Server Administration Guide for DirectConnect

Enterprise Connect Data Access and Mainframe Connect

12.6

[Microsoft Windows and UNIX]

DOCUMENT ID: DC35365-01-1260-01

LAST REVISED: February 2005

Sybase, Inc., One Sybase Drive, Dublin, CA 94568.

Contents

About This Book

This book describes how to use a Sybase® DirectConnect™ server, including how to start and stop the server, configure server properties, set up service name redirection, and troubleshoot server operations using log and trace files.

Audience

This book is written for:

- Application Programmers, who develop organization-specific programs using the major features of DirectConnect.
- System Administrators, who install and test DirectConnect. When DirectConnect is running, System Administrators provide ongoing administration support, disaster recovery, and troubleshooting support.
- System Programmers, who install and test DirectConnect. System Programmers also provide product administration, troubleshooting, and disaster recovery.

How to use this book

This book contains the following topics:

- Chapter 1, "Introduction" introduces the DirectConnect server and describes the server external files.
- Chapter 2, "Creating a Server" describes how to use the new server command.
- Chapter 3, "Starting and Stopping the Server and Access Services" provides the command line syntax necessary to start the server and the basic procedures for stopping it.
- Chapter 4, "Configuring the Server" describes how to configure DirectConnect Server properties.
- Chapter 5, "Setting up SSL and a Windows Service" describes how to set up Secure Sockets Layer (SSL) on a DirectConnect server, and how to set up and run DirectConnect as a Windows service.
- Chapter 6, "Using Service Name Redirection" explains how service name redirection works and provides detailed examples of redirection files.

- Chapter 7, “Log and Trace Files” explains the logging and tracing processes and provides examples of each.
- Chapter 8, “Managing Server Security with DirectConnect Manager” describes how to manage security using DirectConnect Manager.
- The glossary provides definitions of technical terms used in this book.

Related documents

To install and set up connectivity for ECDA products, see the Enterprise Connect Data Access *Installation Guide* for your platform.

To configure and administer DirectConnect products, use the following guides:

- The appropriate Enterprise Connect Data Access *User’s Guide for Access Service’s* for your database system
- Mainframe Connect for z/OS option *User’s Guide for Transaction Router Service*, if applicable

To configure and administer DirectConnect Manager, see the online help feature that is available when you install DirectConnect Manager.

For additional references, use the following documents:

- Open Client™ *Client-Library/C Reference Manual*
- Open Client *Client-DB-Library/C Reference Manual*
- Open Server™ *Server-Library/C Reference Manual*
- Open Server and Software Developer’s Kit *Installation Guide* For Microsoft Windows or UNIX
- Open ClientConnect and Open ServerConnect *Messages and Codes*
- Adaptive Server® Enterprise *Installation Guide* for HP-UX
- Adaptive Server Enterprise *Installation Guide* for IBM RS6000
- Adaptive Server Enterprise *Installation Guide* for Microsoft Windows
- Adaptive Server *Enterprise Reference Manual*

Other sources of information

Use the Sybase Getting Started CD, the Sybase Technical Library CD, and the Technical Library Product Manuals Web site to learn more about your product:

- The Getting Started CD contains release bulletins and installation guides in PDF format, and may also contain other documents or updated information not included on the Technical Library CD. It is included with your software. To read or print documents on the Getting Started CD, you need Adobe Acrobat Reader (downloadable at no charge from the Adobe Web site, using a link provided on the CD).
- The Technical Library CD contains product manuals and is included with your software. The DynaText reader (included on the Technical Library CD) allows you to access technical information about your product in an easy-to-use format.

  Refer to the *Technical Library Installation Guide* in your documentation package for instructions on installing and starting the Technical Library.
- The Technical Library Product Manuals Web site is an HTML version of the Technical Library CD that you can access using a standard Web browser. In addition to product manuals, you will find links to EBFs/Maintenance, Technical Documents, Case Management, Solved Cases, newsgroups, and the Sybase Developer Network.

  To access the Technical Library Product Manuals Web site, go to Product Manuals at http://www.sybase.com/support/manuals/.

Sybase certifications on the Web

Technical documentation at the Sybase Web site is updated frequently.

❖ Finding the latest information on product certifications

1. Point your Web browser to Technical Documents at http://www.sybase.com/support/techdocs/.
2. Select Products from the navigation bar on the left.
3. Select a product name from the product list and click Go.
4. Select the Certification Report filter, specify a time frame, and click Go.
5. Click a Certification Report title to display the report.

❖ Creating a personalized view of the Sybase Web site (including support pages)

Set up a MySybase profile. MySybase is a free service that allows you to create a personalized view of Sybase Web pages.

1. Point your Web browser to Technical Documents at http://www.sybase.com/support/techdocs/.
2. Click MySybase and create a MySybase profile.

Sybase EBFs and software maintenance

❖ Finding the latest information on EBFs and software maintenance

1. Point your Web browser to the Sybase Support Page at http://www.sybase.com/support.
2. Select EBFs/Maintenance. If prompted, enter your MySybase user name and password.
3. Select a product.
4. Specify a time frame and click Go. A list of EBF/Maintenance releases is displayed.

   Padlock icons indicate that you do not have download authorization for certain EBF/Maintenance releases because you are not registered as a Technical Support Contact. If you have not registered, but have valid information provided by your Sybase representative or through your support contract, click Edit Roles to add the “Technical Support Contact” role to your MySybase profile.
5. Click the Info icon to display the EBF/Maintenance report, or click the product description to download the software.

Style conventions

The following table explains the style conventions used in this book.

Table 1: Style conventions

Name	Example
Files, directories	*econnect\ServerName\cfg*
Programs, utilities, procedures, commands	the set statement
Properties	Allocate
Options	connect
Code examples, text on screens	`** Prepare the statement`
Variables in command line displays (integer, in this example)	`ClientIdleTimeout=`*integer*
Syntax statements that display options for a command	sp_columns *table_name* [, *table_owner*] [, *table_qualifier*] [, *column_name*]

Syntax conventions

The following table explains the syntax conventions used in this book.

Table 2: Syntax conventions

Symbol	Convention
( )	Include parentheses as part of the command.
{ }	Braces indicate that you must choose at least one of the enclosed options. Do not type the braces when you type the option.
[ ]	Brackets indicate that you can choose one or more of the enclosed options, or none. Do not type the brackets when you type the options.
|	The vertical bar indicates that you can select only one of the options shown. Do not type the bar in your command.
,	The comma indicates that you can choose one or more of the options shown. Separate each choice by using a comma as part of the command.

If you need help

Each Sybase installation that has purchased a support contract has one or more designated people who are authorized to contact Sybase Technical Support. If you cannot resolve a problem using the manuals or online help, please have the designated person contact Sybase Technical Support or the Sybase subsidiary in your area.

CHAPTER 1 Introduction

This chapter introduces basic DirectConnect concepts and describes the DirectConnect server external files.

This chapter contains the following topics:

DirectConnect overview

DirectConnect is a Sybase product that provides basic connectivity to non-Sybase data sources. Specifically, it provides access management, copy management, and remote systems management.

DirectConnect consists of:

- DirectConnect Enterprise server
- DirectConnect service libraries
- DirectConnect access services

The following sections describe the components.

DirectConnect Enterprise server

The DirectConnect Enterprise server provides management and support functions for DirectConnect Enterprise service libraries, such as:

- Routing client connections to the appropriate access service based on user ID, requesting application, and access service name
- Providing a single log file and a trace file for access services
- Logging server, access service, and client messages

- Tracing server, access service, and client events
- Providing configuration management of all installed services

The server is designed to operate with a variety of related Sybase software products, including:

- Open Client
- Custom Open Client applications
- MainframeConnect™
- Enterprise Application Server (EAServer)
- Adaptive Server Enterprise
- Adaptive Server Enterprise/Component Integration Services (ASE/CIS)
- Replication Server®
- DirectConnect Manager
- jConnect™for JDBC
- Open ServerConnect

For more information about how each of these Sybase products operates, see the appropriate documentation.

DirectConnect service libraries

Residing on the DirectConnect server, a service library is a set of configuration properties that describes how all of its access services will function. The following service libraries reside on the DirectConnect server:

- Access Service Library
- Administrative Service Library

Access Service Library

An access service is the client connection point for a DirectConnect server. You can think of it as the pairing of an access service library with a set of specific values for the configuration properties. You must define at least one access service for every Access Service Library.

Access services allow clients to access data from a specific target. Each access service is a specific set of configuration properties that:

- Transforms SQL
- Converts datatypes
- Transfers data between the target database and other servers accessible through Open Client™
- Supports Catalog Stored Procedures (CSPs) and system stored procedures
- Supports remote stored procedures (RSPs) and host-resident requests

Administrative Service Library

The Administrative Service Library provides specific administrative services for all DirectConnect libraries, including writing to logs and allowing remote configuration of DirectConnect access services through DirectConnect Manager.

DirectConnect access services

An access service is the named pairing of an access service library and a set of specific configuration properties. Clients connect to access services by their access service name. Within each access service library, a collection of configuration sets contain properties that define how each access service behaves.

The following rules define the relationship between an access service library and an access service:

- A single Access Service Library can support multiple access services.
- Each access service has one configuration set and is always associated with a specific access service library.
- All access services within a given access service library share the same configuration properties.
- You can assign different configuration property values to different access services within an access service library.

How DirectConnect routes access service requests

The DirectConnect server routes each client request for an access service to the appropriate access service library. The routing process can take one of two forms:

- With direct resolution, you specify the exact name of the access service. If the access service is defined correctly, the DirectConnect server matches the request with the access service.
- With access service name redirection, you can map your access service connections to allow client requests to be routed to assigned access services based upon user profiles. This feature allows you to centrally manage client access to access services.

For information on access service name redirection and examples of how it works, see Chapter 6, “Using Service Name Redirection.”

Configuring DirectConnect properties

You can configure DirectConnect properties on the server level, the access service library level, or on an individual access service level. To help you do this, DirectConnect configuration properties are grouped as follows:

- Server configuration files, which consist of the properties that manage a particular DirectConnect server.
- Access Service Library configuration files, which consist of general library configuration values and configuration sets for all access services associated with a particular access service library.
- Access service configuration properties that define a particular access service are stored in the access service library configuration file.

When you install a DirectConnect server, the default configurations allow the server to run. For each access service you create within each server, you must provide additional configuration properties that define the connectivity to your target database system. For information on configuring the DirectConnect server, see Chapter 4, “Configuring the Server.”

You can set access services to be enabled at start-up through a configuration setting. If this value is set to no, then you need to manually enable the access service before it can be used. For information on configuring access service libraries and access services, including instructions on creating new access services, see the appropriate ECDA Options *User’s Guide for Access Services* for your database system.

You can configure properties using DirectConnect Manager or a text editor. Sybase recommends using DirectConnect Manager for the following reasons:

- Changes that you make with a text editor do not take effect until you restart the server. However, most changes that you make with DirectConnect Manager can be made to take effect immediately.
- You can use DirectConnect Manager as a guide to the properties that can be changed, as well as the valid values for each property.

The following section provides more information about DirectConnect Manager.

DirectConnect Manager

DirectConnect Manager is a Sybase Central plug-in application that allows you to configure and manage Sybase DirectConnect servers interactively from a client machine. DirectConnect Manager connects to a DirectConnect server and communicates with it through an application programming interface (API) rather than by manipulating the configuration files directly.

You can perform the following tasks with DirectConnect Manager:

- Perform all of its management functions remotely: you do not need physical access to the DirectConnect server machine or directory.
- Provide management services to multiple servers at the same time, including the ability to copy access service configurations from one server to another.
- Change configuration properties of DirectConnect servers, access service libraries, and access services.
- Create an access service by copying an existing access service and giving it a unique name.
- Create new servers using DCDirector.
- Start and stop existing servers using DCDirector.
- Start, stop, and delete access services. (From a remote site, using DirectConnect Manager is the only way you can start an access service.)
- Test the availability of a data source by creating a connection to it.
- Retrieve a DirectConnect server log file or a subset of the log, and view log file messages.
- View the status of an access service and data source on the desktop.

DirectConnect Manager graphically represents each DirectConnect object on a tree list or an “icon map,” a customizable workspace where you can add or remove objects. When you add a DirectConnect server to DirectConnect Manager, its server name, access service library, and any access services appear on the tree list or the icon map.

Using DirectConnect Manager

DirectConnect Manager graphically represents each DirectConnect object on a tree list or an “icon map,” a customizable workspace where you can add or remove objects. When you add a DirectConnect server to DirectConnect Manager, its server name, access service library, and any access services appear on the tree list or the icon map.

DirectConnect Manager communicates with DirectConnect servers asynchronously, which means you can continue to use DirectConnect Manager while a command is being processed.

You can configure properties using DirectConnect Manager or a text editor. Sybase recommends using DirectConnect Manager for the following reasons:

- Changes that you make with a text editor do not take effect until you restart the server.
- Most changes that you make with DirectConnect Manager can be made to take effect immediately.
- You can use DirectConnect Manager as a guide to the properties that can be changed, as well as the valid values for each property.
- DirectConnect Manager can perform all of its management functions remotely. With DirectConnect Manager, you do not need physical access to the DirectConnect server machine or directory.
- DirectConnect Manager provides management services to multiple servers at the same time, including the ability to copy access service configurations from one server to another.

For more information about DirectConnect Manager features, use the DirectConnect Help available under the online Help menu option.

Server external files

The DirectConnect server manages external files that reside in various subdirectories. For information on the DirectConnect directory structure for your installation, see the appropriate *Installation Guide* for your database system and platform.

Following are brief descriptions of the server-managed external files:

License file

The license file contains licensing information entered by the client for the products and features that are being used. This site-specific file contains descriptions of server nodes that can run the license daemons, various vendor daemons, and licenses for the features and the supported products.

Log file

The log file is an active log file that contains operational information that you can use to correct problems. Although the file is maintained in U.S. English, any logged client messages appear in the client language. The log file resides in the server *log* subdirectory. For logging information, see Chapter 7, “Log and Trace Files.”

Server configuration file

The server configuration file *server.cfg* contains all server configuration information. It resides in the server *cfg* subdirectory. For more information on server configuration, refer to Chapter 4, “Configuring the Server”.

Access Service Library files

A dynamically loaded shared library that represents each access service library. The DirectConnect server identifies the library by the file name. To install, load, or access a library, verify that the executable file for that library exists in the server *$SYBASE_ECON/svclib* subdirectory for UNIX, or the *%SYBASE_ECON%\svclib* subdirectory for Windows.

Access Service Library configuration file

This file contains information for the access service library and all of its access services. Each access service library has a configuration collection. The server defines the file format, but each configuration property is defined by the access service library, regardless of whether the property is managed at the access service library or the access service level. The configuration files reside in the server *cfg* subdirectory.

For information on configuring access service library properties, see the appropriate DirectConnect *Access Service Guide* for your database system.

Service name redirection file

This optional file contains all information necessary to redirect incoming requests for access service names to other access services. The file resides in the server *cfg* subdirectory. For information on access service name redirection, see Chapter 6, “Using Service Name Redirection.”

Trace file

This file is the only active trace file for the system and it provides debugging information for Sybase Product Support Engineers and Technical Support personnel. You can turn it on and off through server configuration. Although the trace file is maintained in U.S. English, any logged client messages appear in the client language. The trace file resides in the *log* subdirectory. For tracing information, see Chapter 7, “Log and Trace Files.”

CHAPTER 2 Creating a Server

This chapter describes how to create a DirectConnect server and covers the following topics:

Types of DirectConnect servers

There are three kinds of DirectConnect servers:

- DCDirector server—a server that performs an administrative role in DirectConnect Manager over other servers that you associate with it.
- Directed server—a server associated with DCDirector that you *can* fully administer remotely through DirectConnect Manager.
- Non-directed server—a server not associated with DCDirector, and thus one that you *cannot* fully administer remotely through DirectConnect Manager.

Whether a server is directed or non-directed, there is no impact on your applications. Both types of servers operate the same and require no specific connection changes to client applications. However, the DCDirector and its associated servers all must reside on the same machine.

The following table describes which functions you can administer to directed and non-directed servers using DirectConnect Manager.

DirectConnect Manager function	Supported for directed servers	Supported for non-directed servers
Create a server	Yes	No
Start a server	Yes	No
Stop a server	Yes	No
Add a server connection	Yes	Yes
Remove a server connection	Yes	Yes
Manage server log file	Yes	Yes

Note DirectConnect for Oracle supports only non-directed servers.

DCDirector server

DCDirector server is used by DirectConnect Manager and allows you to remotely administer DirectConnect servers. It also gives you a logical view of a group of servers. The DCDirector server can be viewed as a container that contains servers. To designate a server as a DCDirector server, a property IsDCDirector which is defined in Chapter 4, “Configuring the Server.”

Only one DirectConnect server on a machine can be designated as a DCDirector server, and when it is selected, only an *admin* library is loaded. A DCDirector performs the sole function of creating, starting, and stopping other servers in the same directory.

Hierarchical tree structure

The following hierarchy tree identifies a typical structure, the server name, machine name, and the port number:

Current hierarchy:

DirectConnect

- server name1 (machine, 4113)
- server name2 (machine, 4114)

New hierarchy:

DirectConnect

- Director name (machineA, 4199)

    - server name1 (machineA, 4113)
    - server name2 (machineA, 4114)
- Director name (machineB, 4199)
    - server name1 (machineB, 4113)
    - server name2 (machineB, 4114)
- server name3 (machineC, 4113)
- server name4 (machineD, 4155)

DirectConnect servers

To continue to manage DirectConnect servers that are not using a Director, DirectConnect Manager provides the ability to connect directly to the server. This allows DirectConnect Manager to connect to any DirectConnect servers that are *not* under the control of a DCDirector.

Differences between a DCDirector server and a non-directed servers

Following are some of the concepts and differences between a DCDirector server and a non-directed server:

- The DCDirector server designation is only a conceptual change in the way that DirectConnect Manager views the servers and can only be used with DirectConnect Manager.
- The DCDirector concept gives DirectConnect Manager some additional administrative functionality.
- You are not required to create or use a DCDirector, but it can give DirectConnect Manager a level of control over the servers.
- Because the DCDirector servers and non-directed servers operate in the same manner, nothing needs to be changed in terms of connecting and using these servers by a client application.

Creating a DirectConnect server

You can create a DirectConnect server by using DirectConnect Manager or by using the AddServer utility.

Warning! You cannot use DirectConnect Manager to create a non-directed sever; you must use the AddServer utility.

Using DirectConnect Manager allows you to create a DCDirector server to fit into the new hierarchy.

Using DirectConnect Manager

When you use DirectConnect Manager to create a new DCDirector server, it performs the following tasks:

- Creates a new server directory, including the *cfg* and *log* directories.
- Populates the *cfg* directory with required configuration files for the server and the *admin* service, and for the Service Name Redirection (*snrf)* table. It also creates access service *.cfg* files.

For instructions on how to use DirectConnect Manager to create a Director server, go to the Server Administration topic of DirectConnect Manager online Help and select "Creating a DCDirector server."

Note Before you can use DirectConnect Manager, you must have installed DirectConnect Manager as outlined in the Installation Guide for your platform, and you must also identify and establish a connection between the server and DirectConnect Manager. This is described in DirectConnect Manager online Help topic, "Connecting DirectConnect Manager to a DirectConnect Server."

Using DirectConnect utilities

To simplify the execution of DirectConnect on multiple platforms, Sybase provides utilities to create and start a new DCDirector server or a non-directed server.

These utilities are c shell scripts (on UNIX) and batch files (on Windows) that can be found and kept in the *$SYBASE/$SYBASE_ECON/bin* (UNIX) or *%SYBASE%\%SYBASE_ECON%\bin* (Window) directory. To run properly, the scripts must be kept in their original directory. It is from this directory that the utilities can find the paths to the other files they need to perform their tasks.

The following describes the utilities:

- AddServer—to create a new server
- DCDirector—create and start a DCDirecor server

For more information about DirectConnect utilities, see the Appendix in the Mainframe Connect DirectConnect for z/OS Option *Installation Guide* version 12.6, or the ECDA *Installation Guides* version 12.6, for your platform.

Creating a server using the *AddServer* utility

Creating a new DirectConnect server is not difficult, but often entries are forgotten in the *interfaces* (UNIX) or *sql.ini* (Windows) file. AddServer is a utility that creates the necessary entries in the *interfaces* or *sql.ini* file before starting the DirectConnect server. AddServer requires two parameters:

- The name of the new server
- The port number for the server to listen on

One important limitation of AddServer is that it does not check the *interfaces* or *sql.ini* file for duplicate server names or ports in the same way that DCDirector will. For more Information regarding the AddServer utility, refer to Appendix B of the DirectConnect installation guide for your platform.

Creating and starting a DCDirector server using the *DCDirector* utility

To create and start a new DCDirector for an installation, use the DCDirector script (on UNIX) or batch file (on Windows).

To designate the new server as a DCDirector server, go to *%Sybase%\DC_12.6\bin* (Windows) or *$SYBASE/DC-12_6/bin* (UNIX) and enter:

```
DCDirector
```

The DCDirector will have the new server name, the machine name of where the server is installed, and a port number equal to 7711.

For more information about DirectConnect utilities, see the appendix in the Mainframe Connect DirectConnect for z/OS Option *Installation Guide* version 12.6, and the ECDA *Installation Guides* version 12.6, for your platform.

How DirectConnect creates new configuration files

When you create a new server, the server configuration files are not created until you start the new server. In addition, the new server is configured for a snrf.tbl table (see Chapter 6, “Using Service Name Redirection”), and the table doe not exist, the server creates one and populates it with the following “* * Service A.” However, the access service name redirection functionality will not work until you edit it with valid information.

For more information about server and access service configuration error conditions, see Chapter 4, “Configuring the Server.”

CHAPTER 3

Starting and Stopping the Server and Access Services

This chapter describes the most common ways to start and stop a DirectConnect server and its access services.

This chapter contains the following topics:

Starting the server

You can start a DirectConnect server using DirectConnect Manager, the DCStart utility, or the DCDirector utility (for DCDirector servers only).

Note For starting a server, Sybase recommends using the new DCDirector server through DirectConnect Manager.

You can use the default configuration setup for the DirectConnect server and for each DirectConnect Access Service Library.

Using DirectConnect Manager

For instructions on how to use DirectConnect Manager to start a server through DCDirector, go to DirectConnect Manager online Help and select Server Administration | Starting a Server.

Note Before you can use DirectConnect Manager, you must have installed DirectConnect Manager as outlined in the installation guide for your platform, and you must also identify and establish a connection between the server and DirectConnect Manager. This is described in the DirectConnect Manager online Help topic, “Connecting to DirectConnect Manager to a DirectConnect Server.”

Using the *DCStart* utility

This utility is similar to using the direct executable. DCStart will automatically “source” the appropriate *DC_SYBASE.csh* (UNIX) file or run the appropriate *DC_SYBASE.bat* (Windows) file to ensure that all the appropriate Sybase-specific environment variables are set properly.

Creating and starting a DCDirector server using the *DCDirector* utility

To create and start a new DCDirector for an installation, use the DCDirector script (on UNIX) or batch file (on Windows) to create and start the DCDirector server.

To designate the new server as a DCDirector server, go to *%Sybase%\DC_12.6\bin* (Windows) or *$SYBASE/DC-12_6/bin* (UNIX) and enter:

```
DCDirector
```

The DCDirector will have the new server name, the machine name of where the server is installed, and a port number equal to 7711.

For more information about DirectConnect utilities, see the Appendix in the Mainframe Connect DirectConnect for z/OS Option *Installation Guide* version 12.6, or the ECDA Options *Installation Guides* version 12.6, for your platform.

Stopping the server

You can stop a server using DirectConnect Manager, the command line, or other platform-specific procedures.

Note To stop a server that is associated with a DCDirector server, Sybase recommends using DirectConnect Manager.

Using DirectConnect Manager

For instructions on how to use DirectConnect Manager to stop a server through DCDirector, go to the Server Administration topic of DirectConnect Manager online Help and select “Stopping a server.”

Warning! You can stop a server using DCManager *only* if it is associated with a DCDirector server. To stop a non-directed server, you must use the command line.

Using command line syntax

As an alternative to using DirectConnect Manager to stop the server, you can use the stopsrvr utility that shuts down the server and terminates all client connections. A password is not required. This will work only if the sa user password was not modified.

Note If you invoke stopsrvr when a client is performing work such as batch processing, the utility fails to stop the server, and both the client and the server will suspend operations.

The stopsrvr format is as follows:

```
stopsrvr [-v|-h] -Sserver_name [-ddelay]
```

where:

- -v displays the program version only.
- -h displays the stopsrvr format.

- -S shows the name of the server to be shut down.
- -d is the delay, in seconds, before client connections are terminated. The default is 3.

Platform-specific procedures

Procedures for stopping the server vary by platform, as described in the following sections.

Stopping the server from Windows systems

You have two options to stop the server on Windows:

- If you started the server from a command line, press Ctrl + C to stop the server.
- If you started the server as a Windows service, perform the following steps to stop it:

  a From the Control Panel, select Services.

  b In the drop-down list, select the DirectConnect *server_name*.

  c Select Stop.

Stopping the server from UNIX systems

❖ **To stop the server**

1. Make the DirectConnect console window the active window.
2. Press Ctrl + C.

Starting an access service

You can start an access service either through DirectConnect Manager or automatically when the server is started.

Starting an access service using DirectConnect Manager

For instructions on how to use DirectConnect Manager to start a service, go to the Managing Access Services topic of DirectConnect Manager online Help and select “Starting a Service.”

Starting an access service at start-up

Because DirectConnect access service libraries operate within the framework of the DirectConnect server, you must start the server to enable an access service.

❖ **To enable an access service at start-up**

1 Set the access service configuration property EnableAtStartup to yes.

2 Start the server.

For information about configuring access service properties, see the appropriate ECDA Options *User’s Guide* for *Access Services* or the Mainframe Connect DirectConnect for z/OS Option *User’s Guide for DB2 Access Services*.

Stopping an access service

The only way to stop an access service is through DirectConnect Manager. For instructions, go to the Managing Access Services topic of DirectConnect Manager online Help and select “Stopping a service.”

Understanding process exit codes

If the server terminates normally, it returns an exit code of 0 (zero) to the operating system. If a start-up error occurs, the server returns one of the following exit codes.

Table 3-1: Description of process exit codes

Code	Description
1	A command line syntax error occurred.
2	The server class constructor failed.
3	The Open Server srv_run function failed.
4	The srv_start event handler failed.
5	Out of memory.

Resolving errors during start-up

If the DirectConnect server encounters server or access service library configuration errors during start-up, it stops. If the server encounters errors in an access service configuration, start-up continues.

Also, if the server is configured for a snrf.tbl table and the snrf table doesn't exist, the server creates one and populates it with the following "* * Service A." However, the access service name redirection functionality does not work until you edit the snrf table with valid information. The following sections describe the errors you might encounter.

Server configuration error conditions

If a server configuration error occurs, you receive an error message that lists the type of error and the line number where the error occurred.

❖ **To display the error messages from start-up**

1 Use a text editor to open the log file or the trace file, as applicable.

2 Search for the LogNotice or TraceNotice sections.

If you use the log parse utility, set it to look for LogNotice or TraceNotice.

Note The log parse utility is a command line program that lets you extract specific information from multiple log and trace records. For instructions on using this utility, see the *README* file located in the *DC-12_6* subdirectory.

Typical start-up errors include the following:

- Configuration file is corrupt.

- Required section name is missing.
- Configuration property value is invalid.

For a list of server error messages, explanations, and recommended actions, see the DirectConnect *Error Message Guide*.

Access service configuration error conditions

Typical access service configuration errors are as follows:

- The client user ID and password are used to log in to the database system. If the access service is unable to log in because of an invalid ID or password, it disconnects from the database system and sends an error to the client application.
- Each DirectConnect access service must have a unique access service name. If the server encounters a duplicate access service name during start-up, it logs a warning message that the duplicate access service is being ignored, and start-up continues.
- If the DirectConnect server detects that a required property is absent or incorrect for a particular service, then it does not enable that access service. The affected access service cannot be enabled until the access service configuration property is edited and is valid.

For instructions on using DirectConnect Manager, start DirectConnect Manager and select from the list of topics in the online help.

For a list of access service error messages, explanations, and recommended actions, see the DirectConnect *Error Message Guide*.

CHAPTER 4 Configuring the Server

This chapter describes how to edit the DirectConnect server configuration properties. The properties cover a variety of administrative and start-up functions.

This chapter contains the following topics:

Note You must configure the server before you attempt to connect it to your backend database.

Editing the configuration file

Note It is feasible to configure your DirectConnect Server using a text editor, but Sybase recommends using DirectConnect Manager to change the properties interactively.

As the System Administrator, you control DirectConnect server configuration through the server configuration file, *server.cfg*, a data group that defines individual properties. You can use DirectConnect Manager or a standard text editor to set, change, add, or delete property values in the configuration file.

Because each server configuration property already has a default value, you do not need to specify a property or a value for that property in the server configuration file unless you want to modify the default. A blank configuration file means that default values are in use for all properties that affect the server process. You add an entry to the configuration file *only* when you change the property value to something other than the default. Descriptions of the configuration properties start on page 26.

Note If you change a value, you must correctly specify the new one to start the server. Be sure that any change you make to a property default value is within the range indicated for that property. Failure to do so will result in an error condition.

Configuration file format

The configuration file format uses the following guidelines:

- A primary section name in brackets identifies the managed object being configured (in this case, the DirectConnect server).
- Subsections in braces identify configuration categories to group configuration properties. The server configuration categories are:
    - Client Interaction
    - Logging
    - Tracing

  Note When you change a server configuration property value, place the property value under the correct category name. If the category is not already shown in the file, you must add it.

- You can include comments. Each comment should be on a separate line that begins with a semicolon or the crosshatch character (#).

Following is an example of a server configuration file that contains client interaction and logging properties:

```
# This is a header comment.
 [Server]
 ; This comment is on line 3.
 {Client Interaction}
 MaxConnections=5
 ServiceRedirectionFile=snrf.tbl
 RemoteSites=3
 {Logging}
 LogToScreen=no
 LogWrap=yes
 LogClientLogin=yes
 LogClientMessages=19
 LogOCOSMessages=19
```

Using DirectConnect Manager

Using DirectConnect Manager, you can edit the Client Interaction properties, the Logging properties, and the Tracing properties without having to restart the server.

For instructions on how to use DirectConnect Manager to change the server properties, go to the Managing Server Configuration topic of DirectConnect Manager online Help and select “Modifying server configuration properties.”

Using the text editor

❖ **To edit or change the server configuration property values using the text editor**

1 Open the server configuration file, *server.cfg*, with a text editor.

2 Change values as applicable.

3 Save the file.

4 Stop the server, then restart it to implement the changes.

Resolving missing configuration files during start-up

If the server encounters a missing configuration file during a normal start-up of a new DirectConnect server (using AddServer), it creates a new configuration file and populates it with default values. However, if the missing file is an access service library file, the server does not create an access service.

In addition, if the server is configured for a snrf.tble table and one does not exist, the server creates a new one and populates it with enough data to allow DirectConnect Manager to create new access services.

Client Interaction server properties

The subsection heading and configuration property list must appear in the server configuration file as follows:

```
{Client Interaction}
CreateSrvcCfg
DefaultServerLanguage
DefQueueSize
Description
IsDCDirectorServer
MaxConnections
NetBufSize
OSCodeSetConvert
RemoteSites
ServiceRedirectionFile
SSLEnabled
SSLServices
SSLTrustedCertificateFile
```

The following sections describe each of these configuration properties.

CreateSrvcCfg

Allows you to control which service libraries are loaded for a particular DirectConnect server.

Syntax	CreateSrvcCfg=[no | yes]
Default	yes

Comments

- If this property value is set to no and there is no configuration file for the DirectConnect service, DirectConnect will not load a service library.
- If this property value is set to yes, DirectConnect loads all service libraries and creates a new configuration file for each DirectConnect access service that does not have one.

Default Server Language

Identifies the language in which client messages that originate in the DirectConnect server are returned.

Syntax

DefaultServerLanguage=*language*

Range

0-255 (characters)

Default

us_english

Comments

- If this property value is set to:
  - OpenServerDefault, the DirectConnect server default language is the language configured in the *locales.dat* file.
  - A specific language, that language becomes the DirectConnect server default.
- A client can specify a different language in its login record. This allows multiple clients to communicate with the server in multiple languages simultaneously.
- If the server does not support the language specified by a client, it connects using the server default language. In all cases, the DirectConnect server preserves the language specified by the login record and sends it to the target.

DefQueueSize

Identifies the deferred event queue size. This is the maximum number of events that can be queued on this Open Server application at any given time.

Syntax

DefQueueSize=*integer*

Range

512–4096

Default

1024

Comment Commas are not allowed.

Description

Describes of the DirectConnect server.

Syntax Description=*text*

Range 0–255 (characters)

Default none

Comment This property lets you place descriptive information about the server in the configuration file.

IsDCDirector

When set to yes, designates a DirectConnect server as a DirectConnect Director (DCDirector), and only loads the *admin* library. A DCDirector performs the sole function of creating, starting, and stopping other servers in the same directory.

Syntax IsDCDirector= [yes | no]

Default no

Comments

- All other DirectConnect servers on this machine must have this property set to no.
- Only the *admin* library is loaded when the property is set to yes. All other libraries are not loaded.

MaxConnections

Identifies the maximum number of clients that you can connect to this Open Server application.

Syntax MaxConnections=*integer*

Range 1–5000

Default 42

Comments

- Commas are not allowed.

- This value must accommodate the MaxSvcConnections for all the access services operating on the same server.

  (*UNIX only*).
  In UNIX, this property prevents your system from crashing, as a result of running out of UNIX file descriptors.

  The following formulas are to be used to determine the MaxConnections *server.cfg* value and the file descriptor setting, in UNIX, for the DirectConnect process.

  ```
  MaxConnections = (((max file descriptors per process
  /1.10) - 50) /3)
  ```

  where:

  - *max file descriptors per process* equals the limit in UNIX.

    This limit is determined by your UNIX administrator and found by using the ulimit -a utility to determine the number of file decriptors that are configured in your UNIX environment for your DirectConnect process. To obtain this information, you must be logged in as the user, have permission to start DirectConnect, and must set the environment variables for DirectConnect to run.

  - *1.10* equals a 10% safety factor to build in extra descriptors.
  - *50* equals the number of file descriptors reserved for the DirectConnect server.
  - *3* equals the number of file decriptors that could potentially be used by each concurrent client connection to DirectConnect.

    Example:

    ```
    (((2000 file descriptors /1.10) - 50) / 3) = 589
    (fractional part is truncated)
    ```

  In this example, MaxConnections will be configured to a maximum of 589 or less. If the number of connections needs to be greater due to the number of users using DirectConnect at a peak time, use the following formula to determine the correct number of file descriptors that your UNIX administrator will configure for the DirectConnect process:

  ```
  File descriptors = (((MaxConnections desired) x 3) +
  50) x 1.10
  ```

  where:

- *3* equals the number of file decriptors that could potentially be used by each concurrent client connection to DirectConnect.
- *50* equals the number of file descriptors reserved for the DirectConnect server.
- *1.10* equals a 10% safety factor to build in extra descriptors.

In the previous example, if the file descriptors constrained the calculated number of MaxConnections to 589 and you need 1000 connections, perform the following calculation:

```
(((1000 MaxConnections desired x 3) + 50) x 1.10) =
3355 (file descriptors)
```

In this example, the calculated 3355 file descriptors will provide you with enough file descriptors to handle your desired number of concurrent connections (1000) for your DirectConnect process. To meet this requirement, your UNIX administrator needs to increase the number of file descriptors available for the DirectConnect process.

NetBufSize

Identifies the maximum size of any CT-Library packet used on the network.

Syntax	NetBufSize=*integer*
Range	512–32768
Default	2048
Comment	Commas are not allowed.

OSCodeSetConversion

Allows for an additional code page translation between the DirectConnect server and the client.

Note OSCodeSetConversion is necessary only for the Microsoft Windows versions of the UDB DB2, Informix, and Microsoft SQL Server products, when the DirectConnect server code page differs from the client code page.

Syntax	OSCodeSetConversion= [no | yes]

Default
no

Values

- yes – code page conversions are performed between the DirectConnect server and the client.
- no – no additional code page conversion takes place between the DirectConnect server and the client.

Comment
Most DirectConnect products perform a single code page translation from the target Database Management System (DBMS) code page to the client code page. However, the Microsoft Windows UDB DB2, Informix, and Microsoft SQL Server products convert the target DBMS code page to the DirectConnect server (platform) code page, potentially requiring an additional code page translation between the server and the client.

RemoteSites

Lists the total number of Adaptive Server servers that can connect simultaneously to this Open Server application.

Syntax
RemoteSites=*integer*

Range
0–32

Default
4

ServiceRedirectionFile

Provides the name of the access service name redirection file.

Syntax
ServiceRedirectionFile=*filename.ext*

Range
ASCII file name, one to eight characters, with a one-to-three-character extension.

Default
None. (Specify a value for this property only if you are using access service name redirection.)

Comments

- The access service name redirection file must reside in the server *cfg* subdirectory, the same subdirectory that contains the *server.cfg* file.
- For information on the access service name redirection feature, see Chapter 6, "Using Service Name Redirection."

SSLEnabled

When a server is started, it allows DirectConnect to check for all the configured access services.

Syntax

SSLEnabled= [yes | no]

Default

no

Values

yes causes DirectConnect to search the following directories for two files, one ending in *.crt* (the *certificate* file), and the other ending in *.pwd* (the *encrypted password* file). For example, *srvname.crt* and *srvname.pwd*. Instructions for creating these files are defined in the DirectConnect *Installation Guides*.

DirectConnect searches the following files:

- *$SYBASE/$$SYBASE_ECON/[server]/certificates*
- $SYBASE/$$SYBASE_ECON/certificates
- $SYBASE/certificates
- $SYBASE_CERTDIR

If SSLEnabled equals yes, the *service name* of the SSLServices property and the *srvname.crt* and *srvname.pwd* must match. If a match is not found, DirectConnect does not start.

If both files are present, DirectConnect passes the path to the *certificate* file, and the contents of the *password* file to Open Server. This initializes the SSL context for DirectConnect.

Warning! Only one SSLEnabled access service can run on a DirectConnect server. This is due to the restrictions of Open Server, which allows only one SSL certificate in a program. Open Client requires the name in the certificate to match the name to which Open Client requested a connection.

Comments

While you can configure DirectConnect to listen on both SSL and non-SSL ports, which allows you to use both non-SSL access service and one SSL access service in the same DirectConnect, Sybase recommends using only one SSL access service for each DirectConnect. This prevents a user from using an unsecured port to gain access to unsecured data within an organization.

Note DirectConnect 12.6 does not support “transfer to” and “transfer from” SSL-enabled Adaptive Server servers.

SSLServices

Identifies the access or TRS service that will use SSL.

Syntax

SSLServices= *service name*

Default

none

Values

service name is a valid TRS or access service name that exists in DirectConnect. The access service identified is to use SSL.

Warning! The *service name* entered must match an existing access service for it to be designated as an SSL service.

Comments

DirectConnect will *not* start if the following is found:

- An invalid TRS or access service
- An invalid certificate or password in the certificates or password file

SSLTrustedCertificateFile

Identifies the path to the file containing the certificate(s) of the Trusted Certificate Authorities (CAs).

Syntax

SSLTrustedCertificateFile= <*certificate file path*>

Default

none

Values

certificate file path is a valid path to the file containing the certificates of the Trusted Certificate Authorities (CAs).

Comments

During initialization, the file path is checked for the existence of the file. If the file does not exist, the server exits and logs an error message.

Logging properties

The subsection heading and configuration property list must appear in the server configuration file as follows:

```
{Logging}
 LogClientLogin
 LogClientMessages
```

```
LogFileName
LogFileSize
LogFlush
LogOCOSMessages
LogToScreen
LogWrap
```

LogClientLogin

When set to yes, the log reports the results of connection successes, connection failures, and access service name redirection results, if any. When you set this property to no, connection activity is not logged.

Syntax LogClientLogin=[no | yes]

Default no

Comments

- For information on logging, see Chapter 7, "Log and Trace Files".
- For information on access service name redirection, see Chapter 6, "Using Service Name Redirection".

LogClientMessages

When set to 0 (zero), the system does not log client messages. When you set this property to any other integer, the system logs messages that have a severity level greater than or equal to the specified value.

Message severities fall in the range of 10–24, inclusive, matching the levels defined for Adaptive Server messages.

Syntax LogClientMessages=*severity*

Range 0–24

Default 17

Comments

- For information on logging, see Chapter 7, "Log and Trace Files."
- For information on messages, including explanations of error message severity levels, see the DirectConnect *Error Message Guide*.

LogFileName

Contains log messages.

Syntax

LogFileName=*filename.ext*

Range

ASCII file name, one to eight characters, with a one-to-three-character extension

Default

ServerName.log

where *ServerName* is the name of the DirectConnect server.

Comments

- The log language is U.S. English, using the native character set of the machine on which the server is running.
- The log file is located in the *log* subdirectory for the server.
- For information on logging, see Chapter 7, “Log and Trace Files”.

LogFileSize

Indicates the maximum size of the body of the log file, not including the header.

Syntax

LogFileSize=*integer*

Range

0–500000000 (bytes)

Default

500000

Comments

- Commas are not allowed.
- When the log file is full, either further logging is disabled or subsequent records begin after the header, or the file is archived, depending upon how you set up the value of the LogWrap configuration property. See also “LogWrap” on page 36.
- For information on logging, see Chapter 7, “Log and Trace Files”.

LogFlush

When set to yes, the system writes each log record as it is generated. When you set this property to no, the system buffers log records and writes them periodically for optimal performance.

Syntax

LogFlush=[no | yes]

Default

no

Comment

If you have several other logging properties active, setting this property to yes results in a small, negative impact on performance, but ensures that the log is complete in the event of a system failure.

LogOCOSMessages

When set to 0 (zero) the system does not log error messages generated internally by Open Client/ and Open Server. When you set this property to any other integer, the system logs the generated messages that have a severity level greater than or equal to the specified value.

Message severities fall in the range of 10–24, inclusive, matching the levels defined for Adaptive Server.

Syntax

LogOCOSMessages=*severity*

Range

0–24

Default

1

Comments

For information on logging, see Chapter 7, “Log and Trace Files.”.

LogToScreen

When set to yes, the system sends log output to the console, as well as to the log file. When you set this property to no, the system sends log output to the log file only.

Syntax

LogToScreen=[no | yes]

Default

no

Comments

- Setting this property to yes can result in a negative impact on performance, depending upon the number and types of other log properties you have set to yes.
- For information on logging, see Chapter 7, “Log and Trace Files”.

LogWrap

Allows you to wrap the log file, stop logging when a maximum is reached, or archive the log file when the maximum size is exceeded.

Syntax

LogWrap= [yes | no | archive]

Default

yes

Values

- yes causes the log file to wrap and for subsequent records to overwrite the earlier entries.
- no disables logging when the maximum allowable size is reached, as determined by the LogFileSize property.
- archive results in the DirectConnect log file being archived when the LogFileSize property value is exceeded. Archived log files use a *mmddyyhhmmss* timestamp in the name.

Note The archive option should be monitored as it can fill a file system.

Comments

- Only the DirectConnect log file is affected by this configuration property.
- See also “LogFileSize” on page 35.
- For information on logging, see Chapter 7, “Log and Trace Files”.

Tracing properties

The subsection heading and configuration property list must appear in the server configuration file as follows:

```
{Tracing}
 Trace_osClient
 Trace_smConfigAccess
 Trace_smConfigManager
 Trace_smConfigProperty
 Trace_smConnection
 Trace_smLocaleFile
 Trace_smMsgCollection
 Trace_smServer
 Trace_smService
 Trace_smSvclib
 Trace_SOstreams
 TraceAsync
 TraceEntryExit
 TraceFileName
 TraceLogMessages
 TraceOpenServer
```

```
TraceOther
TraceToScreen
```

For information on tracing, see Chapter 7, “Log and Trace Files”.

Warning! Use the tracing properties *only* when Sybase Technical Support instructs you to do so.

Trace_osClient

Allows the system to trace OS client internal data.

Syntax

Trace_osClient=[no | yes]

Default

no

Comments

yes – the system traces OS client internal data.

Trace_smConfigAccess

Allows the system to trace configuration access internal data.

Syntax

Trace_smConfigAccess=[no | yes]

Default

no

Comments

yes – the system traces configuration access internal data.

Trace_smConfigManager

Allows the system to trace related configuration manager internal data.

Syntax

Trace_smConfigManager=[no | yes]

Default

no

Comments

yes – the system traces configuration manager internal data.

Trace_smConfigProperty

Allows the system to trace configuration property internal data.

Syntax	Trace_smConfigProperty=[no | yes]
Default	no
Comments	yes – the system traces configuration property internal data.

Trace_smConnection

Allows the system to trace connection internal data.

Syntax	Trace_smConnection=[no | yes]
Default	no
Comments	yes – the system traces certain connection internal data.

Trace_smLocaleFile

Allows the system to trace locale file internal data.

Syntax	Trace_smLocaleFile=[no | yes]
Default	no
Comments	yes – the system traces *locale* file internal data.

Trace_smMsgCollection

Allows the system to trace message collection internal data.

Syntax	Trace_smMsgCollection=[no | yes]
Default	no
Comments	yes – the system traces message collection internal data.

Trace_smServer

Allows the system to trace server internal data.

Syntax	Trace_smServer=[no | yes]
Default	no

Comments yes – the system traces server internal data.

Trace_smService

Allows the system to trace service internal data.

Syntax Trace_smService=[no | yes]

Default no

Comments yes – the system traces service internal data.

Trace_smSvclib

Allows the system to trace service library internal data.

Syntax Trace_smSvclib=[no | yes]

Default no

Comments yes – the system traces service library internal data.

Trace_SOstreams

Allows the system to trace OS stream internal data.

Syntax Trace_SOstreams=[no | yes]

Default no

Comments yes – the system traces OS stream internal data.

TraceAsync

Allows tracing of asynchronous events, such as interrupts and timer notification.

Syntax TraceAsync=[no | yes]

Default no

Comments yes – asynchronous event tracing is enabled.

TraceEntryExit

Traces entry and exit from most DirectConnect internal functions.

Syntax	TraceEntryExit=[no | yes]
Default	no
Comments	yes – entry and exit from internal functions are traced.

TraceFileName

Allows you to identify the name of the file containing trace messages.

Syntax	TraceFileName=*filename.ext*
Range	ASCII file name, one to eight characters, with a one-to-three-character extension
Default	*ServerName.trc*, where *ServerName* is the name of the DirectConnect Server.
Comments	*filename.ext* is the file name that contains the trace messages.

TraceLogMessages

Allows you to duplicate log records in the trace file.

Syntax	TraceLogMessages=[no | yes]
Default	no
Comments	yes – the system duplicates log records in the trace file.

TraceOpenServer

Corresponds to the Open Server SRV_S_TRACEFLAG property. Values comprised of the bitwise or of the SRV_TR defined values in *ospublic.h*. For information on configuring this property, see the Open Server *Server-Library/C Reference Manual*.

Syntax	TraceOpenServer=*bitflags*
Range	0–65535
Default	0

Comments *bitflags* is an integer.

TraceOther

Allows you to trace data for debugging.

Syntax TraceOther=[no | yes]

Default no

Comments yes – traces data for debugging.

TraceToScreen

Allows you to send the trace output to the console, as well as to the trace file.

Syntax TraceToScreen=[no | yes]

Default no

Comments

- Setting this property to yes sends the trace output to the console and trace file but can result in a negative impact on performance, depending upon the number and types of other log and trace properties you have set to yes.
- See also “LogToScreen” on page 36.

CHAPTER 5 Setting up SSL and a Windows Service

This chapter describes how to set up Secure Sockets Layer (SSL) on a DirectConnect server, and how to set up and run DirectConnect as a Windows service.This chapter contains the following topics:

Setting up SSL on the DirectConnect server

The following sections describe the procedures for setting up SSL on the DirectConnect server for Windows and UNIX.

Setting up SSL on the server (for Windows)

Warning! Only one SSL-enabled access service can run on a DirectConnect server. This is due to the restrictions of Open Server, which allows only one SSL certificate in a program. Open Client requires the name in the certificate to match the name to which Open Client requested a connection.

Although you can configure DirectConnect to accept SSL and non-SSL connections (for example, use non-SSL access services and one SSL access service in the same DirectConnect), Sybase recommends using only one SSL access service. This prevents a user from using an unsecured port to access data over an unsecured transport medium.

To set up DirectConnect (except DirectConnect for Oracle) for SSL to provide encryption of data sent over the network, and to authenticate clients and their passwords using digital certificates, perform the following tasks:

- To create the certificate of authority files
- To create the certificate of authority files for the specific DirectConnect server and service
- To create DirectConnect certificates directory, enable SSL, and verify the log files

Note DirectConnect 12.6 does not support “transfer to” and “transfer from” SSL-enabled Adaptive Server servers.

❖ To create the certificate of authority files

1. Add the following to the path of the environment variable in *%SYBASE%\SYBASE.bat* file:

   ```
   %SYBASE%\OCS-12_5\lib3p
   ```

2. Set the environment by running the following from a command window:

   ```
   %SYBASE%\SYBASE.bat
   ```

3. Enter the following to go to the *certreq* directory:

   ```
   cd %SYBASE%\OCS-12_5\bin
   ```

4. Execute the setsslreq utility, one time only, on Windows to set SSL registry key information for Open Server.

   Note If you have previously created or obtained a certificate of authority, skip steps 2-8.

5. Create the Certificate Authority (CA) *CA.in* file. (For the parameters, refer to the ASE *Utilities Guide* document for certreq.) Enter the parameters for the CA certificate that you are going to use with the certreq utility, as shown:

   ```
   C:\sybase\OCS-12_5\bin>type CA.in
   ```

```
req_certtype=Server
req_keytype=RSA
req_keylength=512
req_country=US
req_state=CO
req_locality=Boulder
req_organization=Sybase
req_orgunit=Security
req_commonname=CA
```

6. Create the *private key* file and a *certificate request* file for the CA certificate:

    ```
    C:\sybase\OCS-12_5\bin>certreq -F CA.in
    -R CA_req.txt -K CA_pkey.txt -Pmycapassword
    ```

    The following message appears:

    ```
    Generating key pair (please wait)...
    ```

7. Create a public key file named *trusted.txt* by using the *CA_req.txt* file with the *private key* file to sign the *public key* file:

    ```
    C:\sybase\OCS-12_5\bin>certauth -r
    -C CA_req.txt -Q CA_req.txt -K CA_pkey.txt
    -P mycapassword -T 365 -O trusted.txt
    -- Sybase Test Certificate Authority
    certauth\12.6.0.1\SWR 9988 IR\P\NT (IX86)\OS 4.0
    \rel12501 \1773/32-bit\OPT\Sat Feb 16 07:18:45 2002
    -- Certificate Validity:
    startDate = Mon Apr 22 17:58:10 2002
    endDate = Tue Apr 22 17:58:10 2003

    CA sign certificate SUCCEED (0)
    ```

❖ To create the certificate of authority files for the specific DirectConnect server and service

1. Enable SSL and identify the name of the access service using the SSLEnabled and SSLServices properties. For a description of these properties and their use, refer to the ECDA and Mainframe Connect *Server Administration Guide* for DirectConnect.

2. Use a text editor to create the *DC.in* file. (For the parameters, refer to the ASE *Utilities Guide* document for certreq.)

    ```
    notepad DC.in
    ```

    ```
    req_certtype=Server
    req_keytype=RSA
    req_keylength=512
    ```

```
req_country=US
req_state=CO
req_locality=Boulder
req_organization=Sybase
req_orgunit=Database
req_commonname=servicename
```

3 Create *private key* and *certificate request* files for the DirectConnect service:

```
C:\sybase\OCS-12_5\bin>certreq
-F DC.in
-R servicename_req.txt
-K servicename_pkey.txt
-P mydcpassword
```

4 Create a DirectConnect *public key* file *<servicename>.crt* using the *<servicename>_req .txt* file with the CA *private key* file to sign the DirectConnect *public key* file:

```
C:\sybase\OCS-12_5\bin>certauth
-C trusted.txt
-Q servicename_req.txt
-K CA_pkey.txt
-P mycapassword -T180 -O servicename.crt

-- Sybase Test Certificate Authority
certauth\12.6.0.1\SWR 9988 IR\P\NT (IX86)\OS 4.0
\rel12501\1773\32-bit\OPT\Sat Feb 6 07:18:45 2002--
Certificate Validity:

startDate = Mon Apr 22 18:18:41 2002

endDate = Sat Oct 19 18:18:41 2002

CA sign certificate SUCCEED (0).
```

5 Append the signed service name *private key* file to the signed *<server name> public key* file:

```
C:\sybase\OCS-12_5\bin>type
servicename_pkey.txt >> servicename.crt
```

6 Copy the *trusted.txt* file to the DirectConnect *<servicename>.txt* file:

```
C:\sybase\OCS-12_5\bin>
copy trusted.txt servicename.txt
```

7 Using the pwdcrypt utility, create and enter an encrypted password for DirectConnect to establish an SSL connection:

```
C:\sybase\OCS-12_5\bin>pwdcrypt
```

Enter your password that will be encrypted. Your encrypted password will be similar to the following example:

Note The password you enter will not be visible. This is the same password (mydcpassword) used in step 3.

```
C:\sybase\OCS-12_5\bin>pwdcrypt
Enter password please:
Enter password again:
The encrypted password:
0x018c2e0ea8cfc44513e8ff06f3a1b20825288d0ae1ce79268
d0e8669313d1bc4c70c
```

8 Insert the encrypted password by copying from the previous step:

```
C:\sybase\OCS-12_5\bin>ECHO
0x018c2e0ea8cfc44513e8ff06f3a1b20825288d0ae1ce79268
d0e8669313d1bc4c70c >servicename.pwd
```

Note When created, an extra space is appended to the password. You must remove the extra space to have a valid password.

9 Copy the *trusted.txt* file to the DirectConnect *srvname.txt* file:

```
C:\sybase\OCS-12_5\bin>
copy trusted.txt srvname.txt
```

10 From the list of files displayed, verify that the following files are present:

```
C:\sybase\OCS-12_5\bin>dir
CA_pkey.txt
CA_req.txt
DC.in
servicename.crt
servicename.pwd
servicename.txt
servicename_pkey.txt
servicename_req.txt
srvname.txt
trusted.txt
```

❖ To create DirectConnect certificates directory, enable SSL, and verify the log files

Note In the following steps, the server name and service name must be the same.

1 Create a DirectConnect directory to hold the certificates:

```
md %SYBASE%\%SYBASE_ECON%\
<server_name>\certificates
```

2 Copy the service*name.crt*, *servicename.pwd*, *servicename.txt*, and the s*vrname.tx*t files into the new DirectConnect certificates directory created in the previous step:

```
copy %SYBASE%\%SYBASE_OCS%\bin\servicename.*
%SYBASE%\%SYBASE_ECON%\server name\certificates
copy %SYBASE%\%SYBASE_OCS%\bin\srvname.txt
%SYBASE%\%SYBASE_ECON%\server name\certificates
```

3 Verify that the files are copied by listing the contents of the DirectConnect *certificates* directory:

```
cd %SYBASE%\%SYBASE_ECON%\
<server_name>\certificates
```

If successful, the following is displayed:

```
servicename.crt
servicename.pwd
servicename.txt
srvname.txt
```

4 Edit the *server.cfg* file to enable the SSL service:

- Enter the name of the DirectConnect service in the SSLServices property that is going to use SSL.
- Enter yes in the SSLEnabled property to enable the SSL feature:

```
notepad server.cfg

{Client Interaction}

SSLServices=servicename
SSLEnabled=yes
```

5 From all the properties displayed, verify that the logging properties are set correctly and match the following:

```
cd %SYBASE%\%SYBASE_ECON%\server name\cfg
type server.cfg
```

If successful, the following is displayed:

```
(Logging)
LogWrap=yes
LogToScreen=yes
LogOCOSMessages=1
```

```
LogFlush=yes
LogFileSize=500000
LogFileName=
LogClientMessages=1
LogClientLogin=yes
```

6 Append “ssl” to the master and query entries in the *sql.ini* file using a text editor:

```
cd %SYBASE%\ini
notepad sql.ini
server name
MASTER = NLWNSCK, machine name, port, ssl
Query = NLWNSCK, machine name, port, ssl
```

7 Execute the following script to start DirectConnect:

```
C:\sybase>DC-12_6\DCStart -Sservername
```

8 Verify that the following log entries are in the *%SYBASE%\%SYBASE_ECON%\<server name>\log \<server name>.log* file:

```
LogHeader...SSL:Checking for servicename.txt...
LogHeader...SSL:Using trusted CA file...
LogHeader...SSL:Checking for servicename.crt...
LogHeader...SSL:Using certificate file...
LogHeader...SSL:Checking for servicename.pwd...
LogHeader...SSL:Using certificate password file...
```

Configuring the SSL Windows client (for Windows)

❖ To configure the client environment to use SSL

1 Set the client Sybase Window variable:

```
set SYBASE=c:\sybase
```

2 Go to the *%SYBASE%\ini* directory:

```
C:\sybase\OCS-12_5\bin>cd %SYBASE%\ini
```

3 Copy the *trusted.txt* file to the *%SYBASE%\.ini* directory:

```
C:\sybase\ini>copy
C:\sybase\OCS-12_5\bin\trusted.txt %SYBASE%\ini
```

4 Edit the *sql.ini* file and append the SSL entry to the Master and Query entries for the DirectConnect services:

```
C:\sybase\ini>notepad sql.ini
```

```
[server name]
Master = NLWNSCK, machine name, port ssl
Query = NLWNSCK, machine name, port ssl
```

5 Go to the Sybase Open Client and Open Server *bin* directory:

```
cd %SYBASE%\%SYBASE_OCS%\bin
```

6 Issue the isql command to connect to the service:

```
isql -Sservice name -Uuid -Ppwd
```

7 When you are finished, stop the server and restart it. If you receive no connection errors, SSL is installed correctly.

For testing, Sybase recommends that you use ssldump to examine SSL handshakes. You can find this free utility at www.rtfm.com/ssldump/.

Setting up SSL on the server (for UNIX)

Note The following procedure to provide SSL encryption and to authenticate clients is not valid for DirectConnect for Oracle. Refer to the ECDA Option for Oracle Server Administration and User's Guide.

DirectConnect for SSL provides encryption of data sent over the network and authenticates clients and their passwords using digital certificates. Setting up DirectConnect involves the following tasks:

- To create the certificate of authority files
- To create the certificate of authority files specific to the DirectConnect server and service
- To create the DirectConnect certificates directory, enable SSL, and verify the DirectConnect log files

Note DirectConnect 12.6 does not support transfer to and transfer from on the SSL-enabled Adaptive Server servers.

❖ To create the certificate of authority files

1 Set up the Sybase environment variables:

```
Source $SYBASE/SYBASE.csh (or .sh)
```

Note If you have previously created or obtained the certificate of authority files, skip steps 2-9.

2 Change to the Sybase Open Client/Server *bin* directory to run the SSL utilities:

```
prompt% cd $SYBASE/$SYBASE_OCS/bin
```

3 Create the Certificate of Authority (CA) *CA.in* file. (For the parameters, refer to the ASE *Utilities Guide* document for certreq.) Enter the parameters for the CA certificate that you are going to use with the certreq utility, as shown:

```
vi CA.in
req_certtype=Server
req_keytype=RSA
req_keylength=512
req_country=US
req_state=CO
req_locality=Boulder
req_organization=Sybase
req_orgunit=Security
req_commonname=CA
```

4 Create a *private key* file and a *certificate request* file for the CA certificate:

```
prompt% certreq -F CA.in -R CA_req.txt
-K CA_pkey.txt -P mycapassword
```

The following message appears:

```
Generating key pair (please wait)...
```

5 Create a *public key* file named *trusted.txt* by using the *CA_req.txt* file with the *private key* file to sign the *public key* file:

```
prompt% certauth -r -C CA_req.txt -Q CA_req.txt
-K CA_pkey.txt -P mycapassword -T 365 -O trusted.txt

-- Sybase Test Certificate Authority
certauth/12.6/SWR 9609 GA/P/Sun_svr4/OS
5.8/main/1647/32-bit/OPT/Fri Jun  1 17:19:08 2001
--Certificate Validity:
startDate = Tue Apr 23 15:01:40 2002
endDate = Wed Apr 23 15:01:40 2003
CA sign certificate SUCCEED (0).
```

❖ To create the certificate of authority files specific to the DirectConnect server and service

Note In UNIX, the name of the server and service must be the same.

1. Input DirectConnect parameters for the CA. (For the parameters, refer to the ASE utilities documentation for certreq.)

   ```
   prompt%
   vi DC.in
   req_certtype=Server
   req_keytype=RSA
   req_keylength=512
   req_country=US
   req_state=CO
   req_locality=Boulder
   req_organization=Sybase
   req_orgunit=Database
   req_commonname=servicename
   ```

2. Create *private key* and *certificate request* files for the DirectConnect service:

   ```
   prompt% certreq -F DC.in -R servicename_req.txt
   -K servicename_pkey.txt -P mydcpassword
   ```

3. Create a DirectConnect *public key* file *<servicename>.crt* using the *<servicename>_req.txt* file with the CA *private key* file to sign the DirectConnect *public key* file:

   ```
   prompt% certauth -C trusted.txt
   -Q servicename_req.txt -K CA_pkey.txt
   -P mycapassword
   -T 180 -O servicename.crt
   ```

   The following appears:

   ```
   -- Sybase Test Certificate Authority
   certauth/12.6/SWR 9609 GA/P/Sun_svr4/OS
   5.8/main/1647/32-bit/OPT/Fri Jun  1 17:19:08 2001 --

   Certificate Validity:
   startDate = Tue Apr 23 15:11:33 2002
   endDate = Sun Oct 20 15:11:33 2002
   CA sign certificate SUCCEED (0)
   ```

4. Append the service name *private key* file to the signed service name *public key* file:

```
prompt% cat servicename_pkey.txt >> servicename.crt
```

5 Verify that the *private key* file is appended and is similar to the following by entering:

```
prompt% cat servicename.crt
```

6 Copy the *trusted.txt* file to the DirectConnect *<servicename>.txt* file:

```
prompt% cp trusted.txt servicename.txt
```

7 Using the pwdcrypt utility, create and enter an encrypted password for DirectConnect to establish an SSL connection:

```
prompt% pwdcrypt
```

Enter your password that is to be encrypted, which will be similar to the following example:

Note The password you enter will not be visible. This is the same password (mydcpassword) that you entered in step 2.

```
Enter password please:
Enter password again:
The encrypted password:
0x018c2e0ea8cfc44513e8ff06f3a1b20825288d0ae1ce79268
d0e8669313d1bc4c70c
```

8 Insert the encrypted password from the previous step into a file:

```
prompt% vi servicename.pwd
```

9 Copy the *trusted.txt* file to the DirectConnect *srvname.txt* file:

```
cp trusted.txt srvname.txt
```

10 From the list of files, verify that all the following files are present:

```
prompt% ls
```

```
CA.in
CA_pkey.txt
CA_req.txt
servicename.crt
servicename.pwd
servicename.txt
servicename_pkey.txt
servicename_req.txt
srvname.txt
trusted.txt
```

❖ To create the DirectConnect certificates directory, enable SSL, and verify the DirectConnect log files

Note In the following steps, the DirectConnect server name and service name must be the same.

1 Create a DirectConnect directory to hold the certificates:

```
mkdir $SYBASE/$SYBASE_ECON/server name/certificates
```

2 Copy the *servicename.crt*, *servicename.pwd*, *servicename.txt*, and the *svrname.txt* files into the new DirectConnect certificates directory created in the previous step:

```
cp $SYBASE/$SYBASE_OCS/bin/servicename.*
$SYBASE/$SYBASE_ECON/server name/certificates
cp $SYBASE/$SYBASE_OCS/bin/srvname.txt
$SYBASE/$SYBASE_ECON/servername/certificates
```

3 Verify that the files are copied by listing the contents of the DirectConnect *certificates* directory:

```
<machine name>% cd $SYBASE/$SYBASE_ECON/
servername /certificates

prompt% ls

servicename.crt
servicename.pwd
servicename.txt
srvname.txt
```

4 Edit the *server.cfg* file to enable the SSL service:

- Enter the name of the DirectConnect service in the SSLServices property that is going to use SSL.
- Enter yes in the SSLEnabled property to enable the SSL feature:

```
prompt% cd $SYBASE/$SYBASE_ECON/servername/cfg

prompt% vi server.cfg

{Client Interaction}
SSLServices=servicename
SSLEnabled=yes
```

5 From the many properties displayed, verify that the logging properties are set correctly and match the following:

```
prompt% cat server.cfg
```

```
{Logging}
LogWrap=yes
LogToScreen=yes
LogOCOSMessages=1
LogFlush=yes
LogFileSize=500000
LogFileName=
LogClientMessages=1
LogClientLogin=yes
```

6 Append “ssl” to the master and query entries in the *interfaces* file:

```
cd $SYBASE
vi interfaces
server name
master tcp ether machine name 12510 ssl
query tcp ether machine name 12510 ssl
```

7 Start DirectConnect by entering the following:

```
prompt% DCStart -S<dcservername>
```

8 Verify that the following log file entries are in the *$SYBASE/$SYBASE_ECON/<server name>/log /<server name>.log* file:

```
LogHeader...SSL: Checking for servicename.txt...
LogHeader...SSL: Using trusted CA file...
LogHeader...SSL: Checking for servicename.crt...
LogHeader...SSL: Using certificate file...
LogHeader...SSL: Checking for servicename.pwd...
LogHeader...SSL: Using certificate password file...
```

Warning! Echo of the password caused a blank character to be appended to the password. Remove the additional space to avoid an error.

Configuring the SSL client (for UNIX)

❖ To configure the client environment to use SSL

1 Set the client Sybase UNIX variable:

```
prompt% source $SYBASE/SYBASE.csh (or .sh)
```

2 Copy the *trusted.txt* file to the *$SYBASE config* directory:

```
prompt%
cp $SYBASE/$SYBASE_OCS/bin/trusted.txt
```

```
$SYBASE/config
```

3 Go to the *$SYBASE* directory:

```
prompt% cd $SYBASE
```

4 Edit the *interfaces* file, and append the SSL entry to the Master and Query entries for the DirectConnect service:

```
prompt% cat interfaces
```

```
servicename
master tcp ether machine name 12510 ssl
query tcp ether machine name 12510 ssl
```

5 Go to the Sybase Open Client and Open Server *bin* directory:

```
prompt% cd $SYBASE/$SYBASE_OCS/bin
```

6 Issue the isql command to connect to the DirectConnect service:

```
isql -Sservicename -Uuid -Ppwd
```

7 When you are finished, stop the server and restart it. If you receive no connection errors, SSL is installed correctly.

Installing a DirectConnect server as a Windows service

DirectConnect no longer automatically creates the server as a Windows service. However, you can run a DirectConnect server as a Windows service. The following describes how to register, configure, start, stop, and remove DirectConnect as a Windows service.

Register DirectConnect as a Windows service

❖ **To register DirectConnect**

- Execute the AddWinService.bat located in the *DC-12_6\bin* directory using your name for the “Windows service name” and the name of the DirectConnect server:

  ```
  AddWinService <Windows_service_name> <server_name>
  ```

  For example:

  ```
  AddWinService dcservice dcserver
  ```

Configure DirectConnect as a Windows service

❖ To configure DirectConnect to run as a Windows service

1 Install DirectConnect on your local hard drive.

> **Warning!** Do not install DirectConnect under a clear case filesystem, or remotely, or in a shared file system, it may cause DirectConnect to fail during file I/O.

2 Turn off all server tracing and logging to the screen (console) by accessing the “Logging’ and “Tracing” configuration properties.

3 Perform DirectConnect licensing as described in Chapter 2, “Sybase Software Asset Management (SySAM),” of the installation guide for your platform.

4 To ensure the environment created by DC_SYBASE.bat is also your system environment. Set SYBASE, SYBASE_ECON, SYBASE_SYSAM, and PATH accordingly.

5 Verify configuration by starting the DirectConnect sever using DCStart.bat at the command line. Verify that only the copyright notice is written to the screen (console). No licensing dialog should be displayed.

Start DirectConnect as a Windows service

❖ To start DirectConnect as a Windows service

1 Open the Windows Services control panel.

2 Right click on the DirectConnect Windows service.

3 Select *start*.

Stop DirectConnect as a Windows service

❖ To stop DirectConnect as a Windows service

1 Open the Windows Services control panel.

2 Right click on the DirectConnect Windows service.

3 Select *stop.*

Remove DirectConnect as a Windows service

❖ **To remove DirectConnect as a Windows service**

1. Using the Windows Services control panel, stop the DirectConnect service.
2. Execute the *RemoveWinService.bat* located in the *DC-12_6\bin* directory using the name of the DirectConnect Windows service. Enter the following:

```
RemoveWinService <Windows_service_name>
```

CHAPTER 6 Using Service Name Redirection

Service name redirection is an optional feature that lets you route client requests for an access service to alternative access service names. This chapter explains how you can set up and use the service name redirection feature.

This chapter contains the following topics:

How service name redirection works

When a client application accesses a service, it specifies an access service name. That name must correspond either to the name of an actual access service or to an entry in the service name redirection file (snrf).

Service name redirection allows you to control access to services using the user profile: requested access service, user ID, and application name. You can assign each user profile to any access service supported by an access service library. The DirectConnect server attempts to match the client request with an entry in the service name redirection file before connecting directly with the access service.

Different users who request the same general service name can be routed to different actual access services. For example, three individuals requesting "AS400" could receive completely different access services, such as:

- One for decision support
- One for copy management
- One for online transaction processing (OLTP)

Thus, you can manage multiple sets of clients with a single Access Service Library. However, you must still configure the *sql.ini* (NT) or *interfaces* (UNIX) file to connect clients to the DirectConnect server. For instructions on editing the *sql.ini* or *interfaces* file, see the appropriate Open ClientConnect *Installation Guide* for your platform.

Editing service name redirection (snrf) tables

You can use DirectConnect Manager or the command line to edit a snrf table.

Using DirectConnect Manager

Using DirectConnect Manager, you can perform editing tasks interactively on the Service Name Redirection Editor (SNRF) dialog box.

For instructions on how to use DirectConnect Manager to edit , go to the Managing Service Name Redirection topic of DirectConnect Manager online Help and select “Understanding service name redirection” or “Adding an item to the SNRF table.”

Using command line syntax

The following sections contain information you need to edit the snrf table using a command line.

Service name redirection file format

You activate the service name redirection file by configuring the ServiceRedirectionFile property described in Chapter 4, “Configuring the Server.” The ServiceRedirectionFile property indicates the name of the *Service Redirection* file, which is a text file consisting of four columns separated by tabs, with as many rows as you need to define the redirections. For additional information , refer to “ServiceRedirectionFile” on page 31.

Using a text editor or DirectConnect Manager, you can change the name of the file.

Note If you use a text editor, be sure that it inserts actual tabs, not just spaces that simulate tabs.

The following shows the format of a service name redirection file:

```
requested_service | user_id | application_name | assigned_service
```

Service name redirection rules are as follows:

- Columns must be separated by a single tab character.
- Wildcard characters (asterisks) are allowed in the requested_service, user_id, and application_name columns.
- Comments are not allowed.
- Blank lines can be added for easier viewing.
- Only the user_id column is case sensitive.

If used, a service name redirection file must be valid for the server to start successfully. A valid file has exactly four columns on each line.

Null service names

Some DB-Library versions do not provide a remote server name for service routing, so requests from these DB-Library applications contain null service names. If you run multiple services with a single server, you must use service name redirection to connect such clients. In particular, consider the following:

- Microsoft DB-Library requests contain null service names, because the specified service name is not passed to Open Server in the internal login record. You must use service name redirection to connect such clients.
- Sybase Open Client DB-Library versions 4.2 and earlier do not provide a remote server name. You must use service name redirection to connect such clients.
- Sybase Open Client DB-Library versions 4.3 and later provide the server name. With these clients, you can use direct routing or service name redirection.

If the requested_service name is a null or empty string, the service name redirection file line for routing that service must begin with a tab character. The following table shows an example of a service name redirection file with a null requested_service name.

Table 6-1: Sample of null service name format

requested_service	user_id	application_name	assigned_service
<tab>	Jane	db-lib	svc_db2

Precedence rules

If you inadvertently create a service name redirection file in which an assigned access service name is not uniquely specified, the system uses the following precedence rules to resolve the problem. The first rule defines the highest precedence, the eighth one the lowest.

Table 6-2: Precedence rules

Rule	Description
1	All columns are explicitly defined.
2	requested_service and user_id are specified; application_name uses a wildcard character.
3	requested_service and application_name are specified; user_id uses a wildcard character.
4	user_id and application_name are specified; requested_service uses a wildcard character.
5	Only requested_service is specified; user_id and application_name use wildcard character.
6	Only user_id is specified; requested_service and application_name use wildcard character.
7	Only application_name is specified; requested_service and user_id use wildcard character.
8	Nothing is specified; requested_service, user_id, and application_name use wildcard character.

Note A null-requested service is treated as any other explicitly-specified service.

Example of precedence ruling

To see how the precedence rules work, assume that you set up the service name redirection file shown in the following table.

Table 6-3: Example of using the precedence rules

requested_service	user_id	application_name	assigned_service
AS400	Bob	isql	as1
AS400	*	isql	as2
AS400	*	Omni	omniA
AS400	*	PowerBuilder™	powerB
DB2	*	Omni	db2omni
DB2	*	*	db2gen
<tab>	*	*	as3
*	*	*	as4

Based upon the preceding table, the following are true:

- If Bob requests service AS400 using an isql command, he is redirected to service “as1.”
- If anyone other than Bob requests AS400 using an isql command, that person is directed to service “as2.”
- Anyone who requests service AS400 using Omni is directed to service “omniA.”
- Anyone who requests service AS400 using PowerBuilder is redirected to service “powerB.”
- Anyone who requests service AS400 using any other application is not redirected. Such requests are connected directly to service “AS400.”
- Anyone who requests service DB2 UDB using Omni is directed to service “db2omni.”
- Anyone who requests service DB2 UDB using any other application is redirected to service “db2gen.”
- All Microsoft and earlier Sybase DB-Library clients for which the requested service name is blank are directed to service “as3.”
- Finally, all other clients are routed to service “as4.”

The snrfck validation utility

Sybase provides a validation utility called snrfck that lets you validate the format of the service name redirection file.

Using the basic command

The snrfck basic command requires only the -i option. When you use this option, snrfck reads the redirection file, validates each line, and flags the first incorrect line it encounters.

For example, suppose you enter:

```
snrfck -ic:\cfg\testfile
```

where:

- *cfg* is the directory containing the service name redirection file.
- *testfile* is the service name redirection file.

Note The path *cfg\testfile* is shown as a PC-based system in this example and in the remainder of the examples in this chapter.

Next, assume the redirection file contains the entries shown in the following table:

Table 6-4: Example of a redirection file with a duplicate entry

requested_service	user_id	application_ name	assigned_service
AS400	Bob	isql	as1
AS400	*	isql	as2
AS400	Bob	isql	as2
AS400	*	Omni	omniA
AS400	*	Power Builder	powerB
DB2	*	Omni	db2omni
DB2	*	*	db2gen
<tab>	*	*	as2

In this example, snrfck returns:

```
c:\cfg\testfile: line3: duplicate/ambiguous row
```

If the file does not contain errors, the rows are sorted in the order used in the redirection operation and printed to the current window.

The following table shows an example of a correctly formatted access service name redirection file, as output by snrfck. The snrfck utility adds line numbers for clarity.

Table 6-5: Example of a correctly formatted redirection file

	requested_service	user_id	application_name	assigned_service
1:	<tab>	root	ksh	svc_ksh
2:	db2	joe	isql	svc_db2a
3:	db2	jane	isql	svc_db2b
4:	db2	sonia	Omni	svc_db2c
5:	db2	ramon	Omni	svc_db2d
6:	db2	sven	*	svc_db2gen
7:	other	*	*	svc_other

Using specified values

You can test the redirection process by supplying values for requested_service, user_id, and application_name, subject to the following restrictions:

- You must specify values for user_id and application_name.
- You can use a null argument for requested_service to allow matching on a null service.

When you supply these values, snrfck displays the sorted entries and the assigned service to which the request would be directed.

For example, suppose you use the preceding sample file and enter the following:

```
snrfck -itestfile -Sdb2 -Ujane -Aisql
```

where:

- *db2* is the requested service.
- *jane* is the user ID.
- *isql* is the application name.

You receive the match shown in the following table:

Table 6-6: Redirection file with an entry match

	requested_ service	user_id	application_name	assigned_service
1:	<tab>	root	ksh	svc_ksh
2:	db2	joe	isql	svc_db2a
3:	db2	jane	isql	svc_db2b
4:	db2	sonia	Omni	svc_db2c
5:	db2	ramon	Omni	svc_db2d

	requested_ service	user_id	application_name	assigned_service
6:	db2	sven	*	svc_db2gen
7:	other	*	*	svc_other

The following is displayed:

```
assigned service for (db2,jane,isql): svc_db2b
```

If the service redirection comparison does not find a match, the value returned for assigned_service is simply the requested_service value.

For example, suppose you use the preceding sample file and enter:

```
snrfck -itestfile -Sdb2 -Uramon -Aisql
```

where:

- *db2* is the requested service.
- *ramon* is the user ID.
- *isql* is the application name.

You receive the failed entry match shown in the following table:

Table 6-7: Redirection file with failed entry match

	requested _service	user_id	application _name	assigned_service
1:	<tab>	root	ksh	svc_ksh
2:	db2	joe	isql	svc_db2a
3:	db2	jane	isql	svc_db2b
4:	db2	sonia	Omni	svc_db2c
5:	db2	ramon	Omni	svc_db2d
6:	db2	sven	*	svc_db2gen
7:	other	*	*	svc_other
assigned service for (db2,ramon,isql): db2				

The following is displayed:

```
assigned service for (db2,jane,isql): svc_db2b
```

Adding lines to a redirection file

You can add lines to the service name redirection file list by specifying the -t option.

When you use this option, snrfck displays the normal redirection file and prompts you to enter new lines consisting of “service,” “user,” “application,” and “assigned_service,” each separated by a tab character. The snrfck utility reads the lines, validates them, adds them to the output file, and displays the amended file.

For example, you use the preceding sample file and enter:

```
snrfck -itestfile -t -onewfile
```

where:

- -t activates the test or update capability.
- -o*newfile* specifies the output file. To save changes to the redirection file, you must use this option.

Note If you use -t without using -o, your additions are displayed but not saved.

You receive a file with instructions for adding lines, as shown in the following table:

Table 6-8: Redirection file with -t option

	requested _service	**user_id**	**application_ name**	**assigned_service**
1:	<tab>	root	ksh	svc_ksh
2:	db2	joe	isql	svc_db2a
3:	db2	jane	isql	svc_db2b
4:	db2	sonia	Omni	svc_db2c
5:	db2	ramon	Omni	svc_db2d
6:	db2	sven	*	svc_db2gen
7:	other	*	*	svc_other

The following is displayed:

```
Enter service name redirection file lines:
service<tab>user<tab>application<tab>assigned_service
end with '.' on line by itself
8:
```

Then, you add the following lines in response to the prompt (snrfck supplies the line numbers):

```
8: db2 rachel * svc_db2gen
9: .
```

The snrfck utility produces a new service name redirection file, as shown in the following table:

Table 6-9: Redirection file with new line added

	requested _service	user_id	application _name	assigned_service
1:	<tab>	root	ksh	svc_ksh
2:	db2	joe	isql	svc_db2a
3:	db2	jane	isql	svc_db2b
4:	db2	sonia	Omni	svc_db2c
5:	db2	ramon	Omni	svc_db2d
6:	db2	sven	*	svc_db2gen
7:	db2	rachel	*	svc_db2gen
8:	other	*	*	svc_other

The snrfck utility adds the new entry and sorts the file.

Using other options

Other snrfck options include displaying a version number and displaying help text.

Version

You can display the snrfck version by using the -v option.

For example, you enter the following:

```
snrfck -v
```

The following returns:

```
Service Name Redirection Check Utility, $Revision: 1.2 $
```

Help text

You can display Help text in one of the following ways:

- Enter snrfck -h.
- Enter snrfck and press Enter.

For example, you enter the following:

```
snrfck -h
```

The following returns:

```
snrfck [-v] [-? | -h] [-t [-ofile] ]
 [ -Ssvc -Uusr -Aappl ] -ifile
```

where:

- -v displays the program version only.
- -? or -h displays this help text.
- -t activates the test or update capability.
- -o*file* specifies the output file (this has no effect if -t is not used).
- -S*svc* (service), -U*usr* (user), -A*appl* (application) are optional arguments to test the redirection search.
- -i*file* specifies the input service redirection file.

In UNIX systems, use the -? argument as follows:

```
snrfck -\?
```

Implementing a service name redirection file

After you use snrfck to create or update a service name direction file, you can implement the modified file on the DirectConnect server, as described in the following sections.

Substitute a modified file

❖ **To implement a new service name redirection file or copy a modified file**

1. Use snrfck to create a new file, or to modify the existing file and validate it.
2. Stop the DirectConnect server.
3. Copy or rename the file, as applicable.
4. Restart the server.

Update a running server

You can use snrfck to create or update a service name redirection file, validate the file, and send it to a running DirectConnect server.

Using this method allows you to replace the contents of the *snrf.tbl* file that is read when the server starts, write the contents to disk, and update the memory table so the changes take effect immediately.

❖ To update to a running server using *snrfck*

1. Use snrfck to create a new file or to modify the existing file and validate it.
2. Send the file to the server using the following syntax:

```
snrfck [-v][-?]|-h] [-t[-oresult]]
 [-Ssvc -Uuser -Aappl] -ifile
or
snrfck -c -Ssrv -Uuser -Ppwd ifile
```

where:

- -v displays the program version only.
- -? or -h displays this message.
- -t tests the update capability.
- -o*result* outputs the file for results of the update test (this has no effect if you do not specify -t).
- -S*svc*, -U*user*, and -A*appl* are optional arguments used to test the redirection search.
- -i*file* indicates the service name redirection file to be tested.
- -c submits the file to the server *srv* for an immediate update, using the specified login *pwd*.
- -S*srv* indicates the server name.
- -U*user* indicates the user name.
- -P*pwd* indicates the password for the user name.

CHAPTER 7 Log and Trace Files

This chapter describes DirectConnect log and trace files. Both file types provide troubleshooting information, but each is intended for a different audience. The distinctions are as follows:

- The log file is a collection of records intended primarily for the System Administrator.
- The trace file is a collection of records intended primarily for Sybase Technical Support personnel.

This chapter contains the following topics:

Log file description

The DirectConnect server provides several facilities for logging and reporting information. It uses these facilities during start-up, setup, and connection routing. The log file begins recording information each time you start the server and continues recording messages the entire time the server runs.

While the actual data in any log file depends upon the product and events, representative log file data can include:

- Performance data and timestamps
- Client connection activity

- Client messages
- Statistics
- SQL language, as received and after transformation
- Host communications
- Host server file information

The maximum size of any DirectConnect log record is 32,767 characters.

You can enable or disable logging on the following levels:

- DirectConnect server
- Service library
- Access service

Refer to the logging and tracing properties in Chapter 4, “Configuring the Server”

Trace file description

Tracing is a tool used by Sybase Technical Support to solve customer problems. In most situations, you enable tracing only in response to a request from Sybase Technical Support.

The actual data in any trace file depends upon the product. Representative trace file data can include:

- Logged messages
- Function entry and exit events
- Failure points
- Data passed between functional layers
- Data transformations

While you can control the degree of tracing through configuration properties, any level of tracing degrades system performance. For this reason, use tracing only in specific controlled situations.

Note The “Tracing” setting in the ODBC section of the *odbc.ini* file is to be set to `0`, for translation purposes. Setting this value to `1` causes a negative impact on performance.

An exception to this rule involves DirectConnect server start-up. If start-up fails, you may want to use the low-level failure details written to the trace file and attempt to solve the problem without Sybase Technical Support assistance.

Configuring logging and tracing properties

The DirectConnect server differentiates between log records and trace records. Each type of data is contained in a separate file. The files are maintained in U.S. English, using the native character set of the machine on which the server is running. However, client messages that are written to the log file appear in the client language.

Using DirectConnect Manager

You also can use DirectConnect Manager to make configuration changes dynamically to the Logging and Tracing properties.

For instructions on how to use DirectConnect Manager to change the server properties, go to the Managing Server Configuration topic of DirectConnect Manager online Help and select “Editing server configuration properties” and “Modifying server configuration properties.”

Using the text editor

You can configure logging and tracing properties by editing server or access service library configuration files. When you make such changes, you must stop the server, then restart it for the changes to take effect. For information about editing configuration properties, see Chapter 4, “Configuring the Server.”

Reading log and trace files

You can use a text editor to read log or trace records from the appropriate DirectConnect server subdirectory. Optionally, you can use the DirectConnect Manager to retrieve and read the record, described in the following section.

Using DirectConnect Manager to retrieve the server log file

DirectConnect Manager allows you to access the server log file, retrieve its messages, and view them in a text editor. You can retrieve the entire server log file, or set criteria to retrieve only a subset on the log file.

For instructions on how to use DirectConnect Manager to access the log file, go to the Managing Server Administration topic of DirectConnect Manager online Help and select “Filtering and retrieving the log.”

File location

The log file resides in the *log* subdirectory. The default log file name is *ServerName.log*, where *ServerName* is the name you assigned to the DirectConnect server during installation. A single log file contains log records from all access services.

The trace file *also* resides in the *log* subdirectory. The default trace file name is *ServerName.trc*, where *ServerName* is the name you assigned to the DirectConnect server during installation.

File structure

The log and trace files are ASCII text files. Each contains start-up data and configuration information in the header section at the beginning of the file.

The log file has a fixed size, which you can configure. If the LogWrap configuration property value is set to yes, the log file wraps when it reaches its configured maximum file size, writing over earlier records with new records. For more information, see Chapter 4, “Configuring the Server”.

The trace file does not have a size limit. If tracing is enabled, the file grows to consume all available disk space. Because limiting the file size can cause a potential loss of data that Sybase Technical Support may need for problem-solving, you cannot specify a maximum trace file size.

Log and trace records are recorded in chronological order. If multiple workstations use DirectConnect servers, the log or trace records for a particular user do not appear consecutively.

The logical end of the log file is indicated by an <END> marker.

Back-up log and trace files

Each time the DirectConnect server starts, it creates new log and trace files. The existing files are renamed as back-up files, using the following format:

mmddyyss.log

where:

- *mm* is a two-digit number, from 1 to 12, that indicates the month.
- *dd* is a two-digit number, from 1 to 31, that indicates the day.
- *yy* is a two-digit number, from 0 to 99, that indicates the year.
- *ss* is a two-digit number, that indicates seconds.

Note Be sure to delete or archive the backup files periodically to conserve disk space.

Log and trace record format

Log and trace records consist of a variable number of columns of data, separated by tab characters.

The following lists the columns of data in a typical log or trace record:

Record Type | DateTime | Object Name | SPID | UserID | Application Name | Specific Information

The following table describes the function of each of the columns.

Table 7-1: Log and trace record columns

Column	Description
Record Type	The configuration property name
DateTime	The date and time the record was published
Object Name	The name of the access service, access service library, or server that generated the record
SPID	The Open Server process ID (if applicable)
User ID	The user ID of the client connection that generated the record (if applicable)
Application Name	The name of the client application through which the client connected (if applicable)
Specific Information	The message text, which may contain embedded tabs to further separate the information in this column

If a access service library logs a message during its start or stop functions, client information is not available. In such cases, the Object Name column contains the access service library name, and the SPID, User ID, and Application Name columns read “NULL.”

Note Log messages that do not originate from the DirectConnect server or an access service library are generated by the access service library in the context of a client connection.

Pre-log start-up messages

The DirectConnect logging facility must have access to the server configuration file *server.cfg* before it can initialize. If the server configuration is invalid, or if an early start-up error occurs, messages are sent to the following substitute locations:

- In all cases, including Windows systems, the server sends messages to *stderr* (the console by default).
- On Windows systems only, messages are also written to the Windows event log.

Because these “pre-log” error messages do not appear in the log file, the most common messages, with explanations, are listed in the following sections.

Messages sent to the console

Any of the following messages can be sent to the console (*stderr*) when the DirectConnect server is started from the command line and encounters an error:

Table 7-2: Start-up messages sent to the console

Console messages	Description
`System info is invalid`	Message indicates that either the system environment variables are not working properly, or the system is low on memory or other resources.
`Memory allocation failure: property`	Message indicates that the system is out of memory.
`Could not load the configuration:{`*`filespec`*`}`	*filespecserver.cfg* The noted configuration file is missing, incorrectly named, in the wrong location, or corrupt. The message text displays the full path and file name of the expected file.

`Invalid configuration property value on line: {line_number}`
` The configuration is invalid: {filespec}`
	server.cfg
	One or more configuration properties contains an invalid value. The message text displays the full path and file name of the erroneous file.

Messages sent to the Windows event log

If a start-up error occurs when the DirectConnect server is started as a Windows service, the following messages are sent to the Windows event log:

Note Messages 2 through 9 indicate fatal errors that terminate the start-up process. Messages 10 and 11 are informational only.

Table 7-3: Start-up error messages sent to the Windows event log

Message	Description
`2 The DirectConnect server service could not be registered with the Windows service manager.`	Indicates an operating system error or a problem with the Windows Registry.

Message	Description
`3 DirectConnect server failure while reporting status to Windows service manager.`	Indicates an operating system error or a problem with the Windows Registry.
`4 DirectConnect server failure creating event for process thread.`	Indicates an operating system error or a system resource problem. Check whether excessive processes are presently running.
`5 DirectConnect server failure launching process thread.`	Indicates an operating system error or a system resource problem. Check whether excessive processes are presently running.
`6 DirectConnect server failure constructing system information.`	Either the system environment variables are not working properly, or the system is low on memory or other resources. Check the system path syntax and the *SYBASE*DSLISTEN and environment variables.
`7 DirectConnect server could not load the server configuration: <filespec> The file may be missing.`	One or more configuration properties contains an invalid value. If the server was started from a command line, the offending line number is indicated. If the server was started on a Windows system as a Windows service, run the product from the command line and add the -t switch to perform a start-up test. Doing this displays the full error information.
`9 DirectConnect server failure constructing the log manager.`	The log manager process could not be started. Make sure that the executable exists in the *\DC-12_6\bin* directory.
`10 DirectConnect server "{`*`server_name`*`}" started.`	This informational message logs when the server starts.
`11 DirectConnect server "{`*`server_name`*`}" stopped.`	This informational message logs when the server stops.

Note Messages 10 and 11 are written every time you start or stop the server. These records are not automatically erased. If you start and stop the server frequently, you should purge your Windows event log periodically.

Sample log records

The example in this section shows log records from a server start-up attempt. The example uses the following conventions:

- The first six columns of each record are omitted because these columns are virtually identical from record to record.
- All of the records are of type *LogHeader*, except the last, which is of type *LogEndHeader*.
- The line numbers are for the explanations that follow the example. The numbers do not appear in an actual log file.

The sample log file is as follows:

```
1 --------------------------------
2 DirectConnect 12.6 B
3 Copyright(c)2000, Sybase, Inc.
4 INTEL x386 Windows 4.1 (807)
5 (CRS 85.0) OPT 17-Mar-1997912:00:00
6-------------------------------
7 *** Initial configuration for: [SRVNAME] ***
8 --- MaxConnections = 2
9 --- RemoteSites = 3
10--- DefQueueSize = 1024
11--- CtlibPacketSize = 512
12--- OpenServerTraceFlags = 0
13--- ServiceRedirectionFile = snrf.tbl
14--- Description = The DirectConnect server.
15--- LogFileName =
16--- LogFileSize = 500000
17--- LogWrap = yes
18--- LogFlush = no
19--- LogToScreen = no
20--- LogClientLogin = yes
21--- LogClientMessages = 0
22--- TraceFileName =
23--- TraceLogMessages = yes
24--- TraceEntryExit = yes
25--- TraceAsync = yes
26--- TraceOther = yes
27--- TraceToScreen = no
28 Service Name Redirection enabled: C:\SQL10\
DC-12_6\SRVNAME\cfg\snrf.tbl
29*** The following localized message files are \
supported:
30--- C:\SQL10\DC-12_6\SRVNAME\locales\
us_english\cp850\server.loc
31 Loading service library file: C:\SQL10\
DC-12_6\SRVNAME\svclib\shutdown.dll
32***Initial configuration for: [Shutdown] ***
33***Initial configuration for: [shutdown] ***
```

```
34--- EnableAtStartup = yes
35 Service loaded: [shutdown]
36*** The following localized message files are \
supported:
37--- C:\SQL10\DC-12_6\SRVNAME\locales\
us_english\cp850\shutdown.loc
38 Successfully initialized service library: \ Shutdown
39 Loading service library file: C:\SQL10\
DC-12_6\SRVNAME\svclib\smtest1.dll
40*** Initial configuration for: [smtest1] ***
41---UserExitFileName = userexit
42*** Initial configuration for: [ServiceA] ***
43--- MaxClients = 10
44--- EnableAtStartup = yes
45--- StopCondition = error
46--- MaxRowCount = 1000
47--- LogReceivedSQL = yes
48 Service loaded: [ServiceA]
49 Service loaded: [ServiceB]
50*** The following localized message files are \
supported:
51--- C:\SQL10\DC-12_6\SRVNAME\
locales\us_english\cp850\smtest1.loc
52 Successfully initialized service library: \ smtest1
53 !READY! Waiting for connections.
```

Following are explanations of the log record entries, by line number:

5: A mnemonic indicates the build or version of the library that was linked with the executable.

7: The server name appears in the brackets.

8-27: The start-up values for the server configuration properties are listed.

19: The system sends log records to the log file but not to the current window.

28: An indication whether service name redirection is to be used, and if so, the path to the file that was loaded.

29-30: The localized message files found for the server and the supported locales are shown.

31, 39: Each access service library module installed in the *\DC-12_6\ServerName\svclib* subdirectory is loaded in turn.

32: The access service library properties are listed. In this example, the [Shutdown] access service library does not have configurable properties.

33-34: The initial configurations of the enabled access services associated with the access service library are listed. In this example, the [shutdown] access service defines one configuration property: EnableAtStartup.

35: An indication that initialization for the specified access service is complete.

36-37, 50-51: The localized message files found for the associated access service library and the supported locales are listed.

38: An indication that initialization for the specified access service library is complete.

49: In this example, [ServiceB] was loaded but not enabled, nor is it able to receive connections. This is noted by the lack of an initial configuration listing (initial access service configuration is always output when the access service becomes enabled). Because this access service was not enabled at start-up, you can enable it only with DirectConnect Manager tool.

53: An indication that server initialization is complete. Clients can now connect to any enabled access service.

Sample trace records

In this example, the information shown after the system-supplied information is free-form. The trace records are separated by tabs so you can easily import them into most query tools.

```
TraceEntryExit 06/30/1995 16:35:57.641 SRVNAME NULL
NULL NULL > evm_StartHandler

TraceEntryExit 06/30/1995 16:35:57.651 SRVNAME NULL
NULL NULL > smServer::LoadSvclib: [C:\sql10\
DC-12_6\SRVNAME\svclib\sample1.dll] linked with
DirectConnect v10.5.0 lib:smr

TraceEntryExit 06/30/1995 16:35:57.771 SRVNAME NULL
NULL NULL > smSvclib::InitCriticalBase

TraceEntryExit 06/30/1995 16:35:57.801 SRVNAME NULL
NULL NULL > smServer::AddSvclib: [Sample1]

TraceEntryExit 06/30/1995 16:35:57.801 SRVNAME NULL
NULL NULL < smServer::AddSvclib

TraceEntryExit 06/30/1995 16:35:57.821 SRVNAME NULL
NULL NULL < smSvclib::InitCriticalBase
```

```
TraceEntryExit 06/30/1995 16:35:57.821 SRVNAME NULL
NULL NULL < smServer::LoadSvclib: [Sample1]


TraceEntryExit 06/30/1995 16:35:57.831 SRVNAME NULL
NULL NULL > smServer::LoadSvclib: [C:\sql10\DC-
12_6\SRVNAME\svclib\sample2.dll] linked with
DirectConnect v10.5.0 lib:smr

TraceEntryExit 06/30/1995 16:35:57.931 SRVNAME NULL
NULL NULL > smSvclib::InitCriticalBase

TraceEntryExit 06/30/1995 16:35:57.961 SRVNAME NULL
NULL NULL > smServer::AddSvclib: [Sample2]

TraceEntryExit 06/30/1995 16:35:57.961 SRVNAME NULL
NULL NULL < smServer::AddSvclib

TraceEntryExit 06/30/1995 16:35:57.991 SRVNAME NULL
NULL NULL < smSvclib::InitCriticalBase

TraceEntryExit 06/30/1995 16:35:57.991 SRVNAME NULL
NULL NULL < smServer::LoadSvclib: [Sample2]

TraceEntryExit 06/30/1995 16:35:57.991 SRVNAME NULL
NULL NULL < evm_StartHandler: !READY!
```

CHAPTER 8

Managing Server Security with DirectConnect Manager

This chapter describes how DirectConnect Manager manages DirectConnect security access, even from a remote site.

This chapter contains the following topics:

Description of DirectConnect security

DirectConnect security uses a user ID/password combination, coupled with a user level, to determine access.

The user level determines the amount of administration functionality that is available to the user. This function is implemented in DirectConnect Manager, as well as at the Administrative Service Library level. The level of access is granted at two levels: “monitor” and “monitor plus change.” These two levels are also referred to as “user” and “admin,” respectively.

Note Servers that do not support security allow full access to all connections.

Security for DirectConnect is implemented using an encrypted password that is stored on the *user.pwd* file of the Administrative Service Library.

The first time the user connects to the Administrative Service Library, the security program detects that the *user.pwd* file does not exist. As a result, the Administrative Service Library creates a *user.pwd* with the following two entries:

Table 8-1: User.pwd file

User ID	Password
sa	
Admin	Password

The entries on the previous table allow you to access the system using the original "sa" user ID without a password. However, if you use DirectConnect Manager to modify the "sa" user ID, a password is required. When you use the current version of DirectConnect Manager to add new users, the new entries are added to the previous list in the table and are stored in the *user.pwd* file in the *cfg* directory for the DirectConnect server.

Note Keep in mind that while the ability of DirectConnect to automatically create *user.pwd* files is convenient for backward compatibility, you need to limit access to this file using standard file security techniques.

Using DirectConnect Manager for security tasks

The DirectConnect Manager tool provides an easy interface for security checks. It filters which user has view-only access, and which user has full access and administrative functions.

For instructions on how to use DirectConnect Manager, go to the Managing Administrators topic of DirectConnect Manager online Help and select "Edit server administrators."

Troubleshooting

This section covers security file problems:

- If you discover that your *user.pwd* security file has problems due to corruption or user error, you can delete the *user.pwd* file, and it will be recreated with default passwords. You can do this while the server is still running.

- After the file has been recreated, you can use DirectConnect Manager to reenter the user information. An alternative would be to keep a back-up copy of the file, then copy over the *user.pwd* file with the backup file of known users.

Glossary

access management

A DirectConnect feature that provides connectivity to non-Sybase targets.

access service

The named set of properties used with a DirectConnect access service library, to which clients connect. Each DirectConnect server can have multiple access services.

access service library

A component of DirectConnect. A service library that provides access to non-Sybase data contained in a database management system or other type of repository. Each such repository is called a “target.” Each access service library interacts with exactly one target and is named accordingly. See also **service library**.

ACSLIB

See **access service library**.

Adaptive Server Enterprise

The server in the Sybase Client-Server architecture. It manages multiple databases and multiple users, tracks the actual location of data on disks, maintains mapping of logical data description to physical data storage, and maintains data and procedure caches in memory.

administrative service library

A service library that provides remote management capabilities and server-side support. It supports a number of remote procedures (invoked as RPC requests) that enable remote DirectConnect management. See also **remote procedure call**, **service library**.

ADMLIB

See **administrative service library**.

bulk copy transfer

A transfer method in which multiple rows of data are inserted into a table in the target database. See also **transfer**. Compare with **destination-template transfer**.

catalog stored procedure

A stored procedure that provides information about tables, columns, and authorizations. It is used in SQL generation and application development.

character set

A set of specific (usually standardized) characters with an encoding scheme that uniquely defines each character. ASCII is a common character set.

Client-Library™

A Sybase library of routines that allows a client application to interact with Open Server.

client/server An architecture in which the client is an application that handles the user interface and local data manipulation functions, while the server provides data processing access and management for multiple clients. See also **client**, **server**, **client application**.

code page An assignment of graphic characters and control function meanings to all code points.

code set See **character set**.

configuration file A file that specifies the characteristics of a system or subsystem.

configuration set A section into which service library configuration files are divided.

connection A network between two systems. For SNA, the path connects a logical unit on one machine to a logical unit on another machine. For TCP/IP, the path connects TCP modules on separate machines.

connection code page The code page to which DirectConnect converts data when the specified code page is not supported by the database. If the DirectConnect configuration contains a default code page for applications, that code page is used. Otherwise, the system code pages from the initialization file are used. See also **code page**.

CSP See **catalog stored procedure**.

CT-Library See **Client-Library**.

database management system A computer-based system for defining, creating, manipulating, controlling, managing, and using databases.

datatype A keyword that identifies the characteristics of stored information on a computer.

DB-Library A Sybase and Microsoft API that allows client applications to interact with Open Data Services applications.

DBMS See **database management system**.

destination-template transfer A transfer method in which source data is briefly put into a template where the user can specify that some action be performed on it before execution against a target database. See also **transfer**. Compare with **bulk copy transfer**.

DirectConnect A Sybase Open Server-based application that provides access management for non-Sybase databases, copy management, and remote systems management. Each DirectConnect consists of a server and one or more service libraries to provide access to a specific data source. DirectConnect replaces the MDI Database Gateway™ and OmniSQL Access Module™.

DirectConnect Director (DCDirector) A DirectConnect server that performs the sole function of creating, starting, and stopping other servers on the same machine.

DirectConnect Manager A Sybase application for Microsoft Windows that provides remote management capabilities for DirectConnect products. These capabilities include starting, stopping, creating, and copying services.

DirectConnect for z/OS A Sybase LAN-based solution that communicates with mainframe host components. It incorporates the functionality of the MDI Database Gateway and the Sybase Net-Library and includes LU 6.2 and TCP/IP support.

DirectConnect server The component that provides general management and support functions (such as log file management) to service libraries.

DirectConnect service A named set of properties, used with a DirectConnect service library, to which clients connect.

DirectConnect Access Service Library The component that provides a set of functions within the DirectConnect server environment.

dll See **dynamic link library**.

dynamic link library A file containing executable code and data bound to a program at load time or run time, rather than during linking. The code and data in a dynamic link library can be shared by several applications simultaneously.

environment variable A variable that describes how an operating system runs and the devices it recognizes.

event log In Windows, the central point for all system and application error and information messages.

Extended Binary-Coded Decimal Interchange Code A coded character set of 256 eight-bit characters.

globalization The combination of internationalization and localization. See also **internationalization**, **localization**.

graphical user interface

A type of computer interface consisting of a visual metaphor of a real-world scene, often of a desktop. Within that scene are icons, representing actual objects, that the user can access and manipulate with a pointing device.

GUI

See **graphical user interface**.

initial program load

For some operating systems, the initialization procedure that causes the system to commence operation.

interfaces file

A UNIX operating system file that must be available on each machine from which connections to DirectConnect or other Sybase products are made. Each entry in the file determines how the host client software connects to the Sybase product. See also **libtcl file**. Compare with **sql.ini file**.

International Organization for Standardization

An organization of national standards bodies from various countries established to facilitate international exchange of goods and services and develop cooperation in intellectual, scientific, technological, and economic activity.

internationalization

The process of extracting locale-specific components from the source code and moving them into one or more separate modules, making the code culturally neutral so it can be localized for a specific culture. See also **globalization**. Compare with **localization**.

IPL

See **initial program load**.

ISO

See **International Organization for Standardization**.

iso_1 charset

A base translation table that corresponds to the ASCII 0819 or the EBCDIC 0500 character sets.

library

A named disk area that can contain programs and related information. A library consists of different sections, called library members.

libtcl file

A file containing a list of installed Net-Library drivers. Client applications use information in this file and the *interfaces* file (UNIX systems) or the *sql.ini* file (Windows systems) to connect to Adaptive Server or Open Server applications. See also **interfaces file**, **sql.ini file**.

LDAP

Lightweight Directory Access Protocol (LDAP) is a protocol for accessing online directory services.

localization The process of preparing an extracted module for a target environment, in which messages are displayed and logged in the user's language, numbers, money, dates, and time are represented using the user's cultural convention; documents are displayed in the user's language. See also **globalization**. Compare with **internationalization**.

MainframeConnect for DB2 UDB A Sybase mainframe solution that provides dynamic access to DB2 UDB data. It replaces the OmniSQL Access Module for DB2 UDB and the functionality of the MDI Access Server™.

Net-Library A Sybase product that lets PC applications become clients of Adaptive Server or Open Server. See also **client**, **Open Server**, **SQL Server**.

ODS See **Open Data Services**.

OLTP See **Online Transaction Processing**.

OmniConnect An add-on product for Sybase Adaptive Server that provides a Transact-SQL® interface to external data sources, including host data files and tables in other database systems. OmniConnect replaces "OmniSQL Gateway" and "OmniSQL Server."

Online Transaction Processing A system that supports database maintenance through an interactive user interface. The user usually requires a response time of less than three seconds.

Open Client A Sybase product that provides customer applications, third-party products, and other Sybase products with the interfaces required to communicate with Open Client and Open Server applications.

Open ClientConnect A Sybase product that allows mainframe clients to send requests to SQL Server, Open Server, MainframeConnect for DB2 UDB, and Open ClientConnect using Client-Library. Open ClientConnect provides capability for the mainframe to act as a client to LAN-based resources.

Open Data Services A product that provides a framework for creating server applications that respond to DB-Library clients. See also **DB-Library**.

Open Database Connectivity A Microsoft API that allows access to both relational and nonrelational databases.

Open Server A Sybase product that provides the tools and interfaces required to create a custom server.

Open ServerConnect™ A Sybase product that provides capability for programmatic access to mainframe data.

parameter

A variable that is given a constant value for a specified application and that can denote the application. Compare with **property**.

Partner Certification Reports

Sybase publications that certify third-party query and development tools to work with Sybase products.

property

A setting for a server or service that defines characteristics, such as how events are logged or how datatypes are converted. Compare with **parameter**.

protocol

A set of rules that governs the behavior of computers communicating on a network.

Registry

The part of the Windows operating system that holds configuration information for a particular machine.

remote procedure call

A stored procedure executed on a different DirectConnect server from the one onto which a user is logged or on which the initiating application resides.

remote stored procedure

A customer-written CICS program that resides on the mainframe and communicates with MainframeConnect for DB2 UDB. Compare with **client services application**.

remote systems management

A feature that allows a System Administrator to manage multiple DirectConnect servers and multiple services from a client.

RPC

See **remote procedure call**.

RSP

See **remote stored procedure**.

Secure Sockets Layer (SSL)

Provides customers with a secure mode of transport. It includes encryption and authenticating clients and servers using digital certification.

server

A functional unit that provides shared services to clients over a network. See also **client/server**. Compare with **client**.

service

A functionality available to DirectConnect applications. It is the pairing of a service library and a set of specific configuration properties.

service library

A set of configuration properties that determine service functionality. Examples of service libraries include access service libraries, administrative service libraries, and transaction router service libraries. See also **access service library**, **administrative service library**, and **Transaction Router Service library**.

service name redirection

A type of service name resolution that allows a system administrator to map alternative connections to services. See also **service name resolution**. Compare with **direct resolution**.

service name redirection file The default name of the file used for the service name redirection feature. See also **service name redirection**.

service name resolution The DirectConnect server mapping of an incoming service name to an actual service. See also **direct resolution**, **service name redirection**.

SNA See **Systems Network Architecture**.

SNRF See **service name redirection file**.

sqledit A utility for creating and editing *sql.ini* files and file entries. See also **sql.ini**.

sql.ini file The Windows file containing definitions for each DirectConnect server to which a workstation can connect. The file must be on every client machine that connects to Adaptive Servers. See also **libtcl file**. Compare with **interfaces file**.

SSL See *Secure Sockets Layer*.

System Administrator The user in charge of server system administration. For DirectConnect, the user responsible for installing and maintaining DirectConnect servers and service libraries.

Systems Network Architecture An IBM proprietary plan for transmitting information units through networks and controlling network configuration and operation.

TCP/IP See **Transmission Control Protocol/Internet Protocol**.

Transaction Router Service The DirectConnect program that accepts requests from workstation-based clients and routes them to Open ServerConnect.

Transaction Router Service library A service library that facilitates access to remote transactions, allowing customers to execute thousands of transactions from virtually any mainframe data source. See also **service library**.

transfer A DirectConnect feature that allows users to move data or copies of data from one database to another. See also **bulk copy transfer**, **destination-template transfer**.

Transmission Control Protocol/Internet Protocol (TCP/IP) A set of communication protocols that supports peer-to-peer connectivity functions for both local and wide area networks.

TRS See **Transaction Router Service**.

TRS Library See **Transaction Router Service library**.

UNIX An operating system developed by Bell Laboratories that allows for multiple concurrent programs and users.

utility program A computer program that supports computer processes, such as diagnostic programs, trace programs, and sort programs.

wild card A special character that represents a range of characters in a search pattern.

Index