Server Administration Guide

Enterprise Connect™ Data Access and Mainframe Connect™

15.0

[Microsoft Windows, Linux, and UNIX]

DOCUMENT ID: DC35365-01-1500-01

LAST REVISED: August 2007

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

Contents

About This Book

This book describes how to use a Sybase® Enterprise Connect™ Data Access Option for ODBC and Mainframe Connect™ DirectConnect™ for z/OS Option 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.

Note This guide is *not* for the ECDA Option for Oracle. For Oracle administration and user information, see Enterprise Connect Data Access Option for Oracle *Administration and User Guide*.

Audience

This book is written for:

- Application Programmers, who develop organization-specific programs using the major features of ECDA.
- System Administrators, who install and test ECDA. When ECDA is running, System Administrators provide ongoing administration support, disaster recovery, and troubleshooting support.
- System Programmers, who install and test ECDA. 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 ECDA Option for ODBC and Mainframe Connect DirectConnect for z/OS Option 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 ECDA as a Windows Service" describes how to set up Secure Sockets Layer (SSL) on a DirectConnect server, and how to set up and run the DirectConnect server 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 and Mainframe Connect Options, see the appropriate *Installation Guide*.

To configure and administer ECDA and Mainframe Connect Options, use the following guides:

- The appropriate Enterprise Connect Data Access *Users Guide for Access Services* for your database system
- Mainframe Connect DirectConnect for z/OS Option *Users Guide for Transaction Router Service*, if applicable
- Enterprise Connect Data Access Option for Oracle *Administration and User Guide*

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
- Mainframe Connect Client Option and Mainframe Connect Server Option *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 SyBooks™ CD, and the Sybase 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 SyBooks CD. It is included with your software. To read or print documents on the Getting Started CD, you need Adobe Acrobat Reader, which you can download at no charge from the Adobe Web site using a link provided on the CD.
- The SyBooks CD contains product manuals and is included with your software. The Eclipse-based SyBooks browser allows you to access the manuals in an easy-to-use, HTML-based format.

  Some documentation may be provided in PDF format, which you can access through the PDF directory on the SyBooks CD. To read or print the PDF files, you need Adobe Acrobat Reader.

  Refer to the *SyBooks Installation Guide* on the Getting Started CD, or the *README.txt* file on the SyBooks CD for instructions on installing and starting SyBooks.
- The Sybase Product Manuals Web site is an online version of the SyBooks 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 Sybase 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. Click Certification Report.
3. In the Certification Report filter select a product, platform, and timeframe and then click Go.
4. Click a Certification Report title to display the report.

❖ Finding the latest information on component certifications

1. Point your Web browser to Availability and Certification Reports at http://certification.sybase.com/.
2. Either select the product family and product under Search by Base Product; or select the platform and product under Search by Platform.
3. Select Search to display the availability and certification report for the selection.

❖ 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.

Accessibility features

This document is available in an HTML version that is specialized for accessibility. You can navigate the HTML with an adaptive technology such as a screen reader, or view it with a screen enlarger.

Enterprise Connect Data Access and the HTML documentation have been tested for compliance with U.S. government Section 508 Accessibility requirements. Documents that comply with Section 508 generally also meet non-U.S. accessibility guidelines, such as the World Wide Web Consortium (W3C) guidelines for Web sites.

Note You might need to configure your accessibility tool for optimal use. Some screen readers pronounce text based on its case; for example, they pronounce ALL UPPERCASE TEXT as initials, and MixedCase Text as words. You might find it helpful to configure your tool to announce syntax conventions. Consult the documentation for your tool.

For information about how Sybase supports accessibility, see Sybase Accessibility at http://www.sybase.com/accessibility. The Sybase Accessibility site includes links to information on Section 508 and W3C standards.

See Section 508 compliance statement for Enterprise Connect Data Access for Voluntary Product Assessment Templates at http://www.sybase.com/detail_list?id=52484.

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 guide covers the ECDA Option for ODBC and Mainframe Connect DirectConnect for z/OS Option. For information about the ECDA Option for Oracle, see the ECDA Option for Oracle *Server Administration and Users Guide*.

Note In ECDA version 15.0, the Option for DB2 UDB and the Option for Microsoft SQL Server have been merged into the ECDA Option for ODBC. In addition, the Option for Informix is no longer available.

ECDA Option for ODBC

The ECDA Option for ODBC provides access management, copy management, and remote systems management. It consists of:

- The DirectConnect server, which provides management and support functions for DirectConnect service libraries
- An access service library, which accesses data from a particular target database, including DB2 UDB, Microsoft SQL Server, and other ODBC-accessible databases
- Access services, which contain specific sets of configuration properties that relate to the target to be accessed and define how each access service behaves

Using the IBM Distributed Relational Database Architecture (DRDA) protocol, ECDA Option for ODBC supports access to DB2 UDB on z/OS, Windows, Linux, and UNIX platforms.

For more information about ECDA architecture, see the Enterprise Connect Data Access *Overview Guide*.

ODBC driver

ECDA Option for ODBC provides basic connectivity to non-Sybase data sources, using the ODBC back-end (server-side) driver that you purchase for your target database, such as IBM or Microsoft SQL. Following the vendor's instructions, you install the ODBC driver on the same server as ECDA Option for ODBC and then configure ECDA Option for ODBC to use that ODBC driver for access to your database.

Note Be sure to verify that your ODBC driver will be compatible with Sybase driver manager software.

Because ODBC drivers have varying degrees of functionality, it is important that when working with non-Sybase-provided, third-party ODBC drivers, you carefully integrate and test them to be sure they meet your needs.

How DirectConnect server 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:

- Accessing the service directly, 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.
- Accessing the service with 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 properties for ECDA Option for ODBC

You can configure ECDA Option for ODBC 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, which define a particular access service and 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.

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 ECDA Option for ODBC *Users 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.

Server external files

The DirectConnect server manages external files that reside in various subdirectories. For information on the ECDA Option for ODBC 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.

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 *<install_dir>/DC-15_0/svclib* subdirectory for UNIX, or the *C:\<install_dir>\DC-15_0\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 *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.

Mainframe Connect options

This section describes options that DirectConnect for z/OS Option interacts with to provide mainframe access for LAN client requests.

Mainframe Connect DB2 UDB Options

The Mainframe Connect DB2 UDB Option for CICS and the Mainframe Connect DB2 UDB Option for IMS can work with Mainframe Connect DirectConnect for z/OS Option to provide access to mainframe data. They perform these functions:

- Supports full read-write, dynamic SQL access to data
- Allows applications to use cursors for flexible and efficient result-set processing
- Permits the use of long-running transactions against mainframe databases
- Allows applications to use dynamic events to map SQL to a static plan

DirectConnect for z/OS Option invokes the Mainframe Connect DB2 UDB Option to access mainframe data on behalf of its Open Client-based clients, such as:

- ASE/CIS
- ASE through RPCs
- Enterprise Application Server
- JDBC or ODBC applications
- Replication Server

Note The Mainframe Connect for DB2 UDB Option for CICS and the Mainframe Connect for DB2 UDB Option for IMS are hereafter referred to as the DB2 UDB Options for CICS and for IMS.

Mainframe Connect Server Option

The Mainframe Connect Server Option is a programming environment that allows you to create mainframe transactions that are accessible to Sybase client applications. To provide this access, Mainframe Connect Server Option uses traditional Open Server APIs.

These transactions provide access to virtually any CICS and IMS data source and are used for a variety of functions, including:

- Accessing existing mainframe applications
- Initiating mainframe batch jobs

- Providing source data for data transfer operations
- Providing data mapped to a table within ASE/CIS, thus allowing results to be accessed or joined with data from other targets

LAN-side client applications access Mainframe Connect Server Option transactions directly through DirectConnect for z/OS Option or indirectly through ASE/CIS or a Sybase Adaptive Server RPC.

Mainframe Connect Client Option

The Mainframe Connect Client Option is a programming environment that allows you to create mainframe applications that access:

- LAN data residing on a Sybase Adaptive Server or other supported data sources
- Other CICS regions

To provide this access, the Mainframe Connect Client Option uses traditional Open Client APIs.

The Mainframe Connect Client Option allows you to treat the mainframe as if it were just another node on a LAN.

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 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.

You can install DirectConnect Manager and its required components from the DC Client CD.

Note When you install a DirectConnect product on a Windows or UNIX platform or machine, you may install DirectConnect Manager on a separate platform or machine. This allows you to control any DirectConnect from any machine.

Globalization

Globalization consists of internationalization and localization of messages.

Internationalization

Internationalization consists of character code set conversion and cultural formatting.

Code set conversion involves converting the hexadecimal representation of a character from a code set in a target database to a code set in a client application, or the reverse.

Cultural formatting involves designating decimal separators, monetary signs, date and time separators, and a 3-digit grouping symbol. Cultural formatting in DirectConnect is performed through the use of configuration properties.

Code page translation

For ODBC-based products, code page translation can take place in two locations:

- Between DirectConnect and the target database
- Between the client and DirectConnect server

For additional information about code page translation, refer to Chapter 2, “Configuring the Access Service Library for DirectConnect” in the ECDA Option for ODBC *Access Service Users Guide*.

Localization

Two sets of messages can be localized:

- Messages generated by the target database manager and passed to the client application without change

  The target database manager can be any application between the DirectConnect server and the target data file, including the ODBC driver.

- Messages generated in ECDA Option for ODBC

  ECDA Option for ODBC does not localize database manager messages. For information on how to set up localization of such messages, see your database manager and the ODBC driver documentation.

CHAPTER 2 Creating a Server

Types of DirectConnect servers

There are two kinds of DirectConnect servers:

- DCDirector server—a server that performs an administrative role in DirectConnect Manager over other servers that you associate with it.
- DirectConnect server—which can be “directed” by DCDirector server or not. Regardless, it can be managed using DirectConnect Manager, but at different levels depending on whether it’s “directed” or not.

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

Why use a DCDirector server?

A DCDirector server allows you to have more control over servers you manage in DirectConnect Manager. It contains only DirectConnect servers that reside in the same directory and performs the sole function of creating, starting, and stopping the servers.

Table 2-1describes which functions you can administer to directed and non-directed servers using DirectConnect Manager.

Table 2-1: Functions supported for directed and non-directed servers

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

Using a DCDirector server also gives you a logical view of a group of servers in DirectConnect Manager. This is a typical view, including server name, machine name, and port number:

```
DirectConnect
    Director name (machine A)
        server name 1 (machine, 4113)
        server name 2 (machine, 4114)
    Director name (machine B)
        server name 1 (machine, 4115)
        server name 2 (machine, 4116)
```

Creating servers

You can create servers using different methods:

- You can create a DCDirector server using DirectConnect Manager, the DCDirector utility, or by modifying the *server configuration* file.
- You can create a *directed* DirectConnect server using DirectConnect Manager or the AddServer utility.
- You can create a *non-directed* DirectConnect server *only* using the AddServer utility.

Using DirectConnect Manager

When you use DirectConnect Manager to create a directed DirectConnect server, it:

- 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.

Note Before you can use DirectConnect Manager, you must install DirectConnect Manager as outlined in the Enterprise Connect Data Access *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 DirectConnect Manager to a DirectConnect Server."

Using ECDA utilities

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

These utilities are C shell scripts (on UNIX) and batch files (on Windows) that can be found in these directories:

- */<install_dir>/DC-15_0/bin* (UNIX)
- *C:\<install_dir>\DC-15_0\bin* (Windows)

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 utilities are:

- DCDirector—creates and starts a DCDirector server
- AddServer—creates a new DirectConnect server

For more information about DirectConnect utilities, see the appendix in the Mainframe Connect DirectConnect for z/OS Option *Installation Guide*, and the Enterprise Connect Data Access *Installation Guide,* for your platform.

Modifying the server configuration file

For details on modifying the *server configuration* file, see Chapter 4, "Configuring the Server."

Note If you modify the *server configuration* file you must stop and restart the server for the changes to take affect

---

Creating and starting a DCDirector server

Using DirectConnect Manager

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

- At

  C:\<install_dir>\DC-15_0\bin (Windows)

  or

  /<install_dir>/DC-15_0/bin (UNIX)

  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.

Modifying the configuration file

To designate a server as a DCDirector server in the server configuration file *server.cfg*, use the IsDCDirector property, as defined in Chapter 2, "Creating a Server."

Creating a DirectConnect server

Using DirectConnect Manager

Using DirectConnect Manager to create a server creates a *directed* server. You must use the AddServer utility to create a *non-directed* server.

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

Note Before you can use DirectConnect Manager, you must have installed DirectConnect Manager as outlined in the Enterprise Connect Data Access *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 the *AddServer* utility

You can use the AddServer utility to create a directed or a non-directed server. To use the AddServer utility, refer to Appendix B of the Enterprise Connect Data Access *Installation Guide* for your platform.

AddServer 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 directed and non-directed server names or ports.

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, if the new server is configured for a snrf.tbl table (see Chapter 6, “Using Service Name Redirection”), and the table does 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

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 */<install_dir>/DC-15_0/DC_SYBASE.csh* (UNIX) file or run the appropriate *C:\<install_dir>\DC-15_0\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 *C:\<install_dir>\DC-15_0\bin* (Windows) or */<install_dir>/DC-15_0/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 ECDA utilities, see the Appendix in the Mainframe Connect DirectConnect for z/OS Option *Installation Guide*, or the Enterprise Connect Data Access Options *Installation Guide*, 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 directed by 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 DirectConnect Manager online Help and select Server Administration | Stopping a Server.

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

Using the 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 suspends 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

❖ **To stop the server using the command line**

- If you started the server from a command line, press Ctrl + C to stop the server.

❖ **To stop the server using Windows**

- If you started the server as a Windows service, perform the following steps to stop it:
  1. From the Control Panel, select Services.
  2. In the drop-down list, select the DirectConnect *server_name*.
  3. 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

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

❖ **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 Enterprise Connect Data Access 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 exit codes in Table 3-1.

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 "* * 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-15_0* subdirectory.

Typical start-up errors include the following:

- Configuration file is corrupt.

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

Access service configuration error conditions

Typical access service configuration errors are:

- 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 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.

CHAPTER 4 Configuring the Server

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

Modifying the configuration file

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.

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

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.

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 ([ ]) square 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
 LogLicense Messages=no
 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*.
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

Note The order of the configuration properties within each sub-section can be changed.

The subsection heading and configuration property list can 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

Values

- no indicates there is no configuration file for the service library, the server will not load a service library.
- yes the server loads all service libraries and creates a new configuration file for each service library 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

Values

- 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.

Comments

- 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

Value	integer (Commas are not allowed.)

Description

Describes the DirectConnect server.

Syntax	Description=*text*
Range	0–255 (characters)
Default	none
Value	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
Values	• 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
Value	integer is the maximum number of clients that you can connect to Open Client.

Comments

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

  For *UNIX only*, 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 ECDA 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 ECDA, and must set the environment variables for ECDA 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.

    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 ECDA at a peak time, use the following formula to determine the correct number of file descriptors that your UNIX administrator will configure for the ECDA 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 ECDA.

- *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 ECDA process. To meet this requirement, your UNIX administrator needs to increase the number of file descriptors available for the ECDA process.

NetBufSize

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

Syntax NetBufSize=*integer*

Range 512–32768

Default 2048

Value integer is the maximum size of *CT-Library* packets.

OSCodeSetConversion

Enables or disables Open Server code page conversion. Must be yes if client code page is different than the Open Server and Direct Connect server's operating system code page.

Syntax OSCodeSetConversion= [no | yes]

Default no

Values

- yes enables Open Server to convert client code page to Open Server code page that must match the operating system code page.
- no disables Open Server code page conversion between the DirectConnect server and the client.

Comment

Most ECDA 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, 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

Value

integer is the number of Adaptive Servers that can connect simultaneously.

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

Value

filename.ext is a value for this property 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 ECDA to check for all the configured access services.

Syntax

SSLEnabled= [yes | no]

Default

no

Values

yes causes ECDA 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 Enterprise Connect Data Access *Installation Guides* for your platform.

ECDA searches these files that are displayed for UNIX (for Windows use the appropriate environment variables):

- */<install_dir>/DC-15_0/server/certificates*
- /<*install_dir*>/DC-15_0/certificates
- /<*install_dir*>/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, ECDA does not start.

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

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 ECDA 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 ECDA, Sybase recommends using only one SSL access service for each DirectConnect server. This prevents a user from using an unsecured port to gain access to unsecured data within an organization.

Note ECDA does not support "transfer to" and "transfer from" SSL-enabled ASE servers.

SSLServices

Identifies the access or TRS service that will use SSL.

Syntax

SSLServices= *service name*

Default

none

Value

service name is a valid TRS or access service name that exists in ECDA. 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

ECDA 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

Value

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 can appear in the server configuration file as follows:

```
{Logging}
 LogClientLogin
 LogClientMessages
 LogFileName
 LogFileSize
 LogFlush
 LogLicenseMessages
 LogOCOSMessages
 LogToScreen
 LogWrap
```

LogClientLogin

Determines whether to log connection activity.

Syntax

LogClientLogin=[no | yes]

Default

no

Values

- yes the log reports the results of connection successes, connection failures, and access service name redirection results
- no the connection activity is not logged.

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

Value severity when set to any integer, the system logs messages that have a severity level greater than or equal to the specified value.

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

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*

Value *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

Value integer indicates the maximum size of the *log* file. (Commas not allowed).

Comments

- 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 37.
- For information on logging, see Chapter 7, “Log and Trace Files”.

LogFlush

Specifies when the system writes each log record.

Syntax LogFlush=[no | yes]

Default no

Values

- yes the system writes each log record as it is generated.
- no the system buffers log records and writes them periodically for optimal performance.

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.

LogLicenseMessages

Enables SySAM License manager message or only errors to the log.

Syntax LogLicenseManagerh=[no | yes]

Default no

Values

- yes allows verbose License manager messages.
- no reports only errors to the log.

LogOCOSMessages

When set to 0 (zero) the system does not log error messages generated internally by Open Client and Open Server. Message severities fall in the range of 0–24, inclusive, matching the levels defined for Adaptive Server.

Syntax

LogOCOSMessages=*severity*

Range

0–24

Default

1

Value

severity causes the system to log severity levels equal to or greater than the value indicated.

Comments

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

LogToScreen

Specifies where the system log output is sent.

Syntax

LogToScreen=[no | yes]

Default

no

Values

- yes the system sends log output to the console and log file
- no the system sends log output to the log file only.

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 ECDA 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 ECDA 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 can 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
Value	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
Value	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
Value	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

Value 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

Value 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

Value 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

Value 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
Value	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
Value	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
Value	yes – asynchronous event tracing is enabled.

TraceEntryExit

Traces entry and exit of major ODBC API functions and from most ECDA internal functions.

Syntax TraceEntryExit=[no | yes]

Default no

Value 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.

Value *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

Value 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

Value

bitflags is an integer that corresponds to the SRV_S_TRACEFLAG trace property in Open Server.

TraceOther

Allows you to trace data for debugging.

Syntax

TraceOther=[no | yes]

Default

no

Value

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

Value

- 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 37.

CHAPTER 5

Setting up SSL and ECDA as a Windows Service

Setting up SSL on the DirectConnect server

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

SSL is supported only for client access to ECDA options for ODBC and Mainframe Connect DirectConnect for z/OS Option. It is *not* supported to the target databases.

Setting up SSL on the Windows server

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

To set up the ECDA Option for ODBC 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 ECDA certificates directory, enable SSL, and verify the log files

---

Note ECDA 15.0 does not support "transfer to" and "transfer from" SSL-enabled ASE servers.

❖ To create the certificate of authority files

1 Add the following to the path of the environment variable in *C:\<install_dir>\DC-15_0\DC_SYBASE.bat* file:

```
C:\<install_dir>\DC-15_0\OCS-15_0\lib3p
```

Note If you have previously created or obtained a certificate of authority, skip steps 2-7.

---

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

```
C:\<install_dir>\DC-15_0\DC_SYBASE.bat
```

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

```
cd C:\<install_dir>\DC-15_0\bin
```

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

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:\<install_dir>\DC-15_0\connectivity\OCS-15_0\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:\<install_dir>\DC-15_0\bin>certreq -F CA.in
-R CA_req.txt -K CA_pkey.txt -P mycapassword
```

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:\<install_dir>\DC-15_0\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\15.0.0.1\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, see Chapter 4, “Configuring the Server.”

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 service:

```
C:\<install_dir>\DC-15_0\bin>certreq
-F DC.in
-R servicename_req.txt
-K servicename_pkey.txt
-P mydcpassword
```

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

```
C:\<install_dir>\DC-15_0\bin>certauth
-C trusted.txt
-Q servicename_req.txt
-K CA_pkey.txt
-P mycapassword -T180 -O servicename.crt

-- Sybase Test Certificate Authority
certauth\15.0.0.1\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:\<install_dir>\DC-15_0\bin>
type servicename_pkey.txt >> servicename.crt
```

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

```
C:\<install_dir>\DC-15_0\bin>
copy trusted.txt servicename.txt
```

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

```
C:\<install_dir>\DC-15_0\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:\<install_dir>\DC-15_0\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:\<install_dir>\DC-15_0\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 ECDA *srvname.txt* file:

```
C:\<install_dir>\DC-15_0\bin>
copy trusted.txt srvname.txt
```

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

```
C:\<install_dir>\DC-15_0\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 ECDA certificates directory, enable SSL, and verify the log files

1 Create a ECDA directory to hold the certificates:

```
C:\<install_dir>\DC-15_0\servers
\<server_name>\certificates
```

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

```
copy C:\<install_dir>\DC-15_0\bin\servicename.*
C:\<install_dir>\DC-15_0\servers\<server
name>\certificates
copy C:\<install_dir>\DC-15_0\bin\srvname.txt
C:\<install_dir>\DC-15_0\servers\<server
name>\certificates
```

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

```
cd C:\<install_dir>\DC-15_0\servers
\<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 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 C:\<install_dir>\DC-15_0\servers\<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 C:\<install_dir>\DC-15_0\connectivity\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 ECDA:

```
C:\<install_dir>\DC-15_0\bin\DCStart -Sservername
```

8 Verify that the following log entries are in the *C:\<install_dir>\DC-15_0 \servers\<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

Note You must reboot after you update the system environment variables.

❖ To configure the client environment to use SSL

1 Set the client Sybase Window variable:

```
set SYBASE= C:\<install_dir>\DC-15_0\connectivity
```

2 Go to the C:\<install_dir>*\DC-15_0\connectivity\ini* directory:

```
C:\<install_dir>\DC-15_0\bin>
cd C:\<install_dir>\DC-15_0\connectivity\ini
```

3 Copy the *trusted.txt* file to the *C:\<install_dir>\DC-15_0\connectivity\.ini* directory:

```
C:\<install_dir>\DC-15_0\connectivity\ini>copy
C:\<install_dir>\DC-15_0\bin\trusted.txt
```

```
C:\<install_dir>\DC-15_0\connectivity\ini
```

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

    ```
    C:\<install_dir>\DC-15_0\connectivity\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 C:\<install_dir>\DC-15_0\connectivity\\OCS-
    15_0\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, to examine SSL handshakes Sybase recommends that you use the ssldump utility at http://www.rtfm.com/ssldump/.

Setting up SSL on the UNIX server

Note The following procedure to provide SSL encryption and to authenticate clients is not valid for the Enterprise Connect Data Access Option for Oracle. Refer to the Enterprise Connect Data Access Option for Oracle *Server Administration and User's Guide*.

SSL provides encryption of data sent over the network and authenticates clients and their passwords using digital certificates. To setup SSL on a UNIX DirectConnect server, you need:

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

- To create the ECDA certificates directory, enable SSL, and verify the ECDA log files

Note ECDA 15.0 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 \<install_dir>\DC-15_0/DC_SYBASE.csh (or
.sh)
```

Note If you have previously created or obtained the certificate of authority files, skip steps 2-5.

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

```
prompt% cd <install_dir>/OCS-15_0/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/15.0/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 ECDA 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 service:

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

3 Create a ECDA *public key* file *<servicename>.crt* using the *<servicename>_req.txt* file with the CA *private key* file to sign the ECDA *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/15.0/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 ECDA <*servicename*>*.txt* file:

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

7 Using the pwdcrypt utility, create and enter an encrypted password for ECDA 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 ECDA *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 ECDA certificates directory, enable SSL, and verify the ECDA log files

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

1 Create a ECDA directory to hold the certificates:

```
mkdir /<install_dir>/DC-15_0/servers/<server name>
/certificates
```

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

```
cp <install_dir>/OCS-15_0/bin/servicename.*
/<install_dir>/DC 15_0/servers/<servername>
/certificates/.
cp <install_dir>/OCS-15_0/bin/srvname.txt
/<install_dir>/DC-15_0/servers/<servername>
/certificates/.
```

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

```
<prompt>% cd /<install_dir>/DC-15_0
/<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 service in the SSLServices property that is going to use SSL.

- Enter yes in the SSLEnabled property to enable the SSL feature:

  ```
  cd /<install_dir>/DC-15_0/servers /<servername>/cfg
  ```

  ```
  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 <install_dir>
vi interfaces
server name
master tcp ether machine name 12510 ssl
query tcp ether machine name 12510 ssl
```

7 Start ECDA by entering the following:

```
cd /<install_dir>/DC-15_0/bin
prompt% DCStart -S<dcservername>
```

8 Verify that the following log file entries are in the */<install_dir> /DC-15_0/servers/<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 UNIX client

❖ **To configure the client environment to use SSL**

1 Set the client SYBASE UNIX variable:

```
source /<install_dir>/DC-15_0.csh (or .sh)
```

2 Copy the *trusted.txt* file to the <*install_dir*> *config* directory:

```
cp /<install_dir>/OCS-15_0/bin/trusted.txt
/<install_dir>/config
```

3 Go to the <*install_dir*> directory:

```
cd <install_dir>
```

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

```
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:

```
cd <install_dir>/OCS-15_0/bin
```

6 Issue the isql command to connect to the 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 an ECDA server as a Windows service

ECDA does not automatically create the server as a Windows service. However, you can run the ECDA Option for ODBC server and the Mainframe Connect DirectConnect for z/OS Option server as a Windows service. The following describes how to add, start, stop, and remove ECDA as a Windows service.

Note If you set up an ECDA server as a Windows service using ServiceWrapper, you must remove the Windows service using the ServiceWrapper utility. The InstallShield Uninstall process does not remove the Windows service.

Add an ECDA server as a Windows service

To add ECDA as a Windows service use the ServiceWrapper utility.

❖ To add an ECDA server as a Windows service

1 Go to the installation directory where the ServiceWrapper is located:

```
c:\<install_dir>\DC-15_0\bin
```

2 Execute the ServiceWrapper utility:

```
ServiceWrapper.exe --install <service_name> --user=<userid>
--password<password> <installation directory>\DC-15_0\bin\DCStart.bat
-S<dc_server_name>
```

where:

- *service_name* is the Windows service name
- *userid* is the user name for the service that will run
- *password* is the password for the *userid*
- *installation directory* is the target installation directory where you installed ECDA
- *dc_server_name* is the name of the DirectConnect server

For example:

```
ServiceWrapper.exe
--install dcw2ksrv
--password=password
```

```
C:\<install_dir>\DC-15_0\bin\DCStart.bat -
Sdcw2ksrvd
```

Note You must specify the *.bat* suffix for DCStart.

Starting an ECDA server as a Windows service

Note Starting and stopping the ECDA server as a Windows service on Windows XP and Windows 2000 may generate the following in the Windows Event Log:

"The description for Event ID (11) in Source ( ) cannot be found. The local computer may not have the necessary registry information or message DLL files to display messages from a remote computer. The following information is part of the event: DC_150."

❖ To start an ECDA server as a Windows service

1. Select Start | Settings | Control Panel | Administrative Tools | Services.
2. Select the name of the Windows service you installed, right click, and select Properties.
3. Select the Log On tab and verify "This Account" is selected and that the user name is the same as you specified while adding the Windows service.
4. If the Windows service fails, use the Recovery tab to specify any actions.
5. Select the General tab to provide a description of the Windows service and to specify whether the Windows service is to start automatically or manually Click Start to start the Windows service.

Stopping an ECDA server as a Windows service

❖ To stop an ECDA server as a Windows service

1. Select Start | Settings | Control Panel | Administrative Tools | Services.
2. Select the name of the Windows service you want to stop and select Stop the service.

Removing the ECDA server as a Windows service

❖ To remove the ECDA server as a Windows service

Note Sybase recommends that you stop the Windows service before you execute the utility to remove it.

1. Go to the installation directory where the ServiceWrapper is located:

   ```
   C:\<install_dir>\DC-15_0\bin
   ```

2. Execute the ServiceWrapper utility:

```
DC-15_0\ServiceWrapper.exe --uninstall <service_name>
```

   where *service_name* is the Windows service name.

CHAPTER 6 Using Service Name Redirection

How service name redirection works

Service name redirection is an optional feature that lets you route client requests for an access service to alternative access service names.

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 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)

Therefore, 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 Mainframe Connect Client Option *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:

- 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 and earlier do not provide a remote server name. You must use service name redirection to connect such clients.
- Sybase Open Client DB-Library 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.Table 6-1 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 precedence rules shown in Table 6-2 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 Table 6-3.

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 Table 6-4:

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.

Table 6-5 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 Table 6-6:

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
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 Table 6-7:

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 Table 6-8:

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 Table 6-9:

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

Log and trace file descriptions

ECDA Log and trace files 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.

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 ECDA 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.

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.

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

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 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 DirectConnect Manager to retrieve and read the record.

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

Table 7-1 describes the function of each of the columns.

Table 7-1: Log and trace record columns

Column	Description
Record Type	The record type, for example Log Header or TraceEntryExit
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 ECDA 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 messages in Table 7-2 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 or set 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 messages in Table 7-3 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.
`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-15_0\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 these 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:

```
--------------------------------
DirectConnect 15.0 B
Copyright(c)2000, Sybase, Inc.
INTEL x386 Windows 5.1 (2600)
(CRS 85.0) OPT 7-May-2007 13:29:55
--------------------------------
*** Initial configuration for: [SRVNAME] ***
--- CreateSrvcCfg = yes
--- DefQueueSize = 1024
--- DefaultServerLanguage = us_english
--- Description = The DirectConnect server.
--- IsDCDirector = no
--- MaxConnections = 42
--- NetBufSize = 2048
--- OSCodeSetConvert = no
--- IntfFilePath = D:\SYB-15_0\DC-15_0\connectivity
\ini\sql.ini
--- OSCodeSetConvert = no
--- RemoteSites = 4
--- ServiceRedirectionFile =
--- SSLEnabled = no
--- SSLServices =
--- SSLTrustedCertificateFile =
--- LogFileName =
--- LogFileSize = 500000
--- LogWrap = yes
--- LogFlush = no
--- LogToScreen = no
--- LogClientLogin = no
--- LogClientMessages = 17
--- LogCapabilities = no
--- TraceAsync = no
--- TraceEntryExit = no
--- TraceFileName =
--- TraceLogMessages = no
--- TraceOther = yes
--- TraceToScreen = no
--- TraceOpenServer = 0
--- Trace_osClient = no
--- Trace_smConfigAccess = no
--- Trace_smConfigManager = no
--- Trace_smConfigProperty = no
--- Trace_smConnection = no
--- Trace_smLocaleFile = no
--- Trace_smMsgCollection = no
--- Trace_smServer = no
--- Trace_smService = no
--- Trace_smSvclib = no
--- Trace_SOstreams = no

Service Name Redirection not requested.
*** The following localized message files were found:
--- D:\SYB-15_0\DC-15_0\Connectivity\locales\unicode
\econnect\english\server.lcu
--- D:\SYB-15_0\DC-15_0\Connectivity\locales\unicode
\econnect\japanese\server.lcu
Open Server specified language.charset
[us_english.iso_1]
Service Manager active language.charset
[us_english.iso_1]
Calling srv_init().  Set LogOCOSMessages=1 for more
verbose errors if startup halts here.
init License SYSAM2
Sysam MesageID: 131228 Severity: 60 Using licenses from:
D:\SYB-15_0\DC-15_0\Connectivity\\..\..\SYSAM-
2_0\licenses;D:\SYB-12-6_1\SYSAM-
1_0\licenses\license.dat
Sysam email notifaction enabled
Loading service library file: D:\SYB-15_0\DC-
15_0\Connectivity\..\svclib\admin.dll

[ Search String ::--::__:: ]/DirectConnect Admin
Service/15.0/B/INTEL x386/Windows 2000 SP4/005/OPT/May
7 2007 13:43:51

*** Initial configuration for: [srvname] ***

--- Description = The administrative service library.

*** The following localized message files were found:
```

```
--- D:\SYB-15_0\DC-
15_0\Connectivity\locales\unicode\econnect\english\adm
in.lcu

--- D:\SYB-15_0\DC-
15_0\Connectivity\locales\unicode\econnect\japanese\ad
min.lcu

*** Initial configuration for: [srvname] ***

--- Description = The administrative service used by
DirectCONNECT Manager.

Service loaded: [srvname]

Successfully initialized service library: srvname

Loading service library file: D:\SYB-15_0\DC-
15_0\Connectivity\..\svclib\dcany.dll

[ Search String ::--::__:: ]/DirectConnect Anywhere
Access Service/15.0/B/INTEL x386/Windows 2000
SP4/005/OPT/May  7 2007 13:19:13

Sysam MesageID: 131281 Severity: 100 Failed to obtain 1
license(s) for DC_ECDA feature from license file(s) or
server(s).

Sysam MesageID: 131074 Severity: 100 Invalid license
file syntax.

Sysam MesageID: 0 Severity: 100 Feature: DC_ECDA

Sysam MesageID: 0 Severity: 100 License path:  D:\SYB-
12-6_1\SYSAM-1_0\licenses\license.dat;D:\SYB-15_0 -

Sysam MesageID: 0 Severity: 100    \DC-
15_0\Connectivity\\..\..\SYSAM-2_0\licenses\*.lic

Sysam MesageID: 0 Severity: 100 FLEXnet Licensing
error:-2,413.  System Error: 2 ""

Sysam MesageID: 0 Severity: 100 For further information,
refer to the FLEXnet Licensing End User Guide,

Sysam MesageID: 0 Severity: 100 available at
http://sybooks.sybase.com/nav/detail.do?docset=833

Checkout failed

License failed for type ECDA, See log for details

Could not load service library: D:\SYB-15_0\DC-
15_0\Connectivity\..\svclib\dcany.dll
```

```
!READY! Waiting for connections.
    Log Manager Process Process ID(in decimal) = 3476


 ----------------- End of Header -------------------
```

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

Table 7-4: Explanation of log record entries by line number

Line number	Log record entries
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-15_0\ServerName\svclib* subdirectory is loaded in a specific order.
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 only enable it by using DirectConnect Manager.

Line number	Log record entries
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-15_0\SRVNAME\svclib\sample1.dll] linked with
DirectConnect v15.0.0 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-
15_0\SRVNAME\svclib\sample2.dll] linked with
DirectConnect v15.0.0 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

Description of ECDA security

ECDA 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 ECDA 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 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 ECDA to automatically create *user.pwd* files is convenient for backward compatibility, you need to limit access to this file using standard file security techniques.

Changing Administrator IDs in DirectConnect Manager

Sybase recommends that for directed DirectConnect servers, the administrator user IDs and passwords be consistent between the Director and its directed DirectConnect servers. If they are not consistent, the login information must be entered repeatedly for each server as it is accessed. While this is feasible and works well, it may become cumbersome and reduce the value of using the DCDirector. If you choose to use different UID/PWD combinations across servers, you can save this information on your local machine by selecting the Permanent Connection option on the login dialog box. This may reduce the impact of using different UID/PWD combinations somewhat but it reduces security.

Troubleshooting file security issues

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

accept

Establishment of a SNA or TCP/IP connection between Mainframe Connect Server Option and Mainframe Connect DirectConnect for z/OS Option.

access service

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

access code

A number or binary code assigned to programs, documents, or folders that allows authorized users to access them.

access service library

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.

Adaptive Server Enterprise/Component Integration Services

Includes a variation of ASE that provides a Transact-SQL interface to various sources of external data. Component Integration Services allows ASE to present a uniform view of enterprise data to client applications.

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 server management. See also **remote procedure call**, **service library**.

ADMLIB

See **administrative service library**.

Advanced Interactive Executive

The IBM implementation of the UNIX operating system. The RISC System/6000, among other workstations, runs the AIX operating system.

advanced program-to-program communication

Hardware and software that characterize the LU 6.2 architecture and its implementations in products. See also **logical unit 6.2**.

AIX

See **Advanced Interactive Executive.**

AMD2

The component of the Mainframe Connect DB2 UDB Option that allows clients to submit SQL statements to DB2 UDB. It is a CICS transaction that receives SQL statements sent from Mainframe Connect DirectConnect for z/OS Option and submits them to DB2 UDB, using the DB2 UDB dynamic SQL facility. It also receives the results and messages from DB2 UDB and returns them to Mainframe Connect DirectConnect for z/OS Option.

American Standard Code for Information Interchange

The standard code used for information interchange among data processing systems, data communication systems, and associated equipment. The code uses a coded character set consisting of 7-bit coded characters (including a parity check, 8 bits).

API

See **application program interface**.

APPC

See **advanced program-to-program communication**.

application program interface

The programming language interface between the user and Mainframe Connect Client Option or Mainframe Connect Server Option. The API for Mainframe Connect Client Option is Client-Library. The API for Mainframe Connect Server Option is Gateway-Library.

ASCII

See **American Standard Code for Information Interchange**.

ASE

See **Adaptive Server Enterprise**.

ASE/CIS

See **Adaptive Server Enterprise/Component Integration Services**.

batch

A group of records or data processing jobs brought together for processing or transmission.

bind

In the Sybase environment, this term has different meanings depending on the context:

- In CICS, it is an SNA command used to establish a connection between LUs, or a TCP/IP call that connects an application to a port on its system.
- In DB2 UDB, it compiles the Database Request Module, the precompiler product that contains SQL statements in the incoming request, and produces an access plan, a machine code version of the SQL statements that specifies the optimal access strategy for each statement.
- In the mainframe access product set, it establishes a connection between a TRS port and a CICS or IMS region.

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

call level interface
A programming style that calls database functions directly from the top level of the code. Contrast with **embedded SQL.**

catalog
A system table that contains information about objects in a database, such as tables, views, columns, and authorizations.

catalog RPC
A component of the Mainframe Connect DB2 UDB Option that allows clients to access DB2 UDB system catalogs. It uses an interface compatible with the catalog interface for the ODBC API.

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

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

CICS
See **Customer Information Control System**.

CICS region
The instance of CICS.

client
In client/server systems, the part of the system that sends requests to servers and processes the results of those requests. See also **client/server**. Compare with **server**.

client application
Software responsible for the user interface that sends requests to applications acting as servers. See also **client/server**.

Client-Library
A library of routines that is part of Mainframe Connect Client Option.

client request
An RPC or language request sent by a client to a server.

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

Client Services Application
A customer-written CICS program initiated on the host that uses the API to invoke the Mainframe Connect Client Option as a client to the DirectConnect server or to ASE. See also **application program interface**, **Client Services for CICS**.

Client Services for CICS
A Sybase host API that invokes the Mainframe Connect Server Option as a client to an access service for DB2 UDB or ASE. See also **application program interface**, **Customer Information Control System**, **Client Services Application**, **Mainframe Connect Server Option**.

clustered index
An index in which the physical order and the logical (indexed) order is the same. Compare with **nonclustered index**.

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

commit
A process that makes permanent all changes made to one or more database files since the initiation of the application program, the start of an interactive session, or the last commit or rollback operation. Compare with **rollback**.

Common Programming Interface
Specifies the languages and services used to develop applications across SAA environments. The elements of the CPI specification are divided into two parts: processing logic and services.

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.

conversion
The transformation between values that represent the same data item but which belong to different datatypes. Information can be lost due to conversion, because accuracy of data representation varies among different datatypes.

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

connection router
A program provided with Mainframe Connect Client Option that directs requests to particular remote servers. Mainframe system programmers use the connection router to define remote servers and server connections to Mainframe Connect Client Option.

Connection Router Table
A memory-resident table maintained by a Mainframe Connect Client Option system programmer that lists servers and the connections that a Client-Library transaction can use to access them.

control section
The part of a program specified by the programmer to be a relocatable unit, all elements of which are to be loaded into adjoining main storage locations.

control statement
In programming languages, a statement that is used to alter the continuous sequential execution of statements. A control statement can be a conditional statement or an imperative statement.

conversation-level security The passing of client login information to the mainframe by TRS when it allocates a conversation.

CSA See **Client Services Application**.

CSP See **catalog stored procedure**.

cursor In SQL, a named control structure used by an application program to point to a row of data.

Customer Information Control System An IBM licensed program that enables transactions entered at remote terminals to be processed concurrently by user-written application programs.

DASD See **direct access storage device**.

data definition statement An IBM mainframe statement used to relate a name with a file.

data definition language A language for describing data and data relationships in a database.

data set name The term or phrase used to identify a data set.

database management system The term or phrase to identify a data set.A computer-based system for defining, creating, manipulating, controlling, managing, and using databases.

database operation A single action against the database. For Mainframe Connect DirectConnect for z/OS Option, a database operation is usually a single SQL statement. One or more database actions can be grouped together to form a request. See also **request**.

Database 2 An IBM relational database management system.

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 ODS applications. See also **application program interface**.

DBMS See **database management system**.

DB2 UDB See **Database 2**.

DDL See **data definition language**.

DD statement See **data definition statement**.

default language The language that displays a user’s prompts and messages.

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** and **express transfer**.

direct access storage device A device in which access time is effectively independent of the location of the data.

direct request A request sent directly from a client workstation through Transaction Router Service to the DirectConnect server without going through ASE. Contract with **indirect request**.

direct resolution A type of service name resolution that relies upon a client application specifying the exact name of the service to be used. See also **service name resolution**. Compare with **service name redirection**.

DirectConnect Manager A Java application from Sybase that can be used in Windows and UNIX environments. It provides remote management capabilities for DirectConnect products, including starting, stopping, creating, and copying services.

DirectConnect server The component of Mainframe Connect DirectConnect for z/OS Option that provides general management and support functions to service libraries.

dll See **dynamic link library**.

DSN See **data set name**.

dynamic link library A file containing executable code and data bound to a program at load time or runtime, rather than during linking.

dynamic SQL The preparation and processing of SQL source statements within a program while the program runs. The SQL source statements are contained in host-language variables rather than being coded directly into the application program. Contrast with **static SQL**.

ECDA See **Enterprise Connect Data Access**.

ECDA Option for ODBC A Sybase solution that allows client applications to access ODBC data. It combines the functionality of the ECDA Option for ODBC architecture with ODBC to provide dynamic SQL access to target data, as well as the ability to support stored procedures and text and image pointers.

ECDA Option for Oracle A Sybase solution that provides Open Client access to Oracle databases. When used in combination with ASE, it provides many of the features of a distributed database system, such as location transparency, copy transparency, and distributed joins.

embedded SQL SQL statements that are embedded within a program and are prepared in the process before the program runs. After it is prepared, the statement itself does not change, although values of host variables specified within the statement might change.

end user A person who connects to a DirectConnect server using an application to access databases and perform transfers. See also **transfer**.

Enterprise Connect Data Access An integrated set of software applications and connectivity tools that allow access to data within a heterogeneous database environment, such as a variety of LAN-based, non-Sybase data sources, as well as mainframe data sources.

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

exit routine A user-written routine that receives control at predefined user exit points.

express transfer A form of bulk copy transfer that uses ODBC bulk APIs to improve performance when transferring bulk data between data sources. Because it uses the same syntax as bulk copy transfer, no modification of applications is required.

external call interface A CICS client facility that allows a program to call a CICS application as if the calling program had been linked synchronously from a previous program instead of started from a terminal.

External Security Manager An add-on security package for the z/OS mainframe, licensed by Computer Associates.

FCT See **forms control table**.

forms control table An object that contains the special processing requirements for output data streams received from a host system by a remote session.

gateway Connectivity software that allows two or more computer systems with different network architectures to communicate.

Gateway-Library A library of communication, conversion, tracing, and accounting functions supplied with Mainframe Connect Server Option.

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

global variable A variable defined in one portion of a computer program and used in at least one other portion of the computer program. Contrast with **local variable**.

handler A routine that controls a program's reaction to specific external events, for example, an interrupt handler.

host The mainframe or other machine on which a database, an application, or a program resides. In TCP/IP, this is any system that is associated with at least one Internet address. See also **Transmission Control Protocol/Internet Protocol**.

host ID In Mainframe Connect Server Option, the ID that the TRS passes to the mainframe with a client request. The host ID is part of the client login definition at the TRS.

host password In Mainframe Connect Server Option, the password that the client passes to the mainframe with a client request.

host request library A DB2 UDB table that contains host-resident SQL statements that can be executed dynamically. See also **host-resident request**.

host-resident request A SQL request that resides in a DB2 UDB table called the host request library. See also **host request library**.

IMS See **Information Management System**.

indirect request A client request that is routed through a stored procedure on a SQL Server, which forwards the request to TRS as an RPC. Compare with **direct request**.

Information Management System A database/data communication system that can manage complex databases and networks.

interfaces file An operating system file that determines how the host client software connects to a Sybase product. An *interfaces* file entry contains the name of any DirectConnect server and a list of services provided by that server.

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**.

keyword A word or phrase reserved for exclusive use by Transact-SQL.

language RPC The name TRS uses to represent a client's language request. TRS treats a language request as a remote procedure call (RPC) and maps it to a language transaction at the remote server.

language transaction The server transaction that processes client language requests. The Mainframe Connect DB2 UDB Option language transaction for CICS is AMD2, which uses the DB2 UDB dynamic SQL facilities to process incoming SQL strings. The Mainframe Connect DB2 UDB Option for IMS uses SYRT by default.

linkage In computer security, combining data or information from one information system with data or information from another system with the intention to derive additional information; for example, the combination of computer files from two or more sources.

linkage editor A computer program that creates load modules from one or more object modules or creates load modules by resolving cross references among the modules, and if necessary, adjusts those addresses.

link-edit To create a loadable computer program by using a linkage editor. See also **linkage editor**.

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, and documents are displayed in the user's language. See also **globalization**.

local variable A variable that is defined and used only in one specified portion of a computer program. Contrast with **global variable**.

logical unit A type of network addressable unit that enables a network user to gain access to network facilities and communicate remotely. A connection between a TRS and a CICS region is a connection between logical units.

logical unit 6.2 A type of logical unit that supports general communication between programs in a distributed processing environment. See also **advanced program-to-program communication**.

login ID In Mainframe Connect Server Option, the ID that a client user uses to log in to the system.

login packet Client information made available to Mainframe Connect Server Option. The client program sets this information in a login packet and sends it to TRS, which forwards it to the mainframe.

long-running transaction A transaction that accepts more than one client request. Whereas short transactions end the communication after returning results to a client, a long-running transaction can await and process another request. Compare with **short transaction**.

LU 6.2 See **logical unit 6.2**.

mainframe access products Sybase products that enable client applications to communicate with mainframes in a client/server environment. See **client/server**.

Mainframe Connect The Sybase product set that provides access to mainframe data.

Mainframe Connect Client Option A Sybase product that, using Client-Library, allows mainframe clients to send requests to SQL Server, Open Server, the Mainframe Connect DB2 UDB Option and Mainframe Connect Server Option. Mainframe Connect Client Option provides capability for the mainframe to act as a client to LAN-based resources in the CICS or the IMS and MVS environment.

Mainframe Connect DB2 UDB Option A Sybase mainframe solution that provides dynamic access to DB2 UDB data. It is available in the CICS or IMS environment. See also **Customer Information Control System**, **Database 2**, **Multiple Virtual Storage**.

Mainframe Connect DirectConnect for z/OS Option A Sybase Open Server application that provides access management for non-Sybase databases, copy management (transfer), and remote systems management.

Mainframe Connect Server Option A Sybase product that provides capability for programmatic access to mainframe data. It allows workstation-based clients to execute customer-written mainframe transactions remotely. It is available for the CICS and the IMS and MVS environments

Multiple Virtual Storage An IBM operating system that runs on most System/370 and System/390 mainframes. It supports 24-bit addressing up to 16 megabytes.

network protocol A set of rules governing the way computers communicate on a network.

nonclustered index An index that stores key values and pointers to data. Compare with **clustered index**.

null Having no explicitly assigned value. NULL is not equivalent to 0 or to blank.

ODBC See **Open Database Connectivity**.

ODS See **Open Data Services**.

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

Open Database Connectivity A Microsoft API that allows access to both relational and non-relational databases. See also **application program interface**.

Open Server A Sybase product that provides the tools and interfaces required to create a custom server. Clients can route requests to the DirectConnect server through an Open Server configured to meet specific needs, such as the preprocessing of SQL statements.

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

Partner Certification Reports Sybase publications that certify third-party or Sybase products to work with other Sybase products.

Password Expiration Management An IBM password management program with CICS Version 3.3 through an optional program temporary fix, and as an integral part of CICS with version 4.1 and higher.

PEM See **Password Expiration Management**.

PL/1 See **Programming Language /1**.

primary database The database management system that the DirectConnect server is always connected to. It is implied in the transfer statement.

Programming Language/1 A programming language designed for use in a wide range of commercial and scientific computer applications.

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

protocol The rules for requests and responses used to manage a network, transfer data, and synchronize the states of network components.

query A request for data from a database, based upon specified conditions.

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

relational database A database in which data is viewed as being stored in tables consisting of columns (data items) and rows (units of information).

relational operators Operators supported in search conditions.

relops See **relational operators**.

remote procedure call A call to execute a stored procedure on a remote server. For Mainframe Connect Server Option, an RPC is a direct request from a client to TRS. For Mainframe Connect Client Option, a Client-Library transaction that calls a procedure on a remote server acts like an RPC.

remote stored procedure A customer-written CICS program using an API that resides on the mainframe and communicates with Mainframe Connect DB2 UDB Option. See also **Customer Information Control System**, **stored procedure**. 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.

Replication Server A Sybase SQL Server application that maintains replicated data and processes data transactions received from a data source.

request One or more database operations an application sends as a unit to the database. Depending upon the response, the application commits or rolls back the request. See also **commit**, **rollback**, **unit of work**.

resource table A main storage table that associates each resource identifier with an external logical unit (LU) or application program.

rollback An instruction to a database to back out of changes requested in a unit of work. Compare with **commit**.

router An attaching device that connects two LAN segments, which use similar or different architectures, at the Open System Interconnection (OSI) reference model network layer. Contrast with **gateway**.

RPC See **remote procedure call.**

RSP See **remote stored procedure**.

SAA See **System Application Architecture**.

secondary connection The connection specified in the transfer statement. It represents anything that can be accessed using Mainframe Connect Client Option, such as ASE or another access service.

secondary database In transfer processing, the supported database that is specified in the transfer statement. Compare with **primary database**.

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

server process ID A positive integer that uniquely identifies a client connection to the server.

service A functionality available in Mainframe Connect DirectConnect for z/OS Option. It is the pairing of a service library and a set of specific configuration properties.

service library In Mainframe Connect DirectConnect for z/OS Option, a set of configuration properties that determine service functionality. See also **access service library**, **administrative service library**, **Transaction Router Service library**, **transfer service library**.

service name redirection A type of service name resolution that allows a system administrator to create an alternative mechanism to map connections with 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 **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**.

session A connection between two programs or processes. In APPC communications, sessions allow transaction programs to have conversations between the partner LUs. See also **advanced program-to-program communication**.

short transaction A mainframe transaction that ends the communication when it finishes returning results to the client. Compare with **long-running transaction**.

SNA See **Systems Network Architecture**.

SNRF See **service name redirection file**.

SPID See **server process ID**.

SQL See **structured query language**.

SQLDA See **SQL descriptor area**.

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

sql.ini The interfaces file containing definitions for each DirectConnect server to which a workstation can connect. The file must reside on every client machine that connects to ASE.

SQL descriptor area A set of variables used in the processing of SQL statements.

SQL stored procedure A single SQL statement that is statically bound to the database. See also **stored procedure**.

static SQL SQL statements that are embedded within a program and prepared during the program preparation process before the program runs. Compare with **dynamic SQL**.

stored procedure A collection of SQL statements and optional control-of-flow statements stored under a particular name. Adaptive Server stored procedures are called “system procedures.” See also **remote stored procedure**, **system procedures**.

structured query language An IBM industry-standard language for processing data in a relational database.

stub A program module that transfers remote procedure calls (RPCs) and responses between a client and a server.

SYRT The component of Mainframe Connect DB2 UDB for IMS that allows clients to submit SQL language requests to DB2 through IMS.

System Administrator The person in charge of server system administration, including installing and maintaining DirectConnect servers and service libraries.

System Application Architecture An IBM proprietary plan for the logical structure, formats, protocols, and operational sequences for transmitting information units through networks and controlling network configuration and operation. See also **advanced program-to-program communication**.

system procedures A stored procedure that ASE supplies for use in system administration. System procedures serve as shortcuts for retrieving information from system tables, or a mechanism for accomplishing database administration. See also **stored procedure**.

Systems Network Architecture An IBM proprietary plan for the structure, formats, protocols, and operational sequences for transmitting information units through networks. See also **advanced program-to-program communication**.

table An array of data or a named data object that contains a specific number of unordered rows. Each item in a row can be unambiguously identified by means of one or more arguments.

Tabular Data Stream A Sybase application-level protocol that defines the form and content of relational database requests and replies.

target A system, program, or device that interprets, rejects, satisfies, or replies to requests received from a source.

target database The database to which the DirectConnect server transfers data or performs operations on specific data.

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

TDS See **Tabular Data Stream**.

transaction A unit of processing initiated by a single request. A transaction consists of one or more application programs that, when executed, accomplish a particular action. In Mainframe Connect Server Option, a client request (RPC or language request) invokes a mainframe transaction. In Mainframe Connect Client Option, a mainframe transaction executes a stored procedure on a remote server.

transaction processing A sequence of operations on a database that is viewed by the user as a single, individual operation.

Transaction Router Service A Mainframe Connect DirectConnect for z/OS Option program used when the mainframe acts as a transaction server to route requests from remote clients to the Mainframe Connect Server Option and return results to the clients.

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

Transact-SQL A Sybase-enhanced version of the SQL database language used to communicate with ASE.

transfer A Mainframe Connect DirectConnect for z/OS Option feature that allows users to move data or copies of data from one database to another.

transfer service library A service library that provides copy management functionality. See also **service library**.

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

trigger A form of stored procedure that automatically executes when a user issues a change statement to a specified table.

TRS See **Transaction Router Service**.

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

T-SQL See **Transact-SQL**.

unit of work One or more database operations grouped under a commit or rollback. A unit of work ends when the application commits or rolls back a series of requests, or when the application terminates. See also **commit**, **rollback**, **transaction**.

user ID User identification. The ID number by which a user is known in a specific database or system.

variable An entity that is assigned a value. Mainframe Connect DirectConnect for z/OS Option has two kinds of variables: *local* and *global*.

view An alternate representation of data from one or more tables. A view can include all or some of the columns contained the table or tables on which it is defined.

Virtual Storage Access Method An IBM-licensed program that controls communication and the flow of data in an SNA network.

Virtual Telecommunications Access Method IBM mainframe software that allows communication on an SNA network between mainframes and allows the mainframe to have multiple sessions per connection.

VSAM See **Virtual Storage Access Method.**

VTAM See **Virtual Telecommunications Access Method**.

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

Index