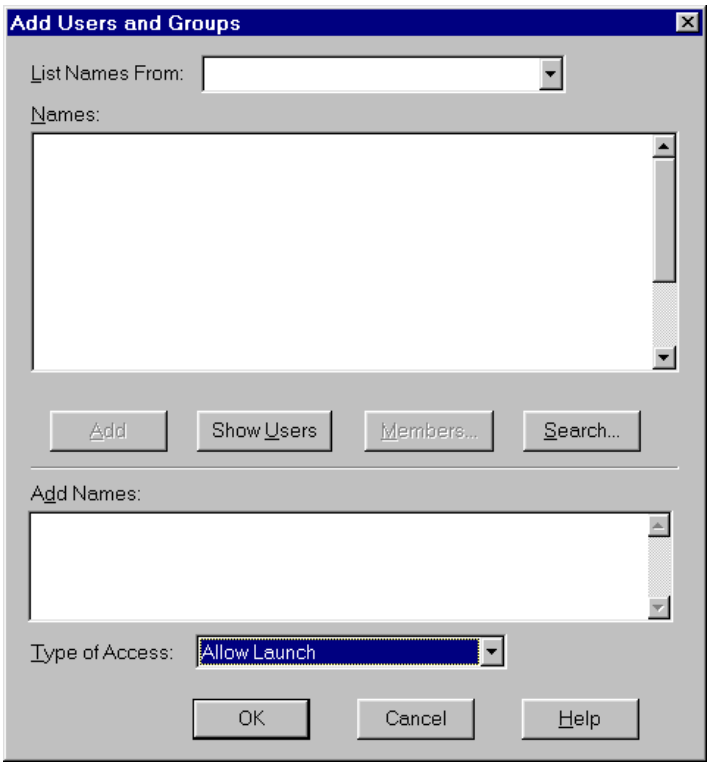
    If you want to rely on the system−wide default launch permissions, select **Use default launch permissions,** click **Apply**, and then continue with Step 13.

    If you want your IOM server application to have its own set of launch permissions:

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



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



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

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

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

COM/DCOM

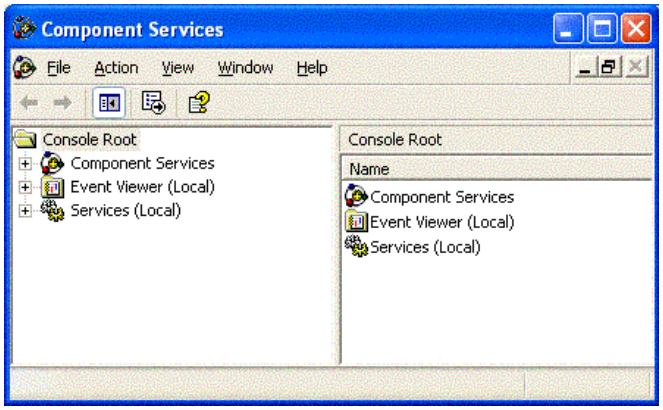
Setting Default COM Security on Windows XP

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

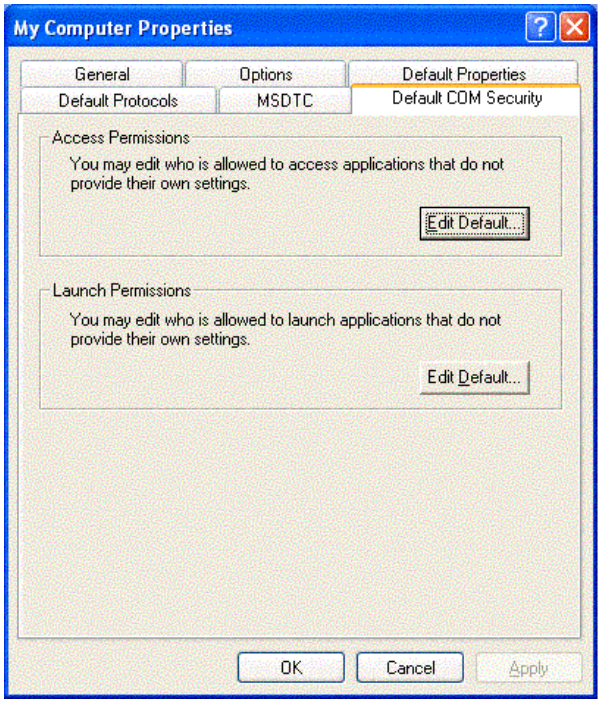
- If authentication is used, client machines that receive events from a DCOM server must include "Everyone" in the default access permissions.
- If authentication is not used, client machines must specify "None" in the default access permissions.

To set default COM security on Windows XP:

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



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

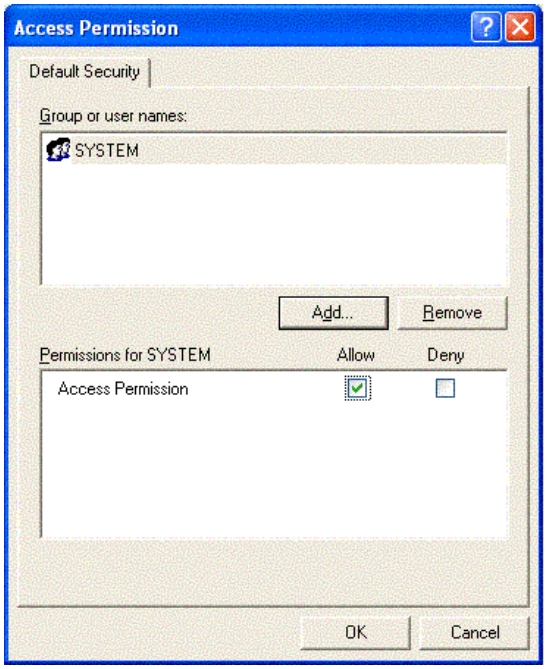


5. From the Default COM Security tab, you must edit the Access Permissions and the Launch Permissions. For details, see

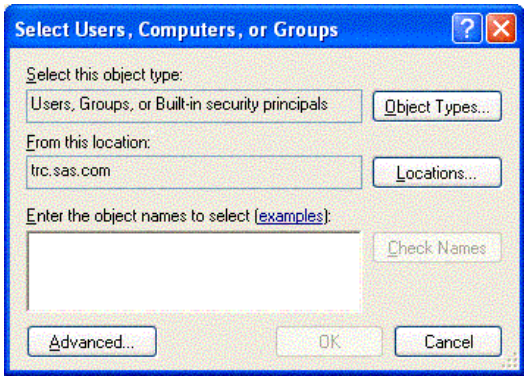
Global Access Permissions

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

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



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

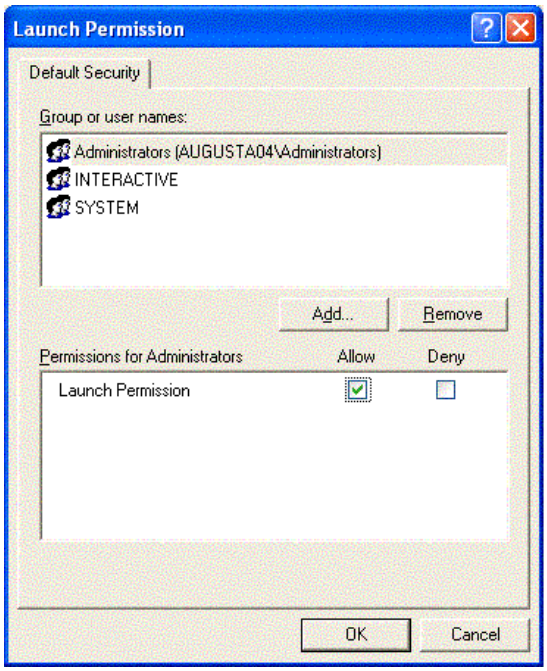


3. Use the Select Users, Computers, or Groups dialog box to identify the users and groups at your site and the type of access (allow or deny access). You should also give access permission to System. For field descriptions, refer to the Windows XP Help. When you are finished, click **OK** and then **OK**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

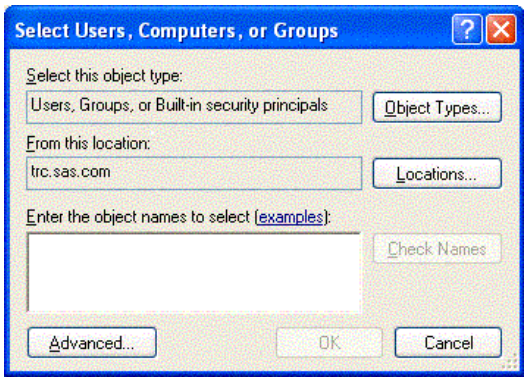
Global Launch Permissions

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

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



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



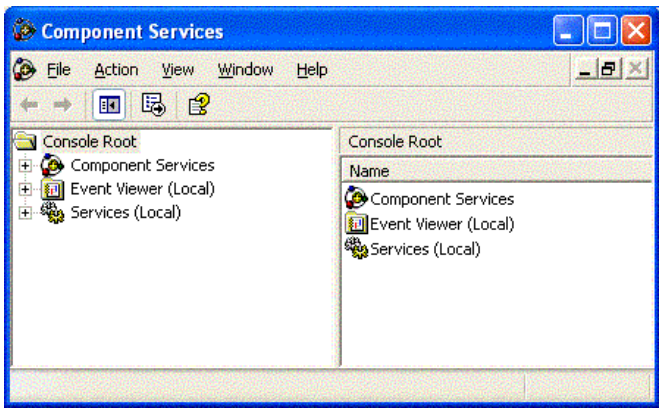
3. Use the Select Users, Computers, or Groups dialog box to identify users and groups at your site and the type of access (allow or deny launch access). It is recommended that you enter the same values that you entered for the Access Permissions. You should also give launch permission to System. When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

COM/DCOM

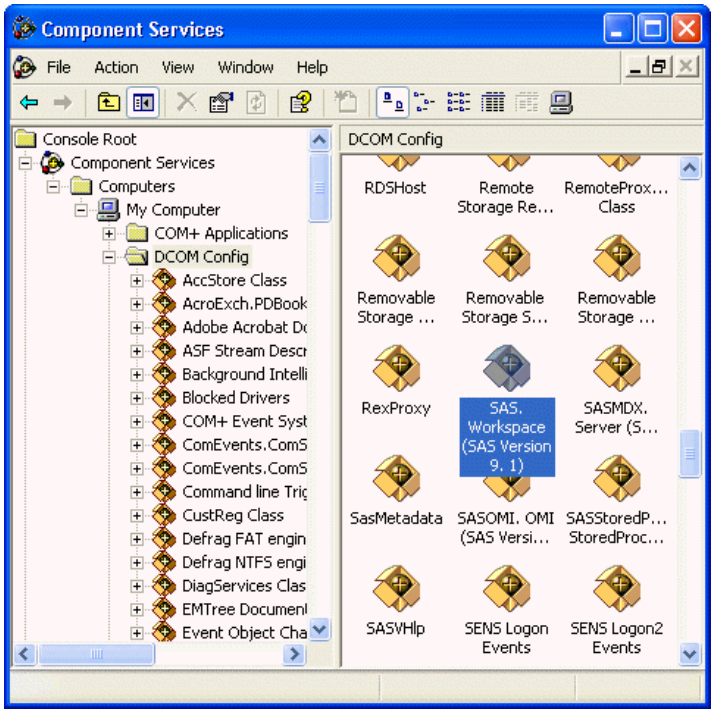
Setting Permissions per Application on Windows XP

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

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



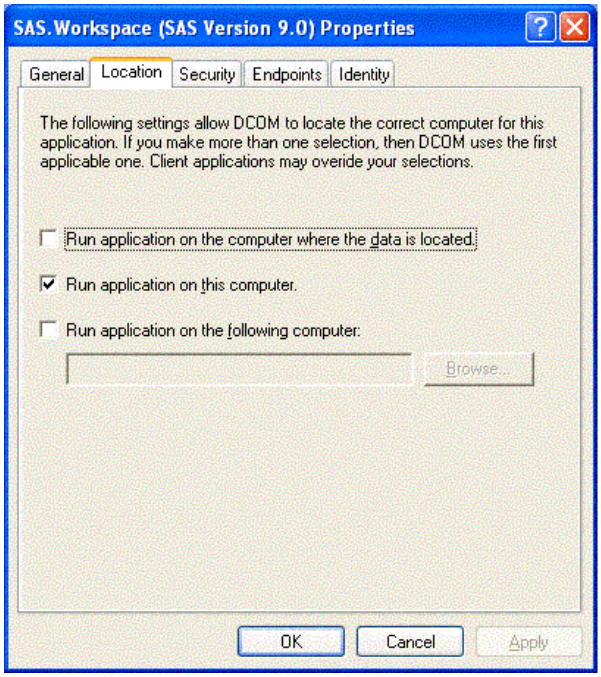
3. Expand the Component Services folder, expand the Computers folder, expand the My Computer folder, and then expand the DCOM Config folder.



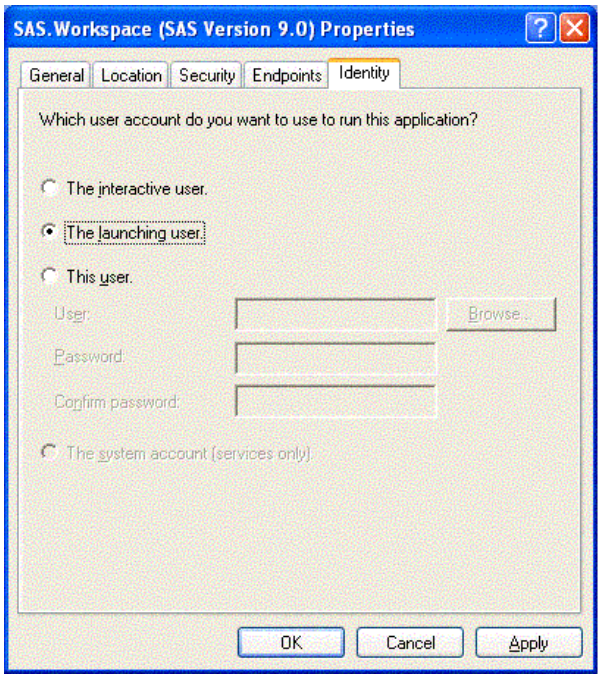
This view shows the AppID description for each DCOM server that can be launched on your machine. (The AppID GUID is shown for servers that register without a description.)

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

6. Select the Location tab.



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



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

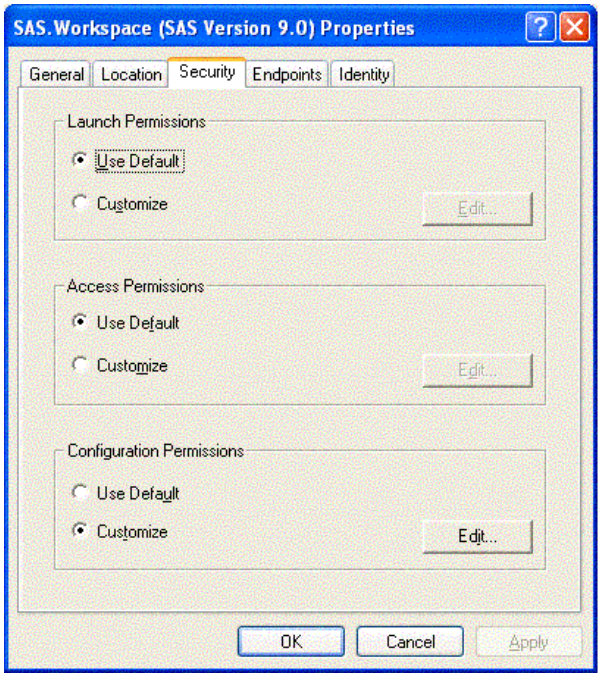
   - For multi−user servers (SAS Metadata Server, SAS System 9 OLAP server), select **This user** and specify the **User**, **Password**, and **Confirm Password** information.

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

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

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

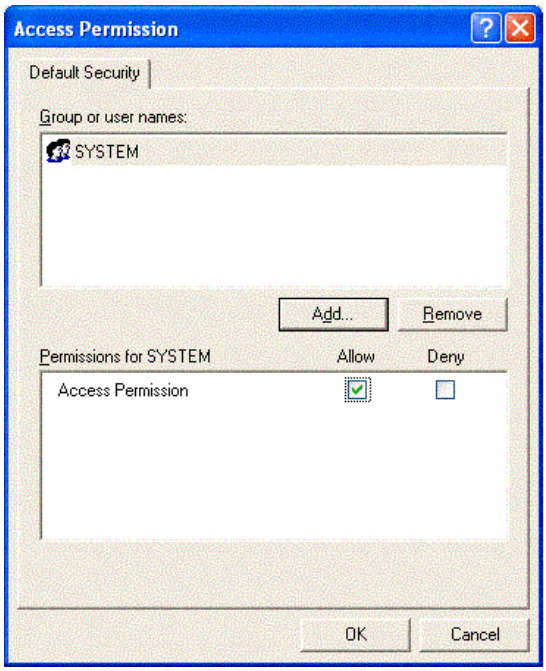
10. Select the Security tab.



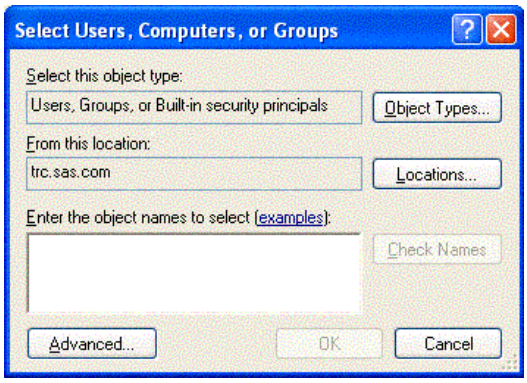
11. If you want to use default access permissions, select **Use Default**, click **OK**, and then continue with Step 12.

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

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



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



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

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

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

    Note: If you grant launch permissions for an application to specific users and groups, you might affect those users who previously had permission to the application through default permissions.
15. Click **OK** in each of the open dialog boxes to save your selections and exit the dcomcnfg utility.

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

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

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

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

COM/DCOM

Configuring COM/DCOM for Active Server Page Access

Note: You can also configure your Active Server Page (ASP) application to access SAS using the IOM Bridge for COM. You might want to use IOM Bridge for COM when

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

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

COM/DCOM Configuration

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

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

   To perform this basic configuration, follow the instructions in Enabling DCOM on the Server and the Client.
2. Additional configuration steps that will enable a Web client to access an IOM server. There are two ways that you can access a Windows Server using COM/DCOM:
   - To configure access to a local COM IOM server, see Accessing a Local COM IOM Server from an Active Server Page.
   - To configure access to a remote DCOM IOM server, see Accessing a Remote DCOM IOM Server from an Active Server Page.

Permissions

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

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

IOM Server	Web Server	DCOM Access Permission (on IOM Server)	DCOM Launch Permission (on IOM Server)	Other Notes
Local COM	IIS 4 All Authentication Methods	System	System	
	IIS 5 using Anonymous Access	IUSR_<*machine_name*> IWAM_<*machine_name*>	IUSR_<*machine_name*> IWAM_<*machine_name*>	The COM+ application can be configured so it is launched as a different user; however, this is not necessary. Refer to Configure your IIS Application to use **High (isolated) Application Protection** for details.
	IIS 5 using Basic Authentication and Integrated	IWAM_<*machine_name*> Any valid NT user	IWAM_<*machine_name*>	

	Windows Authentication	account that will be accessing the ASP		
Remote DCOM	IIS 4 All Authentication Methods	System Network	System Network	If you are setting up the remote DCOM IOM server on a Windows 2000 or XP computer, you must configure the DCOM server to run as a different user than the launching user.
	IIS 5 All Authentication Methods	User account launching IIS COM+ application Network	User account launching IIS COM+ application	The COM+ application must be configured so it is launched as a user that exists on both the Web server and DCOM IOM server. Refer to Configure your IIS Application to use **High (isolated) Application Protection** for details.

COM/DCOM

Accessing a Local COM IOM Server from an Active Server Page

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

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

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

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

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

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

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

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

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

There are two different types of authentication, Anonymous Access and Basic Authentication.

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

Anonymous Access

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

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

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

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

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

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

Basic Authentication

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

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

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

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

1. Start **dcomcnfg** and modify the properties for **SAS.Workspace (SAS Version 9.1)**.
2. Add launch and access permissions (Launch IIS Process Account) for the IWAM_<*machine name*> user.

3. Add access permissions for any user that will be accessing the ASP through the Web. To add access permissions for users, use **dcomcnfg** to either

    - add each user individually
    - create a group of users and then add that group.

COM/DCOM

Accessing a Remote DCOM IOM Server from an Active Server Page

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

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

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

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

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

   - System (the operating system)
   - Network (users accessing this object remotely)
4. If your DCOM IOM server is on Windows NT 4, this configuration is sufficient.

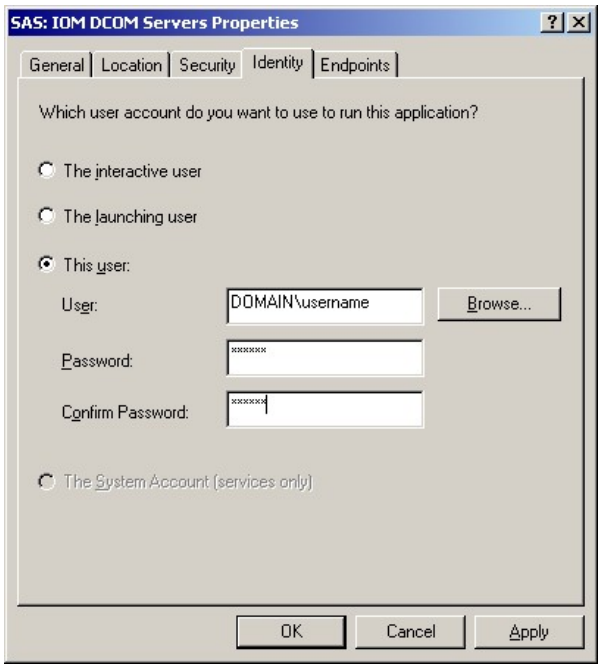
   If your DCOM IOM server is on Windows 2000 or XP, you must change the identity of the user that will run the DCOM server process. The NT Anonymous Logon user account on Windows NT 4 does not have sufficient permission to run SAS on a Windows 2000 or XP server.

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

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

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

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



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

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

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

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

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

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

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

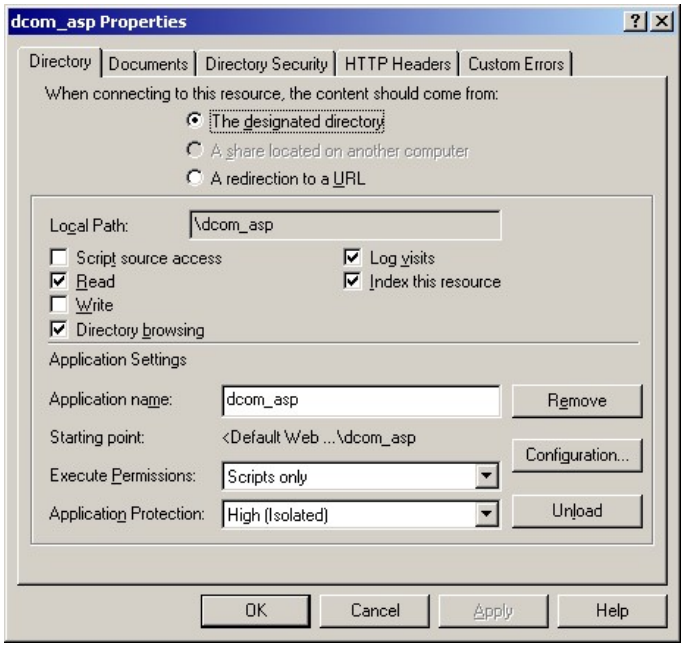
To set up remote DCOM and COM+:

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

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

To run your application as an isolated process:

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

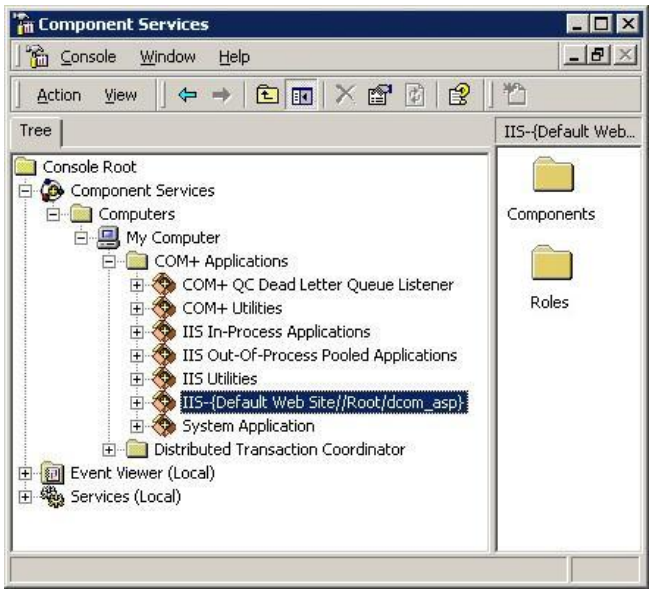


Configure your COM+ Application

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

To configure the COM+ application:

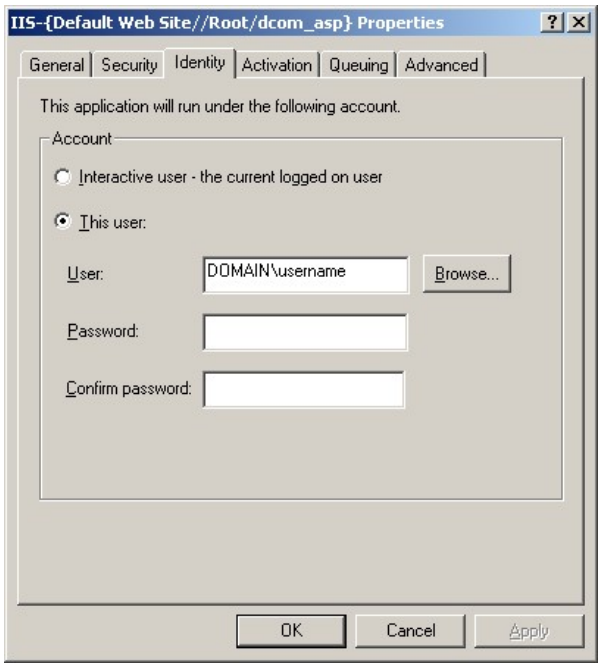
1. Click **Start → Settings → Control Panel**.
2. Open **Administrative Tools** and click **Component Services**.
3. Expand the Component Services folder, expand the Computers folder, expand the My Computer folder, and then expand the COM+ Applications folder.



4. Find the newly created COM+ application for your IIS application. It will be named **IIS−−{Default Web Site//Root/<*iis_application*>}** where <*iis_application*> is the name of your IIS application.
5. Right−click the appropriate COM+ application, and select **Properties**.
6. Select the Identity tab, and do one of the following:

   - Indicate a specific user account for the application.
   - Use the interactive user if the interactive user exists on both machines.

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



Setting Access and Launch Permissions for the User

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

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

   ♦ Network (users accessing this object remotely)

   found in the \\<*machine name*>* domain.

More Information

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

- Designing Secure Web−Based Applications for Microsoft® Windows® 2000
- Microsoft® Windows® 2000 Server Resource Kit: Microsoft Internet Information Services 5.0 Resource Guide
- COM+: Security, Communication, and Configuration
- ASP Component Guidelines
- HOWTO: Accessing Network Files from IIS Applications (Q207671)

COM/DCOM

Using the SAS Integration Technologies Configuration Utility (ITConfig)

The SAS Integration Technologies configuration utility (ITConfig) lets you generate metadata configuration files and test Integrated Object Model (IOM) connections between client machines and SAS. Using ITConfig, you can

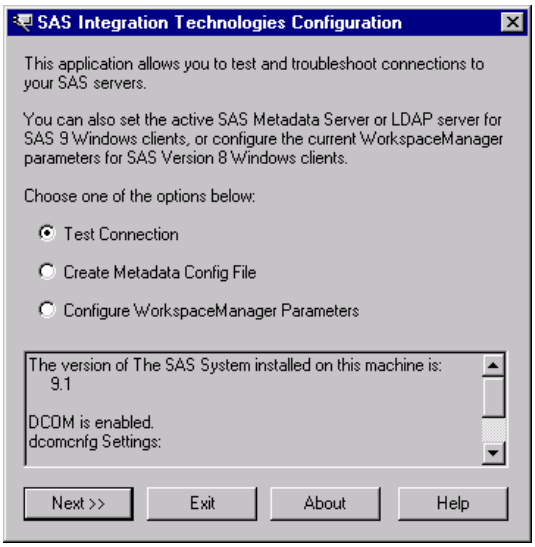
- create metadata configuration files that can be used to access an LDAP server or SAS Metadata Server.
- test and diagnose IOM connections to SAS servers. The application can test COM, DCOM, and IOM Bridge connection types.
- set the registry parameters that are used by the workspace manager on an LDAP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

Starting the Application

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

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

The Integration Technologies Configuration window appears.



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

- create metadata configuration files (Create Metadata Config File)
- view and change the LDAP parameters for the Workspace Manager (Configure WorkspaceManager Parameters)
- test the connection to a server (Test Connection)

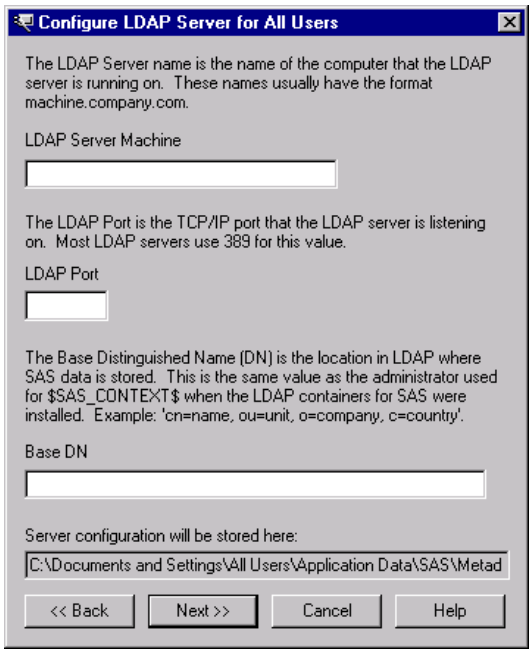
COM/DCOM

Using ITConfig to Create Metadata Configuration Files

To access definitions on a metadata server, you must first connect to the metadata server. For connections to the LDAP server, the Object Manager and SAS can use metadata configuration files that contain information about how to connect to the server.

To create the metadata configuration files

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



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

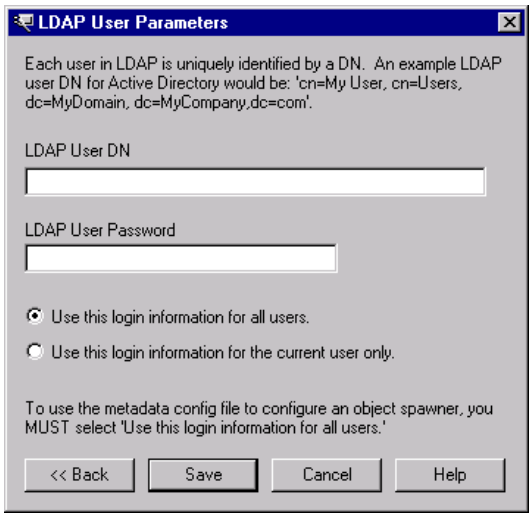
   Enter the following system configuration information:

   LDAP Server Machine
   : The fully−qualified name of the machine that the LDAP Server runs on.

   LDAP Server Port
   : The port used by the LDAP Server machine for receiving requests. A typical value is 389.

   Base DN
   : The distinguished name for the location in the LDAP hierarchy under which SAS directory entries are stored. The value for this field is the same as the value for the $SAS_CONTEXT$ parameter that was specified when the SAS containers were installed in the LDAP directory.

   Select **Next**. The LDAP User Parameters windows appears.



5. Enter the following information:

   LDAP User DN
   : The distinguished name of a user who will be accessing the LDAP server. Because the parameter information is stored in the client machine's registry, specify the DN of the client machine's user.

   LDAP User Password
   : The password required for the specified user to log onto the LDAP server.

6. If you selected **All users of this machine** for the configuration type, select one of the following:

   Use this login information for all users
   : specifies that the server and login information are stored in a single system configuration file that is common to all users.
   Note: You must select this option if you plan to use your configuration file with the object spawner.

   Use this login information for the current user only
   : specifies that the server information is stored in a system configuration file that is common to all users and that the login information is stored in a user configuration file that is specific to the current user.

   If you selected **Current user** for the configuration type, the server and login information are stored in a single system configuration file that is specific to the current user.
7. Select **Next**. ITConfig creates the configuration file(s) and the XML File Written dialog box appears.
8. To return to the main ITConfig screen, select **OK**.

Names and Locations for Configuration Files

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

Default Paths for Windows NT:

Common system configuration file
: \WINNT\Profiles\All Users\Application Data\SAS\MetadataServer\oms_serverinfo.xml

User−specific system configuration file
: \WINNT\Profiles*username*\Application Data\SAS\MetadataServer\oms_serverinfo.xml

User configuration file
: \WINNT\Profiles*username*\Application Data\SAS\MetadataServer\oms_userinfo.xml

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

Common system configuration file
\Documents and Settings\All Users\Application Data\SAS\MetadataServer\oms_serverinfo.xml
User−specific system configuration file
\Documents and Settings*username*\Application Data\SAS\MetadataServer\oms_serverinfo.xml
User configuration file
\Documents and Settings*username*\Application Data\SAS\MetadataServer\oms_userinfo.xml

Note: The location(s) and filename(s) are displayed in the Configure LDAP Server window and in the XML File Written dialog box.

Sample System Configuration File Format for an LDAP Server

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

```
<?xml version="1.0" encoding="UTF−8" ?>
<Redirect>
   <LogicalServer Name="LDAP Server"
    ClassIdentifier="440196D4−90F0−11D0−9F41−00A024BB830C">
      <UsingComponents>
         <ServerComponent Name="LDAP Server" ProductName="LDAP">
            <SourceConnections>
               <TCPIPConnection Name="LDAP Server" Port="389"
                HostName="dtd.pc.sas.com" ApplicationProtocol="LDAP">
                  <Domain>
                     <AuthenticationDomain Name="domainName">
                        <Logins>
                           <Login Name="test" UserID="cn=Mister
                            LDAP,cn=Users,dc=dtd−dom,dc=sas,dc=com"
                            Password="{base64}cGFzc3dvcmQ=" />
                        </Logins>
                     </AuthenticationDomain>
                  </Domain>
               </TCPIPConnection>
            </SourceConnections>
            <Properties>
               <Property Name="basedn"
                DefaultValue="cn=SAS,cn=Applications,dc=dtd−dom,dc=sas,dc=com"
                PropertyName="BaseDN">
               </Property>
            </Properties>
         </ServerComponent>
      </UsingComponents>
   </LogicalServer>
</Redirect>
```

Sample User Configuration File Format for an LDAP server

Use a text editor to edit your metadata configuration files. The following XML code shows a sample user configuration file for a connection to an LDAP Server.

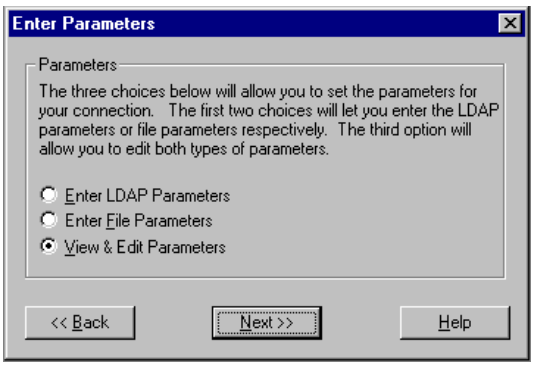
```
<?xml version="1.0" encoding="UTF−8" ?>
<AuthenticationDomain Name="domainName">
   <Logins>
```

```
     <Login Name="domainName\abc" UserID="domainName\abc1"
      Password="{base64}cGFzc3dvcmQ="/>
   </Logins>
</AuthenticationDomain>
```

COM/DCOM

Using ITConfig to Configure Workspace Manager Parameters

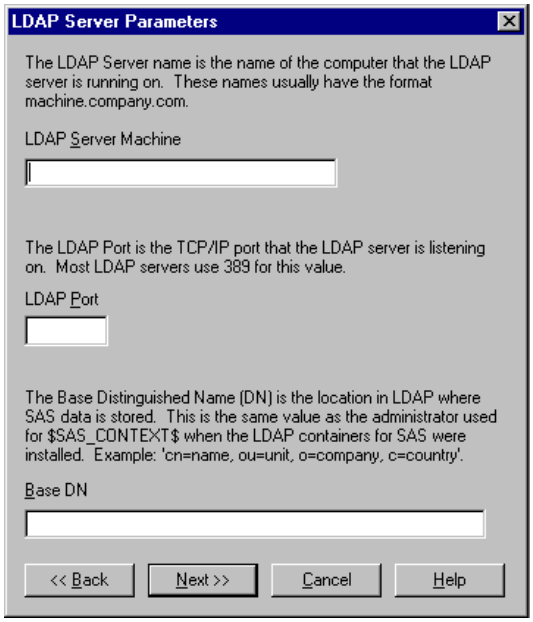
To view and edit connection parameters, select **Configure WorkspaceManager Parameters** from the Configuration window. The Enter Parameters window appears.



Specify whether you want to enter the LDAP connection parameters, enter the name of an LDIF file containing connection parameters, or view and edit all currently defined connection parameters.

Entering LDAP Parameters and Testing Connections

1. To enter connection parameters in the registry for a connection to an LDAP server, select **Enter LDAP Parameters** from the Enter Parameters window and click **Next**. The LDAP Server Parameters window appears.



2. Enter the following information:

LDAP Server Machine

Specifies the fully−qualified name of the machine on which the LDAP server runs.

LDAP Port

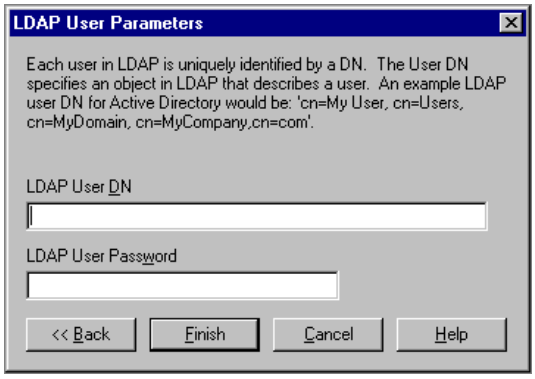
Specifies the port used by the LDAP server machine for receiving requests. A typical value is 389.

Base DN

Specifies the distinguished name for the location in the LDAP hierarchy under which SAS directory entries are stored. The value for this field is the same as the value for the $SAS_CONTEXT$ parameter that was specified when the SAS containers were installed in the LDAP directory.

Note: If you enter a Base DN that does not include cn=sas, then cn=sas is added to the Base DN that you entered.

Click **Next**. The LDAP User Parameters window appears.



3. Enter the following information:

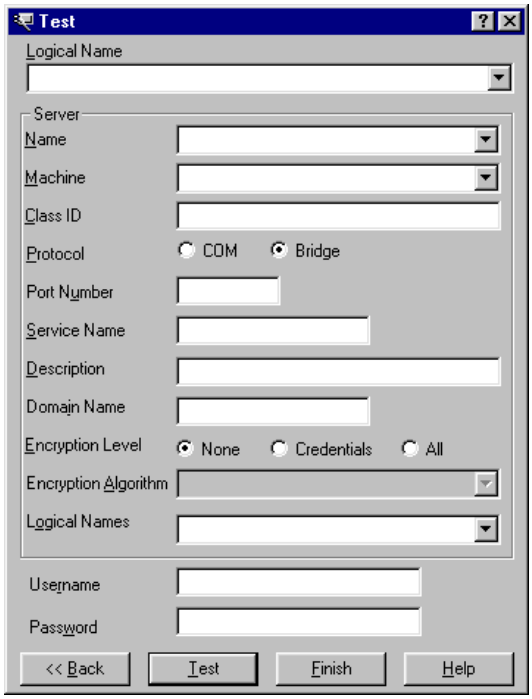
LDAP User DN

Specifies the distinguished name of a user who will be accessing the LDAP server. Because the parameter information is stored in the client machine's registry, specify the DN of the client machine's user.

LDAP User Password

Specifies the password required for the specified user to log onto the LDAP server.

4. Click **Next**. The application writes the data to the registry and the Test window appears.

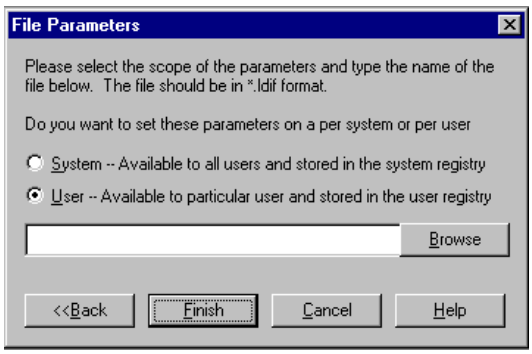


5. To test a connection to a server defined on the LDAP server, select the **Logical Name** of the machine for which you want to test a connection.
6. Click **Test** to test the connection. If the program establishes a DCOM connection to the specified server, the Connection Successful window appears.
7. To return to the main ITConfig screen, click **Finish**.

Entering File Parameters

1. To specify an LDIF file to use for connection parameters, select **Enter File Parameters** from the Enter Parameters window and click **Next**.

   The File Parameters window appears.

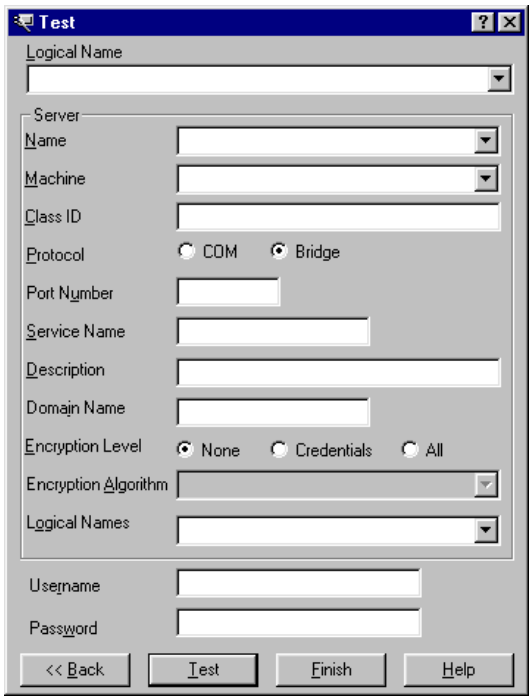


2. Specify whether the parameters apply to all users of this machine or only to a specific user.

   System
   Specifies the parameters contained in the LDIF file apply to all users of the client machine. The parameters will be stored in the system registry.

   User
   Specifies the parameters contained in the LDIF file apply only to a specific user. The parameters will be stored in the registry for the specified user.

3. Click **Next**. The application writes the data to the registry and the Test window appears.



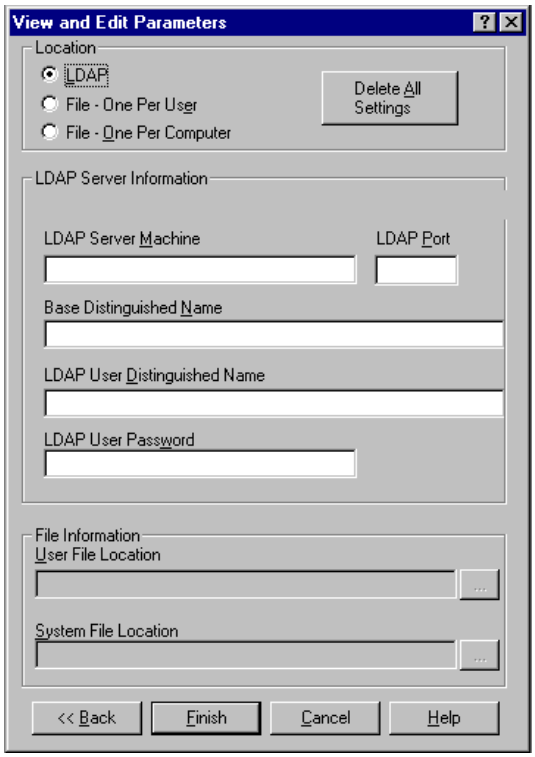
4. If you want to test a connection to a server defined on the LDAP server, select the **Logical Name** of the server connection you wish to test. Click **Test** to test the connection.

   If the program establishes a connection to the specified server, the Connection Successful window appears.
5. To return to the main ITConfig screen, click **Finish**.

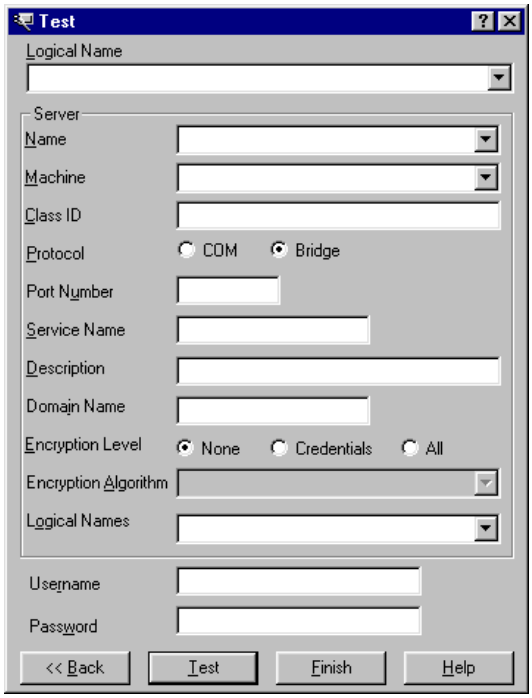
Viewing and Editing All Parameters

1. To work with all of the parameter information, whether from LDAP or a file, select **View and Edit Parameters** from the Enter Parameters window.

   The View and Edit Parameters window appears.



2. This window lets you view and specify the same information as on the LDAP and file parameters windows.
3. Select **LDAP** to view and edit the LDAP parameter information.
4. Select **File − One Per User** to view and edit the LDIF file used to define access for a particular user. The **User File Location** field is then enabled.
5. Select **File − One Per Computer** to view and edit the LDIF file used to define access for all users on the current machine. The **System File Location** field is then enabled.
6. Select **Delete All Settings** to clear all configuration information for all locations.
7. Click **Next**. The Test window appears.



8. To test a connection to a server defined on the LDAP server, select the **Logical Name** of the server connection you wish to test.

   Click **Test** to test the connection. If the program establishes a connection to the specified server, the Connection Successful window appears.
9. To return to the main ITConfig screen, click **Finish**.

COM/DCOM

Using ITConfig to Test Connections

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

The test program used by ITConfig is a small SAS program that verifies the following information about the server environment:

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

Testing a Connection that is Defined on a Metadata Server

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

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

Testing a Local COM Connection

To test a local COM connection to a SAS server:

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

Testing a Manually Defined DCOM Connection

To test a DCOM connection to a SAS server:

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

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

COM/DCOM

Troubleshooting a COM/DCOM Connection

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

- Make sure you observe COM/DCOM requirements:

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

- Make sure the registry settings are correct:

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

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

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

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

- Make sure the working directory is correct:

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

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

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

- Make sure the permissions are correct:

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

    To view the audited information, use the Event Viewer and select Log → Security. This will show you what user ID attempted to access the files specified through the user manager.
  - An error message that states "Server execution failed" when trying to connect to the IOM server can be caused by many things including trying to connect to an IOM server with an expired license or having an invalid username/password in the dcomcnfg identity settings.
  - Events work by having the IOM server make a call on an interface that the client provides to SAS. In order for SAS to make a call on that interface, the client must grant permission to SAS to make the call.

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

- Make sure the authentication is correct:

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

COM/DCOM

AppIDs for Configuring DCOM

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

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

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

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

COM/DCOM

Object Server Parameters

The following table lists the object server parameters that you can use to override or add to the object server parameters that the spawner uses to launch SAS.

Object Server Parameters			
Object Server Parameter	**Value**	**Connection Type**	**Definition**
CLIENTENCRYPTIONLEVEL **Alias:** CEL	none credentials everything	IOM Bridge	Specifies the degree of encryption to use when making outbound calls. This option is used only by the bridge protocol engine.
JNLSTRMAX	Numeric value	IOM Bridge COM/DCOM	Specifies the maximum length of string printed in the IOM server journal. This option can be used to adjust the amount of material included in an IOM trace. A value greater than 500 is recommended.
LOGFILE **Alias:** LOG	Path in which to create the IOM server trace log.	IOM Bridge COM/DCOM	Provides an alternative to the SAS Log for IOM server trace output. **Note:** The user who starts the server must have execute and write permissions for the log destination path.
PORT	TCP/IP port number	IOM Bridge	Specifies the value for the bridge protocol engine to use as the port in which to start listening for client connections.
PROTOCOL	bridge com (com,bridge)	IOM Bridge COM/DCOM	Specifies the protocol engine(s) to launch in server mode. Server mode indicates that the protocol engine(s) will listen for client connections. By default, Windows servers use the COM protocol engine and all other servers use the Bridge protocol engine.] It you specify (com, bridge) a multiuser server can simultaneously support clients using different protocols.
SECURITY | NOSECURITY	N/A	IOM Bridge COM/DCOM	Specifies whether client authorization is required. When security is enabled, the bridge protocol engine requires a username and password; the COM protocol engine is integrated with the single−signon security of the Windows networking environment. Authorization decisions are controlled by the server application. If "nosecurity" is

			specified, these security mechanisms are bypassed.
TIMEOUTSECONDS	Numeric value	IOM Bridge COM/DCOM	Indicates the timeout in seconds before an inactive session is terminated. If this option is not set, inactive sessions are deleted when closed or fully released by the client.
V8ERRORTEXT	N/A	IOM Bridge COM/DCOM	Indicates that the MVA components should return Version 8 style error messages instead of the Version 9 XML style error messages.

COM/DCOM

Attributes for sasServer

The sasServer object class contains startup and connection information for an instance of a SAS object server. The sasServer object class is defined using the attributes listed in the following table. For each attribute, the table shows:

- The name that identifies the attribute on the LDAP server (or in the configuration file). Under each attribute name, the table shows the corresponding tab and field name in the IT Administrator application.
- "Required" or "Optional" to indicate whether the attribute is required.
- The type of server configuration (COM/DCOM or IOM Bridge) for which the attribute is used.
- A definition of the attribute.

Note: The following attributes which appear in the IT Administrator interface are not used in Version 9 of SAS Integration Technologies

- sasTpName
- sasPluName
- sasProtocol=corba
- sasMultiUserObject

If you are using Version 9, do not use these attributes for your configuration.

Note: For the z/OS, you can now use the sasCommand attribute to launch SAS as an object server. Because the IT Admin interface has not changed for Version 9, you can specify the launch command in the **Command for non OS/390** IT Admin field.

For step−by−step instructions on defining the metadata for a server, refer to Using the IT Administrator Wizard to Define a Server and Spawner or Using IT Administrator to Define a Server. If you are not using an LDAP server, you can use a configuration file to define the server. For instructions, see Using a Configuration File to Define the Metadata (IOM Bridge) or Using a Configuration File to Define the Metadata (COM/DCOM).

sasServer Attribute Definitions			
Attribute Name	**Required/ Optional**	**Server Type**	**Definition**
description *In IT Administrator:*: **Description**	Optional	COM/DCOM, IOM Bridge	Text to summarize why this object definition exists.
objectClass *In IT Administrator:* **N/A**	Required	COM/DCOM, IOM Bridge	The object class identifier. For sasServer objects, this is always `sasServer`. If you use IT Administrator, this identifier is assigned automatically.
sasClientEncryptionAlgorithm *In IT Administrator:* **Encryption (IOM) ➔ Client Algorithm**	Optional	IOM Bridge	The encryption algorithm that is supported on the client side of the connection. Valid values are: RC2, RC4, DES, Triple DES, and SAS Proprietary, depending on the country in which the SAS software is licensed. See SAS/SECURE for more

			information regarding this attribute.
sasCommand *In IT Administrator:* **Commands → Command for non OS/390**	Required	IOM Bridge	The command used to launch SAS as an object server. With the command, specify the path relative to the directory in which the spawner will be started. If you are using a configuration file instead of LDAP, then paths with embedded blanks must be in quotation marks (or, for Windows platforms, double quotation marks). For more information about the server command, see Server Startup Command.
sasDomainName *In IT Administrator:* **Connections → Domain**	Optional	COM/DCOM, IOM Bridge	The security domain in which the sasServer definition participates. In IOM bridge servers configurations, the spawner definition must have the same domain name as the server definition. The spawner uses the domain name, along with the machine name and logical name, to determine which server(s) it services. The lack of a domain is considered a domain; therefore, if the server definition has no domain name, it will be associated only with spawners that have no domain name.
sasLogicalName *In IT Administrator:* **Logical Names**	Optional	COM/DCOM, IOM Bridge	The logical names associated with this sasServer definition. In IOM bridge servers configurations, the spawner uses logical names (along with machine names and domain names) to determine which server(s) it services. If logical names are specified, then only those sasSpawner instances that include *all of the logical names* that are defined here will support this sasServer. If you are using a configuration file instead of LDAP, specify each logical name as a separate attribute and value pair. For a general discussion of logical names, refer to Assigning Logical Names.
sasMachineDNSName *In IT Administrator:* **Machines**	Required	COM/DCOM, IOM Bridge	The DNS (domain name service) name(s) and IP address(es) for the machine(s) on which this server definition may execute. Multiple values can be assigned to this attribute. The machine name must be the official network name (for example,

			machine.corp.com). The string `localhost` can be used to signify the host on which the associated spawner is executing.
sasMaxPerWorkspacePool *In IT Administrator:* **Workspace Pool → Maximum Workspaces per Workspace Pool**	Optional	COM/DCOM, IOM Bridge	If you are using connection pooling, specifies the maximum number of workspaces that should be available for a workspace pool that is established on this server. A good starting place for this number is the number of CPUs that are available on the machine that is running SAS.
sasNetEncrAlg *In IT Administrator:* **Encryption (IOM) → Server Algorithms**	Optional	IOM Bridge	The encryption algorithms that are supported by the launched object server. Multiple values can be assigned to this attribute. Valid values are: RC2, RC4, DES, TRIPLEDES, and SASPROPRIETARY, depending on the country in which the SAS software is licensed. See SAS/SECURE for more information regarding this attribute.
sasObjectServerParms	Optional	COM/DCOM, IOM Bridge	The object server parameters that the spawner uses to launch SAS. This field allows you to override or add to the object server parameters. For a list of object server parameters, see Object Server Parameters.
sasPort *In IT Administrator:* **Connections → IOM Bridge → Port**	Required if server will have Java clients	IOM Bridge	The port on which to connect to this object server. If neither sasPort nor sasService is specified, the spawner will attempt to use the service name `sasobjspawn` as the sasService. If `sasobjspawn` has been used already, the spawner will remove this sasService definition from its list. The port number is required if the server will have Java clients.
sasProtocol *In IT Administrator:* **Connections → Protocol**	Required	COM/DCOM, IOM Bridge	The protocol (bridge, com) that clients may use for connection. The protocol `bridge` must be used for servers that are serviced by the spawner. These include all non−Windows servers, as well as Windows servers that will be accessed by Java clients.
sas−RecycleActivationLimit *In IT Administrator:* **Workspace Pool → Recycle**	Optional	COM/DCOM, IOM Bridge	If you are using connection pooling, the number of times a server is used before the process is disposed of and a new process is used in pooling. A value of 0 indicates that

Activation Limit			the process will have no limit.
sasRequiredEncryptionLevel *In IT Administrator:* **Encryption → Encrypt**	Optional	COM/DCOM, IOM Bridge	The level of encryption to be used between the client and the object server. **None** means no encryption is performed; **Credentials** means that only user credentials (id and password) are encrypted; and **Everything** means that all communications between the client and server are encrypted.
sasServercn *In IT Administrator:* **Name**	Required	COM/DCOM, IOM Bridge	The unique name for this sasServer object.
sas−ServerRunForever *In IT Administrator:* **Workspace Pool → Server Process Shutdown → Leave running when idle**	Optional	COM/DCOM, IOM Bridge	If you are using connection pooling, specifies that an idle server should always remain running. If the value of this attribute is true, the server always remains running. If the value is false, the idle server runs for the length of time specified in the sasServerShutdownAfter attribute.
sas−ServerShutdownAfter *In IT Administrator:*: **Workspace Pool → Server Process Shutdown → Minute until idle shutdown**	Optional	COM/DCOM, IOM Bridge	If you are using connection pooling, the number of minutes after which an idle server should be shut down. The value must be between 0 and 1440. The default value is 3. This attribute is ignored if the value of sasServerRunForever is true.
sasService *In IT Administrator:* **Connections → IOM Bridge → Service**	Optional	IOM Bridge	The service in which to connect to this object server. If you specify a value for both sasService and sasPort, then the value for sasService will be ignored. If neither sasPort nor sasService is specified, the spawner will attempt to use the service name `sasobjspawn` as the sasService. If `sasobjspawn` has been used already, the spawner will remove this sasService definition from its list. Note: If the server will have Java clients, specify a sasPort instead of a sasService.

COM/DCOM

Attributes for sasLogicalNameInfo

The sasLogicalNameInfo object class contains information for an instance of a SAS logical name. The sasLogicalNameInfo object class is defined using the attributes listed in the following table. For each attribute, the table shows:

- The name that identifies the attribute on the LDAP server (or in the configuration file).
- "Required" or "Optional" to indicate whether the attribute is required.
- The type of server configuration (COM/DCOM or IOM Bridge) for which the attribute is used.
- A definition of the attribute.

For general information about the use of logical names, refer to Assigning Logical Names. When you use IT Administrator to add a logical name to a server or spawner definition, IT Administrator automatically creates a sasLogicalName object.

If you are not using an LDAP server, you can use a configuration file to define the logical name. For instructions, see Using a Configuration File to Define the Metadata. The spawner does not use this object class. However, if your site uses logical names, it is recommended that sasLogicalNameInfo instance be created.

sasLogicalName Attribute Definitions			
Attribute Name	**Required/ Optional**	**Server Type**	**Definition**
description *In IT Administrator*:: **Description**	Optional	COM/DCOM, IOM Bridge	Text to summarize why this object definition exists. This attribute is not used by the spawner.
objectClass	Required	COM/DCOM, IOM Bridge	The object class identifier. For sasLogicalNameInfo objects, this is always `sasLogicalNameInfo`.
sasLogicalName	Required	COM/DCOM, IOM Bridge	The logical name that is being defined

IOM Bridge Servers

Setting up an IOM Bridge Server and Spawner

An IOM Bridge server configuration enables client access using the IOM Bridge for COM or IOM Bridge for Java.

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

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

When to Use an IOM Bridge Server Configuration

You must use an IOM Bridge server configuration if:

- The object server will run on a non−Windows machine (for example, a UNIX−based machine); or if
- The object server will be accessed by Java client applications

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

Components of an IOM Bridge server configuration

An IOM Bridge Server configuration consists of:

- A server machine which hosts Version 9 of the SAS base software and the SAS Integration Technologies software. The spawner program, which is part of Integration Technologies, must be running on the server machine in order for clients to obtain access.
- A client application, which can run on the same machine as the server or on a remote machine. To connect to the object server via TCP/IP, client applications must use the IOM Bridge for COM or IOM Bridge for Java utilities provided with Integration Technologies. To request specific services from the object server, client applications use Application Program Interfaces (APIs), also known as distributed objects, that are provided with Integration Technologies.
- An LDAP server, which is a central repository that client and server software can access to obtain metadata (or configuration information) about the object server. For IOM Bridge server configurations, the metadata includes definitions for server objects and spawner objects. Optionally, the metadata can also include definitions for login objects and logical name objects (which define groupings of servers, spawners, and other associated objects.)

  Note: If your configuration is very simple (that is, consisting of only one or two servers and clients) and does not require strict security, you can use configuration files instead of an LDAP server to store the metadata for servers and spawners.

  Note: You can also supply the server parameters for the configuration directly in the application program.

How an IOM Bridge Server Works

In order for an IOM Bridge server to be available to clients, the spawner program must be running on the server machine. When a client application needs to access the object server, it uses Integration Technologies distributed objects to submit a request to the object spawner for a SAS workspace. To obtain information about the server (and for metadata, the spawner), the application either

- accesses metadata from the LDAP server.
- accesses metadata from the configuration file.
- imbeds the server information in the application program.

On receiving the request, the object spawner authenticates the user and launches an object server. The client then uses server configuration information from the LDAP server or configuration file to requests a workspace on the server.

After acquiring a workspace, the client application uses Integration Technologies distributed objects to issue one or more requests for SAS language services, data services, file services, or utilities. The SAS software processes these requests in the workspace and returns information to the client.

When the client application is finished using the server, it issues a request to close the workspace.

IOM Bridge Servers

Quick Start: Simple Server and Spawner

The following steps help you to get a simple server and spawner IOM Bridge connection up and running on a Windows or UNIX platform. For details about setting up more complex configurations, see Summary of Setup Steps (IOM Bridge Server).

Windows

To set up and test a simple IOM Bridge server and spawner configuration on Windows:

1. Install SAS Version 9 (including SAS Integration Technologies) on the server machine. Refer to the SAS documentation for the details of this procedure.
2. Use a Windows editor to create an LDIF configuration file and save it as `c:\objspawn.cfg`. Use the following example of a minimal LDIF configuration file: Configuration File Example: Minimal Configuration. Replace the `sasCommand` statement with the Windows location of the SAS system. For example:

   ```
   sasCommand: "c:\Program Files\SAS\SAS 9.1\sas"
      -config "c:\Program Files\SAS\SAS 9.1\sasv9.cfg"
   ```
3. **Note:** This step is only necessary if you are not starting the spawner as a service.

   Define Windows User Rights for the administrator. (For detailed instructions about defining Windows User Rights, see Starting the Spawner on Windows). The user who invokes the spawner must be an administrator and must have the following user rights:

   - act as part of the operating system (Windows NT and Windows 2000)
   - increase quotas (Windows NT and Windows 2000)
   - replace the process level token

4. Define Windows User Rights for each client. For each client that connects to the spawner, specify

   - log on as batch job

5. Start the Object Spawner with the `objspawn.cfg` configuration file using one of the following methods:

   - To start the object spawner with the `objspawn.cfg` configuration file, from a DOS command window enter the following command:

     ```
     c:\Program Files\SAS\SAS 9.1> objspawn
        -configFile  C:\objspawn.cfg
     ```
   - To install the object spawner as a service and update the registry to hold the options that are specified (in this case −configFile), from a DOS command window enter the following command:

     ```
     c:\Program Files\SAS\SAS 9.1> objspawn
        -configfile c:\objspawn.cfg  -install
     ```

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

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

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

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

For the complete list of spawner invocation options, see Spawner Invocation Options.

6. Test your server connection using the Integration Technologies Configuration application (ITConfig), which is installed with the Integration Technologies Windows Client Development Component. See Using the Integration Technologies Configuration Application.

Unix

To set up and test a simple IOM Bridge server and spawner configuration on Unix:

1. Install SAS Version 9 (including SAS Integration Technologies) on the server machine. Refer to the SAS documentation for the details of this procedure.
2. If the `setuid` root bit is not set for `sasrun`, `sasauth`, and `elssrv`, the spawner will not be able to launch SAS sessions. To set the `setuid` root bit, see Changing the `setuid` Permissions to Root
3. Locate a user or temporary directory where you can write and save a configuration file. For example,

   ```
   /users/myid
   ```
4. Use an editor to create an LDIF configuration file and save it as `objspawn.cfg`. Use the following example of a minimal LDIF configuration file: Configuration File Example: Minimal Configuration. Replace the `sasCommand` statement with the location of the SAS system.
5. To start the object spawner with the `objspawn.cfg` configuration file, from a Unix system prompt enter the following command:

   ```
   /sasv9/utilities/bin/objspawn
      −configfile /users/myid/objspawn.cfg
   ```

   For the complete list of spawner invocation options, see Spawner Invocation Options.

6. Test your server from a Windows machine using the Integration Technologies Configuration application (ITConfig), which is installed with the Integration Technologies Windows Client Development Component. See Using the Integration Technologies Configuration Application.

IOM Bridge Servers

Summary of Setup Steps (IOM Bridge Server)

To set up an IOM Bridge server:

1. Install SAS Version 9 (including Integration Technologies) on the server machine. Refer to the SAS documentation for the details of this procedure.
2. If your applications or spawners need to access metadata that describes your IOM Bridge server configuration, you must create the necessary definitions for server objects, spawner objects, and (optionally) SAS login and logical name objects. For details, see Creating Metadata for an IOM Bridge Server. The method for creating the metadata depends on whether you are using an LDAP server or a configuration file for your metadata repository:

    - If you are using an LDAP server, you can use IT Administrator to create the necessary metadata. For details, see Using the IT Administrator Wizard to Define an IOM Bridge Server or Using IT Administrator to Define the Metadata .
    - If you are not using an LDAP server, you must create a configuration file that contains the necessary metadata, and then install the configuration file on the server machine. For details, see Using a Configuration File to Define the Metadata.
3. Set up and start the spawner:
    - If you are using a z/OS server, configure and start the spawner. Refer to Configuring and Starting the Object Spawner on z/OS.
    - If you are not using a z/OS server, launch the spawner. Refer to Invoking (Starting) the Spawner for examples and special security considerations. The command syntax varies based on the server platform:

        - If you are using a Windows server, refer to Starting the Spawner on Windows.
        - If you are using a UNIX server, refer to Starting the Spawner on UNIX.

    For all platforms, refer to the list of Spawner Invocation Options.
4. Install the necessary components on each client machine.

    - For Windows Clients:

        - Install the IT software for Windows clients. For instructions, refer to Developing Windows Clients in the *Developer's Guide*.
        - If you are not using an LDAP server, you must also copy the configuration file (created in Step 2) to the client machine.
    - For Java Clients:

        - Install the IT software for Java clients. For instructions, refer to Developing Java Clients in the *Developer's Guide*.
        - If you are not using an LDAP server and you are using the Java Connection Factory, you must also create a server definition in `com.sas.services.connection.BridgeServer`. This is necessary in order to obtain a reference to an IOM object. Refer to Creating a Server Object with Java for an example. For more information, see Using the Java Connection Factory in the *Developer's Guide*.
        - If you are not using an LDAP server and you are using the Java Workspace Factory, you must also create a server definition in java.util.Properties. This is necessary in order to obtain a reference to an IOM workspace. Refer to Creating a Server Object with Java for an example. For more information, see Using the Java Workspace Factory in the *Developer's Guide*.

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

IOM Bridge Servers

Spawner Overview

The spawner is a program that runs on the server host and listens for requests. When a request is received, the spawner accepts the connection and performs the action that is associated with the port or service on which the connection was made. A connection to a spawner may

- request a SAS object server
- initiate the Administrator Interface
- request a Universal Unique Identifier (UUIDGEND).

Request a SAS Object Server

When a connection is made on a port or service that is associated with a sasServer object, the spawner authenticates the client connection. The spawner then launches a SAS object server for use by the connecting client.

To launch the server, the spawner locates the appropriate object server definition(s) by using the following rules:

- The domain (sasDomainName) must match. The lack of a domain is considered a domain.
- The sasServer definition must have a protocol (sasProtocol) of bridge.
- The logical names specified in the sasServer definition must be a subset of the logical names that are specified in the sasSpawner definition.

Initiate the Administrator Interface

When a connection is made on the port or service that is identified as the sasOperatorPort or sasOperatorService attribute in the spawner definition, the spawner initiates the administration interface. Only one administrator can be active at a given time. For more information about the administration interface, see Monitoring the Spawner.

Request a Universal Unique Identifier (UUIDGEND)

A spawner may be configured to support UUID generation; or it may be configured solely as a UUID generator daemon. In either case, when a connection is made on the port or service that is identified as a sasUUIDPort or sasUUIDService attribute in the spawner definition, the spawner initiates UUID generation. For more information, see Configuring a UUID Generator.

IOM Bridge Servers

Spawner Requirements

Hardware Requirements

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

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

Software Requirements

Install the following software on the server machine:

- SAS Software Version 9 (or later)
- SAS Integration Technologies
- SAS/SECURE (optional)
- Metadata server (optional)

IOM Bridge Servers

Metadata Overview (IOM Bridge)

The metadata that supports Integration Technologies consists of objects, each defined by a collection of attributes that define the object. If you require metadata, the metadata for an IOM Bridge server configuration must include the following classes of objects:

- **sasServer** object. A sasServer object contains startup and connection information for a particular instance of a SAS object server. For each instance of an object server, the following information is defined in the metadata:
    - ♦ Server name
    - ♦ Machine name
    - ♦ Connection information, including service or port
    - ♦ Other information including logical names, encryption information, commands, and pooling information, as required.

  For detailed information about the attributes included in the metadata for a server, see the sasServer Attributes List.
- **sasSpawner** object. A sasSpawner object contains configuration information for a particular instance of a SAS spawner. The spawner is a program that must run on a server machine when an IOM bridge server configuration is used. The spawner listens for and authenticates incoming client requests and launches server instances as needed. For each spawner instance, the following information is defined in the metadata:
    - ♦ Spawner name
    - ♦ Machine name
    - ♦ Other information, including logical names and encryption information, as required

  For detailed information about the attributes included in the metadata for a spawner, see the sasSpawner Attributes List.

The following classes of objects are optional:

- The **sasLogin** object is a convenient tool to provide credentials. A SAS login may need to be available in order to start a SAS session on a server or to connect to a client. For each sasLogin instance, the following information is defined in the metadata:
    - ♦ Login name
    - ♦ Person references for authorized users or groups of users
    - ♦ The SAS login and password needed to access the SAS object server
    - ♦ Domain name
    - ♦ Logical name

  For detailed information about the attributes included in the metadata for a SAS login, see the sasLogin Attributes List.
- The **sasLogicalNameInfo** object is created automatically if you use IT Administrator to assign a logical name to a spawner or server. Logical names are used to create resource groupings. For more information, refer to Assigning Logical Names and sasLogicalNameInfo Attributes List.

The following example configuration files demonstrate the basic metadata needed to create a working IOM Bridge server configuration:

- Example Minimal Configuration
- Example Server and Spawner
- Example Using Logical Names

- Example UUID Generator

IOM Bridge Servers

Creating the Metadata for an IOM Bridge Server

If your applications or spawners need to access metadata from the LDAP server or from a configuration file, you must create the metadata that describes your IOM bridge server configuration. If you need to create metadata, use the appropriate method depending on whether you use an LDAP server or a configuration file to store your metadata:

- **If you are using an LDAP server:**
    - ♦ You can use the IT Administrator Wizard to create definitions for the server, spawner, and logical name objects. For instructions, see Using the IT Administrator Wizard to Define a Server and Spawner (IOM Bridge).
    - ♦ You can use the IT Administrator interface to create and modify the object definitions. For instructions, see Using IT Administrator to Define the Metadata (IOM Bridge).
- **If you are *not* using an LDAP server,** you can create and install configuration files that contain the object definitions. For instructions, see Using a Configuration File to Define the Metadata (IOM Bridge).

  Note: If your configuration requires more than one or two servers, or if multiple clients will be using the servers, we strongly recommend the use of LDAP as a central metadata repository. The use of LDAP also gives you the ability to use access control lists to control access to the servers in your enterprise.

As you create the metadata, you can refer to the following example configuration files, which demonstrate the basic metadata needed to create a working IOM Bridge server configuration:

- Example Minimal Configuration
- Example Server and Spawner
- Example Using Logical Names
- Example UUID Generator

IOM Bridge Servers

Using the IT Administrator Wizard to Define a Server and Spawner (IOM Bridge)

The SAS Integration Technologies Administrator provides a wizard to guide you through the process of creating LDAP−based metadata for an IOM Bridge server and spawner. (Alternatively, you can create this metadata using the regular IT Administrator interface.) For general information about IT Administrator, refer to Using the Integration Technologies (IT) Administrator.

Before beginning this procedure, be sure that you have:

- Installed an LDAP directory server and configured it for use with SAS software. For detailed instructions, refer to Setting up an LDAP Directory Server.
- Installed the Integration Technologies (IT) Administrator application. For details, see IT Administrator Installation and Startup.

Note: The following attributes which appear in the Commands tab of the IT Administrator interface are not used in Version 9 of SAS Integration Technologies

- Transaction Program Name
- Partner Logical Unit Name
- Logical Unit Name

The Multi−User Process ID Login attribute which appears in the Comnnections tab of the IT Administrator interface is not used in Version 9 of SAS Integration Technologies

If you are using Version 9, do not use these attributes for your configuration.

To define an IOM Bridge server object and spawner object using the wizard:

1. Start IT Administrator.
2. Select the **SAS Configuration** button in the Manager Bar.
3. Click the **Wizard** button (). The welcome screen for the wizard appears.
4. On each screen of the wizard, follow the instructions given; then select the **Next** button to move to the next screen. The **Next** button remains grayed out until you enter the required information. If you need to change information you have already entered, select the **Back** button. Select the **Help** button on any screen of the Wizard to receive instructions for the screen currently displayed. Select the **Cancel** button to exit the wizard at any time; if you select **Cancel**, no server object or spawner object will be created.

   Each wizard screen is described below. For detailed information about a field, click on the corresponding attribute name, which is shown in parentheses. The appropriate definition on the sasServer Attributes List or sasSpawner Attributes List will be displayed. You can then use your browser's **Back** button to return to this page. The wizard screens are as follows:

   - **Select a Logical Grouping**. On this screen, select a defined logical name (sasLogicalName) from the list. If you want to create a new logical name, select the **Create** button; the wizard will help you create a new sasLogicalNameInfo object. For more information about logical names, refer to Assigning Logical Names.
   - **Create a New Server**. On this screen, enter a server name (sasServercn), (optionally) a description (description), and a domain (sasDomainName).

- **Specify a Protocol**. On this screen, select the **IOM Bridge** protocol (sasProtocol).
- **Specify IOM Bridge Connection Information**. On this screen, specify a fully−qualified host name (sasMachineDNSName). Also enter a service (sasService) and/or a port number (sasPort). The port number is required if the server will have Java clients.
- **Specify the IOM Bridge Server Type**. On this screen, select the type of server platform: either **OS/390** (z/OS) or **Other**.
- **Set the Command for this Server**. Enter the command (sasCommand) used to start the object server. Include the path relative to the directory in which the spawner will be started. This screen appears only if the server is a machine other than a z/OS.
- **Specify the OS/390 Command Information.** For Version 9, these fields are not used; however, you must enter a value. This screen appears only if the server is a z/OS machine.
- **Done**. On this screen, select the **Advanced** button if you want to add support for encryption. (For Version 9, multi−user logins are not supported.) Otherwise, select **Next** to skip to the Configure a New Spawner window. If you select **Cancel**, the server object will not be created.
- **Specify IOM Bridge Server Security Settings**. On this screen, specify client encryption algorithms and server (sasNetEncrAlg) encryption algorithms. Also make a selection to indicate what to encrypt (sasRequiredEncryptionLevel).
- **Configure a New Spawner**. On this screen, select the **Next** button to begin configuring the spawner. If you select **Cancel**, the server object will not be created.
- **New Spawner**. This screen displays the domain, host name, and logical name information that are associated with the new server you have just configured. Review this information and, if necessary, select the **Back** button to make needed corrections. If the information is correct, select the **Next** button to proceed. If you select **Cancel**, the server and spawner object will not be created.
- **Modify Existing Spawner**. This screen displays a list of existing spawner definitions that contain the same domain, host name, and logical name as the new server. If you want to use an existing spawner, select the spawner's name and the **Next** button; the wizard will skip to the Done screen, where you can finish defining the new server. If you want to create a new spawner definition using the displayed domain, host name, and logical name, select the **Create** button.
- **Enter a Name and Description for the Spawner**. On this screen, enter a name (sasSpawnercn) and description (description) for the new spawner.
- **Specify the Operator Information**. On this screen, enter the service (sasOperatorService) and/or port (sasOperatorPort) and the password (sasOperatorPassword) to allow a user to connect to the new spawner for administration purposes.
- **Specify the Encryption Information**. On this screen, select the key length (sasNetEncrKey).
- **Done**. On this screen, select the **Advanced** button if you want to add UUID support, or if you want to configure logging (For Version 9, dependent spawners are not supported). Otherwise, select **Next** to skip to the Done screen, where you can finish defining the new server and spawner. If you select **Cancel**, the server and spawner definitions will not be created.
- **Specify the UUID Information**. If you are setting up the spawner for UUID generation, specify the service (sasUUIDService), port (sasUUIDPort), and node (sasUUIDNode) to allow SAS to connect to the spawner and obtain UUID information. For more information about UUID generation, see Configuring a UUID Generator.
- **Specify the Spawner Logging Information**. On this screen, specify the log file name and path (sasLogFile), and select the checkbox to indicate whether you want to use verbose logging (sasVerbose).
- **Specify the OS/390 Logical Unit Name**. Version 9 does not use the Logical Unit Name field. This screen appears only if the server is a z/OS machine.
- **Done**. On this screen:

    ◊ You must select **Finish** to create the new server and spawner definitions and exit the wizard.

- ◊ If you would like to add another host machine to the server before exiting, select the **Add host** button.
- ◊ If you would like to define another server before exiting, select the **Do Another** button.
- ◊ If you select the **Cancel** button, the server and spawner definitions will not be created.

When you complete all of the steps in the above procedure, your LDAP directory will contain fully defined sasServer, sasSpawner, and sasLogicalNameInfo objects. You can now start the spawner and begin using the server. For details, see Invoking/Starting the Spawner.

IOM Bridge Servers

Using IT Administrator to Define the Metadata (IOM Bridge)

If you are using LDAP as the metadata repository, you can use the Integration Technologies Administrator graphical user interface to create and modify the metadata for your IOM Bridge server configuration. For general information about IT Administrator, including installation procedures and general usage instructions, refer to Using the Integration Technologies (IT) Administrator.

The following sections provide specific instructions for using IT Administrator to set up an IOM Bridge server configuration:

- Using IT Administrator to Define a SAS Login (IOM Bridge)
- Using IT Administrator to Define an Object Server (IOM Bridge)
- Using IT Administrator to Define a Spawner (IOM Bridge)

Alternatively, you can use the IT Administrator wizard to perform the server and spawner definition. For details, see Using the IT Administrator Wizard to Define an IOM Bridge Server.

IOM Bridge Servers

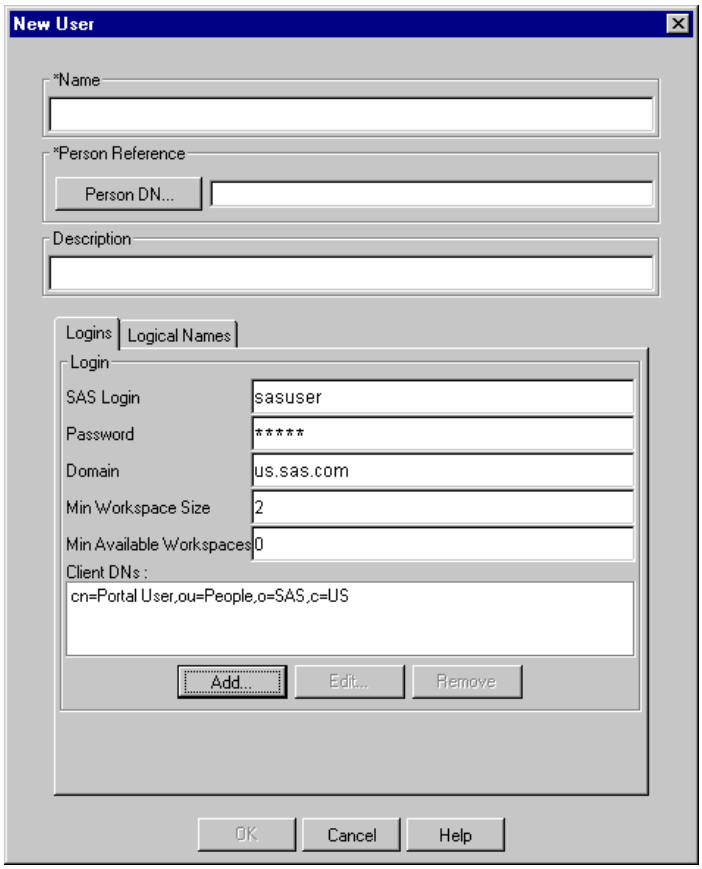
Using IT Administrator to Define a SAS Login (IOM Bridge)

A SAS login may need to be available in order to start a SAS session on a server or to connect to a client. Each SAS login definition contains a user name, password, and domain, as well as a pointer to the user's person reference entry in the LDAP directory.

SAS logins may be used to provide credentials when creating a client connection. Whether or not SAS logins are required depends on the method calls used to start the server or create the connection. For example, you might need logins for pooling. If you do not use logins, you must track and specify the user credentials manually.

To define a SAS login using IT Administrator:

1. Open IT Administrator.
2. In the manager bar, select **SAS Configuration**.
3. In the tree view, select the **SAS Logins** folder (found under the **SAS Servers** folder); then select the **New** button ( ) on the toolbar. Alternatively, you can select **File** → **New** → **SAS Login** from the menu bar. The following window appears:



4. Enter the necessary attributes. The attribute fields that are marked with an asterisk (*) are required. The **OK** button will remain grayed out until you have entered all of the required fields. Select the **Help** button on any tab to display entry instructions. Brief entry instructions are provided below. For detailed information about a field, click on the corresponding attribute name, which is shown in parentheses. The appropriate definition on the sasLogin Attributes List will be displayed. You can then use your browser's **Back** button to return to this page.

a. Enter a unique name (sasLogincn) for the login.
b. Optionally, enter a description (description).
c. Specify the person reference (sasreferenceDn) in the LDAP directory that corresponds to this login. To do so, click the **Select** button to display the Person Index window, which contains a tabbed list of all the person references in the LDAP directory. Select the appropriate reference, and click **OK**. If you need to deselect an item, press the **Ctrl** key and click the mouse button.
d. On the **Logins** tab:

    i. Enter the SAS Login (sasLoginName), or user name, for this login and the associated Password (sasUserPassword).
    ii. Enter the Domain (sasDomainName). Note that the login must use the same domain as the server on which SAS sessions will be established.
    iii. Optionally, enter the Min Workspace Size (sasMinSize) and Min Available Workspaces (sasMinAvail).
    iv. Under Client DNs, click **Add** and enter the distinguished name (sasAllowedClientDN) of each user or group of users that will be authorized to enter to a workspace pool.

e. On the **Logical Names** tab, select one or more logical names (sasLogicalName) that this login is to be associated with. If you want to create a new logical name, select the **Add** button to create a new sasLogicalNameInfo object. For more information about logical names, refer to Assigning Logical Names.

5. When you are finished entering information in the fields, select **OK**. The new spawner object appears in the tree view.

To modify a login definition using IT Administrator:

1. Open IT Administrator.
2. In the manager bar, select **SAS Configuration**.
3. In the tree view, find the **SAS Logins** folder (found under the **SAS Servers** folder) and click the plus sign to open it.
4. Select the SAS login object that you wish to modify. The login's current attributes will be displayed in the property view in the right portion of the window.
5. Select the appropriate tabs, and enter the necessary changes. For a description of the fields, refer to the sasLogin Attributes List.
6. When you are finished, select the **Save** icon ( ) on the toolbar; or select **File → Save** from the menu bar. (If you skip this step, IT Administrator will prompt you to save your changes when you attempt to navigate to another object.)

IOM Bridge Servers

Using IT Administrator to Define a Server (IOM Bridge)

The SAS Integration Technologies Administrator provides a graphical user interface that allows you to create or modify a definition for an IOM Bridge server. (Alternatively, you can use the IT Administrator wizard to perform the initial creation. For details, see Using the IT Administrator Wizard to Define an IOM Bridge Server.)

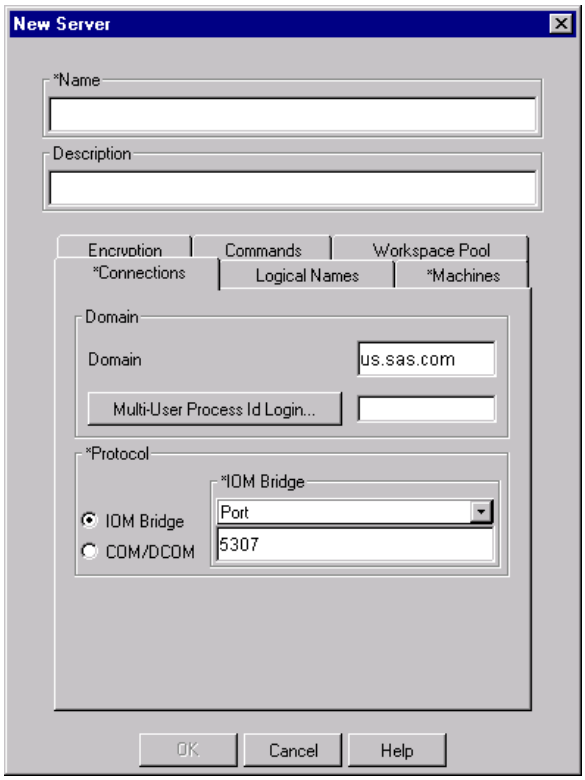
Note: The following attributes which appear in the Commands tab of the IT Administrator interface are not used in Version 9 of SAS Integration Technologies

- Transaction Program Name
- Partner Logical Unit Name

The Multi−User Process ID Login attribute which appears in the Connections tab of the IT Administrator interface is not used in Version 9 of SAS Integration Technologies. If you are using Version 9, do not use these attributes for your configuration.

To define an IOM Bridge server object using IT Administrator:

1. Open IT Administrator.
2. In the manager bar, select **SAS Configuration**.
3. In the tree view, select the **Servers** folder (found under the **SAS Servers** folder); then select the **New** button ( ) on the toolbar. Alternatively, you can select **File → New → Server** from the menu bar. The following window appears:



4. Enter the necessary attributes. The attribute fields that are marked with an asterisk (*) are required. The **OK** button will remain greyed out until you have entered all of the required fields. Select the **Help** button on any tab to display entry instructions. Brief entry instructions are provided below. For detailed information about a field, click on the corresponding attribute name, which is shown in parentheses. The appropriate definition on

the sasServer Attributes List will be displayed. You can then use your browser's **Back** button to return to this page.

- Enter a unique name (sasServercn) for the server. Optionally, enter a description (description).
- On the **Connections** tab:

    a. Enter a Domain (sasDomainName). Note that you must use the identical domain name if you define a spawner for this server.
    b. Select the **IOM Bridge** protocol (sasProtocol).
    c. From the pull−down menu that appears, select either **Port** or **Service** and enter either the service (sasService) or the port number (sasPort) in the field below the menu. The port number is required if the server will have Java clients.
- On the **Logical Names** tab, select one more logical names (sasLogicalName) that this server is to be associated with. If you want to create a new logical name, select the **Add** button to create a new sasLogicalNameInfo object.
- On the **Machines** tab, select the **Add** button, and then enter the fully qualified host name (sasMachineDNSName) for the machine on which the server is to run. Repeat for each additional host machine. To change an entry, highlight the machine name and select **Edit**. To remove an entry, highlight the machine name and select **Remove**.
- On the **Encryption** tab, specify client encryption algorithm (sasClientEncryptionAlgorithm) and server (sasNetEncrAlg) encryption algorithms. Also make a selection to indicate what to encrypt (sasRequiredEncryptionLevel).
- On the **Commands** tab, complete the **Command for non OS/390** field with the command (sasCommand) used to start the object server. Include the path relative to the directory in which the spawner will be started.

    Note: For the z/OS, you can now use the (sasCommand) attribute to launch SAS as an object server. Since the IT Admin interface has not changed for Version 9, you can specify the launch command in the **Command for non OS/390** IT Admin field.
- You can use the **Workspace Pool** tab to set up workspace pooling for the server. Specify the Maximum Workspaces per Workspace Pool (sasMaxPerWorkspacePool) and the Recycle Activation Limit (sasMaxPerWorkspacePool) in the fields provided. (Select the **Default** button if you want to reset the Recycle Activation Limit to its default value of 10.) Then select **Leave running when idle** (sas−ServerRunForever); or select **Minutes until idle shutdown** and enter the number of minutes (sas−ServerShutdownAfter) after which an idle server should be shut down.

5. When you are finished entering information in the fields, select **OK**. The new server object appears in the tree view.

To modify an IOM Bridge server object using IT Administrator:

1. Open IT Administrator.
2. In the manager bar, select **SAS Configuration**.
3. In the tree view, find the **Servers** folder (found under the **SAS Servers** folder) and click the plus sign to open it.
4. Select the server object that you wish to modify. The server's current attributes will be displayed in the property view in the right portion of the window.
5. Select the appropriate tabs, and enter the necessary changes. For a description of the fields, refer to the sasServer Attributes List.
6. When you are finished, select the **Save** icon ( ) on the toolbar; or select **File → Save** from the menu bar. (If you skip this step, IT Administrator will prompt you to save your changes when you attempt to navigate to another object.)

IOM Bridge Servers

Using IT Administrator to Define a Spawner (IOM Bridge)

The SAS Integration Technologies Administrator provides a graphical user interface that allows you to create or modify a definition for a spawner to run on an IOM Bridge server. (Alternatively, you can use the IT Administrator wizard to perform the initial creation of a server and spawner. For details, see Using the IT Administrator Wizard to Define an IOM Bridge Server.)

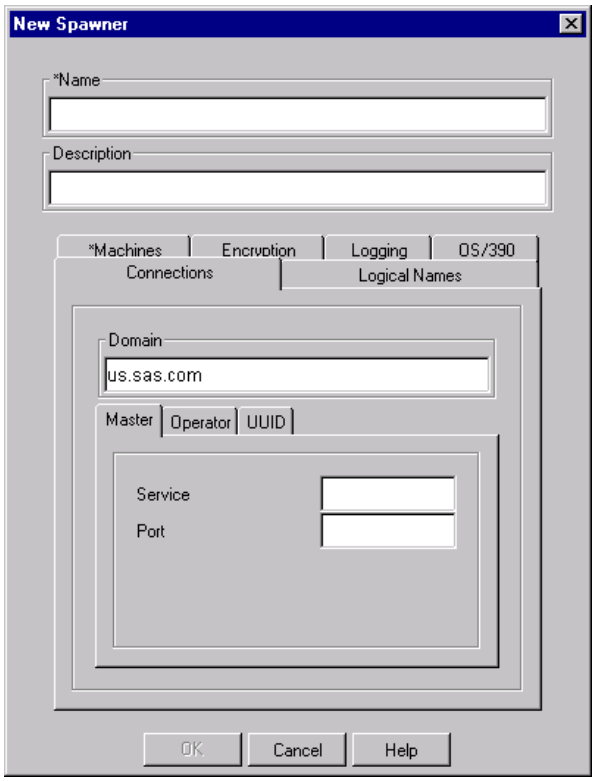
Note: The following attributes which appear in the IT Administrator interface are not used in Version 9 of SAS Integration Technologies

- Encryption Modules Path
- Logical Unit Name
- Master tab

If you are using Version 9, do not use these attributes for your configuration.

To define an IOM Bridge spawner object using IT Administrator:

1. Open IT Administrator.
2. In the manager bar, select **SAS Configuration**.
3. In the tree view, select the **Spawners** folder (found under the **SAS Servers** folder); then select the **New** button ( ) on the toolbar. Alternatively, you can select **File → New → Spawner** from the menu bar. The following window appears:



4. Enter the necessary attributes. The attribute fields that are marked with an asterisk (*) are required. The **OK** button will remain greyed out until you have entered all of the required fields. Select the **Help** button on any tab to display entry instructions. Brief entry instructions are provided below. For detailed information about a field, click on the corresponding attribute name, which is shown in parentheses. The appropriate definition on

the sasSpawner Attributes List will be displayed. You can then use your browser's **Back** button to return to this page.

a. Enter a unique name (sasSpawnercn) for the spawner. Optionally, enter a description (description).
b. On the **Connections** tab:

i. Enter a Domain (sasDomainName). The spawner must use the same domain as the server with which it connects.
ii. On the **Operator** subtab, enter the service (sasOperatorService) or port (sasOperatorPort) and the password (sasOperatorPassword) that operators will use to connect to the spawner in order to perform administration tasks (such as checking status).
iii. On the **UUID** subtab, enter the service (sasUUIDService) or port (sasUUIDPort) and the password (sasUUIDNode) with which to connect to request UUID generation. See the section titled Request a Universal Unique Identifier (UUIDGEND) for more information regarding UUID generation by the spawner.

c. On the **Logical Names** tab, select one more logical names (sasLogicalName) that this spawner is to be associated with. If you want to create a new logical name, select the **Add** button to create a new sasLogicalNameInfo object. For more information about logical names, refer to Assigning Logical Names.
d. On the **Machines** tab, select the **Add** button, and then enter the fully qualified host name (sasMachineDNSName) for the machine on which the spawner is to run. Repeat for each additional host machine. To change an entry, highlight the machine name and select **Edit**. To remove an entry, highlight the machine name and select **Remove**.
e. On the **Encryption** tab, select the key length (sasNetEncrKey).
f. On the **Logging** tab, specify the log file name and path (sasLogFile), and select the checkbox to indicate whether you want to use verbose logging (sasVerbose).

5. When you are finished entering information in the fields, select **OK**. The new spawner object appears in the tree view.

To modify a spawner definition using IT Administrator:

1. Open IT Administrator.
2. In the manager bar, select **SAS Configuration**.
3. In the tree view, find the **Spawners** folder (found under the **SAS Servers** folder) and click the plus sign to open it.
4. Select the spawner object that you wish to modify. The spawner's current attributes will be displayed in the property view in the right portion of the window.
5. Select the appropriate tabs, and enter the necessary changes. For a description of the fields, refer to the sasSpawner Attributes List.
6. When you are finished, select the **Save** icon ( ) on the toolbar; or select **File → Save** from the menu bar. (If you skip this step, IT Administrator will prompt you to save your changes when you attempt to navigate to another object.)

IOM Bridge Servers

Using a Configuration File to Define the Metadata (IOM Bridge)

For LDAP, if you do not use a metadata server as the metadata repository, you can create a flat configuration file that contains the object definitions for an IOM Bridge server configuration. The configuration file must then be installed on the server and on each client machine.

Note: If your configuration requires more than one or two servers, or if multiple clients will be using the servers, we strongly recommend the use of LDAP as a central metadata repository. The use of LDAP also gives you the ability to use access control lists to control access to the servers in your enterprise.

To define an IOM Bridge server configuration using a configuration file:

1. Use a text editor to code the configuration file. At a minimum, the file must define a server object and a spawner object. You can also define one more SAS login objects, and/or logical name objects. To create the file:
    - Refer to the attribute descriptions for each object type:
        - Attributes for sasLogin
        - Attributes for sasLogicalNameInfo
        - Attributes for sasServer
        - Attributes for sasSpawner
    - Refer to the following examples:
        - Example Minimal Configuration
        - Example Server and Spawner
        - Example Using Logical Names
        - Example UUID Generator
    - Use the LDAP Data Interchange Format (LDIF), format, which has the following syntax rules:
        - Start each entry in column one.
        - To indicate a comment line, place '#' in column one.
        - Use the following general format for each entry: "attribute: value".
        - If an entry spans multiple lines, insert a blank in the first column of each continuation line. The blank in column one is a continuation character and is consumed by the LDIF file parser. Therefore, it should not be considered part of the entry.
        - A blank line must precede a distinguished name (exclude 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 spawner's LDIF parser relies on this requirement in order to separate object class definitions. The DN name can be any value.
        - Two consecutive blank lines indicate the end of the configuration file definitions.
2. Save the file with a name of your choice (for example, objspawn.cfg).
3. Install the file on the server machine and on each client machine.

You can now start the spawner. For instructions, refer to Starting the Spawner.

IOM Bridge Servers

Configuring a UUID Generator

Currently, only SAS on Windows can generate unique UUIDs. The UUID Generator Daemon (UUIDGEND) generates unique UUIDs for SAS sessions that execute on hosts without native UUID generation support.

Installing UUIDGEND

If your SAS application executes on a platform other than Windows and your application requires unique UUIDs, install UUIDGEND and identify its location (see SAS UUIDGENDHOST and UUIDCOUNT options documentation) to your executing SAS application. If you install UUIDGEND on a host other than Windows, you need to contact SAS Technical Support to obtain a UUID node. The UUID node must be unique per UUIDGEND installation in order for UUIDGEND to guarantee truly unique UUIDs.

Configuring the Spawner for UUIDGEND

UUIDGEND is implemented in the spawner. You can execute a separate spawner to support UUIDGEND only, or you can update an existing spawner instance to support UUIDGEND along with its sasServer definitions. UUIDGEND is configured by the sasSpawner object class sasUUIDPort/sasUUIDService and sasUUIDNode (when it is not installed on Windows) attributes. All other sasSpawner definition requirements must be met.

For an example of the metadata needed to define a spawner for UUID generation, refer to the UUID Generator configuration file example.

IOM Bridge Servers

Configuring and Starting the Object Spawner on z/OS

On a z/OS server machine, the spawner starts a SAS object server session in response to a request from a client.The client communicates first with the spawner, and then with the object server, using TCP/IP. The object spawner runs as a started task; before it can handle client requests, you must start it using a started task procedure.

The following setup tasks are required:

1. Configure TCP/IP
2. Create the object spawner started task
3. Create a SAS startup command

Note: This page is intended to serve as an outline of the process, rather than a step−by−step guide, for setting up a spawner on a z/OS platform.

Task 1: Configure TCP/IP

The overall configuration of TCP/IP is outside the scope of this discussion. Assuming that a functioning TCP/IP link is already in place between the client and the z/OS server, the following additional step is required to support the object spawner:

- Make sure that the SAS/C Transient Runtime Library (CTRANS), IBM TCPIP.DATA, and TCP/IP SERVICES configurations are available to both the object spawner and its object servers.

If you specify TCP/IP service names rather than ports in the spawner configuration, you must define the services in the TCP/IP services file. For example, the default spawner operator listen service name is `sasobjoper` and the default spawner server listen service name is `sasobjspawn`. To define these in the TCP/IP services file, add the following two lines:

```
sasobjoper       8582/tcp
sasobjspawn      8581/tcp
```

Task 2: Create the Object Spawner Started Task

The object spawner runs as a started task (STC). Its job is to listen for requests from clients and pass them to the startup command associated with the service/port in which there is activity. The startup command will start a server session. You must create a procedure in a system PROCLIB library (SYS1.PROCLIB, for example).

Create the Procedure

Because z/OS Job Control Language has a parameter line length restriction of 100 characters, you can use DDNames to identify filenames in object spawner parameters. When a file pathname is 8 characters or less, the file pathname is first checked to see if it matches a DDName. If so, the DDName is used. If DDNames are not used for the config file and log file, you need to specify a config file and log file in the UNIX file system.

If you need to specify more than 100 characters for command line parameters, put the additional parameters in a z/OS data set or UNIX file and reference it using the =<//DDN:PARMS parameter.

The following procedure explicitly specifies the pathname for the config file and uses a DDName to reference the log file in the command line parameters for the object spawner.

```
//OBJSPAWN PROC PROG=OBJSPAWN,
//  OPTIONS='-CONFIGFILE /usr/lpp/SAS/objspawn.cfg ',
//  OPT2='-SASVERBOSE -SASLOGFILE LOGFILE'
//OBJSPAWN EXEC PGM=&PROG,REGION=512M,
//        PARM='&OPTIONS &OPT2 =<//DDN:PARMS'
//STEPLIB  DD DISP=SHR,DSN=SYS2.SAS.LIBRARY
//CTRANS   DD DISP=SHR,DSN=SYS2.SASC.TRANSLIB
//PARMS    DD DISP=SHR,DSN=SYS2.OBJSPAWN.PARMS
//TKMVSJNL DD PATH='/tmp/objspawn/JNL.&LYYMMDD..&LHHMMSS..txt',
//         PATHMODE=(SIRUSR,SIWUSR,SIRGRP,SIROTH),
//         PATHOPTS=(OWRONLY,OCREAT,OTRUNC)
//LOGFILE  DD PATH='/tmp/objspawn/LOG.&LYYMMDD..&LHHMMSS..txt',
//         PATHMODE=(SIRUSR,SIWUSR,SIRGRP,SIROTH),
//         PATHOPTS=(OWRONLY,OCREAT,OTRUNC)
```

Remember that the STC has access to the SAS/C Transient Runtime Library (CTRANS).

The -CONFIGFILE parameter identifies the LDIF format configuration file that the spawner is to use.

The -SASVERBOSE and -SASLOGFILE options in the STC procedure provide useful information for diagnosing connection problems. It is a good idea to include these options until you are satisfied that everything is working correctly.

Create the LDIF Configuration File

The DDname support does not allow for empty lines in the LDIF configuration file. Instead of using an empty line in the LDIF file, use a line containing at least one blank.

The following LDIF configuration file is a generic template:

```
dn: sasSpawnercn=MySpawner
objectClass: sasSpawner
sasSpawnercn: MySpawner
sasMachineDNSName: localhost
sasOperatorPort : 5306

# This is a comment line!
dn: sasServercn=MyServer
objectClass: sasServer
sasCommand: /usr/bin/startsas.sh --
sasPort: 5307
sasProtocol: bridge
sasMachineDNSName: localhost
sasServercn: MyServer
```

In the previous configuration file:

- the first paragraph provides the definitions for the object spawner (the STC). It contains the name of the operator service that is defined in the TCP/IP services file, and the operator password.
- the second paragraph defines the object server. It gives details about the TCP/IP port that it is listening to, and the command to use to launch an object server.

For a list of all of the available options, refer to the Attributes for Spawner and Attributes for Server. In general, simple procedures work best. Any option can become the source of problems, so include only the options that you need.

After you have created the STC procedure, you can start the object spawner by issuing an operator command:

Define the Object Spawner System Security Configuration

The z/OS system considers the object spawner a daemon process. Therefore, if the BPX.DAEMON profile of the RACF Facility class is active and RACF program control is enabled, then the SAS and SAS/C load libraries specified in the STC procedure must be program controlled. However, the userid under which the object spawner runs does not require RACF READ access to the BPX.DAEMON profile.

If the following messages appear in the z/OS system log when a client attempts to connect, then a necessary library is not program controlled.

```
ICH420I  PROGRAM program-name [FROM LIBRARY dsname]
         CAUSED THE ENVIRONMENT TO BECOME UNCONTROLLED.
BPXP014I ENVIRONMENT MUST BE CONTROLLED FOR DAEMON (BPX.DAEMON)
         PROCESSING
```

Start the Object Spawner

```
START OBJSPAWN
```

For a list of all available spawner invocation options, see Spawner Invocation Options. If there are no configuration errors, the object spawner will assume a listening state by entering a detected wait state (DW).

Task 3: Create a SAS Startup Command

Create the Startup Command

The startup command is meant to build a parameter string that is capable of launching SAS. The startup command in the spawner configuration must end with '--' to indicate the end of the user specified parameters. Here is a sample shell script (startsas.sh):

```
#!/bin/sh
#
#  foundDashDash is a boolean.  When TRUE,
#  we found the string "--" in our arguments.
#
foundDashDash=0

#
#  Construct our arguments
#
args=''
for arg in "$@" ; do
   if [ "$arg" != "--" ]; then
      tmp="$arg ";
   else
      tmp="SRVOPTS('')";
```

```
      foundDashDash=1;
   fi
   args="$args$tmp"
done

#
#  If we found a "--", we need to close the SRVOPTS
#  option.
#
if [[ $foundDashDash -ne 0 ]]; then
   args="$args '')"
fi

#
#  Construct the command line...
#
cmd="/bin/tso -t EX 'SYS2.TSO.CLIST(SPWNSAS)'"
cmd="$cmd 'nosasuser $args'"

#
#  Set environment variables...
#  Account data can be used to place SAS in the correct
#  WLM service class. SYSPROC specifies the data set
#  containing the SAS CLIST/REXX.
#
export _BPX_ACCT_DATA=MYNAME1
export SYSPROC=SYS2.TSO.CLIST

#
#  Start up SAS
#
exec $cmd
```

The sample invokes the /bin/tso/ UNIX command to execute the CLIST `SYS2.TSO.CLIST(SPWNSAS)`. Replace the CLIST data set name SYS2.TSO.CLIST with the name appropriate to your site. The control (CNTL) data set that you created for your SAS install contains an example CLIST for use in launching IOM server sessions.

Note: The SAS CLIST requires the following parameters:

- NOSASUSER to allow more than one concurrent SAS session per user. NOSASUSER suppresses allocation of a SASUSER data set.
- SRVOPTS() in order to pass in the objectserver options.

Specify Account Data

The IOM spawner on z/OS uses the Unix System Services spawn function to initiate a process to run an IOM server. This process runs in a USS initiator (BPXAS). By default, the process runs with the default Work Load Manager (WLM) service class that was assigned to OMVS work during installation. The default service class might have been defined with a goal of providing USS shell commands with good response times. This default service class assumes the requests are relatively short. Because work associated with IOM requests might require more time, it might be desirable to assign IOM servers to a different service class.

You can use MVS accounting data to assign the work to a specific Work Load Manager service class. To set the accounting data, use the _BPX_ACCT_DATA environment variable in the startsas.sh script that starts that SAS IOM server session. The server session then runs with the accounting data. For example:

```
export _BPX_ACCT_DATA=MYNAME1
```

To assign a Work Load Manager service class based on the accounting data, use the WLM AI classification rule. For example (in the WLM ISPF dialog):

```
            Qualifier                     Class
   Type       Name     Start             Service   Report
                              DEFAULTS: OMVSSHRT  ________
1  AI         MYNAME1  1                 OMVSLONG  ________
```

For more information about using accounting information with USS processes, consult *Unix System Service Planning*. For information about defining WLM service classes with appropriate characteristics, and for information about specifying classification rules to use these classes, see *MVS Planning: Workload Management*.

Because you might define different IOM servers, in order to segregate different work loads, you may also specify that these servers run in different service classes. To specify different service classes, create a separate server definition for each class of work in the spawner config file, and assign client requests to the listen port associated with each server.

IOM Bridge Servers

Invoking (Starting) the Spawner

After you have used IT Administrator or configuration files to define a server and spawner, you are ready to invoke and administer the defined spawners. Refer to the appropriate startup procedures for your server platform:

- Starting the Spawner on Windows
- Starting the Spawner on UNIX
- Starting the Spawner on Alpha/VMS

As you use these instructions, refer to the list of Spawner Invocation Options that are available.

After you have started the spawner, you can connect to the spawner as an administrator (operator) to monitor and control the spawner's operation. For instructions, see Monitoring the Spawner Using Telnet.

Security Considerations

The spawner can be launched with the *−nosecurity* option. However, this option should be used with caution, since it will allow any client connecting to the spawner to obtain an object server using the same user ID that launched the spawner. This means that any client that can manipulate the host file system can obtain an object server as if the client had the user ID that launched the spawner.

Note: If you use the *−nosecurity* option, the *−install* option is ignored.

Example Commands

The following are examples of the spawner command in the UNIX and Windows NT environments:

- UNIX example using an LDAP server:

```
prompt> /sasv9/utilities/bin/objspawn −sasverbose
        −saslogfile /usr/logs/obj.log
        −ldaphost ldapsrv.alphalite.com  −ldapport 389
        −sasSpawnercn Finance
        −ldapbase "sascomponent=sasServer,cn=ldapsrv,
           o=Alphalite Airways,c=us"
```

- UNIX example using a configuration file:

```
prompt> /sasv9/utilities/bin/objspawn −configFile objspawn.cfg
```

- Windows NT example for launching a spawner that connects to a secure LDAP server with a default port of 389. The −sasSpawnercn option tells the spawner to use the sasSpawner definition of IOMSpawner. The −ldapBase tells the spawner where in the tree to start searching for the server, spawner, and login definitions. This −ldapbinddn option tells the spawner to bind to the LDAP server with the user "uid=sasiom1,ou=People,o=ABC Inc,c=US"

```
c:\sasv9> objspawn −ldapHost machine.abc.sas.com −ldapPort 389
  −ldapBase "sasComponent=sasServer,cn=SAS,o=ABC Inc,c=US"
  −sasSpawnercn IOMSpawner −sasverbose
  −ldapbinddn "uid=johndoe,ou=People,o=ABC Inc,c=US"
  −ldapPw mypassword −sasLogFile objspawnldap.log
```

- Windows NT example for launching a spawner with an LDIF configuration file:

```
c:\sasv9>objspawn −configFile c:\sasv9\objspawn.cfg
```

Notes:

1. In the first and third examples, the command line options point the spawner to the LDAP server and base distinguished name where the configuration parameters are located.
2. In the second and fourth examples, the command line options point the spawner to a configuration file (objspawn.cfg) where the configuration parameters are located.
3. The invocation options vary depending on the platform. Refer to the Spawner Invocation Options for details.
4. On Windows, in most cases you should install the spawner as an NT service using the −install option.
5. If you do not specify the −sasSpawnerCn option, the object spawner uses the first sasSpawner definition (on the LDAP server) that has the same machine name as the current host.

IOM Bridge Servers

Starting the Spawner on Windows

To start the spawner on a Windows host:

1. **Note:** This step is only necessary if you are not starting the spawner as a service.

   Define the user rights for the administrator. The user who invokes the spawner, in addition to being an administrator, must have the following user rights:

   - replace the process level token.
   - act as part of the operating system (Windows NT and Windows 2000, not needed if you install the spawner as a service)
   - adjust memory quotas for a process (Windows XP only, not needed if you install the spawner as a service)
   - increase quotas (Windows NT and Windows 2000, not needed if you install the spawner as a service)

   To set the administrator's user rights on Windows NT:

   a. Select **Start** ➧ **Programs** ➧ **Administrative Tools** ➧ **User Manager**.
   b. From the Policies pull−down menu, select **User Rights**.
   c. Click the **Show Advanced User Rights** check box.
   d. Add rights using the **Right:** drop−down menu.

   To set the administrator's user rights on Windows 2000:

   a. Select **Start** ➧ **Settings** ➧ **Control Panel** ➧ **Administrative Tools** ➧ **Local Security Policy**.
   b. Select **Security Settings** ➧ **Local Policies** ➧ **User Rights Assignment**.
   c. Add rights by double−clicking on each right and assigning the appropriate users.

   To set the administrator's user rights on Windows XP:

   a. Select **Start** ➧ **Settings** ➧ **Control Panel** ➧ **Administrative Tools** ➧ **Local Security Policy**.
   b. Expand the tree for Local Policies and select **User Rights Assignment**.
   c. Add rights by double−clicking on each right and assigning the appropriate users.

2. Define the user rights for each client that connects to the spawner. Similar to the administrator, each client that connects to the spawner must have the following user right: **log on as batch job**.
3. Restart Windows to apply the new user rights.
4. Start the spawner program (called objspawn.exe) using a command that specifies the appropriate options. The spawner program is installed in your installed SAS folder. In most cases, you should install the spawner as a service. Refer to the Spawner Invocation Options for a complete list of valid options for the command.

   In the following examples, c:\sasv9 is the installed SAS folder and `objspawn.cfg` is an LDIF configuration file. In the last example, `spawner2.xml` is an XML metadata configuration file that was created using the Integration Technologies Configuration application.

   Note: Some of the example commands are broken into more than one line for presentation purposes. However, the command must be entered as a continuous text stream on the command line.

   - The following command installs the spawner as an NT service and updates the registry to hold the options that are specified (in this case `−configFile`):

```
c:\sasv9> objspawn −configFile c:\sasv9\objspawn.cfg
   −install
```

When the spawner is started as an NT service, it will self configure utilizing the options that are placed in the registry at install time.

- The following command installs the spawner as an NT service, specifies service dependencies, and names the service:

```
c:\sasv9> objspawn −installDependencies "service1;service2"
  −name serviceName −configFile c:\sasv9\objspawn.cfg −install
```

When the spawner is started as an NT service, it will self configure utilizing the options that are placed in the registry at install time.

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

```
c:\sasv9> objspawn −deinstall
```

- The following command launches a spawner with an XML metadata configuration file (created Using the Integration Technologies Configuration Application) that contains information for accessing the LDAP server:

```
c:\sasv9> objspawn −xmlconfigFile c:\sasv9\spawner2.xml
```

- The following command launches the spawner on a nonsecure LDAP server with a default port of 389. The −ldapBase tells the spawner where in the tree to start searching for the server, spawner, and login definitions. This command assumes that there is only one spawner definition in the ldap server. (If there are multiple spawner definitions, the user should specify the −sasSpawnercn option. If the −sasSpawnercn option is not specified, the spawner uses the first spawner definition that it finds in the server.)

```
c:\sasv9> objspawn −ldapHost machine.abc.sas.com
   −ldapBase "sasComponent=sasServer,cn=SAS,o=ABC Inc,c=US"
```

- The following command launches the spawner on a nonsecure LDAP server with a default port of 389. The −ldapBase tells the spawner where in the tree to start searching for the server, spawner, and login definitions. The −sasSpawnercn option tells the spawner which spawner definition to use.

```
c:\sasv9> objspawn −ldapHost machine.abc.sas.com
  −ldapBase "sasComponent=sasServer,cn=SAS,o=ABC Inc,c=US"
  −sasSpawnercn mySpawner
```

- The following command launches a spawner that connects to a nonsecure LDAP server listening on port 12345. The −sasSpawnercn option tells the spawner to use the spawner definiton WNTSpawner. The −ldapBase tells the spawner where in the tree to start searching for the server, spawner, and login definitions.

```
c:\sasv9> objspawn −ldaphost machine.abc.sas.com
  −ldapport 12345 −sasspawnercn "WNTSpawner"
  −ldapbase "sasComponent=sasServer,cn=SAS,o=ABC Inc,c=US"
```

- The following command start an Active Directory spawner that is set up as secure. The −sasSpawnercn option tells the spawner to use the sasSpawner definition of WNTSpawner. The −ldapBase tells the spawner where in the tree to start searching for the server, spawner, and login definitions. This −ldapbinddn option tells the spawner to bind to the LDAP server with the username and password found in the login definition CN=John Doe,CN=Users,DC=dtd−dom,DC=sas,DC=com.

```
c:\sasv9> objspawn −ldaphost machine.abc.sas.com
  −ldapport 389 −sasspawnercn "WNTSpawner"
  −ldapbase "cn=AdHoc,cn=Applications,dc=dtd−dom,dc=sas,dc=com"
```

```
  −ldap_binddn "CN=John Doe,CN=Users,DC=dtd−dom,DC=sas,DC=com"
  −lpw mypassword
```

- The following command start a spawner that connects to a secure LDAP server with a default port of 389. The −sasSpawnercn option tells the spawner to use the sasSpawner definition of IOMSpawner. The −ldapBase tells the spawner where in the tree to start searching for the server, spawner, and login definitions. This −ldapbinddn option tells the spawner to bind to the LDAP server with the user "uid=sasiom1,ou=People,o=ABC Inc,c=US"

```
c:\sasv9> objspawn −ldapHost machine.abc.sas.com
  −ldapPort 389 −sasSpawnercn IOMSpawner
  −ldapBase "sasComponent=sasServer,cn=SAS,o=ABC Inc,c=US"
  −ldapbinddn "uid=johndoe,ou=People,o=ABC Inc,c=US"
  −ldapPw mypassword −sasLogFile objspawnldap.log −sasverbose
```

- The following command launches a spawner with an LDIF configuration file:

```
c:\sasv9> objspawn −configFile c:\sasv9\objspawn.cfg
```

Note: After the spawner is started, a message is written to the application event log indicating whether objspawn initialization completed or failed.

IOM Bridge Servers

Starting the Spawner on UNIX

The SAS object server is launched in the client's home directory (as specified in the client's password entry). If the client has a directory in its home directory that is named the same as its user ID, SAS will use that directory as the SAS session's SASUSER path.

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

```
export DISPLAY=<machine name>:0.0
```

Ensure that the setuid root bit is set for elssrv, sasauth, and sasrun. If the setuid root bit is not set for these utilities, objspawn will not be able to launch SAS sessions. For details about setting the setuid root bit, see Changing the `setuid` Permissions to Root.

Start the spawner program (called objspawn) using a command that specifies the appropriate options. Refer to the Spawner Invocation Options for a complete list of valid options for the command. The following examples use "/sasv9/" as the directory in which SAS was installed.

Note: Some of the example commands are broken into more than one line for presentation purpose. However, the command must be entered as a continuous text stream on the command line.

- The following command launches the spawner with a configuration file:

  ```
  prompt> /sasv9/utilities/bin/objspawn
     -configFile objspawn.cfg
  ```
- The following command launches the spawner with the configuration information on an LDAP directory server:

  ```
  prompt> /sasv9/utilities/bin/objspawn
     -ldapHost machine.abc.com
     -ldapBase "sasComponent=sasServer,
        cn=SAS,o=ABC Inc,c=US"
  ```
- The following command launches the spawner, specifying the sasSpawner definition to use, with the configuration information on an LDAP server:

  ```
  prompt> /sasv9/utilities/bin/objspawn
     -ldapHost machine.abc.com
     -ldapBase "sasComponent=sasServer,
        cn=SAS,o=ABC Inc,c=US"
     -sasSpawnercn mySpawner
  ```

Note: After the spawner is started, an attempt is made to write a message to stdout indicating whether objspawn initialization completed or failed.

Changing the `setuid` Permissions to Root

You can change the `setuid` permissions of files in `!SASROOT/utilities/bin` to root using either of the following methods.

Method 1: Using SAS Setup

1. Log in to the root account.

   ```
   $ su root
   ```
2. Run SAS Setup from `!SASROOT/sassetup`.
3. Select **Run Setup Utilities** from the SAS Setup Primary Menu.
4. Select **Perform SAS System Configuration.**
5. Select **Configure User Authorization.**

Method 2: Using the Command Line

From a Unix prompt, type the following:

```
$ su root
# cd !SASROOT/utilities/bin
# chown root elssrv sasauth sasperm sasrun
# chmod 4755 elssrv sasauth sasperm sasrun
# exit
```

IOM Bridge Servers

Starting a Spawner on Alpha/VMS

If the spawner is to service more than one client user ID, the spawner should run under an account that has the following privileges:

```
IMPERSONATE  NETMBX READALL TMPMBX
```

These privileges are required in order for the spawner to create a detached process with the connecting client as the owner.

If the spawner is to service one client, the spawner may be launched under that client's user ID.

Note: If you are printing or using SAS/GRAPH procedures, you must set the display to a machine running an X server. For example:

```
set display/create/transport=tcpip/node=
   <ip address of machine running X server>
```

Included as part of the Base SAS installation are some sample DCL files that demonstrate how to start the daemon as a detached process. The files listed here are all located in SAS$ROOT:[MISC.BASE]. Make a backup copy of these files before making any modifications.

OBJSPAWN_STARTUP.COM
: executes OBJSPAWN.COM as a detached process.

OBJSPAWN.COM
: runs the spawner. OBJSPAWN.COM also includes other commands that your site may need in order to run the appropriate version of the spawner, to set the display node, to define a process level logical pointing to a template DCL file (OBJSPAWN_TEMPLATE.COM), and perform any other actions needed before the spawner is started.

OBJSPAWN_TEMPLATE.COM
: performs setup that is needed in order for the client process to execute. The spawner first checks to see if the logical SAS$OBJSPAWN_TEMPLATE is defined. If SAS$TKELS_TEMPLATE is defined, when the server first starts the corresponding template file is executed as a DCL command procedure. You are not required to define the template file.

OBJSPAWN.CFG
: provides a sample configuration file for the spawner.

Note: After the spawner is started, an attempt is made to write a message to stdout indicating whether objspawn initialization completed or failed.

IOM Bridge Servers

Spawner Invocation Options

The following options can be used in the command to start up the spawner for an IOM Bridge server. Note that the spawner must be stopped and restarted in order to reflect configuration updates.

−*allowxcmd*
: Enables host commands and PIPE commands for all servers that are started by the spawner. By default, the spawner starts all servers with the −NOXCMD SAS system option. When you specify `−allowxcmd`, the spawner no longer specifies −NOXCMD when launching server sessions.

  Caution: When you specify `−allowxcmd`, clients can use host commands to perform potentially harmful operations such as file deletion.

−*authproviderdomain*
: Associates a default domain with the host. For example,

```
authproviderdomain (hostuser:Raleigh)
```

  The `−authproviderdomain` option has the following syntax:

```
authproviderdomain  (hostuser:<domain>)
```

  For more details about the `−authproviderdomain` option, see AUTHPROVIDERDOMAIN Option in the *SAS Language Reference: Dictionary*.

  This option can be abbreviated as `−authpd`.

−*configFile*
: Specifies a fully qualified path to the file that contains spawner configuration information. Enclose paths with embedded blanks in quotation marks. On Windows, enclose paths with embedded blanks in double quotation marks. On z/OS, specify filenames similarly to //dsn:myid.objspawn.log for MVS and //hfs:filename.ext for OpenEdition files.

  This option may be abbreviated as `−cf`.

−*deinstall*
: Windows only. Instructs the spawner to deinstall as a Windows Service. This option may be abbreviated as `−di`.

  Note: If you specified a service name when you installed the spawner service, you must specify the same name when you deinstall the service.

−*install*
: Windows only. Instructs the spawner to install as a Windows service. This option may be abbreviated as `−i`. When asked to install as a service, the spawner records all options specified at install time in the registry under the following key:

```
"SYSTEM\CurrentControlSet\Services\service-name\Parameters"
```

  You may also specify options in the Startup Parameters when you manually start the spawner service from the Services dialog box.

−*installDependencies*

Windows only. Specifies the Windows services that must be started before the spawner service starts. The `-installdependencies` option has the following syntax:

−INSTALLDEPENDENCIES "*service1*<;*service2*><;*service3*>"

This option may be abbreviated as `-idep`.

−*ldapBase*

Specifies the Distinguished Name (DN) in which to base all subsequent LDAP searches. This option may be abbreviated as `-lb`.

−*ldap_binddn*

Specifies the Distinguished Name (DN) in which to bind to the LDAP server. This, along with ldap_bindpw, might be needed when your network directory is secured. This option may be abbreviated as `-lbd`.

−*ldapHost*

Specifies the DNS Name of the machine that is hosting the network directory that contains the spawner's configuration information. This option may be abbreviated as `-lh`.

−*ldapPort*

Specifies the port on the ldapHost on which to connect. The default is 389. This option may be abbreviated as `-lp`.

−*ldap_bindpw*

Specifies the password that is associated with the ldap_binddn or ldapURL given. This, along with the ldap_binddn or bind information in the ldapURL, might be needed when your network directory is secured. This option may be abbreviated as `-lpw`.

−*ldapURL*

Specifies the LDAP URL that contains the host, port, base, bind, and, optionally, password information in which to utilize in order to obtain the spawner configuration information. This option may be abbreviated as `-lu`.

−*name*

Windows only. Specifies a service name to use when installing the spawner as a service. The default value is `SAS Object Spawner Daemon II`.

Note: If you install more than one spawner as a service on the same machine, you must use the −name option to give each spawner service a unique name.

−*noSecurity*

Instructs the spawner not to authenticate clients. Clients will execute as the user that launched the spawner. This option is useful during development.

WARNING: Because clients connected to the −noSecurity spawner execute as the user that launched the spawner, it is strongly suggested that the host in which the spawner is executing not be connected to a network. Otherwise, data that is accessible by the user that launched the spawner is at risk.

Note: If you use the −noSecurity option, the −install option is ignored.

−*sasLogFile*

Specifies a fully qualified path to the file in which to log spawner activity. Enclose paths with embedded blanks in quotation marks. On z/OS, specify filenames similar to UNIX file paths due to the requirement for z/OS Unix System Services. This option may be abbreviated as `-slf`.

Note: If you specify a log destination in the configuration metadata rather than the startup command, you might miss some messages that are generated before the log destination is set.

−*sasSpawnerCn*

Specifies the name (used in the SAS Management Console configuration) of the spawner object to utilize for this spawner invocation configuration. If you do not specify −sasSpawnerCn, the object spawner uses the first

spawner definition (on the LDAP server) with the same machine name as the current host.

Note: If none of the spawner definitions contain a host name of the current host, you must specify the `-sasSpawnercn` option to designate which spawner definition to use.

If you specify a spawner name that contains embedded blanks, you must specify the name in quotes (" "). This option can be abbreviated as `-ssc`.

-sasVerbose

When present, this option causes the spawner to record more detail in the log file (sasLogFile). This option may be abbreviated as `-sv`.

-servPass

Windows only. Specifies a password for the username specified in the `-servUser` option. This option can be abbreviated as `-sp`.

-servUser

Windows only. Specifies a username that the service will run under, when you also specify the `-install` option. This option can be abbreviated as `-su`.

-xmlConfigFile

Specifies a fully qualified path to a system configuration file containing an LDAP server definition to connect to for the complete configuration. On Windows, enclose paths with embedded blanks in double quotation marks. On z/OS, specify filenames similar to UNIX file paths due to the requirement for z/OS Unix System Services. This option may be abbreviated as `-xcf`.

To create a system configuration file for the LDAP server, see Using the IT Configuration Application.

IOM Bridge Servers

Using Telnet to Administer the Spawner

The spawner may be controlled and monitored using a telnet client connected to the operator port or service.

Connecting to a Spawner

To connect to an executing spawner, telnet to the operator interface port/service that is specified in the spawner definition.

The following example, run on UNIX, assumes 6337 was specified as the port for the operator:

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

After the telnet conversation is active, enter the operator password that is specified. If the operator password was not specified, use `sasobjspawn` as the password.

Note: You will not be prompted for the password. For example:

```
sasobjspawn
Operator conversation established
```

You may now interact with the executing spawner by issuing any of the Available Commands.

Available Commands

The following is a list of commands that are available via the spawner's operator interface:

btrace *filename*	Begin trace. filename is a fully qualified path to the file in which to log spawner activity.
bye	Terminate the spawner execution.
etrace	End trace.
help	List available operator commands.
list	List all known servers that are supported by this spawner.
quit	Exit operator conversation.

IOM Bridge Servers

Using the SAS Integration Technologies Configuration Utility (ITConfig)

The SAS Integration Technologies configuration utility (ITConfig) lets you generate metadata configuration files and test Integrated Object Model (IOM) connections between client machines and SAS. Using ITConfig, you can

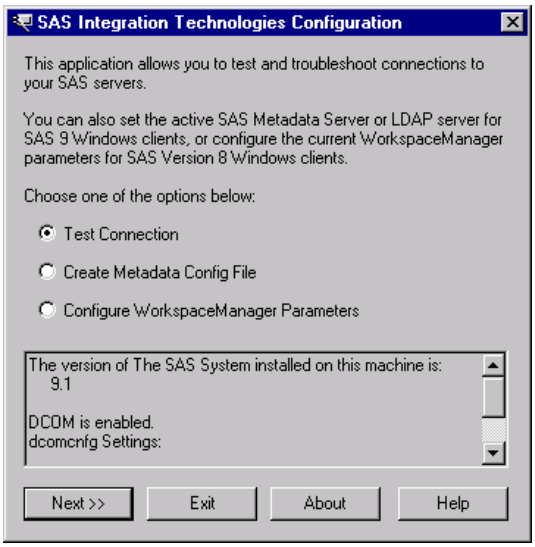
- create metadata configuration files that can be used to access an LDAP server or SAS Metadata Server.
- test and diagnose IOM connections to SAS servers. The application can test COM, DCOM, and IOM Bridge connection types.
- set the registry parameters that are used by the workspace manager on an LDAP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

Starting the Application

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

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

The Integration Technologies Configuration window appears.



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

- create metadata configuration files (Create Metadata Config File)
- view and change the LDAP parameters for the Workspace Manager (Configure WorkspaceManager Parameters)
- test the connection to a server (Test Connection)

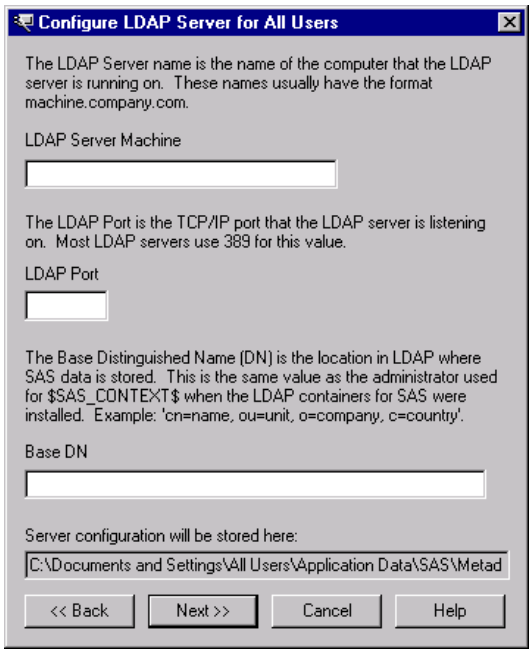
IOM Bridge Servers

Using ITConfig to Create Metadata Configuration Files

To access definitions on a metadata server, you must first connect to the metadata server. For connections to the LDAP server, the Object Manager and SAS can use metadata configuration files that contain information about how to connect to the server.

To create the metadata configuration files

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



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

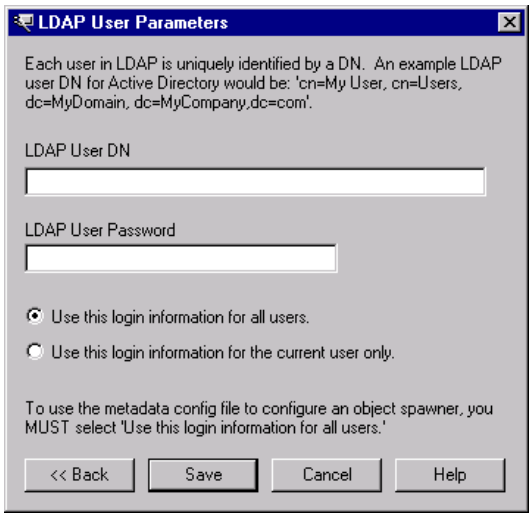
   Enter the following system configuration information:

   LDAP Server Machine
   : The fully−qualified name of the machine that the LDAP Server runs on.

   LDAP Server Port
   : The port used by the LDAP Server machine for receiving requests. A typical value is 389.

   Base DN
   : The distinguished name for the location in the LDAP hierarchy under which SAS directory entries are stored. The value for this field is the same as the value for the $SAS_CONTEXT$ parameter that was specified when the SAS containers were installed in the LDAP directory.

   Select **Next**. The LDAP User Parameters windows appears.



5. Enter the following information:

   LDAP User DN
   : The distinguished name of a user who will be accessing the LDAP server. Because the parameter information is stored in the client machine's registry, specify the DN of the client machine's user.

   LDAP User Password
   : The password required for the specified user to log onto the LDAP server.

6. If you selected **All users of this machine** for the configuration type, select one of the following:

   Use this login information for all users
   : specifies that the server and login information are stored in a single system configuration file that is common to all users.
   Note: You must select this option if you plan to use your configuration file with the object spawner.

   Use this login information for the current user only
   : specifies that the server information is stored in a system configuration file that is common to all users and that the login information is stored in a user configuration file that is specific to the current user.

   If you selected **Current user** for the configuration type, the server and login information are stored in a single system configuration file that is specific to the current user.
7. Select **Next**. ITConfig creates the configuration file(s) and the XML File Written dialog box appears.
8. To return to the main ITConfig screen, select **OK**.

Names and Locations for Configuration Files

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

Default Paths for Windows NT:

Common system configuration file
: \WINNT\Profiles\All Users\Application Data\SAS\MetadataServer\oms_serverinfo.xml

User−specific system configuration file
: \WINNT\Profiles*username*\Application Data\SAS\MetadataServer\oms_serverinfo.xml

User configuration file
: \WINNT\Profiles*username*\Application Data\SAS\MetadataServer\oms_userinfo.xml

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

Common system configuration file
\Documents and Settings\All Users\Application Data\SAS\MetadataServer\oms_serverinfo.xml
User−specific system configuration file
\Documents and Settings*username*\Application Data\SAS\MetadataServer\oms_serverinfo.xml
User configuration file
\Documents and Settings*username*\Application Data\SAS\MetadataServer\oms_userinfo.xml

Note: The location(s) and filename(s) are displayed in the Configure LDAP Server window and in the XML File Written dialog box.

Sample System Configuration File Format for an LDAP Server

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

```
<?xml version="1.0" encoding="UTF−8" ?>
<Redirect>
   <LogicalServer Name="LDAP Server"
    ClassIdentifier="440196D4−90F0−11D0−9F41−00A024BB830C">
      <UsingComponents>
         <ServerComponent Name="LDAP Server" ProductName="LDAP">
            <SourceConnections>
               <TCPIPConnection Name="LDAP Server" Port="389"
                HostName="dtd.pc.sas.com" ApplicationProtocol="LDAP">
                  <Domain>
                     <AuthenticationDomain Name="domainName">
                        <Logins>
                           <Login Name="test" UserID="cn=Mister
                            LDAP,cn=Users,dc=dtd−dom,dc=sas,dc=com"
                            Password="{base64}cGFzc3dvcmQ=" />
                        </Logins>
                     </AuthenticationDomain>
                  </Domain>
               </TCPIPConnection>
            </SourceConnections>
            <Properties>
               <Property Name="basedn"
                DefaultValue="cn=SAS,cn=Applications,dc=dtd−dom,dc=sas,dc=com"
                PropertyName="BaseDN">
               </Property>
            </Properties>
         </ServerComponent>
      </UsingComponents>
   </LogicalServer>
</Redirect>
```

Sample User Configuration File Format for an LDAP server

Use a text editor to edit your metadata configuration files. The following XML code shows a sample user configuration file for a connection to an LDAP Server.

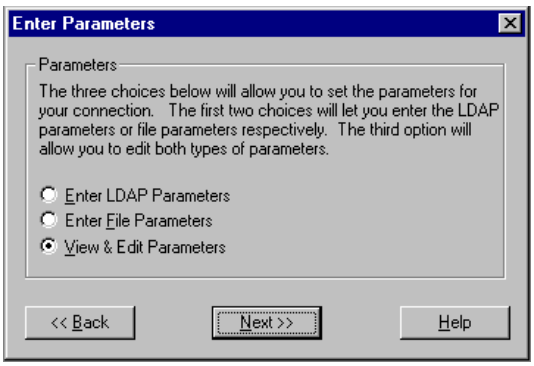
```
<?xml version="1.0" encoding="UTF−8" ?>
<AuthenticationDomain Name="domainName">
   <Logins>
```

```
     <Login Name="domainName\abc" UserID="domainName\abc1"
      Password="{base64}cGFzc3dvcmQ="/>
   </Logins>
</AuthenticationDomain>
```

IOM Bridge Servers

Using ITConfig to Configure Workspace Manager Parameters

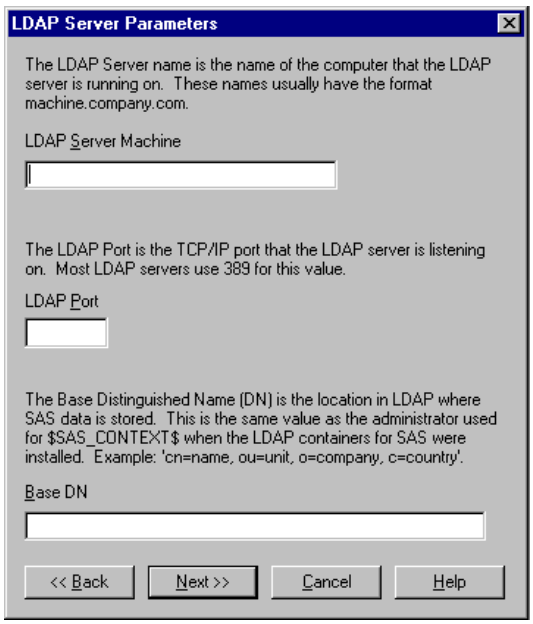
To view and edit connection parameters, select **Configure WorkspaceManager Parameters** from the Configuration window. The Enter Parameters window appears.



Specify whether you want to enter the LDAP connection parameters, enter the name of an LDIF file containing connection parameters, or view and edit all currently defined connection parameters.

Entering LDAP Parameters and Testing Connections

1. To enter connection parameters in the registry for a connection to an LDAP server, select **Enter LDAP Parameters** from the Enter Parameters window and click **Next**. The LDAP Server Parameters window appears.



2. Enter the following information:

LDAP Server Machine

Specifies the fully−qualified name of the machine on which the LDAP server runs.

LDAP Port

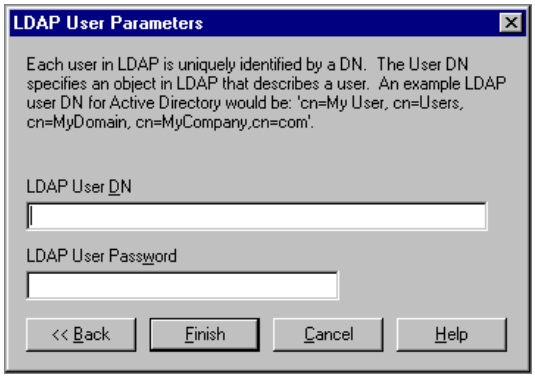
Specifies the port used by the LDAP server machine for receiving requests. A typical value is 389.

Base DN

Specifies the distinguished name for the location in the LDAP hierarchy under which SAS directory entries are stored. The value for this field is the same as the value for the $SAS_CONTEXT$ parameter that was specified when the SAS containers were installed in the LDAP directory.

Note: If you enter a Base DN that does not include cn=sas, then cn=sas is added to the Base DN that you entered.

Click **Next**. The LDAP User Parameters window appears.



3. Enter the following information:

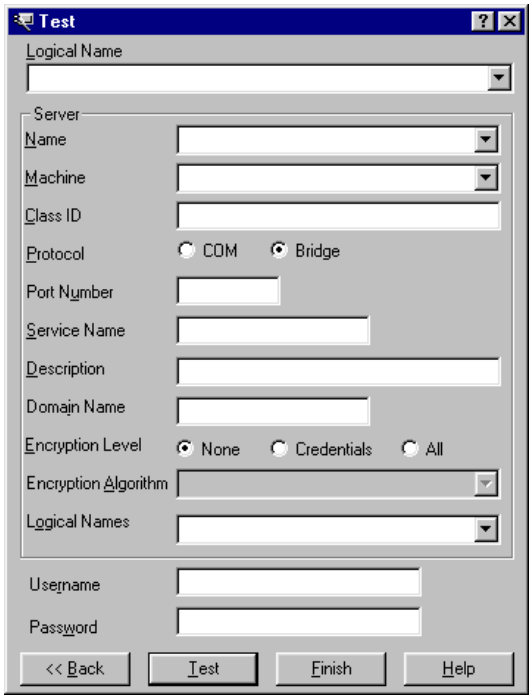
LDAP User DN

Specifies the distinguished name of a user who will be accessing the LDAP server. Because the parameter information is stored in the client machine's registry, specify the DN of the client machine's user.

LDAP User Password

Specifies the password required for the specified user to log onto the LDAP server.

4. Click **Next**. The application writes the data to the registry and the Test window appears.

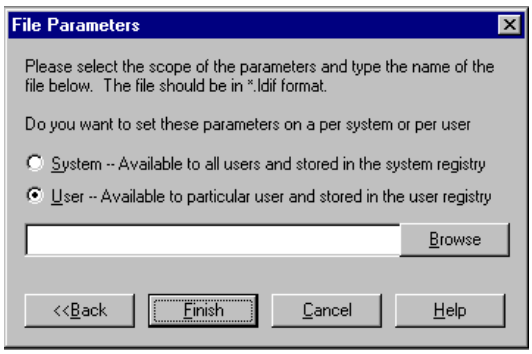


5. To test a connection to a server defined on the LDAP server, select the **Logical Name** of the machine for which you want to test a connection.
6. Click **Test** to test the connection. If the program establishes a DCOM connection to the specified server, the Connection Successful window appears.
7. To return to the main ITConfig screen, click **Finish**.

Entering File Parameters

1. To specify an LDIF file to use for connection parameters, select **Enter File Parameters** from the Enter Parameters window and click **Next**.

   The File Parameters window appears.

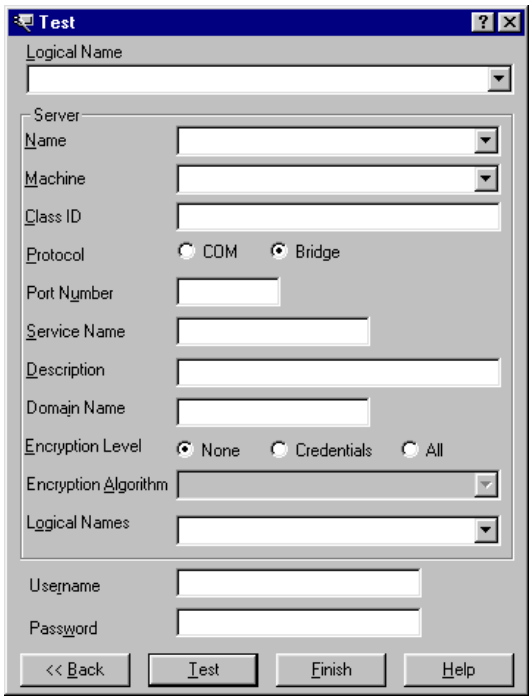


2. Specify whether the parameters apply to all users of this machine or only to a specific user.

   System
   Specifies the parameters contained in the LDIF file apply to all users of the client machine. The parameters will be stored in the system registry.

   User
   Specifies the parameters contained in the LDIF file apply only to a specific user. The parameters will be stored in the registry for the specified user.

3. Click **Next**. The application writes the data to the registry and the Test window appears.



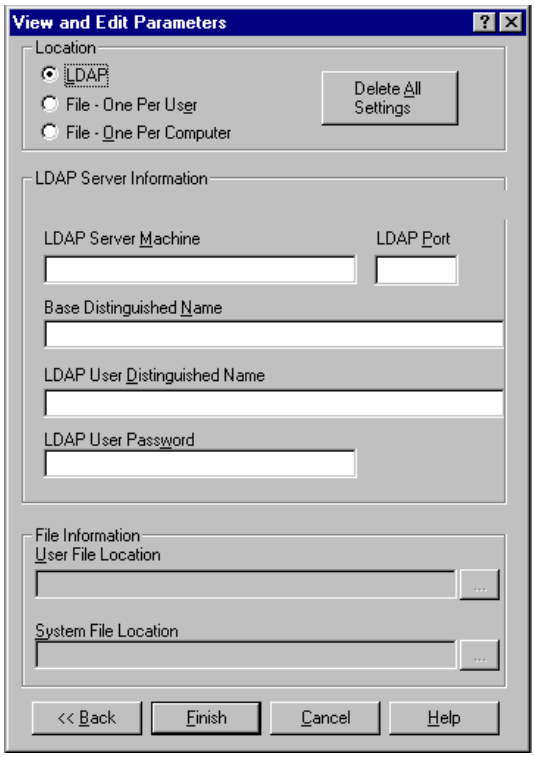
4. If you want to test a connection to a server defined on the LDAP server, select the **Logical Name** of the server connection you wish to test. Click **Test** to test the connection.

   If the program establishes a connection to the specified server, the Connection Successful window appears.
5. To return to the main ITConfig screen, click **Finish**.

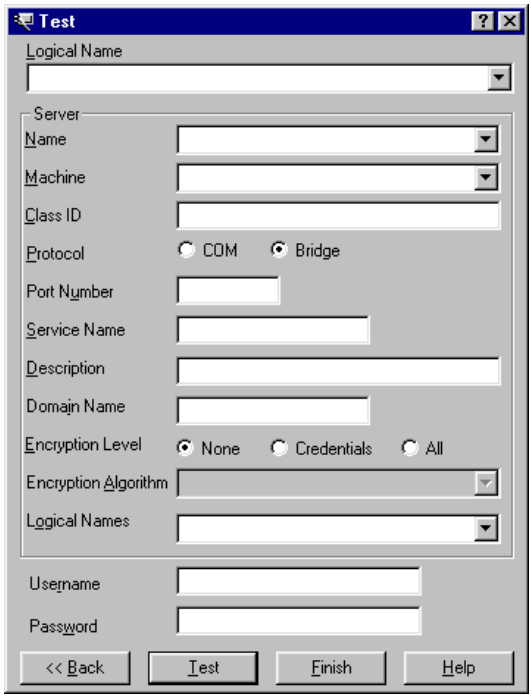
Viewing and Editing All Parameters

1. To work with all of the parameter information, whether from LDAP or a file, select **View and Edit Parameters** from the Enter Parameters window.

   The View and Edit Parameters window appears.



2. This window lets you view and specify the same information as on the LDAP and file parameters windows.
3. Select **LDAP** to view and edit the LDAP parameter information.
4. Select **File − One Per User** to view and edit the LDIF file used to define access for a particular user. The **User File Location** field is then enabled.
5. Select **File − One Per Computer** to view and edit the LDIF file used to define access for all users on the current machine. The **System File Location** field is then enabled.
6. Select **Delete All Settings** to clear all configuration information for all locations.
7. Click **Next**. The Test window appears.



8. To test a connection to a server defined on the LDAP server, select the **Logical Name** of the server connection you wish to test.

   Click **Test** to test the connection. If the program establishes a connection to the specified server, the Connection Successful window appears.
9. To return to the main ITConfig screen, click **Finish**.

IOM Bridge Servers

Using ITConfig to Test Connections

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

The test program used by ITConfig is a small SAS program that verifies the following information about the server environment:

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

Testing a Connection that is Defined on a Metadata Server

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

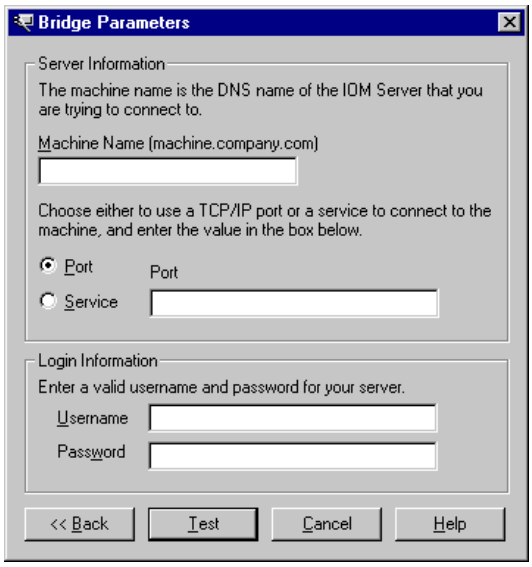
1. Select **Test Connection** from the main Integration Technologies Configuration window and click **Next.** The Choose How to Test window appears.
2. Select **Retrieve logical server definitions from the currently configured metadata server**, then click **Next**. The Test window appears.
3. Select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

7. Click ***Test*** to test the connection again, or click ***Cancel*** to return to the main Integration Technologies Configuration window.

Testing a Manually Defined IOM Bridge Connection

To test an IOM Bridge connection, follow these steps:

1. Select **Test Connection** from the main Integration Technologies Configuration window, then click **Next**. The Choose How to Test window appears.
2. Select **Enter a SAS server definition manually** and click **Next**. The Run Tests window appears.
3. Select the type of server to test and select **Bridge**, then click **Next**. The Bridge Parameters window appears



4. Enter the fully−qualified machine name in the Machine Name field. Examples of fully−qualified names are

   - machine1.alphaliteair.com
   - server.us.alphaliteair.com

5. Select either **Port** or **Service** to specify the method used to connect to the server.
6. Enter either the port number or the service name in the **Port** or **Service Name** field. The title of the field changes depending on whether you selected Port or Service as the connection method.
7. Enter a valid user name and password in the **Username** and **Password** fields.
8. Click **Test** to submit the test program through the connection. If the program establishes an IOM Bridge connection to the specified server, the Connection Successful window appears.
9. Click **Copy Text** to copy the test results to the clipboard. Click **OK** to return to the Bridge Parameters window.

10. Click ***Test*** to test the connection again, or click ***Cancel*** to return to the main Integration Technologies Configuration window.

IOM Bridge Servers

Spawner Error Messages

Here are error messages that might be reported by objspawn and explanations to correct their cause.

If you are still unable to correct the error, you might want the spawner to begin tracing its activity. See the administrator command section or use the `-slf` option to specify a log file when launching the spawner. For details, see Invoking (Starting) the Spawner.

Note: If an error occurs when the −slf option is not in effect, the spawner sends error messages to the SAS Console Log. This is a host−specific output destination. For details about the SAS Console Log, see the SAS Companion for your operating environment.

[Service Name] is already installed as a service. Deinstall the service, then reissue the install request

Host: *Windows*

Explanation:

An attempt was made to install the spawner as a Windows service when it was already installed.

Resolution:

If attempting to install with a different configuration, deinstall the spawner then reissue your install command.

A client that does not support redirection has connected to a server that requires redirection. The client connection will be closed.

Host: *All*

Explanation:

A down level IOM Bridge for Java client is attempting to connect to a server that has been defined within a load balancing cluster.

Resolution:

Upgrade the client's IOM Bridge for Java support.

A duplicate configuration option [duplicated option] was found.

Host: *All*

Explanation:

The displayed option was specified more than once.

Resolution:

Remove the redundant option and reissue your command.

A true socket handle cannot be obtained.

Host: *All*

Explanation:

The spawner was unable to retrieve the TCP/IP stack socket identifier from the runtime.

Resolution:

Contact SAS Technical Support.

A valid sasSpawner definition cannot be found.

Host: *All*

Explanation:

The spawner failed to find the named spawner definition. Or, if no name was given, a spawner definition that referenced the host in which the spawner is executing.

Resolution:

If a spawner name was specified at invocation, ensure the name is correct. Otherwise, correct the configuration source to define a valid spawner containing the correct host name.

Also known as:

Host: *All*

Explanation:

The host in which objspawn is executing is also known under the aliases listed.

Resolution:

N/A

An accepted client connection cannot be registered.

Host: *All*

Explanation:

The spawner was unable to place the socket associated with a connected client in a select.

Resolution:

Review the spawner log file to determine if the spawner is resource constrained. Ensure the client is still connected.

An NLS pipeline ([encoding identifier] −to− [encoding identifier]) cannot be created.

Host: *All*

Explanation:

The spawner was unable to initialize an internal transcoding object.

Resolution:

Ensure the SAS installation is complete/correct.

An unknown option ([option]) was specified .

Host: *All*

Explanation:

The spawner encountered an invocation option that is invalid.

Resolution:

Remove the invalid option and reissue the spawner command.

An unsupported UUID request version ([invalid version]) was received.

Host: *All*

Explanation:

A connection to the UUID listen port/service specified an invalid UUID protocol version.

Resolution:

Ensure that the IOM server clients are not connecting to the wrong port/service.

Cannot install objspawn with the NOSECURITY option.

Host: *Windows*

Explanation:

Due to the security exposure associated with the `nosecurity` option, the spawner will not install as a Windows service when `nosecurity` is specified.

Resolution:

Remove the `nosecurity` option and reissue the install command.

Communication cannot be established with the launched session.

Host: *All*

Explanation:

The spawner was unable to forward client information to the IOM server launched on behalf of the client.

Resolution:

Contact SAS Technical Support.

Deinstall objspawn then reissue the install request.

Host: *Windows*

Explanation:

The spawner is already installed as a service.

Resolution:

Deinstall the spawner then reissue your install command.

Ignoring attribute [attribute name].

Host: *All*

Explanation:

The named attribute is not applicable to the spawner.

Resolution:

N/A

Line [line−number] in the configuration file is not in LDIF format and is being skipped.

Host: *All*

Explanation:

A line in your configuration file is not in LDIF format and cannot be used.

Resolution:

Ensure that each line in your configuration file follows the syntax rules for LDIF format.

Load Balancing did not authorize server [server] to start and is disregarding the AddServer request.

Host: *All*

Explanation:

The spawner is using Load Balancing and started a server without Load Balancing instructing it to do so. This request is thrown out and the spawner should continue to function.

Resolution:

Review the configuration via SAS Management Console to ensure that all servers are set up correctly.

No configuration was specified.

Host: *All*

Explanation:

The spawner was invoked without a configuration source.

Resolution:

Reissue spawner command with a configuration source.

Objspawn cannot be deinstalled.

Host: *Windows*

Explanation:

The spawner was unable to deinstall as a Windows service.

Resolution:

Review the spawner log file to determine the cause of failure. Ensure the spawner is currently installed as a Windows service.

Objspawn cannot be installed.

Host: *Windows*

Explanation:

The spawner was unable to install as a Windows service.

Resolution:

Review the spawner log file to determine the cause of failure. Ensure the spawner is not currently installed as a Windows service.

Objspawn encountered [number of errors] error(s) during command−line processing.

Host: *All*

Explanation:

The spawner was unable to start.

Resolution:

Reissue the spawner invocation command with a valid log file destination. Review the contents of the generated log file to determine why the spawner failed to start.

Objspawn encountered errors during results processing.

Host: *All*

Explanation:

The spawner was unable to complete configuration processing.

Resolution:

Review the spawner log file to determine the configuration error details.

Objspawn encountered errors while attempting to start. To view the errors, define the DD name TKMVSJNL and restart objspawn with the sasVerbose option.

Host: *z/OS*

Explanation:

The spawner encountered errors and was unable to start.

Resolution:

Define the DD name TKMVSJNL and restart objspawn with the sasVerbose option to create a log file. Review the contents of the log file to determine why the spawner failed to start.

Objspawn encountered errors while attempting to start. View the application event log for the name of the log file containing the errors.

Host: *Windows*

Explanation:

The spawner encountered errors and was unable to start.

Resolution:

View the application event log to determine the name of the log file. Review the contents of the log file to determine why the spawner failed to start.

Objspawn failed to reinitiate the multi−user server listen. Objspawn is removing the server definition.

Host: *All*

Explanation:

The spawner was unable to restart a multi−user server listen when the previously launched multi−user server exited.

Resolution:

Ensure that there is not a port/service conflict.

Objspawn has completed initialization.

Host: *All*

Explanation:

The spawner is operational.

Resolution:

N/A

Objspawn has detected a bridge protocol over the operator conversation socket. Objspawn is closing the operator conversation with the peer (%s).

Host: *All*

Explanation:

An IOM Bridge client has connected to the operator listen port/service instead of a port/service belonging to a server definition.

Resolution:

Update the client to connect to the proper server definition port/service.

Objspawn is being terminated by the operating system.

Host: *z/OS*

Explanation:

The operator or operating system has requested that the spawner exit. The spawner will exit after this message is displayed.

Resolution:

N/A

Objspawn is executing on host [fully qualified host name] ([string IP address for fully qualified host name]).

Host: *All*

Explanation:

The host in which the spawner is executing returned the displayed fully qualified host name that resolved to the displayed IP address. These two strings plus the string "localhost", and any names or IP addresses listed after the alias message, are used by the spawner to locate the appropriate spawner and server definitions.

Resolution:

If the spawner fails to locate a spawner or server definition, ensure the spawner and/or server definitions specify one of the listed name or IP addresses.

Objspawn is exiting as a result of errors.

Host: *All*

Explanation:

The spawner was unable to start.

Resolution:

Reissue the spawner invocation command with a valid log file destination. Review the contents of the generated log file to determine why the spawner failed to start.

Objspawn lost connection with the launched session.

Host: *All*

Explanation:

The spawner was unable to complete startup of the launched IOM server.

Resolution:

If the message is identified as an error, contact SAS Technical Support.

Objspawn may not have been installed.

Host: *Windows*

Explanation:

The spawner was unable to deinstall as a Windows service. This might be due to the spawner not being installed as a Windows service.

Resolution:

Ensure that the spawner is installed as a Windows service.

Objspawn running as service [service name].

Host: *Windows*

Explanation:

Indicates which service the spawner is running as.

Objspawn service ([name of deinstalled spawner service]) was deinstalled successfully.

Host: *Windows*

Explanation:

The spawner is no longer installed as a Windows service.

Resolution:

N/A

Objspawn service ([name of installed spawner service]) was installed successfully.

Host: *Windows*

Explanation:

The spawner successfully installed as a Windows service. Subsequent boots of Windows will start the spawner automatically.

Resolution:

N/A

Objspawn version [major].[minor].[delta] is initializing.

Host: *All*

Explanation:

The version of the spawner being invoked.

Resolution:

N/A

Objspawn was unable to locate a server definition. Objspawn is exiting.

Host: *All*

Explanation:

The spawner was unable to find a server definition in the configuration source specified that was valid for this machine and the spawner definition's domain and logical name.

Resolution:

Ensure there is a valid server definition that meets the requirements stated. If you are using an LDIF configuration file and the configuration file contains a valid server definition, ensure that there are not two or more blank lines located before the server definition. In LDIF format, two contiguous blank lines signify the end of the definitions that will be used.

Objspawn was unable to open the configuration file ([file path]).

Host: *All*

Explanation:

The spawner was unable to open a configuration file at the specified location.

Resolution:

Ensure that the configuration file exists at the location specified. Ensure the configuration file is readable by the spawner.

Objspawn was unable to read data from the operator conversation socket. The returned error number is [errno] and the text associated with that error is ([errno description]).

Host: *All*

Explanation:

The spawner encountered a TCP/IP read error while attempting to converse with a connected operator.

Resolution:

Ensure the operator is still connected.

Objspawn was unable to send data over the operator conversation socket, The returned error number is [errno] and the text associated with that error is ([errno description]).

Host: *All*

Explanation:

The spawner encountered a TCP/IP write error while attempting to converse with the operator.

Resolution:

The operator might have terminated their connection.

Port [port number] will be ignored, and service [service name] will be used.

Host: *All*

Explanation:

The spawner encountered both port and service attributes in the current definition. The service definition takes precedence.

Resolution:

Remove the attribute that is redundant or incorrect.

The [attribute name] attribute requires an argument.

Host: *All*

Explanation:

An attribute present in the configuration requires a value.

Resolution:

Supply a value for the attribute and restart the spawner.

The [attribute name/description] attribute is either missing or is mismatched.

Host: *All*

Explanation:

The spawner encountered an attribute that did not have a required value.

Resolution:

Correct the configuration.

The [object class name] attribute [attribute name] is no longer supported.

Host: *All*

Explanation:

The spawner encountered an attribute within the specified configuration source that is no longer supported.

Resolution:

If the configuration source is not shared by earlier versions of the spawner, remove the named attribute from the configuration source.

The [option name] option requires an argument.

Host: *All*

Explanation:

The displayed option requires a value.

Resolution:

Reissue the spawner command and specify a value for the displayed option.

The [spawner utility name] service cannot be loaded.

Host: *All*

Explanation:

The spawner was unable to locate the specified support.

Resolution:

Ensure the SAS installation is complete/correct.

The [tracker name] resource tracker cannot be created.

Host: *All*

Explanation:

The spawner was unable to create an internal object repository.

Resolution:

Review the spawner log file to determine if the spawner is resource constrained.

The client ([Client child process identifier]) specified by the launched session could not be located.

Host: *All*

Explanation:

The spawner was unable to locate the connection information associated with the client definition in which an IOM server was launched.

Resolution:

Ensure that the command associated with the launched session is correct and that the IOM server is successfully launching.

The configuration source ([source]) conflicts with the previously specified configuration source ([source]).

Host: *All*

Explanation:

More than one configuration source was specified.

Resolution:

Determine which configuration source is correct and remove the others from your spawner invocation.

The connection to the LDAP server failed.

Host: *All*

Explanation:

The spawner was unable to contact the LDAP server specified.

Resolution:

Ensure that the LDAP server is running and that the proper credentials were specified.

The connection with the UUID generator session was lost.

Host: *All*

Explanation:

The spawner lost contact with the UUID generator client.

Resolution:

Ensure the client did not terminate.

The duplicate [attribute name] attribute will be ignored.

Host: *All*

Explanation:

The named attribute was encountered more than once.

Resolution:

N/A

The entry ([object class name]) is no longer supported and will be ignored.

Host: *All*

Explanation:

The spawner encountered an object class within the specified configuration source that is no longer supported.

Resolution:

If the configuration source is not shared by earlier versions of the spawner, remove the named object class definition from the configuration source.

The entry ([object class name]) was defined incorrectly and will be ignored.

Host: *All*

Explanation:

The spawner encountered an object class within the specified configuration source that is not defined correctly.

Resolution:

Review the spawner log file to determine which values in the object class definition are invalid, then correct the object class definition.

The exit handler cannot be installed.

Host: *z/OS*

Explanation:

The spawner was unable to install an exit handler.

Resolution:

Contact SAS Technical Support.

The IOM run−time subsystem cannot be initialized.

Host: *All*

Explanation:

The spawner was unable to locate the IOM server runtime.

Resolution:

Ensure the SAS installation is complete and correct.

The launched session did not accept forwarded requirements. The reply is [reply error number].

Host: *All*

Explanation:

The launched IOM server could not process the client requirements presented.

Resolution:

Ensure that the spawner and the server being launched are compatible releases.

The LDAP support extension cannot be loaded.

Host: *All*

Explanation:

The spawner was unable to locate the LDAP support.

Resolution:

Ensure the SAS installation is complete/correct.

The log file ([file path]) already exists. Please erase this file and restart.

Host: *All*

Explanation:

The spawner was unable to create a log file. A file which is not a spawner log file already exists at the named location.

Resolution:

Either delete the file at the named location or specify a different location for the spawner log file.

The log file ([file path]) cannot be created.

Host: *All*

Explanation:

The spawner was unable to create a log file at the given file path location.

Resolution:

Ensure the given file path is correct. Ensure that there is not a file at the specified location that is not a spawner log file.

The logged−in user does not have the appropriate user permissions to invoke [Windows service name].

Host: *Windows*

Explanation:

The spawner was not able to install or deinstall as a Windows service due to the launching user not having the appropriate Windows User Rights.

Resolution:

Ensure that the invoking user is an administrator on the Windows host and that the user holds the appropriate Windows User Rights.

The multi−user login ([login identifier]) that was specified for server ([server name]) cannot be found.

Host: *All*

Explanation:

The spawner was unable to locate the login definition associated with a multi−user server definition.

Resolution:

Correct the configuration source to properly define the missing login definition then reissue the spawner command.

The old client cannot be redirected as a result of IP address issues.

Host: *All*

Explanation:

The spawner cannot format the redirect IP address into a format suitable by a back level client.

Resolution:

Update the client IOM Bridge for COM or IOM Bridge for Java.

The operator communication buffer cannot be allocated.

Host: *All*

Explanation:

The spawner was unable to allocate a buffer in which to process operator conversations.

Resolution:

Review the spawner log file to determine if the spawner is resource constrained.

The operator conversation cannot be allocated.

Host: *All*

Explanation:

The spawner was unable to allocate a descriptor in which to process operator conversations.

Resolution:

Review the spawner log file to determine if the spawner is resource constrained.

The operator conversation was terminated by peer.

Host: *All*

Explanation:

The administration session was disconnected by the administrator.

Resolution:

N/A

The operator listen definition cannot be allocated.

Host: *All*

Explanation:

The spawner was unable to allocate a descriptor in which to process the operator listen definition.

Resolution:

Review the spawner log file to determine if the spawner is resource constrained.

The operator listen socket cannot be created. The returned error is [errno] and the text associated with that error is ([errno description]).

Host: *All*

Explanation:

The spawner was unable to create a TCP/IP socket for use as an operator listen socket.

Resolution:

Review the spawner log file to determine if the spawner is resource constrained.

The operator password specified by [string IP address] is invalid.

Host: *All*

Explanation:

The password received by a session originating from the displayed IP address was not correct.

Resolution:

Reissue operator session and specify the correct password.

The port or service for UUID generator TCP/IP definition is missing.

Host: *All*

Explanation:

The TCP/IP connection definition associated with the UUID generation did not contain a port or service definition.

Resolution:

Correct the TCP/IP connection definition.

The process cannot be launched for client [client username].

Host: *All*

Explanation:

The spawner was unable to launch an IOM server on behalf of the named client.

Resolution:

Ensure the command associated with the server definition is correct. Review the spawner log file to determine the cause of failure.

The process definition cannot be tracked for the server [server name].

Host: *All*

Explanation:

The spawner was unable to insert a server definition object into its repository.

Resolution:

Review the spawner log file to determine if the spawner is resource constrained.

The server [name of server] cannot be placed in a resource track.

Host: *All*

Explanation:

The spawner was unable to insert the internal server definition object in its repository.

Resolution:

Review the spawner log file to determine if the spawner is resource constrained.

The server [server name] listen cannot be registered.

Host: *All*

Explanation:

The spawner was unable to place the socket associated with a server listen in a select.

Resolution:

Review the spawner log file to determine if the spawner is resource constrained.

The server connection definition cannot be created.

Host: *All*

Explanation:

The spawner was unable to create an internal launched IOM server connection object.

Resolution:

Review the spawner log file to determine if the spawner is resource constrained.

The server definition cannot be allocated.

Host: *All*

Explanation:

The spawner was unable to allocate a server definition.

Resolution:

Review the spawner log file to determine if the spawner is resource constrained.

The server launch command cannot be allocated.

Host: *All*

Explanation:

The spawner was unable to allocate the server's launch command.

Resolution:

Review the spawner log file to determine if the spawner is resource constrained.

The server must define the available encryption algorithm(s) when an encryption level is set.

Host: *All*

Explanation:

The server definition specifies an encryption level, but does not specify which encryption algorithms are available.

Resolution:

Specify the available encryption algorithms.

The server name [server−name] is not unique. Therefore this server definition will not be included.

Host: *All*

Explanation:

The spawner was unable to process a server definition because another server definition has the same name.

Resolution:

Change the server name in the server definition.

The session socket for the UUID generator was not accepted.

Host: *All*

Explanation:

The spawner was unable to process a new UUID generator client.

Resolution:

Review the spawner log file to determine if the spawner is resource constrained.

The socket−access method handle cannot be acquired.

Host: *All*

Explanation:

The spawner was unable to locate the IOM protocol TCP/IP driver.

Resolution:

Ensure the SAS installation is complete and correct.

The specified [attribute name] value is invalid ([invalid attribute value]).

Host: *All*

Explanation:

The displayed value for the displayed attribute is not valid.

Resolution:

Correct the attribute value and reissue command.

The specified TCP/IP definition protocol is invalid.

Host: *All*

Explanation:

A TCP/IP connection definition specifies a protocol that is not supported by the spawner.

Resolution:

Correct the TCP/IP connection protocol attribute value.

The TCP/IP accept call failed to process the client connection.

Host: *All*

Explanation:

The spawner was unable to process a new client.

Resolution:

Review the spawner log file to determine if the spawner is resource constrained.

The TCP/IP accept call failed to process the operator connection. The returned error number is [errno], and the text associated with that error is ([errno description]).

Host: *All*

Explanation:

The spawner was unable to process a new operator.

Resolution:

Review the spawner log file to determine if the spawner is resource constrained.

The TCP/IP accept call failed to process the session conversation request.

Host: *All*

Explanation:

The spawner was unable to process a new IOM server connection.

Resolution:

Review the spawner log file to determine if the spawner is resource constrained.

The TCP/IP bind call for the server [server name] listen port failed. The text associated with that error is ([reason of failure]).

Host: *All*

Explanation:

The spawner was unable to establish the named server listen.

Resolution:

Ensure the port/service specified for use by the spawner is defined and not in use. If in use, ensure the spawner is not already executing.

The TCP/IP bind call for the session conversation port failed. The returned error number is [errno], and the text associated with that number is ([errno description]).

Host: *All*

Explanation:

The spawner was unable to bind to any port in order to establish a listen for use by launched IOM servers.

Resolution:

Contact SAS Technical Support.

The TCP/IP bind call for the UUID listen port failed. The returned error number is [errno] and the text associated with that error is ([errno description]).

Host: *All*

Explanation:

The spawner was unable to establish the UUID generator listen.

Resolution:

Ensure the port/service specified for use by the spawner is defined and not in use. If in use, ensure the spawner is not already executing.

The TCP/IP listen call for the operator listen port failed. The returned error number is [errno], and the text associated with that error is ([errno description]).

Host: *All*

Explanation:

The spawner was unable to establish the operator listen.

Resolution:

Ensure the port/service specified for use by the spawner is defined and not in use. If in use, ensure the spawner is not already executing.

The TCP/IP listen call for the server [server name] listen port failed. The text associated with that error is ([reason of failure]).

Host: *All*

Explanation:

The spawner was unable to establish the server listen.

Resolution:

Ensure the port/service specified for use by the spawner is defined and not in use. If in use, ensure the spawner is not already executing.

The TCP/IP listen call for the session conversation port failed. The returned error number is [errno] and the text associated with that error is ([errno description]).

Host: *All*

Explanation:

The spawner was unable to establish the launched IOM server listen.

Resolution:

Contact SAS Technical Support.

The TCP/IP listen call for the UUID listen port failed. The returned error number is [errno] and the text associated with that error is ([errno description]).

Host: *All*

Explanation:

The spawner was unable to establish the UUID generator listen.

Resolution:

Contact SAS Technical Support.

The URI support extension cannot be loaded.

Host: *All*

Explanation:

The spawner was unable to locate the URI parsing support.

Resolution:

Ensure the SAS installation is complete/correct.

The UUID listen definition cannot be created.

Host: *All*

Explanation:

The spawner was unable to create an internal UUID listen object.

Resolution:

Review the spawner log file to determine if the spawner is resource constrained.

The UUID service name ([service name]) cannot be resolved.

Host: *All*

Explanation:

The host TCP/IP stack was unable to resolve the displayed TCP/IP service name.

Resolution:

Ensure the given service name is correct and defined to the spawner host installation.

The wait event for the objspawn cannot be created.

Host: *All*

Explanation:

The spawner was unable to create an internal synchronization object.

Resolution:

Review the spawner log file to determine if the spawner is resource constrained.

The Windows [Windows routine name] call failed. ([reason for failure]).

Host: *All*

Explanation:

The spawner encountered an Windows SDK error while invoking the given method.

Resolution:

Contact the system administrator to determine the meaning of the error text.

The Windows [Windows routine name] call failed. GetLastError() = [GetLastError() return value].

Host: *All*

Explanation:

The spawner encountered an Windows SDK error while invoking the given method.

Resolution:

Contact the system administrator to determine the meaning of the GetLastError() return code.

Unable to create client definition.

Host: *All*

Explanation:

The spawner was unable to allocate and initialize a descriptor for the connected client.

Resolution:

Review the spawner log file to determine if the spawner is resource constrained.

Unable to create the session conversation definition.

Host: *All*

Explanation:

The spawner was unable to create an internal launched IOM server conversation object.

Resolution:

Review the spawner log file to determine if the spawner is resource constrained.

Unable to generate requested UUIDs.

Host: *All*

Explanation:

The spawner encountered an error while attempting to fulfill a UUID generator request.

Resolution:

Review the spawner log file to determine if the spawner is resource constrained.

Unable to obtain the session conversation port. The returned error number is [errno], and the text associated with that error is ([errno description]).

Host: *All*

Explanation:

The spawner was unable to retrieve the port associated with the session conversation listen.

Resolution:

Contact the system administrator to determine if there are issues with the TCP/IP implementation.

Unable to read the server ([server name]) client update information.

Host: *All*

Explanation:

The spawner encountered a TCP/IP read error while attempting to converse with a launched IOM server.

Resolution:

The server might have exited.

Unable to resolve "localhost".

Host: *All*

Explanation:

The spawner could not resolve the local IP address.

Resolution:

Ensure that your TCP/IP configuration settings are correct.

Unable to retrieve the [object class] definition or definitions.

Host: *All*

Explanation:

The spawner was unable to retrieve the named object class definitions from the LDAP server specified.

Resolution:

Ensure the spawner was directed to the proper LDAP server. Ensure the LDAP server metadata is complete.

You can only specify one of the following choices: install or deinstall.

Host: *Windows*

Explanation:

Both the `install` and `deinstall` commands were specified.

Resolution:

Remove the option that should not be specified.

You cannot specify the ldapurl when one or more of the following options have been specified: ldapbase, ldaphost, or ldapport.

Host: *All*

Explanation:

The `ldapurl` option overlaps in functionality with another configuration source option.

Resolution:

Determine which configuration source definition is correct and remove the other configuration source definitions from your spawner invocation.

IOM Bridge Servers

Configuration File Example: Minimal Configuration

The following LDIF file can be used as a minimal configuration file for your IOM Bridge server. It contains a definition for the spawner and the object server.

These definitions assume that the spawner and the object server are running on a UNIX machine. If your server is running on a different platform, then the value for the *SASCommand* attribute will need to be changed.

```
#
# Spawner Definition
#
dn: cn=Finance,o=AlphaliteAirways,c=US
objectClass: sasSpawner
sasSpawnercn: mySASObjectSpawner
sasMachineDNSName: localhost
sasOperatorPort: 5306
description: SAS Object Spawner

#
# Object Server Definition listening on port 5307;
#
dn: cn=Finance,o=AlphaliteAirways,c=US
objectClass: sasServer
sasServercn: mySASObjectServer
sasPort: 5307
sasMachineDNSName: localhost
sasProtocol: bridge
sasCommand: /sasv9/usrlibsas/sas
description: SAS Object Server
```

Note: Ensure that there is only one blank line, excluding comments (#), between the spawner and server definitions.

IOM Bridge Servers

Configuration File Examples: Server and Spawner

Here is a sample configuration file that defines a spawner named Simple, which services one object server named SimpleServer:

```
#
##  Define a Simple spawner to run on our
##  localhost machine.
#
dn: sasSpawnercn=Simple
objectClass: sasSpawner
sasMachineDNSName: localhost
sasSpawnercn: Simple
sasOperatorPort: 5306
description: Example spawner executing on our localhost machine

#
##  SimpleServer only handles clear text connections
#
dn: sasServercn=SimpleServer
objectClass: sasServer
sasServercn: SimpleServer
sasCommand: /sasv9/usrlibsas/sas
sasMachineDNSName: localhost
sasPort: 5307
sasProtocol: bridge
description: Example server that does not handle encrypted connections
```

Here is a sample configuration file that defines a spawner named Example that services two object servers; ExampleServer1 and ExampleServer2:

```
#
##  Define the Example spawner to run on our
##  demo machine.
#
dn: sasSpawnercn=Example
objectClass: sasSpawner
sasMachineDNSName: demo.unx.abc.com
sasSpawnercn: Example
sasOperatorPort: 5306
description: Example spawner executing on our demo machine

#
##  ExampleServer1 handles encrypted connections
#
dn: sasServercn=ExampleServer1
objectClass: sasServer
sasServercn: ExampleServer1
sasCommand: /sasv9/usrlibsas/sas
sasMachineDNSName: demo.unx.abc.com
sasNetEncrAlg: sasproprietary
sasNetEncrAlg: des
sasNetEncrAlg: tripledes
sasNetEncrAlg: rc2
sasNetEncrAlg: rc4
sasPort: 5307
sasProtocol: bridge
# Don't require encryption
```

```
sasRequiredEncryptionLevel: none
description: Example server that handles encrypted connections

#
##  ExampleServer2 only handles clear text connections
#
dn: sasServercn=ExampleServer2
objectClass: sasServer
sasServercn: ExampleServer2
sasCommand: /sasv9/usrlibsas/sas
sasMachineDNSName: demo.unx.abc.com
sasPort: 5308
sasProtocol: bridge
description: Example server that does not handle encrypted connections
```

The example above illustrates these points:

- Comments are ignored. Therefore, they do not affect processing that is associated with empty lines.
- The distinguished name attribute, preceded by a blank line, identifies the beginning of the next object definition.

IOM Bridge Servers

Configuration File Example: Using Logical Names

Logical names provide a mechanism to identify similar functionality. They are specified via the sasLogicalName attribute.

For instance, your installation may want to stage a new application without altering its production applications. To do this, the Object Spawner Daemon and SAS Object Server definitions specify the same logical name. Here is a configuration file that illustrates how this is accomplished:

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

#
##  Define our test MyApplication Object Spawner Daemon.
##  We also log activity into /tmp/myid.objspawn.log
#
dn: sasSpawnercn=test,sascomponent=sasServer,cn=SAS,o=ABC Inc,
c=US
objectClass: sasSpawner
sasSpawnercn: test
sasDomainName: mvs.abc.com
sasLogFile: /tmp/myid.objspawn.log
sasLogicalName: stage
sasMachineDNSName: bigiron.mvs.abc.com
sasOperatorPort: 6342
sasOperatorPassword: myPassword
description: Test MyApplication Object Spawner Daemon

#
##  Define our production MyApplication SAS Object Server
#
dn: sasServercn=MyApplication,sascomponent=sasServer,cn=SAS,o=ABC Inc,
c=US
objectClass: sasServer
sasServercn: MyApplication
sasDomainName: mvs.abc.com
sasMachineDNSName: bigiron.mvs.abc.com
sasPort: 6341
sasProtocol: bridge

#
##  Define our test MyApplication SAS Object Server
#
dn: sasServercn=testApplication,sascomponent=sasServer,cn=SAS,o=ABC Inc,
c=US
objectClass: sasServer
sasServercn: testApplication
```

```
sasDomainName: mvs.abc.com
sasLogicalName: stage
sasMachineDNSName: bigiron.mvs.abc.com
sasPort: 6343
sasProtocol: bridge
```

A spawner locates its object server definitions by using the following rules:

- The domain (sasDomainName) must match. The lack of a domain is considered a domain.
- The sasServer must have a protocol (sasProtocol) of bridge.
- The sasServer logical names must be a subset of the logical names that are specified in the sasSpawner definition.

IOM Bridge Servers

Configuration File Examples: UUID Generator

Here is an example UUIDGEND setup configuration file for a host other than Windows NT:

```
#
##  Define our UUID Generator Daemon.  Since this UUIDGEND is
##  executing on a non−Windows NT host, we contacted SAS Technical
##  Support for the sasUUIDNode specified.
#
dn: sasSpawnercn=UUIDGEND,sascomponent=sasServer,cn=SAS,o=ABC Inc,
c=US
objectClass: sasSpawner
sasSpawnercn: UUIDGEND
sasDomainName: unx.abc.com
sasMachineDNSName: medium.unx.abc.com
sasOperatorPassword: myPassword
sasOperatorPort: 6340
sasUUIDNode: 0123456789ab
sasUUIDPort: 6341
description: SAS Session UUID Generator Daemon on UNIX
```

Here is an example UUIDGEND setup configuration file for a Windows NT host:

```
#
##  Define our UUID Generator Daemon.  Since this UUIDGEND is
##  executing on a Windows NT host, we do not need to specify
##  the sasUUIDNode.
#
dn: sasSpawnercn=UUIDGEND,sascomponent=sasServer,cn=SAS,o=ABC Inc,
c=US
objectClass: sasSpawner
sasSpawnercn: UUIDGEND
sasDomainName: wnt.abc.com
sasMachineDNSName: little.wnt.abc.com
sasOperatorPassword: myPassword
sasOperatorPort: 6340
sasUUIDPort: 6341
description: SAS Session UUID Generator Daemon on NT
```

IOM Bridge Servers

Attributes for sasLogicalNameInfo

The sasLogicalNameInfo object class contains information for an instance of a SAS logical name. The sasLogicalNameInfo object class is defined using the attributes listed in the following table. For each attribute, the table shows:

- The name that identifies the attribute on the LDAP server (or in the configuration file).
- "Required" or "Optional" to indicate whether the attribute is required.
- The type of server configuration (COM/DCOM or IOM Bridge) for which the attribute is used.
- A definition of the attribute.

For general information about the use of logical names, refer to Assigning Logical Names. When you use IT Administrator to add a logical name to a server or spawner definition, IT Administrator automatically creates a sasLogicalName object.

If you are not using an LDAP server, you can use a configuration file to define the logical name. For instructions, see Using a Configuration File to Define the Metadata. The spawner does not use this object class. However, if your site uses logical names, it is recommended that sasLogicalNameInfo instance be created.

sasLogicalName Attribute Definitions			
Attribute Name	**Required/ Optional**	**Server Type**	**Definition**
description *In IT Administrator*:: **Description**	Optional	COM/DCOM, IOM Bridge	Text to summarize why this object definition exists. This attribute is not used by the spawner.
objectClass	Required	COM/DCOM, IOM Bridge	The object class identifier. For sasLogicalNameInfo objects, this is always `sasLogicalNameInfo`.
sasLogicalName	Required	COM/DCOM, IOM Bridge	The logical name that is being defined

IOM Bridge Servers

Attributes for sasLogin

A SAS login may need to be available in order to start a SAS session on a server or to connect to a client. Each SAS login definition contains a user name, password, and domain, as well as a pointer to the user's person reference entry in the LDAP directory.

SAS logins may be used to provide credentials when creating a client connection. Whether or not SAS logins are required depends on the method calls used to start the server or create the connection. For example, you might need logins for pooling. If you do not use logins, you must track and specify the user credentials manually.

The sasLogin object class is defined using the attributes listed in the following table. For each attribute, the table shows:

- The name that identifies the attribute on the LDAP server (or in the configuration file). Under each attribute name, the table shows the corresponding tab and field name in the IT Administrator application.
- "Required" or "Optional" to indicate whether the attribute is required.
- The type of server configuration for which the attribute is used. **Note:** SAS logins are used only in IOM Bridge configurations. Therefore, IOM Bridge is listed as the server type for each attribute.
- A definition of the attribute.

For step−by−step instructions on defining the metadata for a SAS login, refer to Using the IT Administrator to Define a SAS Login. If you are not using an LDAP server, you can use a configuration file to define a SAS login. For instructions, see Using a Configuration File to Define the Metadata.

sasLogin Attribute Definitions			
Attribute Name	**Required/ Optional**	**Server Type**	**Definition**
description *In IT Administrator:*: **Description**	Optional	IOM Bridge	Text to summarize why this object definition exists.
objectClass *In IT Administrator:* **N/A**	Required	IOM Bridge	The object class identifier. For sasLogin objects, this is always `sasLogin`. If you use IT Administrator, this identifier is assigned automatically.
sasAllowedClientDN *In IT Administrator:* **Logins ➧ Client DNs**	Optional	IOM Bridge	The distinguished names of the users or groups of users who are to be allowed access to a workspace that is created with this sasLogin.
sasDomainName *In IT Administrator:* **Logins ➧ Domain**	Optional	IOM Bridge	The security domain in which the sasLogin definition participates. The login definition must have the same domain name as the server on which SAS sessions will be established. The lack of a domain is considered a domain; therefore, if the login definition has no domain name, it will be associated only with servers that have no domain name.
sasLogicalName	Optional	IOM	The logical names associated with this sasLogin definition.

In IT Administrator: **Logical Names**		Bridge	For more information about logical names, refer to Assigning Logical Names.
sasLogincn *In IT Administrator:* **Name**	Required	IOM Bridge	The unique name for this sasLogin object.
sasLoginName *In IT Administrator:* **Logins ➧ SAS Login**	Required	IOM Bridge	The user name, or login ID, that the spawner is to use when launching the object server. The user name must be valid for the server on which SAS sessions will be established.
sasMinAvail *In IT Administrator:* **Logins ➧ Min Available Workspaces**	Optional	IOM Bridge	Specifies the minimum number of workspaces using this login definition that need to be available. This value includes only idle connections.
sasMinSize *In IT Administrator:* **Logins ➧ Min Workspace Size**	Optional	IOM Bridge	Specifies the minimum number of workspaces using this login definition that are created when the workspace pool is created. This value includes both connections that are in use and connections that are idle. The default value is 0.
sasReferenceDn *In IT Administrator:* **Person Reference**	Optional	IOM Bridge	The distinguished name of a person or group entry. This attribute is not used by the object spawner.
sasUserPassword *In IT Administrator:* **Logins ➧ Password**	Required	IOM Bridge	The user login password for starting a SAS session. The password must be valid for the server on which SAS sessions will be established.

IOM Bridge Servers

Attributes for sasServer

The sasServer object class contains startup and connection information for an instance of a SAS object server. The sasServer object class is defined using the attributes listed in the following table. For each attribute, the table shows:

- The name that identifies the attribute on the LDAP server (or in the configuration file). Under each attribute name, the table shows the corresponding tab and field name in the IT Administrator application.
- "Required" or "Optional" to indicate whether the attribute is required.
- The type of server configuration (COM/DCOM or IOM Bridge) for which the attribute is used.
- A definition of the attribute.

Note: The following attributes which appear in the IT Administrator interface are not used in Version 9 of SAS Integration Technologies

- sasTpName
- sasPluName
- sasProtocol=corba
- sasMultiUserObject

If you are using Version 9, do not use these attributes for your configuration.

Note: For the z/OS, you can now use the sasCommand attribute to launch SAS as an object server. Because the IT Admin interface has not changed for Version 9, you can specify the launch command in the **Command for non OS/390** IT Admin field.

For step−by−step instructions on defining the metadata for a server, refer to Using the IT Administrator Wizard to Define a Server and Spawner or Using IT Administrator to Define a Server. If you are not using an LDAP server, you can use a configuration file to define the server. For instructions, see Using a Configuration File to Define the Metadata (IOM Bridge) or Using a Configuration File to Define the Metadata (COM/DCOM).

sasServer Attribute Definitions			
Attribute Name	**Required/ Optional**	**Server Type**	**Definition**
description *In IT Administrator:*: **Description**	Optional	COM/DCOM, IOM Bridge	Text to summarize why this object definition exists.
objectClass *In IT Administrator:* **N/A**	Required	COM/DCOM, IOM Bridge	The object class identifier. For sasServer objects, this is always `sasServer`. If you use IT Administrator, this identifier is assigned automatically.
sasClientEncryptionAlgorithm *In IT Administrator:* **Encryption (IOM) ➔ Client Algorithm**	Optional	IOM Bridge	The encryption algorithm that is supported on the client side of the connection. Valid values are: RC2, RC4, DES, Triple DES, and SAS Proprietary, depending on the country in which the SAS software is licensed. See SAS/SECURE for more

			information regarding this attribute.
sasCommand *In IT Administrator:* **Commands → Command for non OS/390**	Required	IOM Bridge	The command used to launch SAS as an object server. With the command, specify the path relative to the directory in which the spawner will be started. If you are using a configuration file instead of LDAP, then paths with embedded blanks must be in quotation marks (or, for Windows platforms, double quotation marks). For more information about the server command, see Server Startup Command.
sasDomainName *In IT Administrator:* **Connections → Domain**	Optional	COM/DCOM, IOM Bridge	The security domain in which the sasServer definition participates. In IOM bridge servers configurations, the spawner definition must have the same domain name as the server definition. The spawner uses the domain name, along with the machine name and logical name, to determine which server(s) it services. The lack of a domain is considered a domain; therefore, if the server definition has no domain name, it will be associated only with spawners that have no domain name.
sasLogicalName *In IT Administrator:* **Logical Names**	Optional	COM/DCOM, IOM Bridge	The logical names associated with this sasServer definition. In IOM bridge servers configurations, the spawner uses logical names (along with machine names and domain names) to determine which server(s) it services. If logical names are specified, then only those sasSpawner instances that include *all of the logical names* that are defined here will support this sasServer. If you are using a configuration file instead of LDAP, specify each logical name as a separate attribute and value pair. For a general discussion of logical names, refer to Assigning Logical Names.
sasMachineDNSName *In IT Administrator:* **Machines**	Required	COM/DCOM, IOM Bridge	The DNS (domain name service) name(s) and IP address(es) for the machine(s) on which this server definition may execute. Multiple values can be assigned to this attribute. The machine name must be the official network name (for example,

			machine.corp.com). The string `localhost` can be used to signify the host on which the associated spawner is executing.
sasMaxPerWorkspacePool *In IT Administrator:* **Workspace Pool → Maximum Workspaces per Workspace Pool**	Optional	COM/DCOM, IOM Bridge	If you are using connection pooling, specifies the maximum number of workspaces that should be available for a workspace pool that is established on this server. A good starting place for this number is the number of CPUs that are available on the machine that is running SAS.
sasNetEncrAlg *In IT Administrator:* **Encryption (IOM) → Server Algorithms**	Optional	IOM Bridge	The encryption algorithms that are supported by the launched object server. Multiple values can be assigned to this attribute. Valid values are: RC2, RC4, DES, TRIPLEDES, and SASPROPRIETARY, depending on the country in which the SAS software is licensed. See SAS/SECURE for more information regarding this attribute.
sasObjectServerParms	Optional	COM/DCOM, IOM Bridge	The object server parameters that the spawner uses to launch SAS. This field allows you to override or add to the object server parameters. For a list of object server parameters, see Object Server Parameters.
sasPort *In IT Administrator:* **Connections → IOM Bridge → Port**	Required if server will have Java clients	IOM Bridge	The port on which to connect to this object server. If neither sasPort nor sasService is specified, the spawner will attempt to use the service name `sasobjspawn` as the sasService. If `sasobjspawn` has been used already, the spawner will remove this sasService definition from its list. The port number is required if the server will have Java clients.
sasProtocol *In IT Administrator:* **Connections → Protocol**	Required	COM/DCOM, IOM Bridge	The protocol (bridge, com) that clients may use for connection. The protocol `bridge` must be used for servers that are serviced by the spawner. These include all non−Windows servers, as well as Windows servers that will be accessed by Java clients.
sas−RecycleActivationLimit *In IT Administrator:* **Workspace Pool → Recycle**	Optional	COM/DCOM, IOM Bridge	If you are using connection pooling, the number of times a server is used before the process is disposed of and a new process is used in pooling. A value of 0 indicates that

Activation Limit			the process will have no limit.
sasRequiredEncryptionLevel *In IT Administrator:* **Encryption → Encrypt**	Optional	COM/DCOM, IOM Bridge	The level of encryption to be used between the client and the object server. **None** means no encryption is performed; **Credentials** means that only user credentials (id and password) are encrypted; and **Everything** means that all communications between the client and server are encrypted.
sasServercn *In IT Administrator:* **Name**	Required	COM/DCOM, IOM Bridge	The unique name for this sasServer object.
sas−ServerRunForever *In IT Administrator:* **Workspace Pool → Server Process Shutdown → Leave running when idle**	Optional	COM/DCOM, IOM Bridge	If you are using connection pooling, specifies that an idle server should always remain running. If the value of this attribute is true, the server always remains running. If the value is false, the idle server runs for the length of time specified in the sasServerShutdownAfter attribute.
sas−ServerShutdownAfter *In IT Administrator:*: **Workspace Pool → Server Process Shutdown → Minute until idle shutdown**	Optional	COM/DCOM, IOM Bridge	If you are using connection pooling, the number of minutes after which an idle server should be shut down. The value must be between 0 and 1440. The default value is 3. This attribute is ignored if the value of sasServerRunForever is true.
sasService *In IT Administrator:* **Connections → IOM Bridge → Service**	Optional	IOM Bridge	The service in which to connect to this object server. If you specify a value for both sasService and sasPort, then the value for sasService will be ignored. If neither sasPort nor sasService is specified, the spawner will attempt to use the service name `sasobjspawn` as the sasService. If `sasobjspawn` has been used already, the spawner will remove this sasService definition from its list. Note: If the server will have Java clients, specify a sasPort instead of a sasService.

IOM Bridge Servers

Object Server Parameters

The following table lists the object server parameters that you can use to override or add to the object server parameters that the spawner uses to launch SAS.

Object Server Parameters			
Object Server Parameter	**Value**	**Connection Type**	**Definition**
CLIENTENCRYPTIONLEVEL **Alias:** CEL	none credentials everything	IOM Bridge	Specifies the degree of encryption to use when making outbound calls. This option is used only by the bridge protocol engine.
JNLSTRMAX	Numeric value	IOM Bridge COM/DCOM	Specifies the maximum length of string printed in the IOM server journal. This option can be used to adjust the amount of material included in an IOM trace. A value greater than 500 is recommended.
LOGFILE **Alias:** LOG	Path in which to create the IOM server trace log.	IOM Bridge COM/DCOM	Provides an alternative to the SAS Log for IOM server trace output. **Note:** The user who starts the server must have execute and write permissions for the log destination path.
PORT	TCP/IP port number	IOM Bridge	Specifies the value for the bridge protocol engine to use as the port in which to start listening for client connections.
PROTOCOL	bridge com (com,bridge)	IOM Bridge COM/DCOM	Specifies the protocol engine(s) to launch in server mode. Server mode indicates that the protocol engine(s) will listen for client connections. By default, Windows servers use the COM protocol engine and all other servers use the Bridge protocol engine.] It you specify (com, bridge) a multiuser server can simultaneously support clients using different protocols.
SECURITY | NOSECURITY	N/A	IOM Bridge COM/DCOM	Specifies whether client authorization is required. When security is enabled, the bridge protocol engine requires a username and password; the COM protocol engine is integrated with the single−signon security of the Windows networking environment. Authorization decisions are controlled by the server application. If "nosecurity" is

			specified, these security mechanisms are bypassed.
TIMEOUTSECONDS	Numeric value	IOM Bridge COM/DCOM	Indicates the timeout in seconds before an inactive session is terminated. If this option is not set, inactive sessions are deleted when closed or fully released by the client.
V8ERRORTEXT	N/A	IOM Bridge COM/DCOM	Indicates that the MVA components should return Version 8 style error messages instead of the Version 9 XML style error messages.

IOM Bridge Servers

Server Startup Command

You can specify the server command in either of the following locations:

- on the server startup command−line
- in the sasCommand attribute of the metadata server definition.

Note: Server parameters that are specified on the command line override parameters that are specified in the server metadata.

In the server command, you can provide the following information:

- **SAS configuration file (required)**. To initialize SAS options, you must specify a SAS configuration file using the `−config` option on the server command line. For example,

  ```
  C:\Program Files\SAS\SAS 9.1\sas.exe
     −config "C:\Program Files\SAS\SAS 9.1\sasv9.cfg"
  ```

  The SAS Configuration file contains SAS options that are automatically executed when the SAS System is invoked. The default configuration is located in the SAS install directory; you can also create your own configuration file.
- **SAS Autoexec File (optional)**. To preassign server settings, specify a SAS autoexec file using the `−autoexec` option on the server command line. For example,

  ```
  C:\Program Files\SAS\SAS 9.1\sas.exe
     −autoexec "C:\Program Files\SAS\SAS 9.1\autoexec.sas"
  ```

  A SAS autoexec file contains SAS statements that are executed as part of the SAS invocation. SAS autoexec files are particularly useful for pre−assigning librefs, filerefs, and macros. When multiple workspaces are used on the same server, each workspace inherits the server properties that are set by the autoexec file. Individual workspaces can override the properties that are inherited from the server by specifying new LIBNAME, FILENAME, or macro statements−−however, these changes only affect the workspace where the new statements are submitted.

  Note: Workspaces do not inherit the server WORK library that is used during autoexec processing.

  To use a single autoexec file for both SAS sessions and IOM servers, you can set up conditional statements in your autoexec file. For example,

  ```
  %macro autsetup;
  %if %sysfunc(getoption(objectserver))=OBJECTSERVER
   %then
    %do;
        <IOM server autoexec statements>
    %end;
  %else
   %do;
       <SAS session autoexec statements>
   %end;
  %mend autsetup;
  %autsetup;
  ```

Important: For some SAS 9.1 hosts, IOM servers process a SAS autoexec file implicitly if the file is stored in the default location. This might cause compatibility issues for existing configurations because IOM servers did not process autoexec files in previous versions of SAS. You can suppress this behavior by specifying `-noautoexec` in the server command.

For more information about the -AUTOEXEC system option, see the SAS documentation for your operating environment.

- **Logging Options (optional)**. To diagnose server problems, specify the `-log` and `-logparm` logging options on the server command line. You specify the logging options in the **Command** field or **Object Server Parameters** field of the server definition.

  For details about object server parameters, see Object Server Parameters. When you specify the logging options, you can also configure the server to create a different log for each process, or switch logs during execution.

  The following command (specified in the **Command** field creates a unique log file (in the server user's home directory) for each instance of this server definition.

  ```
  C:\Program Files\SAS\SAS 9.1\sas.exe  -log "test%v.log"
     -logparm "rollover=session"
  ```

  For example, when the spawner starts the first server, a log named `test1.log` is created; when the spawner starts the second server, a log named `test2.log` is created.

  For information about system logging options, see *SAS 9.1 Language Reference: Dictionary*.

  Note: Specifying logging options can cause performance degradation in your server; therefore, you should only specify logging options to diagnose problems with your server connections.

  Note: If you specify a log destination in the configuration metadata rather than the startup command, you might miss some messages that are generated before the log destination is set.

IOM Bridge Servers

Attributes for sasSpawner

The sasSpawner object class contains information for an instance of a SAS spawner. The sasSpawner object class is defined using the attributes listed in the following table. For each option, the table shows:

- The name that identifies the attribute on the LDAP server (or in the configuration file). Under each attributes name, the table shows the corresponding tab and field name in the IT Administrator application.
- "Required" or "Optional" to indicate whether the attribute is required.
- The type of server configuration for which the attribute is used. **Note:** Spawners are used only in IOM Bridge configurations. Therefore, IOM Bridge is listed as the server type for each option.
- A definition of the attribute.

Note: The following attributes which appear in the IT Administrator interface are not used in Version 9 of SAS Integration Technologies

- sasEncryptionModulesPath
- sasMasterPort
- sasMasterService
- sasLUName

If you are using Version 9, do not use these attributes for your configuration.

For step−by−step instructions on defining the metadata for a spawner, refer to Using the IT Administrator Wizard to Define a Server and Spawner or Using IT Administrator to Define a Spawner. If you are not using an LDAP server, you can use a configuration file to define a spawner. For instructions, see Using a Configuration File to Define the Metadata.

sasSpawner Attribute Definitions			
Attribute Name	**Required/ Optional**	**Server Type**	**Definition**
description *In IT Administrator:*: **Description**	Optional	IOM Bridge	Text to summarize why this object definition exists. This attribute is not used by the spawner.
objectClass *In IT Administrator:* **N/A**	Required	IOM Bridge	The object class identifier. For sasSpawner objects, this is always `sasSpawner`. If you use IT Administrator, this identifier is assigned automatically.
sasDomainName *In IT Administrator:* **Connections** ➔ **Domain**	Optional	IOM Bridge	The security domain in which the sasSpawner definition participates. The spawner definition must have the same domain name as the server with which it connects. The spawner uses the domain name, along with the machine name and logical name, to determine which server(s) it services. The lack of a domain is considered a domain; therefore, if the spawner definition has no domain name, it will be associated only with servers that have no domain name.

sasLogFile *In IT Administrator:* **Logging → Log File**	Optional	IOM Bridge	A fully qualified path to the file in which spawner activity is to be logged. Paths with blanks must be in quotation marks. On Windows, paths with embedded blanks must be in double quotation marks. On z/OS, specify filenames similar to UNIX file paths due to the requirement for z/OS Unix System Services.
sasLogicalName *In IT Administrator:* **Logical Names**	Optional	IOM Bridge	The logical names associated with this sasSpawner definition. When a spawner receives a request to start a SAS session, it looks for servers in its own domain and on its own machine that have the same logical name as the spawner. If logical names are specified, then the spawner supports only those sasServer instances whose logical names are a subset of the logical names defined here. If you are using a configuration file instead of LDAP, specify each logical name as a separate attribute and value pair.
sasMachineDNSName *In IT Administrator:* **Machines**	Required	IOM Bridge	The DNS name(s) and IP address(es) for the machine(s) on which this spawner definition may execute. Multiple values can be assigned to this attribute. The machine name must be the official network name (for example, machine.corp.com). The string `localhost` can be used to signify the host on which the spawner is executing.
sasNetEncrKey *In IT Administrator:* **Encryption → Key Length**	Optional	IOM Bridge	A numeric value (0, 40, or 128) that specifies the encryption key length. See SAS/SECURE for more information regarding this attribute.
sasOperatorPassword *In IT Administrator:* **Connections → Operator → Password**	Optional	IOM Bridge	The password that an operator must enter to connect to the spawner in order to perform administration tasks. The default password is sasobjspawn.
sasOperatorPort *In IT Administrator:* **Connections → Operator → Port**	Optional	IOM Bridge	The port on which to connect to the spawner in order to perform administration tasks (such as checking status). If neither sasOperatorPort nor sasOperatorService is specified, the service name sasobjoper is used as the sasOperatorService. The spawner will not start without an Administrator listen port or service.
sasOperatorService *In IT Administrator:* **Connections → Operator → Service**	Optional	IOM Bridge	The service in which to connect to the spawner in order to perform administration tasks. If neither sasOperatorPort nor sasOperatorService is specified, the service name sasobjoper is used as the sasOperatorService. The spawner will not start without an Administrator listen port or service.

sasSpawnercn *In IT Administrator:* **Name**	Required	IOM Bridge	The unique name for this sasSpawner object. When specified at spawner invocation, its value identifies which sasSpawner definition to use.
sasUUIDNode *In IT Administrator:* **Connections → UUID → Node**	Optional	IOM Bridge	The 12−character string that represents this spawner's node portion of its generated UUIDs. See the section titled Request a Universal Unique Identifier (UUIDGEND) for more information regarding UUID generation by the spawner.
sasUUIDPort *In IT Administrator:* **Connections → UUID → Port**	Optional	IOM Bridge	The port on which to connect to request UUID generation.
sasUUIDService *In IT Administrator:* **Connections → UUID → Service**	Optional	IOM Bridge	The service on which to connect to request UUID generation.
sasVerbose *In IT Administrator:* **Logging → Verbose**	Optional	IOM Bridge	When present, this attribute causes the spawner to record more details in the log file (sasLogFile or slf).

IOM Bridge Servers

Initializing UNIX Environment Variables for SAS Workspace Servers

In UNIX environments, many third−party databases require access information such as the default server address to be set as environment variables. To make these environment variables available to a SAS Workspace Server, you must create the workspace using a *wrapper script* that defines the variables before invoking SAS.

The following code is an example script.

```
#!/bin/ksh −p

# Purpose: Runs database setup scripts before invoking SAS.
#          Called by objspawn.

# Restore quotation marks around arguments that have multiple tokens.

function quoteme { #arg

   if [[ $# −gt 1 ]]; then
      quoteme="\"$*\""
   else
      quoteme=$1
   fi

   echo $quoteme
}

#  Run database setup scripts or set required environment
#  variables here.

 <script calls or export commands>

#  Reconstruct and execute the original SAS command.

cmd=''
for arg in "$@" ; do
   tmp="$(quoteme $arg)";
   cmd="$cmd $tmp"
done

eval exec $cmd
```

To use this script:

1. Add your `export` statements or script calls and save the file as `objspawn.setup`.
2. Set the execute bits for the file. You can do this using the following command:

   ```
   chmod 755 objspawn.setup
   ```
3. Add `objspawn.setup` to the start of your sas command in the server definition. For example:

   ```
   objspawn.setup sas
   ```

Stored Processes

Stored Processes

A stored process is a SAS program that is stored centrally on a server. A client application can then execute the program, optionally supply name/value parameters, and receive and process the results. For details about creating a stored process and processing the results, refer to Stored Processes in the Integration Technologies Developer's Guide.

To make a stored process accessible to client applications, you can use IT Administrator to create metadata that describes the stored process and its location. The process includes:

- defining a stored process path to enable applications to determine where the stored process is located
- creating a stored process object that contains a name for the stored process and information about associated parameters

Stored Processes

Creating a Stored Process Path

A stored process path is a location where SAS source programs are stored by name. A client can then use the name of the process to execute the program and receive the results.

The SAS Integration Technologies Administrator provides a graphical user interface that allows you to create or modify a stored process path. For general instructions, see Using IT Administrator.

To create a new stored process path using IT Administrator:

1. Open IT Administrator.
2. Select **Applications** in the Manager Bar.
3. Select the **Stored Process Paths** folder in the tree view.
4. Select the **New** button (). A dialog box appears requesting property information for the new stored process path.
5. Enter the name and path, and select one or more logical names that should be associated with the stored process path.
6. When you are finished entering information in the fields, select **OK**. The new stored process path appears in the tree view.

To modify a stored process path using IT Administrator:

1. Open IT Administrator.
2. Select **Applications** in the Manager Bar.
3. Expand the Stored Process Paths folder, then select the path you want to modify. The properties area displays:

   Stored Process Path
   : The path where the stored process definitions are stored. Enter any changes directly in the Stored Process Path field.

   Logical Names
   : The logical names with which this path is associated. To associate the stored process path with a logical name, select the **Add** button to display the Logical Name Info window. Use this window to select an existing logical name or define a new one, then select **OK** to add the logical name to the stored process path definition.
4. When you are finished, select the **Save** button () to save the changes to the server. If you attempt to navigate to another object, the administrator application will prompt you to save any changes you made to the path.

Stored Processes

Creating a Stored Process Object

To enable a stored process to be invoked in an application, you can use IT Administrator to create metadata that defines the name of the stored process and provides information about associated parameters. For general instructions on using IT Administrator, see Using IT Administrator.

To create a new stored process object using IT Administrator:

1. Open IT Administrator.
2. Select **Applications** in the Manager Bar.
3. Select the **Stored Process Paths** folder in the tree view.
4. Select the node representing the stored process path for the new stored process. If the path is a folder, open the folder and select any stored process node beneath the folder.
5. If you are creating the first stored process under the selected stored process path, you must select **File** ➔ **New** ➔ **Stored Process** in order to create the process. Otherwise, click the **New** button (). The New Stored Process window appears requesting property information for the new stored process.
6. Enter the name and description of the process. This information identifies the process in the directory. In the SAS Stored Process section, enter the following

   Descriptive Label
   : A short label for the stored process. If you specify a value for this field, you must also specify a value for the File field.

   File
   : The name of the stored SAS program. If you specify a value for this field, you must also specify a value for the Descriptive Label field.

   Portal JSP
   : The java server page used to access the program

   In the Parameters field, enter the parameters that are passed to the SAS program upon execution. You can have more than one parameter set.

   Valid parameters are:

   DATA_SOURCE
   : Specifies the distinguished name of a data source used to generate a list of valid values for the parameter. Used together with the REPORT_QUERY parameter.

   DEFAULT_VALUE
   : Specifies the default value for the parameter.

   DESCRIPTION
   : Provides a description of the parameter.

   LABEL
   : Specifies the label to display when the user sees the parameter.

   NAME
   : Specifies the name of the parameter.

   REPORT_QUERY
   : An SQL statement used together with DATA_SOURCE to generate a list of valid values for the parameter.

   REQUIRED
   : Specifies whether the parameter is required (value is true) or not required (value is false).

   VALUES

List the possible values for the parameter. If VALUES is not specified, DATA_SOURCE and REPORT_QUERY are used to generate the list.

7. When you are finished entering information in the fields, click **OK**. The new stored process appears in the tree view.

   If the **OK** button remains gray, check the Descriptive Label and File fields. You must either enter values in both fields or neither field. If you enter a value in one field but not the other, the **OK** button will remain grayed.

To modify or view details about a stored process using IT Administrator:

1. Open IT Administrator.
2. Select **Applications** in the Manager Bar.
3. Expand the Stored Process Paths folder, then select the stored process whose details you want to modify or view. The properties appear in the properties area.

   The properties area displays these fields

   Description
   A description of the stored process
   File
   The name of the stored SAS program
   Portal JSP
   The java stored page used to access the program
   Descriptive Label
   A short description of the process
   Parameters
   The parameters that are passed to the SAS program upon execution. You may have more than one set of parameters for each stored process. To add a parameter set, select the **Add** button. The Add Value dialog box appears, in which you can add the new parameter set. Select a parameter set and select the **Edit** button to change the parameter values, or **Remove** to remove the set.
4. When you are finished, select the **Save** button ( ) to save the changes to the server. If you attempt to navigate to another object, the administrator application will prompt you to save any changes you made to the process.

Publishing

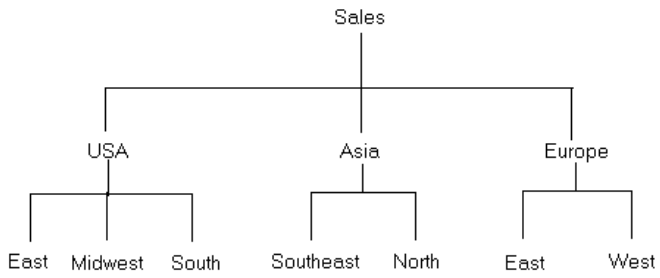
Administering the Publishing Framework (Publish and Subscribe Planning and Implementation Guide)

In order to set up a publish and subscribe solution using SAS Integration Technologies, you can follow these steps to help you plan and implement your solution. The following is a list of tasks that are required to set up a publish and subscribe system using Integration Technologies:

1. **Design information channels**

   Designing a successful publish and subscribe implementation starts with an understanding of why your organization is implementing the system. You will need to know, at a very basic level, what kind of information needs to be distributed to users and how widely that information needs to be distributed.

   For example, you could start the planning process by understanding that your organization needs to disseminate sales information throughout the marketing organization and inventory data to the production organization. Starting with this base level of knowledge, you begin the process of breaking down the general categories of information into specific information channels by using a hierarchical model.



   How you divide and subset the categories depends on your organization's needs, but you should work toward creating information channels as tightly focused as possible, without making them too tightly focused to be useful. Channels that are broadly defined leave users not knowing whether information delivered over the channel will be useful to them; channels that are too narrowly defined force users to subscribe to a long list of channels in order to ensure that they receive the information that they need.

   To help focus the information that users receive, set up policies for name/value keywords. Name/value pairs are attributes that are specified when a package is published and that help to identify the package contents. Each subscriber definition can include a name/value filter that only allows packages that meet the subscriber's needs to be delivered.

   For example, if you publish a package with a name/value attribute of `market=(Mexico)`, that package is only seen by those subscribers whose name/value filter indicates that they are interested in information about the Mexican market. Although the names and associated values can be anything that your organization finds useful, you must establish a list of acceptable keywords and values for those keywords. This list is essential for publishers to be able to provide consistent metadata that identifies published content and for subscribers to be able to filter published content in order to focus on the information they need.

   When you define your information channels, you must also consider the users that will be accessing those channels as well as any restrictions that need to be placed on the channels. Although these aspects of planning

are discussed separately and in more detail in the following two topics, in practice they are examined at the same time as you are defining your channels. You cannot define an information channel without first knowing who needs to see the information and how that information should be restricted.

2. **Identify initial subscriptions**

   When you plan an initial set of information channels, you must identify the users, groups, and applications that are initially subscribed to those channels. The information to set up these subscriptions is taken from the information you collected when you planned the channels. An understanding of your organization's need for a publish and subscribe system must include not only what information needs to be published, but also who needs to see that information.

   However, you do not have to determine every piece of information that every individual needs to see. Rather, the process of planning initial subscriptions focuses on wider distributions of information, such as identifying the essential information that departments and groups of users need. How closely you follow this guideline depends on your organization's needs − there might be a few critical users who need to receive specific information, and there might be a need to subscribe a group of users to a tightly focused channel. In general, however, the initial subscriptions that you plan cater to distributing essential information to the largest number of users. Subscribers can set up subscriptions to tightly focused channels themselves as the need arises.

   After you have determined the list of initial subscribers for each channel, you must determine how the information is to be distributed to each user (whether by text e−mail, HTML e−mail, or through a queue) and identify their address information. The address information is essential for setting up both subscriber entries and the LDAP directory.

3. **Analyze information security requirements**

   When you plan information channels you must also consider security for your publish and subscribe implementation in order to ensure that the information that is published on each planned channel is uniformly sensitive. For example, if you plan for a single channel to distribute accounting information throughout your organization, you will encounter a security problem when the accounting department needs to publish sensitive information (such as employee salaries). With only a single, unrestricted channel, you cannot publish the information to a specific set of users. In your consultations with users, you must identify information channels whose access needs to be controlled.

   Your plan must address both methods that Integration Technologies uses to implement security − authentication and access control.

   Authentication security involves the process of users connecting to the LDAP server. Because the LDAP server contains all of the definitions for Integration Technologies objects (including subscribers, channels, and servers), a user must be able to connect to the LDAP directory in order to make any changes to the LDAP definitions. This level of security is controlled when users supply a distinguished name and corresponding password when they connect to the directory.

   Access control security controls the information channels that users have access to. Without any security, users are able to subscribe to any information channel in your organization and access sensitive information. To prevent this, you must to create access control lists (ACLs) in the LDAP directory in order to specify what definitions and attributes users have access to. To plan for implementing access control security, you must consider what kinds of users access the directory and what kinds of information they should have access to. As an example, the following are some possible user classes and questions you must consider for each one:

   General subscriber

Should subscribers be able to change their own password? Should they be able to change their own subscriptions? Should there be channels that not all subscribers are allowed to subscribe to?

Management−level user

Should someone at management level be able to modify user subscriptions? Should a manager be able to access management−only channels?

Administrator

Should the administrator be able to access all attributes of a subscriber's definition, including the password? Should the administrator be able to add and delete subscribers from a channel?

In addition, your initial information channel planning must identify some channels whose distribution is limited to certain user classes. You must make sure that the user classes that you identify when you consider security planning match those that you identified during initial channel planning.

After you determine rules for access control for the user groups in your organization, you must work toward simplifying and consolidating the rules as much as possible. Rather than having many specific rules for each group, try to develop a general rule that accomplishes the same result and can be applied to the user base as a whole.

For example, if your organization has five classes of users (A through E) and you want to define the rules for access to channel definitions, you can write the following rules:

- user class A is not allowed access
- user class B is not allowed access
- user class C is not allowed access
- user class D is allowed access
- user class E is not allowed access.

Each rule is then applied to each user group separately. To simplify, you can define the following rule:

- Only user class D is allowed access.

This rule, applied one time to the whole user base, accomplishes the same result as the previous list of rules.

4. **Configure the LDAP server**

   After you complete your initial planning, you can begin implementing the publish and subscribe solution. The first step is to identify and configure the LDAP server. You must start by installing the LDAP directory server software, if you have not already done so. See Setting Up an LDAP Server for information on the process of installing and configuring an LDAP server. If you are using Microsoft's Active Directory, you must install the LDAP schema to support that server. See Installing the LDAP Schema for Microsoft Active Directory for more information.

   When you install and configure the LDAP server, you must create person entries in the directory in order to supply identifying information for persons and groups in your organization. Make sure that you create a person entry for each person and group in your organization that you have identified as a subscriber. If a person or group does not have an entry in the LDAP directory, you will not be able to create a subscription for them.

5. **Configure channels and subscribers**

   After you install and configure the LDAP directory server, use the Integration Technologies Administrator application to define the channels and subscriptions that you identified during the planning phase. Begin by defining the subscriber entries. Defining the subscribers first gives you the ability to select a channel's subscribers at the time that you define the channel. See Creating Subscribers for information.

After you define the subscribers, define the channels and associate subscribers with the channels. See Creating Channels and Creating Subscriptions for information.

6. **Implement LDAP directory security**.

   After you define the channels and subscriptions, implement the security plan that you previously devised. To implement access control security, you must add Access Control Lists (ACLs) to your LDAP directory. The ACLs are composed of access−control information (ACI) statements, each of which specifies the access policy for a particular target.

   Simplify your security policies by consolidating your ACI statements as you write them. Establishing a simple security structure that has a relatively small number of ACIs makes security easier to maintain, allows for change as the system changes, and helps prevent conflicts between ACI statements.

   Consult the documentation for your LDAP directory server for details on creating ACI statements and ACLs.

7. **Develop applications that deliver content**

   After you set up the publish and subscribe infrastructure and implement the mechanisms that deliver content to a selected set of users, you must develop or modify applications that will be used to create the content to be published. These applications can take the form of standalone applications that are written in a visual programming language or SAS programs. See Publishing Framework in the *Integration Technologies Developer's Guide* for information about the tools that are available to create a publishing application. See SAS Publisher in the *Integration Technologies Developer's Guide* for information on using the SAS Publisher application to create and publish packages.

8. **Make client applications available**

   After you develop or modify the applications that publish content, the initial structure of the publish and subscribe implementation is complete. Your next step is to make these applications available to users in your organization. Using the information that you gathered during initial planning, make the appropriate applications available to each user or group. Publishers must obtain or install the appropriate publishing application for their needs. For example, an individual or department that needs to publish data−intensive reports on a regular basis might use a SAS program for publishing, while a user who needs to send information to a changing number of users on an occasional basis might use the SAS Publisher application.

   Subscribers must also obtain or install any appropriate software that is required to view published content. In particular, each subscriber must install the SAS Package Reader application in order to be able to view the contents of published SAS packages. See SAS Package Reader in the *Developer's Guide* for more information. If the subscribers receive information through queues, they must also install the SAS Retriever. See SAS Package Retriever in the *Developer's Guide* for more information.

   It is recommended that subscribers install the SAS Subscription Manager Java applet. This applet enables subscribers to subscribe to and unsubscribe from channels as well as change how content is delivered. Giving subscribers the ability to change their own information lessens the burden on the administrator and lets the administrator concentrate on administering channels. See SAS Subscription Manager in the *Developer's Guide* for more information.

9. **Announce solution and train users**

   After the publishers and subscribers install the necessary applications, you can announce your implementation to your organization. You will also need to follow up the announcement with training for both publishers and subscribers, with training broken down by publishing methods, publishing needs, and subscriber applications.

Publishing

Creating Channels

A channel is a topic or identifier that acts as a conduit for related information. The channel carries the information from the publishers who created it to the subscribers who want it.

Channels have a name, a description, a subject, key words, reference keys, and archive paths associated with them. This information is used by search facilities in the subscription manager to help users locate channels that are of interest to them and are also used in the administrator application to locate specific channels for administration purposes. Channels will also have subscribers (including subscriber groups) associated with them.

Each association of a subscriber to a channel is a subscription. A subscription enables the information that is published to a channel to be delivered to the interested (subscribed) subscribers. Note that, although you can associate a group to a channel, only the members of the group that are also (and separately) identified as subscribers will receive the published content.

Administrators should create a channel for each distinct topic or audience. For instance, users of a particular application may want a channel for discussion and data exchange, while the programmers of that application may want another channel to discuss technical problems and future enhancements. Although the topic is the same application, the discussion and data exchanged will be very different, so two separate channels would probably best serve the needs of the two groups of users.

The SAS Integration Technologies Administrator provides a graphical user interface that allows you to create a channel object with the specified attributes on the LDAP server. For general instructions, see Using IT Administrator.

To create a new channel using IT Administrator:

1. Open IT Administrator.
2. Select **Publish Framework** in the Manager Bar.
3. Select the **Channels** folder in the tree view.
4. Click the **New** button (). A dialog box appears requesting property information for the new channel.
5. Enter the name, description, and subject of the channel.

   Channel names must be unique, and are limited to 40 characters. They cannot contain a comma (,) or a forward slash (/).

   Users that publish information and users that receive information may later search for these items. Therefore, meaningful and concise entries are important. In particular, the channel should have a meaningful name so that users of the subscription manager can easily discern the type of information that will be distributed via that channel.

   For example

   ```
   Name: Conversion Tool Users
   Description: User discussion of the conversion tool
   Subject: sgml2html
   ```
6. Add subscribers to the channel from the **Subscribers** tab by selecting the **Add** button and selecting the subscribers and groups from the Add Subscribers window. If you add a group as a subscriber to a channel, make sure that every member of the group is also defined as a subscriber. Group members that are not defined as subscribers will not receive content.

You cannot deliver information on the channel until you define subscribers. See Creating Subscriptions for more information.

7. Enter any keywords or reference keys from the **Advanced** tab.

   Keywords enable you to describe the channel beyond the description and subject, and are used in keyword searching.

   Reference keys are usually assigned programmatically to associate a channel with another object. For example, SAS/Warehouse Administrator uses reference keys to associate a channel with an object in a data warehouse. When information is published from the warehouse, the warehouse software can locate associated channels by searching the LDAP server for channels that have matching reference keys. Reference keys do not usually contain meaningful information for a user, but are provided as a debugging aid.

   To enter multiple keywords or references, press Return between each entry. Specifying keywords and references enables you to specify a wide range of values for which you can later search.

   To delete a keyword or reference that is already entered, select it and press the Delete key.
8. When you are finished entering information in the fields, select **OK**. The new channel appears in the tree view.
9. If you want to create an archive path for the channel, select the channel in the tree view and select **File → New → Archive Path**. The New Archive Path window appears, where you can enter details about the path. When you finish, the definition is placed below the level of the channel in the tree view.
10. If you want to assign one of the archive paths you created as the default for the channel, select the channel in the tree view and select the Advanced tab in the properties window. Select the Select button under Default Archive Path. The Select Default window appears and lists all of the archive paths currently assigned to the channel, as well as the selection <none>. If you have not assigned any archive paths to the channel, <none> will be the only selection. Select the archive path and select OK. If you select <none>, the channel will not have a default archive path.

    The selected archive path appears in the Default Archive Path field.

    Note: When writing SAS programs that use the Publish Package Interface, to use the default archive path, you must include the ARCHIVE_PATH property in the PUBLISH_PACKAGE call routine and leave the value blank. For more information, see Publish Package Interface in the *SAS Integration Technologies Developer's Guide*.

Certain channels may deliver information that should be restricted to a particular audience. The publication administrator must work with the LDAP administrator to configure the access permissions for restricted channels. If a subscriber does not have read access to a particular channel, the channel will not appear as a selection when they run the subscription manager applet.

Publishing

Creating an Archive Path

When publishers create packages to be published, they can specify that the package is archived when it is published. The publisher then stores a copy of the package in a specified location, either for later retrieval or for archival purposes.

The SAS Integration Technologies Administrator provides a graphical user interface that allows you to define a new archive location. For general instructions, see Using IT Administrator.

To define a new archive location using IT Administrator:

1. Open IT Administrator.
2. Select **SAS Archiving** in the Manager Bar.
3. Select the **Archive Paths** folder in the tree view.
4. Click the **New** button (). A dialog box appears requesting property information for the new archive path.
5. Enter the name and path, and select one or more logical names that should be associated with the archive path.
6. When you are finished entering information in the fields, select **OK**. The new archive path appears in the tree view.

Publishing

Creating Subscribers

The publication administrator must define a subscriber for each user before the user can receive information from a channel or use the subscription manager applet to configure their subscriptions. A subscriber must have a SASperson object (containing a name and a delivery transport) defined for them when they are created.

It is important to understand how the level of authentication being used by the LDAP server affects the creation of subscribers. The administrator application and the subscription manager applet will view and interact with the LDAP server and the subscriber in different ways depending on the authentication setting. Authentication is configured in the site.cfg file.

If LDAP server authentication is set to "none", the publication administrator must create a subscriber with the administration application for each user. This creates a subscriber object in the publication framework. The information in this entry is not used by the LDAP server for authentication (since authentication is set to "none"), but the information is used by the subscription manager applet to identify a particular subscriber at login so that the information particular to that subscriber can be retrieved.

If the LDAP server authentication is set to "simple", the publication administrator should create a subscriber **only** for those users who have an identity defined on the LDAP server outside of the Publish/Subscribe framework. The LDAP administrator must create these identities with a valid user name and password. The name of the subscriber created by the publication administrator in the publication framework must match **exactly** the name of the identity on the LDAP server created by the LDAP administrator.

When configured properly for simple authentication, the subscription manager applet will prompt a user for a user name and password which will then be used to authenticate the user against the LDAP server. After the user has been authenticated against the LDAP server, the applet attempts to retrieve the user's subscription information using the same name. The applet expects the name in the identity defined on the LDAP server outside the publication framework to be identical to the name of the subscriber created by the publication administrator.

The delivery transport defines how information is delivered to a subscriber. Information can be provided via an e−mail message or a message queue. The publication administrator can configure subscriptions for any type of subscriber, regardless of whether the subscriber is a person or a program. Most human users will want information delivered to them through e−mail, while subscribers that are programs will likely be configured to receive information in a queue for further processing.

For example, SAS Integration Technologies provides support for message queues like MQ SERIES and MSMQ. A SAS program could be written to wait for information to be delivered on a message queue and then consume that information. While the program would be defined as a subscriber and the delivery mechanism would be a message queue, the identity of such a subscriber is not used for LDAP server authentication. Subscribers that are programs are administered by the publications administrator and do not use the subscription manager applet.

If you create a subscription for a person, that subscriber can deactivate (remove) the subscription or change the delivery mechanism.

If you create a subscription for a group, you must also make sure that every member of that group is defined as a subscriber. Group members that are not defined as subscribers will not receive published content.

The SAS Integration Technologies Administrator provides a graphical user interface that allows you to create a subscriber. When you create a subscriber via the administrator application, the subscriber object with the specified attributes is stored on the LDAP server. For general instructions, see Using IT Administrator

To create a new subscriber using IT Administrator:

1. Open IT Administrator.
2. Select **Publish Framework** in the Manager Bar.
3. Select the **Subscribers** folder in the tree view.
4. Click the **New** button (). A dialog box appears requesting property information for the new subscriber.
5. Enter the name and description of the subscriber.

   Subscriber names must be unique, and are limited to 40 characters. They cannot contain a comma (,) or a forward slash (/). The name of the subscriber is used by the publication framework to reference the subscriber, including the association of subscribers to channels.
6. Enter the distinguished name of a SAS person reference.

   Select the **Select** button to display the Person Index window. This window displays all of the person reference entries in the LDAP directory, organized under a set of alphabetic tabs. Select the person reference and select **OK** to display the person reference in the New Subscriber window.

   A person reference points to an LDAP object that describes the subscriber in more detail (such as password, phone number and room number).
7. Choose the delivery transport for the subscriber, either email, queue, or none.
8. If you choose email as the delivery transport, specify the format (HTML or text), the email address, and the delivery method.
9. If you choose queue as the delivery transport, specify the queue setting and the delivery method.
10. Optionally, in the Advanced tab, you can define
    - Name/Value filters to filter at the package level
    - Entry filters to determine the types of entries you receive
    - MIME Type filters to determine the types of files you receive.

11. When you are finished entering information in the fields, select **OK**. The new subscriber appears in the tree view.

Although defined, the new subscriber is not yet subscribed to a channel or part of a group. See Creating Subscriptions for more information.

Publishing

Creating Subscriptions

When you associate a subscriber to a channel, you create a subscription. This association enables information published to the channel to be delivered to the subscriber.

The SAS Integration Technologies Administrator provides a graphical user interface that allows you to create subscriptions to a channel.

To create subscriptions to a channel using IT Administrator:

1. Open IT Administrator.
2. Select **Publish Framework** in the Manager Bar.
3. Select the channel in the tree view. The channel's properties appear.
4. Click the **Add** button in the property view. The Add dialog box appears.
5. From each tab in the Add dialog box, select the subscribers and groups to subscribe to the channel.

   Control−click to make non−contiguous selections. Shift−click to select a range. Selections in one tab persist when you move to the other tab.

   Note: Although you can subscribe a group to a channel, only the members of the group that have been identified as subscribers will receive published content. See Creating Subscribers for more information on identifying users as subscribers.
6. Click **OK** to add the selected subscribers or groups to the channel.
7. When you are finished, click the **Save** button ( ) to save the changes to the channel. If you attempt to navigate to another object, the administrator application will prompt you to save any changes you made to the channel.

Publishing

Name/Value Filters

Publishers can specify name/value pairs describing the information being published. This enables subscribers to filter the packages they receive. Publishers are responsible for providing accurate name/value pairs for the information they publish.

Name/value filters use comparison operators such as = (equals), != (not equals), and ? (contains) and logical operators such as & (and) and | (or) to specify the information that will be passed by the filter. Some example filters are:

- market=(US, Asia, Europe)
- type=report & forecast
- priority=high | medium

For detailed information on name/value filters, see Specifying Name/Value Filters in the *Integration Technologies Developer's Guide*.

Entering Name/Value Filters

To enter name/value pairs, type the filter string into the NameValue field and press Enter.

Deleting Name/Value Filters

To delete a name/value pair, select the entry and press the Delete key.*Publishing*

Entry Filters

Publishers send information in packages which can be described with name/value pairs. Each package contains one or more entries that can also be described with name/value pairs. When publishers provide such descriptive data for entries, subscribers can filter the entries, providing another level of control over the information they receive. As always, publishers are responsible for providing accurate name/value pairs for the information they publish.

Entering Entry Filters

To enter Entries, type the entry into the Inclusion or Exclusion field and press Enter.

Deleting Entry Filters

To delete an Entry, select it and press the Delete key.

Publishing

MIME Type Filters

MIME types provide details about the information being published. For example, specifying the MIME type "audio/basic" indicates that the file is an audio file and requires software that can interpret such content. It is the responsibility of publishers to specify the appropriate MIME type parameters for the information they publish.

Subscribers can filter the type of information they receive by typing a MIME type into the Inclusion or Exclusion field. For example, if a subscriber is connecting via a modem, some data types may be too large or unwieldy to use, such as movies or audio. By excluding those MIME types, the subscriber never encounters those types of information.

Entering MIME type filters

To enter MIME type filters, type the content string into the Inclusion or Exclusion field and press Enter.

Deleting MIME types

To delete a MIME type, select the entry and press the Delete key.

Some common MIME types:

- application/msword
- application/octet−stream
- application/pdf
- application/postscript
- application/zip
- audio/basic
- image/jpeg
- image/gif
- image/tiff
- model/vrml
- text/html
- text/plain
- text/richtext
- video/quicktime
- video/mpeg

Publishing

Creating Overrides

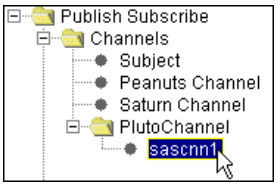
An override is created when a subscriber's delivery information is changed by a subscriber via the subscription manager applet to differ from

- the default delivery information as specified by the subscriber's properties when subscribed directly to a channel.
- the default delivery information specified by the subscriber's delivery properties when subscribed to a channel via a group.

When the default delivery information for a subscriber is overridden, the channel's node changes to a folder. The channel folder contains an override for each subscriber whose delivery information differs from their default delivery information, or the default delivery information of a group through which they are subscribed.

For example, in the following image the PlutoChannel appears as a node.

However, after the delivery information for user sascnn1 has been changed from the default delivery information for that subscriber, the PlutoChannel node displays as a folder containing all overrides for that channel. The following image displays the sascnn1 override in the PlutoChannel Channel.



In essence, overrides are a special type of subscriber, and can be edited or removed just as a subscriber can be edited or removed.*Data Sources*

SAS Data Sources

Data sources are SAS libraries, SAS tables, and columns within SAS tables which are accessible to users through client applications. You can use IT Administrator to create objects containing metadata for these data sources. You can define:

- library data sources
- table data sources
- column data sources

Client applications can then use the resulting LDAP definitions to access SAS data.

Data Sources

Creating a Library Data Source

A library data source is an existing SAS library that is accessible to users through client applications.

The SAS Integration Technologies Administrator provides a graphical user interface that allows you to create an LDAP object definition for a library. Client applications can then use the LDAP definition to create a LIBNAME statement that can be used to access a data source. The logical names that you assign to the library associate that library with the servers on which the library resides.

To create a new library data source using IT Administrator:

1. Open IT Administrator.
2. Select **SAS Data Sources** in the Manager Bar.
3. Select the **Libraries** folder in the tree view.
4. Click the **New** button (). The New Library window appears and requests that you enter property information for the new library.
5. Enter a name and description for the library. The name identifies the library in the tree view, and the description is visible when the object properties are viewed.
6. On the Library tab (selected by default), you must enter a SAS library reference (libref) name in the **Libref** field. If the library is preassigned by the server, then the library must already exist and must use the name that is entered in this field.
7. Enter values for any of the other Library fields. You can find information about the values that are required in each field by selecting the **Help** button in the New Library window.
8. To associate the library with a logical name, select the **Add** button. The Logical Name Info window appears where you can select an existing logical name or create a new one. The logical name that you choose identifies the server on which the library resides. After you have chosen a logical name, select **OK** to close the Logical Name Info window.
9. When you are finished entering information in the fields, select **OK**. The new library definition appears in the tree view.

Data Sources

Creating a Column Data Source

A column data source is a column in an existing SAS table that is accessible to users through client applications. You must create a data source definition for the column's table before you create the column definition.

The SAS Integration Technologies Administrator provides a graphical user interface that allows you to create an LDAP object definition for a column. Client applications can then use the LDAP definition to access the column.

To add a new column data source using IT Administrator:

1. Open IT Administrator.
2. Select the **SAS Data Sources** button in the Manager Bar.
3. Select the **Tables** folder in the tree view.
4. Select the definition for the table that contains the column you want to use as a data source.
5. Select **File** ➔ **New** ➔ **Column**. The New Column window appears and requests that you enter property information for the new column.
6. Enter a name and description for the column. The name identifies the column in the tree view, and the description is visible when the object properties are viewed.
7. In the Column fields, you must enter the following information:

   Name
   Enter the name of the column as it is identified in the SAS table. The column must already exist and must use the same name as the one that is specified in this field.

   Type
   Enter the column type − either N (numeric) or C (character).

   Maximum Length
   Enter the maximum specified length for values in the column.

8. Enter values for any of the other Column fields as needed. You can find information about the values that are required in each field by selecting the **Help** button in the New Column window.
9. When you are finished entering information in the fields, select **OK**. The new column definition appears in the tree view.

Data Sources

Creating a Table Data Source

A table data source is an existing SAS table that is accessible to users through client applications.

The SAS Integration Technologies Administrator provides a graphical user interface that allows you to create an LDAP object definition for a table. Client applications can then use the LDAP definition to access the table.

To add a new table data source using IT Administrator:

1. Open IT Administrator.
2. Select the **SAS Data Sources** button in the in the Manager Bar.
3. Select the **Tables** folder in the tree view.
4. Click the **New** button (). The New Table window appears and requests that you enter property information for the new table data source.

   An alternative method for creating a table is to select the **Tables** folder and select **File** ➔ **New** ➔ **Table**.
5. Enter a name and description for the data source. The name identifies the table in the tree view, and the description is visible when the object properties are viewed.
6. On the Table tab (selected by default), you must enter the table name in the **Member Name** field. Note that the table must exist and must use the name that is entered in this field.
7. Enter values for any of the other Table fields as needed. You can find information about the values that are required in each field by selecting the **Help** button in the New Table window.
8. To associate the table with a logical name, select the **Add** button. The Logical Name Info window appears where you can select an existing logical name or create a new one. The logical name that you choose identifies the server on which the table resides. After you have chosen a logical name, select **OK** to close the Logical Name Info window.
9. When you are finished entering information in the fields, select **OK**. The new table definition appears in the tree view.

Security

Security

For the LDAP server and SAS Integration Technologies, you can implement security using authentication and authorization mechanisms. **Authentication** is the process of verifying that a person is who they say they are. **Authorization** is the process of evaluating whether a given user has permission to perform a task (such as read or write) on a given resource.

SAS Integration Technologies 9.1 supports Sun ONE Directory Server Version 5.1, Netscape Directory Server 4.12 (also owned by Sun Microsystems, and previously sold under the name iPlanet Directory Server), and IBM Secureway Server Version 3.2.2.

To implement security for SAS Integration Technologies, follow these steps:

1. **Define Person Entries for Authentication.** To enable authentication against the LDAP server, you must set up your person entries on the LDAP server. For details, see Defining Person Entries
2. **Implement Server Security (optional)**. If you are using an IOM Bridge server, you can use a SAS Login definition to ensure that only authorized users obtain access to SAS data and processes. The login definition specifies which specific users or groups of users can access the server. For more information, see Defining a SAS Login.
3. **Define Access Controls for Authorization**. You can update access controls on the LDAP server. Authorization in SAS Integration Technologies is accomplished using access control information (ACI) rules (for the Sun ONE Directory Server and Netscape Directory Server) and access control permissions (for the IBM Secureway Directory Server).

   For general information about access control, see Sun ONE and Netscape Directory Server Access Control Overview and Secureway Directory Server Access Control Overview. You can also refer to the Sun Product Documentation Web site. (The Sun ONE Directory Server is referred to as iPlanet Directory Server on this page.)

   For information about using the Integration Technologies Administrator to specify ACI rules for the iPlanet Directory Server, see Setting Access Permissions for an Object and Specifying Bind Rules. For information about using the Integration Technologies Administrator to set access control for the Secureway Directory Server, see Setting Access Control for Objects.

Security

Adding Person Entries to the Directory

Before you define person entries, you should have already started the directory server, updated the schema, and set the basic access control. (For details, see Setting up an LDAP Directory Server.)

Person entries are needed in order to make the directory userful to SAS applications. For example, when you update access control, access decisions are based on the Distinguised Name (DN) that the person binds to the directory.

SAS software also uses person entries to identify users and to obtain information such as user ID and e−mail address. Some of the options for user data are object class, directory structure, and DN.

To add person entries to the directory, follow these steps:

1. Select an object class to use for the entries. A common choice is the `inetOrgPerson` class, which accepts many useful attributes. If you need to add attributes to your person entries and the attributes are not allowed by `inetOrgPerson`, you can create your own object class using `inetOrgPerson` as a parent class.
2. Enter the person entries in the directory. Follow these guidelines to help your person data work better with SAS software:

   - Keep common names unique. Some SAS applications use the common name when associating a person entry with other entries in the SAS application entries.
   - Include the user ID and e−mail address in the person entry. Applications need to look up the user ID.
   - When you load the directory with person entries for the first time, add a default `userpassword` attribute. This attribute allows users to bind to this DN when they use the directory.
3. Decide how the person data is laid out in the directory. The two most popular options are as follows:

   Flat structure
   : puts all of the data in one place in the directory. The benefit is that you do not have to move the entries if users change organizations within the company.

   Organizational unit structure
   : places the entries in a subtree according to the organizational unit within the company. This structure can resemble the company's organization, which allows you to visualize the relationships between entries.
4. Decide on the structure of the distinguished names for your person entries. Although your selection of the attribute for the relative distinguished name is not critical, you must be consistent. Two acceptable choices are common name and user ID. If you use a flat structure for the person data, then use user ID for the DN, because common names are duplicated more often than user IDs.

Security

Sun ONE and Netscape Directory Server Access Control Overview

When LDAP was first developed, it was only a protocol for accessing data in an X500 directory server. Therefore, many specifics of how the server itself was supposed to work were left out. Even when the new standard was written for LDAP version 3 (RFC2251), some important issues were left out simply because the people involved had already picked an implementation, and didn't want a new standard to force them to redesign and reimplement large portions of their servers. Access control was one of those important issues. That is why when discussing access control, it is important to remember that each vendor has a different mechanism, and very little is portable from one server to another.

Access control starts with authentication. There are several mechanisms to accomplish the authentication, but all must eventually resolve to a distinguished name (DN) that exists in the directory. This distinguished name is then used to determine the access that is granted to a user. The process of associating a distinguished name with a user is called binding. A user can bind to a server using the DN and a password, or they can bind anonymously, providing no credentials.

Authentication is accomplished using access control information (ACI) rules. An ACI rule specifies the LDAP object to which the rule applies, whether the rule allows the specified permission or denies it, the users who are permitted or denied access, and what type of permission is being allowed or denied. For information on using the Integration Technologies Administrator to specify ACI rules, see Setting Access Permissions for an Object and Specifying Bind Rules.

The Sun ONE and Netscape mechanism for administering access controls is flexible and powerful, but can also be complex. Some basics may make things clearer.

- By default, no access is allowed to the directory except to the directory manager. The directory manager bypasses all access control checks, and is used to administer the directory. A new directory with no access control information is unreadable by any user except the directory manager. This type of control is different from an explicit deny, which will be discussed later in this document.
- All access control information propagates down from its target to all the children under that target.
- All access control is cumulative.
- You cannot limit the scope of access control information.
- If conflicting access control information exists, deny always overrides allow. In other words, if there are access control lists that allow a user access, and another access control instruction that denies access, the deny will always be preferred.

Sun ONE and Netscape Syntax

```
(target="ldap:///dn")(targetattr="attrname")
   [(targetfilter="rfc2254-style filter")]
   ( version 3.0; acl "name"; (allow | deny)
   (read, write, search, compare, selfwrite, add, delete )
   (userdn | groupdn)="ldap:///dn";)
```

Details about each element in the syntax are as follows:

Target

The target specifies the entry where ACI rule will be effective. Usually, the target is the same as the entry where the ACI attribute exists. In other words, if the ACI attribute is added to the entry with DN `cn=SAS,o=SAS Institute,c=US`, then the target will be `ldap:///cn=SAS,o=SAS Institute,c=US`. The target parameter can point at an entry which is a direct descendant, but that can become hard to manage. The parameter can also cause the server to process access control information that does not apply to the search it is carrying out.

Targetattr

Targetattr specifies one or more attributes that the access control information applies to. Access control rules can apply to specific attributes. This parameter provides more strict access to attributes such as userpassword.

Targetfilter

This is an optional parameter that can be used to apply to specific entries based on a filter. The filter has the same syntax as a filter provided to the ldapsearch command. This type of ACI rule is expensive to process and should be used sparingly.

Permissions

The permissions parameter consists of the version number (currently always 3.0), the ACI name, the operation (either allow or deny), the permissions and the subject to which to apply the rule (typically either a user or a group). Most ACI rules are written to allow permissions, because deny ACI rules can quickly become complex. The following permissions are supported:

Read
: Allows data to be returned to the user

Search
: Allows user operations to search the data

Compare
: The user is allowed to use the data for filter comparisons

Write
: The user may write to the data item

Add
: The user may add the data item

Delete
: The user may delete the data

Selfwrite
: The user may write their own DN to the data item

Some of these permission make the most sense when they are applied to an attribute. For example, allowing selfwrite of the member attribute above the area where groups are stored will allow a user to add himself to or remove himself from a group.

The subject specification will normally be made using the userdn or groupdn keyword. The value will be an ldap URL with the distinguished name of the user or group. Wildcard characters can be used in the DN to specify more than one user. Also, there are two special strings: `ldap:///all` specifies all bound users, and `ldap:///anyone` specifies any users, including anonymous users. If the groupdn keyword is used, the DN points to a groupOfUniqueNames. If the DN of the bound user exists in the group as a uniqueMember attribute, the rule is applied.

There are two other subject specifications that are sometimes used: userdnattr and groupdnattr. These specify that an attribute on the entry will contain a DN, and if the DN matches the bound user, it will apply the rule.

ACI Rule Considerations

Since ACI rules are cumulative, it is important to be careful granting access at a node that has a deep tree under it. For example, if read access is granted to all users at the node `cn=SAS,o=SAS Institute, c=US`, then it becomes difficult to restrict that access further down, such as at the container where logins are stored. Therefore, it is important to look at the whole directory tree before deciding on an access policy.

From an efficiency point of view, the fewer access control instructions the better, as long as the data is secured in a meaningful way. Using groups is a good way to accomplish this, and make access control easier to manage at the same time. It is a lot easier to add or remove a user from a group than try to find all of the ACI rules that reference that user's DN, or figure out all of the different kinds of access a user requires.

Access Control Examples

```
ACI: (target="ldap:///o=SAS Institute,c=US")
  (targetattr=*)( version 3.0; acl
  " allow portal user"; allow (all)
  userdn="ldap:///cn=Portal User,ou=People,
  o=SAS Institute,c=US";)
```

This example allows the user with a DN of `cn=Portal User,ou=People,o=SAS Institute,c=US` all access to everything in the directory. This level of access is unusual, but acceptable for this example because the Portal User identity is used by an application that performs very specific operations in the directory.

```
ACI: (target="ldap:///o=SAS Institute,c=US")
   (targetattr=*)(targetfilter="(| (objectclass=sascontainer)
   (objectclass=sascomponentcontainer)) ")(version 3.0; acl
   "see sascontainers"; allow (compare, read, search)
   userdn="ldap:///all";)
```

This example is somewhat unusual, but it has a specific purpose. It allows all non−anonymous users to see sascontainers and sascomponentcontainers in order to facilitate browsing.

```
ACI: (target="ldap:///cn=sassubscribers,
   sascomponent=saspublishsubscribe,cn=sas,o=sas institute,
   c=us")(targetattr="*")(version 3.0; acl "owner has all
   rights"; allow (all) userdnattr = "saspersondn";)
```

This example is also unusual, because it uses the userdnattr specifier. This ACI rule grants all permissions to a user whose bind DN is found in the saspersondn attribute of an entry below cn=sassubscribers,sascomponent=saspublishsubscribe,cn=sas,o=sas institute,c=US. This is important to allow users to update their own subscriber entries.

```
ACI: (target="ldap:///saschannelcn=Orders for Manufacturing
    Materials,cn=saschannels,sascomponent=sasPublishSubscribe,
    cn=SAS,o=SAS Institute,c=us")(targetattr="*")(version 3.0;
    acl "allow valid users"; allow (compare,add,read,search)
    groupdn = "ldap:///cn=IDBGroup,ou=Groups,o=SAS Institute,
    c=US||ldap:///cn=Sales,ou=Groups,o=SAS Institute,c=US"; )
```

This example allows access to two groups: cn=IDBGroup,ou=Groups,o=SAS Institute,c=US and cn=Sales,ou=Groups,o=SAS Institute,c=US. The permissions allow read, search, compare, and add. The add permission is important because it allows members of these groups to create archive entries under the channel Orders for Manufacturing Materials. If the administrator wished, the add permission could have been moved down to the archivepath entry. This would have restricted who could publish archives to be stored under the channel.

SAS application requirements

There are several places in the SAS hierarchy that do not require any special access control. There are other places, however, that require careful thought when applying the ACI rule. The following are the places in the hierarchy that require special attention.

Cn=SAS

This is the top of the SAS application tree. If some accommodation for browsing is desired, a filtered ACI rule that allows reading sascontainer and sascomponentcontainer by all bound users can be placed here.

Cn=sasSubscribers,sascomponent=sasPublishSubscribe

No sensitive data is contained in this part of the tree, but an ACI rule that allows a user to update their own subscriber information is useful. Using the userdnattr on saspersondn allows this function.

Cn=saschannels,sascomponent=sasPublishSubscribe

The level of control you implement here depends on how you want to secure channels, archive paths and archives. The way publishing works, if a user can read a channel entry, they can publish TO_SUBSCRIBERS, which sends email to all of the users subscribed to a channel. This is allowed even if the publishing user is not subscribed to the channel and therefore cannot write to the channel. This situation can be covered by only allowing read access to specific groups. The other consideration is how archive paths are secured. Allowing add access to an archive path means a user can create archives under that path (assuming they have permission on the physical path). An ACI rule needs to be created to allow users to manage the archives they create, based on the saspublisher attribute. An ACI rule with a userdnattr=saspublisher that allows all is recommended.

Cn=sasArchivePaths,sascomponent=Archiving

This area has the same considerations as the channels. To define the ACI rule, you must decide how you want to protect the global archive paths.

Cn=saslogins,sascomponent=sasserver

This requires careful consideration, because the saslogins below this container contain user names and passwords. The sasreferencedn and sasallowedclientdn attributes contain the distinguished names of the user represented by this login, as well as the DNs of clients who are allowed to use the login. ACI rules should be written to allow read and write access based on these attributes. Using the userdnattr and groupdnattr is appropriate in this case. The groupdnattr is necessary because the sasreferencedn can refer to a group (for a group login).

Cn=sasStoredProcessPaths,sascomponent=sasApplications

It is likely that each storedprocesspath entry will have its own ACI rule set. This is because the stored processes will generate information that will be intended for a certain audience. The stored processes should be grouped under a sasstoredprocesspath according to the group that needs access to them.

Sascomponent=sasPortal

The SAS Information Delivery Portal Installation Guide contains guidelines for setting the access controls on portal entries.

Sascomponent=sasDataSources

This is another location that requires careful consideration. Libraries, tables, infomarts and other data sources may all have individual security requirements. The most important thing to remember is to not place access controls at the container level unless you want that access to apply to all of the entries below it.

Other locations

Other areas can be opened for read by any bound user, but you must make sure you do not put the ACI rule too far up in the tree. For instance, the container cn=sasservers,sascomponent=sasServer can be opened for read by all, but granting that access at sascomponent=sasServer gives access to logins.

Deny ACI Rules

Using deny ACI rules is a useful tool in certain situations, but it can be dangerous. If you want to limit access to a segment of the tree, when higher−level ACI rules have allowed access, you can use deny to accomplish that. Remember, though, that a deny cannot be undone. In other words, if you deny access at a directory entry to all users who are not in a specified group, you cannot then allow a user who is not in that group to access the directory at a lower level in the tree.

You must also remember that an explicit deny is not the same as an implicit allow. If you deny everyone except one group, it does not necessarily mean that everyone in that group is allowed. If no explicit allow was ever specified, the users in that group still do not have access. Deny ACI rules are usually most useful when used with a != operator on the subject, for example:

```
ACI: (target="ldap:///sasUniqueName=Security Group A −
     A000000E.WHSECGRP.A00001X7,cn=sasContentObjects,
     sasmetadatacn=A0000001.WHDW.A000000E,cn=sasMetadata
     Repositories,sascomponent=sasMetadataRepository,
     cn=sas,o=sas institute,c=us") (targetattr="*")
     (version 3.0; acl "Security Policy"; deny (all)
     groupdn != "ldap:///cn=Distributed Technologies,
         ou=groups,o=SAS Institute,c=US||ldap:///cn=IDBGroup,
         ou=groups,o=SAS Institute,c=US" ;)
```

This rule denies access to everything below this entry to everyone that is not a member of the IDBGroup group. The rule to remember is this: do not use deny unless there is no other way to accomplish what you need to do.

Security

Setting Access Permissions for an Object

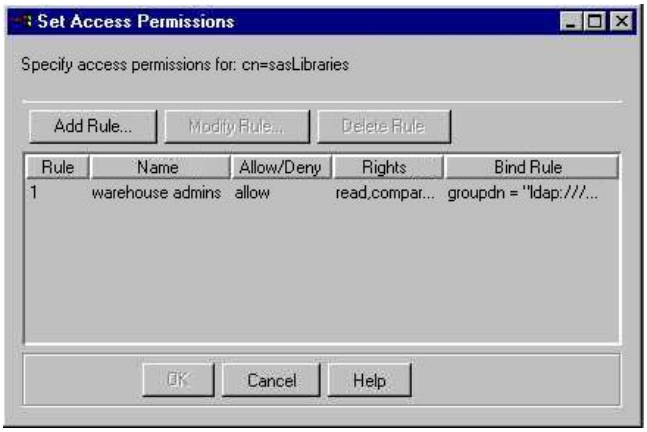
You can use the IT Administrator application to set permissions for objects in directories which reside on Sun ONE and Netscape directory servers. Using these permissions, you can allow or deny access to objects or groups of objects by users or classes of users. A well−planned security strategy allows users to access objects that they need to use (for example, personal subscriptions) while restricting access to sensitive information (for example, a SAS table that contains salary information). See Sun ONE and Netscape Directory Server Access Control Overview for more information on authentication and access control.

The SAS Integration Technologies Administrator provides a graphical user interface that allows you to set permissions for an object in the directory. For general instructions, see Using IT Administrator.

To set permissions for an object in the directory using IT Administrator:

1. Open IT Administrator.
2. In the tree view, select an object or a folder whose permissions you want to set. If you set permissions on a folder, you are also setting permissions for all objects in that folder.
3. Select the **Set Access Permissions** tool on the toolbar. If the tool is grayed out, you cannot set permissions for the selected object.

   When you select the tool, the main Administrator window disappears and the Set Access Permissions window appears.

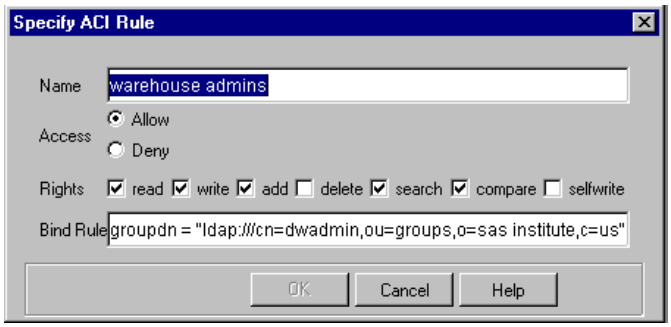


4. The Set Access Permissions window lists all of the existing access rules for the selected object.

   To create a new access rule, select the **Add Rule** button.

   To modify an existing rule, select the rule and then select the **Modify Rule** button.

   To delete a rule, select the rule and select the **Delete Rule** button.
5. If you selected **Add Rule** or **Modify Rule**, the Specify ACI Rule window appears.



Enter or specify the following:

Name
is the name of the rule.

Access
specifies whether the rule is to allow permissions or deny permissions.

Rights
indicates the specific actions that are to be allowed or denied. The rights available are

Right	***Description***
Read	Directory data may be read.
Write	Directory data may be changed, created, or deleted.
Add	Child objects may be created under the specified object.
Delete	The selected object may be deleted.
Search	Directory data may be searched. For example, denying search rights for a user login object prevents users from searching for a particular user login name.
Compare	Directory data may be used for comparisons. Unlike searches, the information is not displayed as a result of the comparison; only an indication as to whether the search was successful is returned.
Selfwrite	Specifies whether users can add or delete themselves from a group.

Bind Rule
specifies the condition that must be met for the rule to take effect. For example, you could specify that the rule be applied if users log on to their own entry in the LDAP directory. See Specifying Bind Rules for details about what information to enter in this field.

Select **OK** to create the rule and close the Specify ACI Rule window.

6. When you finish creating or modifying the access permissions, select **OK** from the Set Access Permissions window.
7. The Set Access Permissions window disappears and the main Administrator window reappears.

NOTE: If any items in the tree view were expanded when you opened the Set Access Permissions window, they are all collapsed when you return to the main Administrator window.

Security

Specifying Bind Rules

The bind rule lets you specify a bind condition under which the access control information (ACI) rule is applied. For example, you could specify that the ACI rule is applied only when a user binds to the directory using their distinguished name (DN).

Note: ACI rules are supported only for the iPlanet (previously Netscape) LDAP server.

Enter the bind rule in the **Bind Rule** field on the Specify ACI Rule window, using one of the following forms:

keyword = expression
: The keyword and expression must match for the statement to be true.

keyword != expression
: The keyword and expression must not match for the statement to be true.

The possible keywords and expressions follow. For detailed information on specifying bind rules, see the *iPlanet Directory Server Administrator's Guide*.

Note: Although bind rules are usually specified as ending with a semicolon, do not put a semicolon on the bind rules in this field. The Administrator application adds the semicolon automatically.

userdn

The expressions that you can use with this keyword are as follows:

userdn = "ldap:///dn"
: Specify a distinguished name or a distinguished name pattern for *dn*. You may use an asterisk as a wildcard. The rule is true if the user binds using the specified distinguished name or pattern. For example, if you specified userdn = "ldap:///uid=*, o=Alphalite Airways" the expression is true if the user binds using uid=jrush, o=Alphalite Airways, but not if the user binds using uid=jrush, ou=sales, o=Alphalite Airways.

userdn = "ldap:///self"
: The rule is true if the user is accessing the entry for the distinguished name that is used when binding to the directory. For example, a user that binds as uid=jrush, o=Alphalite Airways could access the uid=jrush object.

userdn = "ldap:///all"
: The rule is true for any valid distinguished name that has successfully bound to the directory.

userdn = "ldap:///anyone"
: The rule is true for anyone. This rule permits anonymous access to the directory.

userdn = "ldap:///uid=dn || ldap:///uid=dn2"
: The rule is valid if the user binds using either of the specified distinguished names. Wildcards are not allowed.

userdn = "ldap:///o=Alphalite Airways???(ou=sales)(ou=accounting)"
: The rule is valid if the user's distinguished name is under either ou=sales o=Alphalite Airways or ou=accounting o=Alphalite Airways.

groupdn

This keyword uses the following expression:

groupdn = "ldap:///dn"

This rule is true if the bind distinguished name is a member of the group that is specified by *dn*. You can specify more than one group. For example, if the rule is specified as groupdn = "ldap:///cn=managers, o=Alphalite Airways", the rule is true if the user's distinguished name is a member of the managers group.

userdnattr

This keyword uses the following expression:

userdnattr = "ldap:///attribute"
: The rule is true if the bind distinguished name is the same as the distinguished name that is specified for *attribute*. As an example, consider a directory object that has uid=nking specified for the "manager" attribute and a bind rule that is specified as userdnattr = "ldap:///manager" User nking could bind to the directory and access the object because the bind distinguished name matches the value of the "manager" attribute.

groupdnattr

This keyword uses the following expressions:

groupdnattr = "ldap:///attribute"
: The rule is true if the bind distinguished name is the same as the distinguished name that is specified for *attribute*. This operates identically to the userdnattr keyword, except that the attribute is specified on a group object.

groupdnattr = "ldap:///dn?attribute"
: This rule is true if the bind distinguished name is the same as the distinguished name that is specified for *attribute*. The group must also be under the distinguished name that is specified by *dn*.

ip

This keyword uses the following expression:

ip = "ip address"
: The rule is true if the user that is accessing the directory uses the specified IP address. You may use asterisks as wildcards. For example, ip = "10.15.67.*"

dns

This keyword uses the following expression:

dns = "dns hostname"
: The rule is true if the user that is accessing the directory is located in the specified domain. You may use asterisks as wildcards. For example, dns = "*.alphalite.com"

timeofday

This keyword uses the following expression:

timeofday operator "time"
: The rule is true if the time that the user accesses the directory matches the time that is specified in the rule. Specify *time* in 24−hour format (0 to 2359). Use the *operator* value to specify whether the access time should

be before, after, or equal to the time that is specified in *time*. The possible values for *operator* are given in the following examples:

- ◊ timeofday = "800" (rule is true if user logs on at 8:00 AM)
- ◊ timeofday != "1030" (rule is true if user logs on at any time other than 10:30 AM
- ◊ timeofday > "1400" (rule is true if user logs on after 2:00 PM)
- ◊ timeofday > = "1400" (rule is true if user logs on or after 2:00 PM)
- ◊ timeofday < "1100" (rule is true if user logs on before 11:00 AM)
- ◊ timeofday < = "1100" (rule is true if user logs on or before 11:00 AM)

dayofweek

This keyword uses the following expression:

dayofweek = "day"

The rule is true if the user accesses the directory on the specified day (the day is determined on the server). The values for *day* are Sun, Mon, Tue, Wed, Thu, Fri, and Sat.

authmethod

This keyword uses the following expression:

authmethod = "authentication method"

The rule is true if the user accesses the directory using the specified authentication method. Values for *authentication method* are none, ssl, and sasl *sasl_mechanism*. For example, the rule authmethod = "simple" is true if the user accesses the directory using a username and password.

Security

SecureWay Directory Server Access Control Overview

When Lightweight Directory Access Protocol (LDAP) was first developed, it was only a protocol for accessing data in an X500 directory server. Therefore, many specifics of how the server itself was supposed to work were left out. Even when the new standard was written for LDAP version 3 (RFC2251), some important issues were left out simply because the people involved had already picked an implementation, and didn't want a new standard to force them to redesign and reimplement large portions of their servers. Access control was one of those important issues. That is why when discussing access control, it is important to remember that each vendor has a different mechanism, and very little is portable from one server to another.

Access control starts with authentication. There are several mechanisms to accomplish the authentication, but all must eventually resolve to a distinguished name (DN) that exists in the directory. This distinguished name is then used to determine the access that is granted to a user. The process of associating a distinguished name with a user is called binding. A user can bind to a server using the DN and a password, or they can bind anonymously, providing no credentials.

Authentication is accomplished using access control permissions that you set for objects in the IBM Secureway directory. For information on using the Integration Technologies Administrator to specify access control permissions, Setting Access Control for Objects.

SecureWay Syntax

There are four attribute types which determine the access that is allowed on an entry:

Attribute	Definition
AclEntry	A multivalue attribute that describes access to attributes of the associated LDAP object, as well as permissions on the object itself.
AclPropagate	A "true" or "false" flag that indicates if this particular ACL should be propagated down the directory hierarchy.
EntryOwner	The owner of this particular directory object. The entryOwner receives complete access to all attributes of the object.
OwnerPropagate	A "true" or "false" flag that indicates if this owner should be propagated down the directory hierarchy.

These attributes can only be modified by the entry owner, or the directory administrator. There are two other attributes which are maintained by the server, and are not user modifiable, but are available to read for informational purposes:

Attribute	Definition
AclSource	An attribute which identifies the directory object from which the ACL information is inherited.
OwnerSource	An attribute which identified the directory object from which the owner information is inherited.

AclEntry

The aclEntry attribute describes the access granted to the entry object. It describes who has rights (the subject), what rights they have to the object itself, and what rights they have to the attributes of the object. The format of the aclEntry attribute is:
<subject−type>:<subjectDN>:<object−access>:<attribute−access>

Subject−type	one of access−id, group, or role. If access−id, then subjectDN should be the DN of a user entry. If group, subjectDN should be the DN of an AccessGroup entry, and role should point to an AccessRole entry.
SubjectDn	The Dn of the subject for the aclEntry.
Object−access	The access rights granted to the object itself. Valid permissions are a to allow the subject to add children under the entry, and d to allow delete permission. The format for object access is "object:*permissions*".
Attribute−access	Specifies the permissions granted to the entry attributes. There are three levels of attribute access: normal, sensitive, and critical. The security level for an attribute is defined in the schema.

AclPropagate

This attribute will have a value of "true" or "false". The default is "true", and indicates that the aclEntry values for this entry should propagate down the hierarchy to apply to any entries below it which don't have their own aclEntry attribute(s).

EntryOwner

Like the subject clause of the aclEntry, the entryOwner has a subject type and a subjectDN. The subject type can be access−id, group, or role. The subjectDN should be the distinguished name of an entity that represents the correct type of entry (person, accessGroup, or accessRole).
Example: access−id:cn=SAS,o=SAS Institute,c=US

OwnerPropogate

A value of "true" or "false" that determines whether the entryOwner value will propogate to down the hierarchy to apply to entries below it.

ACI Rule Considerations

AclEntry attributes in SecureWay propogate down (assuming aclPropogate is true) to all its subordinates until the aclEntry is overridden. Any aclEntry will override all the values of the previous aclSource. That is to say, if you want to add access by another individual or group at a given point in the tree, while retaining the access controls specified higher up, you must copy the aclEntry attribute values that the entry is already inheriting, and add the new values. Just creating an aclEntry with the new value will revoke the access provided by the previous aclSource.

Access Control Examples

```
AclEntry: access-id:cn=Portal User,ou=People,o=SAS Institute,
c=US:object:ad:normal:rwsc:sensitive:rwsc:critical:rwsc
```

Note: The previous code should be entered as a single line.

This example allows the user with a DN of `cn=Portal User,ou=People,o=SAS Institute,c=US` all access to everything in the directory. This level of access is unusual, but acceptable for this example because the Portal User identity is used by an application that performs very specific operations in the directory.

```
AclEntry: group:cn=Sales,ou=Groups,o=SAS Institute,c=US:object:
a:normal:rsc
```

Note: The previous code should be entered as a single line.

This example allows access to the group: cn=Sales,ou=Groups,o=SAS Institute,c=US. The permissions allow read, search, compare, and add. The add permission is important because it allows members of these groups can create archive entries under the channel Orders for Manufacturing Materials. If the administrator wished, the add permission could have been moved down to the archivepath entry. This would have restricted who could publish archives to be stored under the channel.

SAS application requirements

There are several places in the SAS hierarchy that do not require any special access control. There are other places, however, that require careful thought when applying the ACI rule. The following are the places in the hierarchy that require special attention.

Cn=SAS

This is the top of the SAS application tree. Special permissions for an administrative user, or application identity can go here, if desired.

Cn=sasSubscribers,sascomponent=sasPublishSubscribe

No sensitive data is contained in this part of the tree, so allowing public read access is acceptable. It's also good to provide a way for users to update their own subscriber entries. The only way to do this is to make the personDn the entryOwner for each entry, or put an aclEntry on each subscriber granting write access to the person.

Cn=saschannels,sascomponent=sasPublishSubscribe

The level of control you implement here depends on how you want to secure channels, archive paths and archives. The way publishing works, if a user can read a channel entry, they can publish TO_SUBSCRIBERS, which sends email to all of the users subscribed to a channel. This is allowed even if the publishing user is not subscribed to the channel and therefore cannot write to the channel. This situation can be covered by only allowing read access to specific groups. The other consideration is how archive paths are secured. Allowing add access to an archive path means a user can create archives under that path (assuming they have permission on the physical path).

Cn=sasArchivePaths,sascomponent=Archiving

This area has the same considerations as the channels. To define the ACI rule, you must decide how you want to protect the global archive paths.

Cn=saslogins,sascomponent=sasserver

This requires careful consideration, because the saslogins below this container contain user names and passwords. These attributes are defined as being critical, but the appropriate user needs to be able to read those attributes in order to access their data. The best answer is to make the entryOwner of individual logins the individual they belong to, and for group logins, set the aclEntry to allow access to critical attributes by the appropriate group, and set the owner to a group administrator.

Cn=sasStoredProcessPaths,sascomponent=sasApplications

It is likely that each storedprocesspath entry will have its own ACI rule set. This is because the stored processes will generate information that will be intended for a certain audience. The stored processes should be grouped under a sasstoredprocesspath according to the group that needs access to them.

Sascomponent=sasDataSources

This is another location that requires careful consideration. Libraries, tables, infomarts and other data sources may all have individual security requirements.

Security

Setting Access Control for Objects

The Administrator implements security by letting you set permissions for objects in the IBM Secureway directory. Using these permissions, you can allow or deny access to objects or groups of objects by users or classes of users. A well−planned security strategy allows users to access objects that they need to use (for example, personal subscriptions) while restricting access to sensitive information (for example, a SAS table that contains salary information).

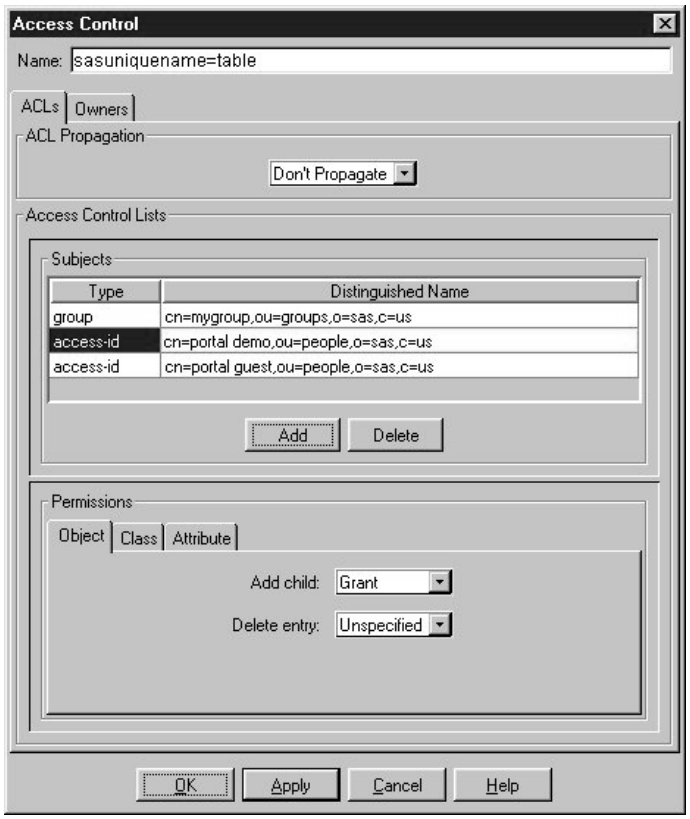
To control access to an object, you must specify the following information:

- The object whose access you want to control
- Whether the access is for the object only, or also for all objects beneath it in the LDAP hierarchy
- The ACL access rule (who has access and what kind of access they have)
- Who is the owner of the object (who has permission to perform any action, regardless of the ACL rule).

To set permissions for an object in the directory, follow these steps:

1. In the tree view, select an object or a folder whose permissions you want to set.
2. Select the **Set Access Permissions** tool on the toolbar. If the tool is grayed out, you cannot set permissions for the selected object.

   When you select the tool, the Access Control window appears.



3. The Access Control window lists all of the existing access rules for the selected object.

   Use the ACL Propagation pull−down menu to specify to what level the ACL rules are applied. Choose

Unspecified
: Inherit rules, clearing any rules explicitly specified for this object.

Propagate
: The rules apply to the object and all objects below it in the LDAP hierarchy.

Don't Propagate
: The rules apply only to the chosen object.

4. To add an access control list, select the **Add** button. A new subject is created with a default permission set. Specify the subject's type and distinguished name. Use the Permissions pane to specify the desired permissions.
5. Under the Subjects pane, click the Type field to display a pull−down menu of the subject types. Select one of the following:

   access−id
   : The rule applies to a specific user or user ID. You should then enter a distinguished name of a specific user, for example, cn=Julieb,o=Alphalite Airways. Specify cn=this specifies the bindDN that matches the object's DN.

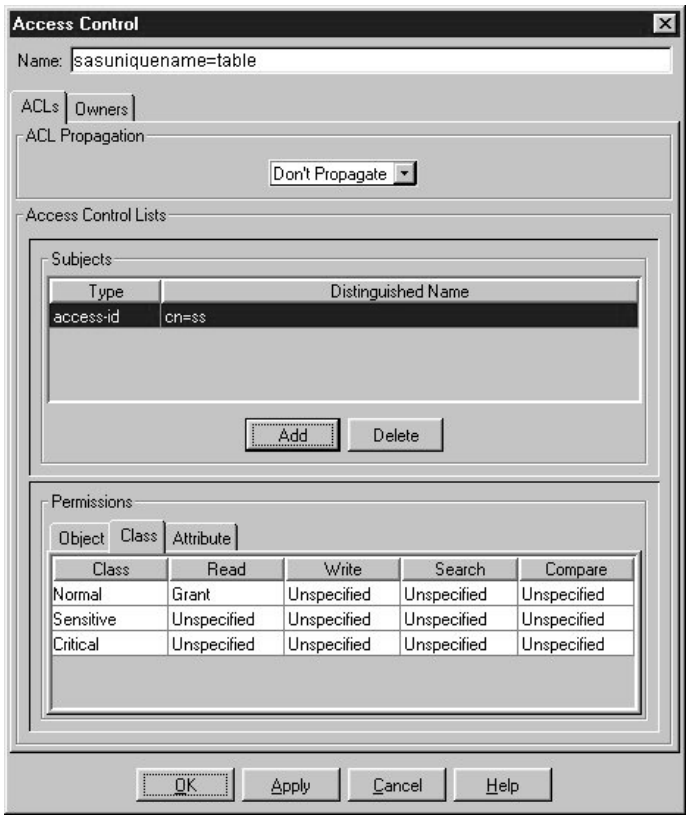
   group
   : The rule applies to a group of users. You should then enter a distinguished name of a group of users, for example, cn=accounting,o=Alphalite Airways. Enter cn=anybody in the Distinguished Name field to specify all users, including unauthenticated users. Enter cn=authenticated to specify any DN that has been authenticated to the directory.

   role
   : The rule applies to a specific user role. You should then enter the distinguished name of a defined access role, for example, cn=administrator,o=Alphalite Airways.
6. Double−click in the Distinguished Name field to enter the DN of the subject as detailed above.
7. To specify the permissions for a subject, select the subject in the Subject pane, then select the tabs in the Permissions pane.

8. Under the Permissions pane, select the Objects tab. Specify the following:

   Add child
   : Specify Unspecified, Grant or Deny. This access control determines whether the subject is allowed to add an object under the current object in the LDAP hierarchy. Choose Unspecified if you want to inherit this setting.

   Delete Entry
   : Specify Unspecified, Grant or Deny. This access control determines whether the subject is allowed to delete the object from the LDAP directory. Choose Unspecified if you want to inherit this setting.
9. To define class level permissions, select the Class tab in the Permissions pane.



Classes are groups of attributes for the object. For example, the userpassword attribute is a member of the Critical class, while the commonName attribute is a member of the Normal class.

10. For each class listed, specify whether the subject is granted or denied read, write, search and compare access for the attributes in the class. Choose Unspecified to inherit the access control setting. Details of the permissions are as follows:

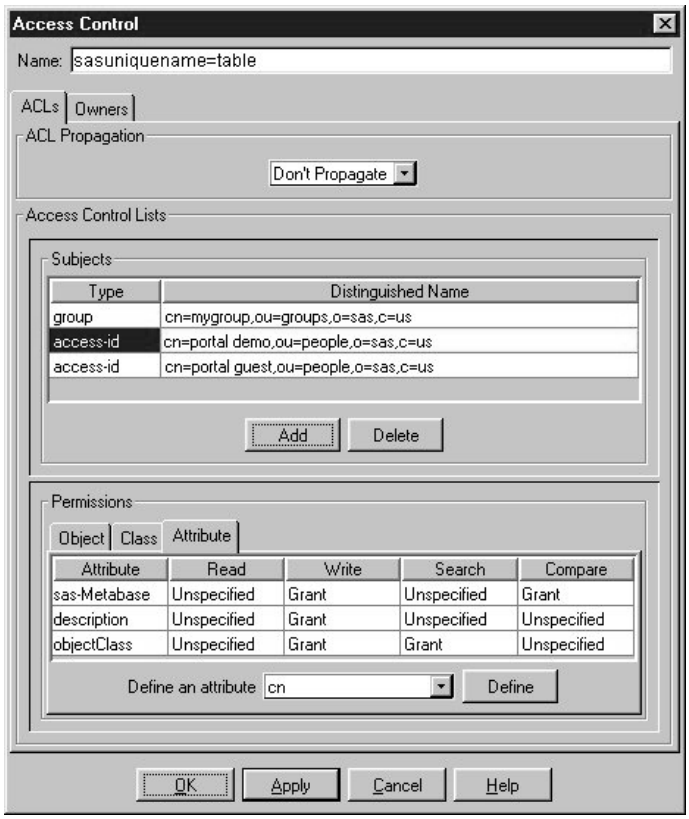
    Read
    : Directory data may be read.

    Write
    : Directory data may be changed, created, or deleted.

    Search
    : Directory data may be searched. For example, denying search rights for a user login object prevents users from searching for a particular user login name.

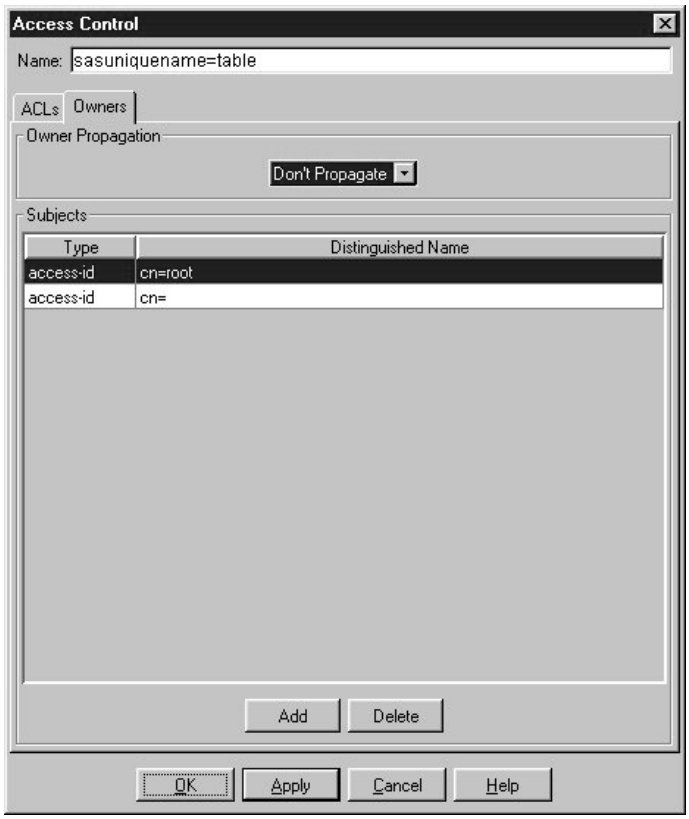
    Compare
    : Directory data may be used for comparisons. Unlike searches, the information is not displayed as a result of the comparison; only an indication as to whether the search was successful is returned.

11. To define permissions for specific attributes, select the Attributes tab in the Permissions pane.



Select an attribute from the Define an Attribute pull down menu and select Define. Select whether the subject should be granted or denied read, write, search and compare access for the specified attribute. Choose Unspecified to inherit the access control setting.

12. To define the owner of the object, select the Owners tab.



Specify the following:

Owner Propagation
: Use the pull−down menu to specify whether the owner of the object is also the owner of all objects under it in the LDAP hierarchy. Select Propagate to specify that the owner also owns the other objects; select Don't Propagate to specify that the owner only owns the current object. Select Unspecified to inherit this access control setting.

Subjects
: Specify who has full access to the object regardless of the ACL rules. To add an owner, select Add. Use the Type pull−down menu to specify whether the owner is a specific user (access−id), a group of users (group), or a type of user (role). Enter the owner's distinguished name in the Distinguished Name field.

13. Select **Apply** to apply the defined access controls and leave the Access Control window open. Select **OK** to apply the access control and close the Access Control window.

    NOTE: If any items in the tree view were expanded when you opened the Access Controls window, they are all collapsed when you return to the main Administrator window.