SYBASE®
An SAP® Company

Server Administration Guide

Enterprise Connect™ Data Access and Mainframe Connect™ 15.7

DOCUMENT ID: DC01881-01-1570-01
LAST REVISED: November 2012

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

Contents

Conventions

These style and syntax conventions are used in Sybase® documentation.

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

Key	Definition
{ }	Curly braces indicate that you must choose at least one of the enclosed options. Do not type the braces when you enter the command.
[ ]	Brackets mean that choosing one or more of the enclosed options is optional. Do not type the brackets when you enter the command.
( )	Parentheses are to be typed as part of the command.
|	The vertical bar means you can select only one of the options shown.
,	The comma means you can choose as many of the options shown as you like, separating your choices with commas that you type as part of the command.

Introduction

Enterprise Connect™ Data Access (ECDA) is an integrated set of software applications and connectivity tools that allows you to access data within a heterogeneous database environment. ECDA gives you the ability to access variety of LAN-based datasources, as well as mainframe datasources.

ECDA Option for ODBC

ECDA Option for ODBC is a Sybase solution that gives client applications ODBC data access.

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.

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

- The DirectConnect™ server, which provides management and support functions for DirectConnect service libraries.
- An access service library, which accesses data from ODBC-accessible target databases, such as DB2 UDB and Microsoft SQL Server.
- Access services, which contain specific sets of configuration properties relating 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 datasources, 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: Verify that your ODBC driver is 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.

DirectConnect Server Routing Process

The DirectConnect server routes each client request for an access service to the appropriate access service library.

The routing process can access the service in one of these ways:

- Directly – 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.
- Using service name redirection – map your access service connections to allow client requests to be routed to assigned access services based on user profiles. This feature allows you to centrally manage client access to access services.

See also

- *Service Name Redirection* on page 57

ECDA Option for ODBC Configuration Properties

You can configure ECDA Option for ODBC properties on the server level, the access service library level, or an individual access service level.

DirectConnect configuration properties are grouped as:

- Server configuration files – consist of the properties that manage a particular DirectConnect server.
- Access service library configuration files – consist of general library configuration values and configuration sets for all access services associated with a particular access service library.
- Access service configuration properties – 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, you must manually enable the access service before it can be used.

For information about configuring access service libraries and access services, including instructions on creating new access services, see the access services users guide for your database system.

You can configure properties using the DirectConnect Manager or a text editor. Sybase recommends that you use DirectConnect Manager for these 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 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.

See also

- *Service Name Redirection* on page 57
- *Configure the DirectConnect Server* on page 21

Server External Files

The DirectConnect server manages external files that reside in various subdirectories.

See the appropriate installation guide for your database system and platform.

Table 1. Server External Files

File Name	Description
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 contains operational information that you can use to correct problems. Although the file is maintained in US English, client messages appear in the client language. The log file resides in the server `log` subdirectory.
Server configuration file	`server.cfg` contains all server configuration information. It resides in the server `cfg` subdirectory.

File Name	Description
Access service library files	This dynamically loaded shared library 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 *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.
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 set it on and off through server configuration. Although the trace file is maintained in US English, client messages appear in the client language. The trace file resides in the log subdirectory.

See also

- *Configure the DirectConnect Server* on page 21
- *Service Name Redirection* on page 57

Mainframe Connect Options

The DirectConnect for z/OS Option interacts with several options to provide mainframe access for LAN client requests.

Table 2. Mainframe Connect Options

Options	Description
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: • Support full read-write, dynamic SQL access to data • Allow applications to use cursors for flexible and efficient result-set processing • Permit the use of long-running transactions against mainframe databases • Allow 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: • Adaptive Server® Enterprise/Component Integration Services (ASE/CIS) • Adaptive Server 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 also referred to as the DB2 UDB Options for CICS and for IMS.

Options	Description
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 datasource 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.

DirectConnect Manager

DirectConnect Manager graphically represents each DirectConnect object on a tree list or an "icon map," which is 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 asynchronously with DirectConnect servers, 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. However, Sybase strongly recommends that you use DirectConnect Manager for these reasons:

- Changes that you make with a text editor do not take effect until you restart the server, while 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 online help menu option.

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

Note: When you install a DirectConnect product on a Windows or UNIX platform or machine, you must install DirectConnect on a separate platform or machine; doing so allows you to control any ECDA product from any machine.

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

For more information about code page translation, see *Configuring the Access Service Library for DirectConnect* in the *ECDA Option for ODBC Access Service Users Guide*.

Localization

You can localize the messages that are generated by the target database manager and passed to the client without changes, and the messages generated in ECDA Option for ODBC.

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

About DirectConnect Servers

Be familiar with the types of DirectConnect servers and their functions.

Types of DirectConnect Servers

The two kinds of DirectConnect servers are the DCDirector server and the DirectConnect server.

- 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 is directed or not.

Whether a server is directed or nondirected 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.

DCDirector Server Functions

A DCDirector server allows you to have control over the servers you manage in DirectConnect Manager.

A DCDirector server contains only DirectConnect servers that reside in the same directory and performs the sole function of creating, starting, and stopping the servers.

Table 3. Functions Supported for Directed and Nondirected 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)
```

Server Creation

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 nondirected DirectConnect server only using the **AddServer** utility.

See also

- *Modifying the Server Configuration File* on page 22

Create a Server 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.

Also, DirectConnect Manager 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 it, and identify and establish a connection between the server and DirectConnect Manager. See the installation guide for your platform, and *Connecting DirectConnect Manager to a DirectConnect Server* in the DirectConnect Manager online help.

Create a Server Using ECDA Utilities

Sybase provides **DCDirector** and **AddServer** utilities to simplify the execution of ECDA on multiple platforms.

- **DCDirector** – creates and starts a DCDirector server.
- **AddServer** – creates a new 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)

Keep these scripts in their original directory. It is from this directory that the utilities find the paths to the other files they need to perform their tasks.

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

See also

- *Modifying the Server Configuration File* on page 22

Creating and Starting a DCDirector Server

Create and start a DCDirector server using DirectConnect Manager, the **DCDirector** utility, or by modifying the configuration file.

DirectConnect Manager	Go to the DirectConnect Manager online help to create a DCDirector server. Select **Server Administration > Creating a DCDirector server**.
DCDirector utility	Use the DCDirector script (on UNIX) or batch file (on Windows) to create and start a new DCDirector for an installation. **1.** To designate the new server as a DCDirector server, go to: • `/<install_dir>/DC-15_0/bin` (UNIX) • `C:\<install_dir>\DC-15_0\bin` (Windows) **2.** Enter `DCDirector`. The DCDirector displays the new server name, the machine name where the server is installed, and a port number equal to 7711. For more information about ECDA utilities, see the *Mainframe Connect DirectConnect for z/OS Option Installation Guide* or the *Enterprise Connect Data Access Options Installation Guide*, for your platform.
Modify the configuration file	To designate a server as a DCDirector server in the server configuration file `server.cfg`, use the `IsDCDirector` property.

Creating a DirectConnect Server

Create a directed server using DirectConnect Manager and a nondirected server using the **AddServer** utility.

DirectConnect Manager	Go to the DirectConnect Manager online help to create a directed server. Select **Server Administration > Creating a DirectConnect server**. Before you can use DirectConnect Manager, you must install it, and identify and establish a connection between the server and DirectConnect Manager. See the installation guide for your platform, and *Connecting DirectConnect Manager to a DirectConnect Server* in the DirectConnect Manager online help.
AddServer utility	To create a nondirected server using the **AddServer** utility, see 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 nondirected server names or ports.

DirectConnect New Configuration Files

When you create a new server, the server configuration files are not created until you start the new server.

If the new server is configured for a `snrf.tbl` table, and the table does not exist, the server creates one and populates it with **"* * Service A**."

However, the access service name redirection functionality does not work until you replace the text above with valid information.

See also

- *Service Name Redirection* on page 57
- *Configure the DirectConnect Server* on page 21

Start and Stop the Server and Access Services

Start and stop server and access services.

Starting the DirectConnect Server

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 that you use the DCDirector server through DirectConnect Manager.

DirectConnect Manager	Go to the DirectConnect Manager online help to start a DirectConnect server. Select **Server Administration** > **Starting a Server**. 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 topic *Connecting to DirectConnect Manager to a DirectConnect Server* of the DirectConnect Manager online help.
DCStart utility	This utility is similar to the **direct** executable. **DCStart** automatically sources the appropriate `/<install_dir>/DC-15_0/DC_SYBASE.csh` (UNIX) file, or runs 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.

See also

- *Creating and Starting a DCDirector Server* on page 14

Stopping the DirectConnect Server

Stop a DirectConnect 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 that you use DirectConnect Manager.

DirectConnect Manager	Go to the DirectConnect Manager online help to stop a DirectConnect server. 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, use the command line.
Command line	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; however, this works only if the sa user password has not been modified. **Note:** If you invoke **stopsrvr** when a client is performing work such as batch processing, the server is not stopped, but both the client and the server suspend operations. The **stopsrvr** format is: `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.
Windows systems	Use one of these methods: • To stop the server using the command line, press **Ctrl+C**. • To stop the server that is started as a Windows service: **1.** Select **Control Panel > Services**. **2.** Select the DirectConnect *server_name*. **3.** Select **Stop**.
UNIX systems	**1.** Make the DirectConnect console window the active window. **2.** Press **Ctrl+C**.

Starting an Access Service

Start an access service at start-up or by using DirectConnect Manager.

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

DirectConnect Manager	Go to the DirectConnect Manager online help to start a DirectConnect server. Select **Managing Access Services > Starting a 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 users guide.

Stopping an Access Service

Go to the DirectConnect Manager online help to stop an acesss service. Select **Managing Access Services > Stopping a service**.

See also

- *Process Exit Codes* on page 79

Configure the DirectConnect Server

Control the DirectConnect server configuration through the server configuration file.

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

The server configuration file, `server.cfg`, is a data group that defines individual properties. As the system administrator, you can use DirectConnect Manager or a standard text editor to set, change, add, or delete property values in the configuration file.

You can use a text editor to configure your DirectConnect server, however, Sybase recommends that you use 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. Add an entry to the configuration file only to change a property value to something other than the default.

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 results in an error condition.

Server Configuration File Format

The configuration file format uses some 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 in which 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. Place each one on a separate line that begins with a semicolon or the crosshatch character (#).

This 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
```

Modifying the Server Configuration File

Modify the server configuration file using different methods.

- Using DirectConnect Manager:
  You can use DirectConnect Manager to edit the client interaction properties, the logging properties, and the tracing properties without having to restart the server.
- 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.

Client Interaction Server Properties

The client interaction server properties subsection consists of a heading and the configuration property list.

Note: You can change the order of the configuration properties within each subsection.

```
{Client Interaction}
CreateSrvcCfg
```

```
DefaultServerLanguage
DefQueueSize
Description
```

```
IsDCDirectorServer
MaxConnections
NetBufSize
```

```
OSCodeSetConvert
RemoteSites
ServiceRedirectionFile
SSLEnabled
```

```
SSLServices
```

```
SSLTrustedCertificateFile
```

See also

- *ServiceRedirectionFile* on page 28

CreateSrvcCfg

The `CreateSrvcCfg` configuration property controls the service libraries that 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 does 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.

DefaultServerLanguage

The `DefaultServerLanguage` configuration property 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

The `DefQueueSize` configuration property identifies the deferred event queue size, which is the maximum number of events that can be queued on the Open Server application at any given time.

Syntax

```
DefQueueSize=integer
```

Range
512–4096

Default
1024

Value
integer (commas are not allowed)

Description

The `Description` configuration property describes the DirectConnect server.

Syntax

```
Description=text
```

Range
0–255 (characters)

Default
none

Value
Descriptive information about the server in the configuration file.

IsDCDirector

The `IsDCDirector` configuration property allows you to specify whether to designate the DirectConnect server as a DirectConnect Director (DCDirector). 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.

MaxConnections

The `MaxConnections` configuration property identifies the maximum number of clients that you can connect to the 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. This value must accommodate the `MaxSvcConnections` for all the access services operating on the same server.

Comments

- For UNIX only, this property prevents your system from failing as a result of running out of UNIX file descriptors.
  Use this formula to determine the `MaxConnections` `server.cfg` value and the file descriptor setting, on UNIX, for the ECDA process:

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

  where:

- *max file descriptors per process* is determined by your UNIX administrator 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** is configured to a maximum of 589. If the number of connections must 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 should 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, use this:

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

The 3355 file descriptors 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 must increase the number of file descriptors available for the ECDA process.

NetBufSize

The `NetBufSize` configuration property 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

The `OSCodeSetConversion` configuration property enables or disables Open Server code page conversion.

Syntax

```
OSCodeSetConversion= [no | yes]
```

Warning! The `OSCodeSetConversion` configuration property value must be set to yes if client code page is different than the Open Server and DirectConnect server's operating system code page.

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

The `RemoteSites` configuration property lists the total number of Adaptive 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

The `ServiceRedirectionFile` configuration property 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.

See also

- *Service Name Redirection* on page 57

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 ECDA installation guide for your platform.

ECDA searches these files 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 that you use 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 Adaptive Servers.

SSLServices

The `SSLServices` configuration property identifies the access or Transaction Router Service (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 does not start with:

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

SSLTrustedCertificateFile

The `SSLTrustedCertificateFile` configuration property identifies the path to the file containing the certificates 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 logging properties subsection consists of a heading and the configuration property list.

```
{Logging}
 LogClientLogin
 LogClientMessages
 LogFileName
 LogFileSize
 LogFlush
```

```
 LogLicenseMessages
 LogOCOSMessages
 LogToScreen
 LogWrap
```

LogClientLogin

The `LogClientLogin` configuration property 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.

See also

- *Service Name Redirection* on page 57
- *Log and Trace Files* on page 67

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.

Syntax

LogClientMessages=*severity*

Range

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

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.

See also

- *Log and Trace Files* on page 67

LogFileName

The `LogFileName` configuration property 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 US 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.

See also

- *Log and Trace Files* on page 67

LogFileSize

The `LogFileSize` configuration property 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

- *Log and Trace Files* on page 67
- *LogWrap* on page 35

LogFlush

The `LogFlush` configuration property 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

The `LogLicenseMessages` configuration property writes SySAM license manager messages to the log.

Syntax

```
LogLicenseManagerh=[no | yes]
```

Default

no

Values

- yes – reports verbose license manager messages.
- no – reports only errors to the log.

LogOCOSMessages

When `LogOCOSMessages` is set to 0 (zero), the system does not log error messages generated internally by Open Client and Open Server.

Syntax

```
LogOCOSMessages=severity
```

Range

Message severities fall in the range of 0–24, inclusive, matching the levels defined for Adaptive Server.

Default

1

Value

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

See also

- *Log and Trace Files* on page 67

LogToScreen

The `LogToScreen` configuration property 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 may result in a negative impact on performance, depending upon the number and types of other log properties you have set to yes.

See also

- *Log and Trace Files* on page 67

LogWrap

The `LogWrap` configuration property 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: Monitor the archive option to prevent it filling the file system.

Comments

Only the ECDA log file is affected by this configuration property.

See also

- *Log and Trace Files* on page 67
- *LogFileSize* on page 32

Tracing Properties

The tracing properties subsection consists of a heading and the configuration property list.

```
{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
```

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

See also

- *Log and Trace Files* on page 67

Trace_osClient

Set the `Trace_osClient` configuration property to trace OS client internal data.

Syntax

```
Trace_osClient=[no | yes]
```

Default

no

Value

yes – the system traces OS client internal data.

Trace_smConfigAccess

Set the `Trace_smConfigAccess` configuration property to trace configuration access internal data.

Syntax

```
Trace_smConfigAccess=[no | yes]
```

Default

no

Value

yes – the system traces configuration access internal data.

Trace_smConfigManager

Set the `Trace_smConfigManager` configuration property 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

Set the `Trace_smConfigProperty` configuration property to trace configuration property internal data.

Syntax
```
Trace_smConfigProperty=[no | yes]
```

Default
no

Value
yes – the system traces configuration property internal data.

Trace_smConnection

Set the `Trace_smConnection` configuration property to trace connection internal data.

Syntax
```
Trace_smConnection=[no | yes]
```

Default
no

Value
yes – the system traces certain connection internal data.

Trace_smLocaleFile

Set the `Trace_smLocaleFile` configuration property to trace locale file internal data.

Syntax
```
Trace_smLocaleFile=[no | yes]
```

Default
no

Value
yes – the system traces `locale` file internal data.

Trace_smMsgCollection

Set the `Trace_smMsgCollection` configuration property to trace message collection internal data.

Syntax
```
Trace_smMsgCollection=[no | yes]
```

Default
no

Value
yes – the system traces message collection internal data.

Trace_smServer

Set the `Trace_smServer` configuration property to trace server internal data.

Syntax
```
Trace_smServer=[no | yes]
```

Default
no

Comments
yes – the system traces server internal data.

Trace_smService

The `Trace_smServer` configuration property 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

Set the `Trace_smSvclib` configuration property to trace service library internal data.

Syntax
Trace_smSvclib=[no | yes]

Default
no

Value
yes – the system traces service library internal data.

Trace_SOstreams

Set the `Trace_SOstreams` configuration property to trace OS stream internal data.

Syntax

```
Trace_SOstreams=[no | yes]
```

Default
no

Comments
yes – the system traces OS stream internal data.

TraceAsync

Set the `TraceAsync` configuration property to trace asynchronous events, such as interrupts and timer notification.

Syntax

```
TraceAsync=[no | yes]
```

Default
no

Value
yes – asynchronous event tracing is enabled.

TraceEntryExit

Set the `TraceEntryExit` configuration property to trace 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

Set the `TraceFileName` configuration property 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

Set the `TraceLogMessages` configuration property 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

The `TraceOpenServer` configuration property corresponds to the Open Server `SRV_S_TRACEFLAG` property. Values comprised of the bitwise or of the `SRV_TR` defined values in `ospublic.h`.

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.

See the *Open Server Server-Library/C Reference Manual*.

TraceOther

Set the `TraceOther` configuration property to trace data for debugging.

Syntax

```
TraceOther=[no | yes]
```

Default
no

Value
yes – traces data for debugging.

TraceToScreen

Set the `TraceOther` configuration property 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 34

Set Up SSL on the DirectConnect Server

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

Note: The tasks related to setting up SSL on the DirectConnect server apply to ECDA options and Mainframe Connect DirectConnect for z/OS Option, unless otherwise indicated. Any platform differences are also noted.

Set Up SSL on Windows Server

SSL on Windows server provides encryption of data sent over the network and authenticates clients and their passwords using digital certificates.

Warning! Only one SSL-enabled access service can run on a DirectConnect server. This is due to 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 or Mainframe Connect to accept SSL and non-SSL connections (for example, use non-SSL access services and one SSL access service in the same ECDA or Mainframe Connect), Sybase recommends that you use only one SSL access service. This prevents a user from using a secured port to access data over an unsecured transport medium.

In the tasks related to setting up SSL, substitute the variables as follows:

- *servicename* is your service's name.
- *srvname* is your server's name.
- *yourcapassword* is the password you create.

In addition, the C drive is used as the installation drive in the examples.

Note: ECDA or Mainframe Connect 15.0 or later does not support **transfer to** and **transfer from** SSL-enabled Adaptive Servers.

Creating Certification Authority Files

Follow these steps to create the certification authority (CA) files.

Note: If you have previously already created or obtained a certificate, skip steps 2 through 6.

1. Set the environment by issuing this command from a command window:

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

where *install_dir* is the directory for your installation. For example:

```
cd C:\sybase\DC-15_0\DC_SYBASE.bat
```

2. Create the `CA.in` file. Enter the parameters for the CA certificate that you are going to use with the **certreq** utility, as shown:
   a) Enter the following, on one line:

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

   b) Using a text editor such as Notepad, create a file called `CA.in` and enter:

   ```
   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
   ```

   Save the file.

   Note: For more information about **certreq** parameters, see the *Adaptive Server Utilities Guide*.

3. 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
   ```

   You see:

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

4. 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:

   ```
   >certauth -r -C CA_req.txt -Q CA_req.txt -K CA_pkey.txt -P
   yourcapassword -T 365 -O trusted.txt
   ```

Following is an example of the expected output:

```
-- Sybase Test Certificate Authority Utility -- -- Certificate
Validity:
      startDate = Thu Mar 02 09:56:43 2008
      endDate = Fri Mar 20 09:58:10 2009
Setting serial number Ox1w7d236819a91a32
Could not sign certificate using signature type 20, error 'No error
string returned.' (3000).
Could not sign certificate using signature type 22, error 'No error
string returned.' (3000)
CA sign certificate SUCCEED using signature type 2, return
'SSLNoErr' (0).
```

Creating Certificates for the Specific DirectConnect Server and Service

Learn the steps 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.
2. From `C:\<install_dir>\DC-15_0\bin`, use a text editor to create the `DC.in` file. (Refer to the *Adaptive Server Utilities Guide* document for **certreq** parameters.)

   ```
   notepad DC.inreq_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
   ```

   Save the file.
3. Create `private key` and `certificate request` files for the service by entering this, on one line:

   ```
   certreq -F DC.in -R servicename_req.txt -K servicename_pkey.txt -
   P yourdcpassword
   ```
4. Create a `public key` file (*servicename*`.crt`) using the *servicename_req*`.txt` file with the CA `private key` file to sign the `public key` file. Enter this on one line:

   ```
   >certauth -C trusted.txt -Q servicename_req.txt
   -K CA_pkey.txt -P yourcapassword -T180 -O servicename.crt
   ```

   Here is an example of the expected result:

   ```
   Setting environment variables for this install....
   ```

   ```
   Using DC_SYBASE.bat Environment file from: C:\Sybase
   \DC-15_0\bin...
   ```

   ```
   1 file(s) copied.
   ```

   ```
   -- Sybase SSL Certificate Authority Utility --
   Certificate Validity:
         startDate = Thu Mar 20 10:21:41 2008
         endDate = Tue Sep 16 11:21:41 2008
   Setting serial number 0x31ab52626efa122f
   Could not sign certificate using signature type 20, error 'No
   error string returned.' (3000).
   Could not sign certificate using signature type 22, error 'No
   error string returned.' (3000)
   ```

```
CA sign certificate SUCCEED using signature type 2, return
'SSLNoErr' (0).
```

5. Append the signed service name `private key` file to the signed *<servicename>* `public key` file:

```
type servicename_pkey.txt >> servicename.crt
```

6. Copy the `trusted.txt` file to the *servicename*`.txt` file:

```
copy trusted.txt servicename.txt
```

7. Create and enter an encrypted password to establish an SSL connection:

```
pwdcrypt
```

Note: You cannot see the password you enter. This is your *yourcapassword*.

```
pwdcrypt
Enter password please:
Enter password again:
The encrypted password:
0x018c2e0ea8cfc44513e8ff06f3a1b20825288d0ae1ce79268d0e8669313d1bc
4c70c
```

8. From the `bin` directory, insert the encrypted password by copying from the previous step. Enter this on one line:

```
ECHO encrypted_password>servicename.pwd
```

Warning! To ensure that *servicename*`.pwd` contains a valid password, do not insert a space between *encrypted_password*, the symbol ">", and *servicename*`.pwd` file name.

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

```
copy trusted.txt srvname.txt
```

10. Verify that these files exist in the `C:\`*<install_dir>*`\DC-15_0\bin` directory:

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

Creating the Certificates Directory, Enabling SSL, and Verifying the Log Files

Learn the steps to create the certificates directory, enable SSL, and verify the log files.

1. Create a directory to hold the certificates:

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

2. Copy the `servicename.crt`, `servicename.pwd`, `servicename.txt`, and the `svrname.txt` files into the new directory:

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

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

3. Verify that the files are copied by listing the contents of the `certificates` directory:

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

   You should see:

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

   Note: The files located in the `\<install_dir>\DC-15_0\servers \<server_name>\certificates` directory include keys and password information. It is important to modify the permission of these files so that they are only viewable and writable by the user account that starts the server. To do this, modify the files' properties accordingly. Consult your IT security officer for further steps.

4. Change to `C:\<install_dir>\DC-15_0\servers\server.css.cfg`. Using a text editor like Notepad, edit the `server.cfg` file to enable the SSL service:
   a) Set the `SSLTrustedCertificateFile` property to the `SSLTrustedCertificateFile` path, for example:

```
SSLTrustedCertificateFile=C:\Sybase\DC-15_0\servers\dkxpsrv
\certificates
```

   b) Enter the name of the service in the `SSLServices` property that will use SSL:

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

   c) Enable SSL:

```
SSLEnabled=yes
```

5. Verify that the logging properties are set correctly:

```
(Logging)
LogWrap=yes
LogToScreen=yes
LogOCOSMessages=1
LogFlush=yes
LogFileSize=500000
LogFileName=
LogClientMessages=1
LogClientLogin=yes
```

Save the `server.cfg` file.

6. Use a text editor to append "ssl" to the `master` and `query` entries in the `sql.ini` file:

```
cd C:\<install_dir>\ini
```

```
notepad sql.ini
[srvname]
```

```
MASTER = NLWNSCK, machine name, port, ssl
Query = NLWNSCK, machine name, port, ssl
```

Save the `sql.ini` file.

7. Start the server:

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

8. Verify that these log entries exist in `C:\<install_dir>\DC-15_0\servers\<srvname>\log\<srvname>.log`:

```
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

Learn the steps to configure the SSL Windows client.

Note: You must restart the server after you update the system environment variables.

1. Set the client Sybase Window environment variable:

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

2. Enter:

```
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 to 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. Change to the Sybase Open Client and Open Server `bin` directory:

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

6. 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.

To test by examining SSL handshakes, Sybase recommends that you use the **ssldump** utility at *http://www.rtfm.com/ssldump/*.

Set Up SSL on UNIX Server

SSL on UNIX server provides encryption of data sent over the network, and authenticates clients and their passwords using digital certificates.

The task that provides SSL encryption and authenticates clients is invalid for the ECDA Option for Oracle. See *Enterprise Connect Data Access Option for Oracle Server Administration and Users Guide*.

Note: ECDA 15.0 or later does not support `transfer to` and `transfer from` on the SSL-enabled Adaptive Server servers.

Creating Certification Authority Files

Learn the steps to create the certification authority (CA) files.

1. Set up the Sybase environment variables:

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

Note: If you have already created or obtained a certificate, skip steps 2 through 5.

2. Change to the Sybase Open Client/Server™ `bin` directory:

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

3. Create the `CA.in` file. 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
```

Note: For more information about **certreq** parameters, see the *Adaptive Server Utilities Guide*.

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 yourcapassword
```

You see:

```
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 yourcapassword -T 365 -O trusted.txt
```

```
-- Sybase Test Certificate Authority Utility -- -- Certificate
Validity:
      startDate = Thu Mar 02 09:56:43 2008
      endDate = Fri Mar 20 09:58:10 2009
Setting serial number Ox1w7d236819a91a32
Could not sign certificate using signature type 20, error 'No
error string returned.' (3000).
Could not sign certificate using signature type 22, error 'No
error string returned.' (3000)
CA sign certificate SUCCEED using signature type 2, return
'SSLNoErr' (0).
```

Creating Certificates Specific to the DirectConnect Server and Service

Learn the steps to create the certificate of authority files for the specific DirectConnect server and service.

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

1. Enter the parameters for the CA.

```
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
```

Note: For more information about **certreq** parameters, see the *Adaptive Server Utilities Guide*.

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 yourdcpassword
```

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

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

You see:

```
-- Sybase SSL Certificate Authority Utility --
Certificate Validity:
      startDate = Wed May 28 11:51:53 2008
      endDate = Mon Nov 24 10:51:53 2008
setting serial number 0xffff940cffff8cb11ab
Could not sign certificate using signature type 20, error 'No
error string returned.' (3000).
Could not sign certificate using signature type 22, error 'No
error string returned.' (3000).
CA sign certificate SUCCEED using signature type 2, return
'SSLNoErr' (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 *<servicename>*`.txt` file:

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

7. Create and enter an encrypted password to establish an SSL connection:

```
prompt% pwdcrypt
```

Enter the password that is to be encrypted.

Note: You cannot see the password you enter. This is your *yourcapassword*.

```
Enter password again:
The encrypted password:
0x018c2e0ea8cfc44513e8ff06f3a1b20825288d0ae1ce79268d0e8669313d1bc
4c70c
```

8. Insert the encrypted password into a file:

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

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

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

10. Verify that all of these 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
```

Creating the Certificates Directory, Enabling SSL, and Verifying the Log Files

Learn the steps to create the certificates directory, enable SSL, and verify the log files.

Note: In this task, the DirectConnect server name and service name must be the same.

1. Create a 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 certificates directory created in the previous step:

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

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

```
cp <install_dir>/DC-15_0/bin/trusted.txt
/<install_dir>/DC-15_0/servers/<srvname> /certificates/.
```

3. Verify that the files are copied by listing the contents of the `certificates` directory:

```
<prompt>% cd /<install_dir>/DC-15_0
/<srvname>/certificates
```

```
prompt% ls
```

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

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

Note: The files located in the `/<install_dir>/DC-15_0/<srvname>/certificates` directory include keys and password information. It is important to modify the permission of these files so that they are only viewable and writable by the user

account that starts the server. To do this, change the files' permission to 600. Consult your IT security officer for further steps.

4. To enable the SSL service, edit the `server.cfg` file:

   - Enter the name of the service in the `SSLServices` property that will use SSL.
   - Enter **yes** in the `SSLEnabled`:

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

```
vi server.cfg
```

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

5. Verify that the logging properties are set correctly:

```
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:

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

8. Verify that these log file entries exist in */<install_dir>/DC-15_0/servers/<server name>/log /<server name>.log*:

```
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

Learn the steps to configure the SSL UNIX client.

1. Set the client SYBASE UNIX environment variable:

   ```
   source /<install_dir>/DC-15_0/DC_SYBASE.csh
   ```

2. Copy the `trusted.txt` file to the *<install_dir>* `/DC-15_0/connectivity/config` directory:

   ```
   cp /<install_dir>/OCS-15_0/bin/trusted.txt /<install_dir>/
   DC-15_0/connectivity/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. To connect to the service, issue:

   ```
   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.

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.

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.

Adding the Server as a Windows Service

Add the server as a Windows service using the **ServiceWrapper** utility.

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> <install_dir>\DC-15_0\bin\DCStart.bat -
S<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*.
- *install_dir* is the target installation directory where you installed the server.
- *server_name* is the name of the server.

For example:

```
ServiceWrapper.exe
--install dcw2ksrv
--password=password
C:\<install_dir>\DC-15_0\INSTALL\RUN_dcw2ksrvd.cmd
```

Starting the Server as a Windows Service

Start the server as a Windows service using the Administrative Tools.

Note: Starting and stopping the server as a Windows service on Windows 2008 R2 and Windows 7 may generate this 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.
```

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

Stopping the Server as a Windows Service

Stop the server as a Windows service using the Administrative Tools.

1. Select **Start > Settings > Control Panel > Administrative Tools > Services**.
2. Select the name of the Windows service you want to stop, and click **Stop the service**.

Removing the Server as a Windows Service

Remove the server as a Windows service using the **ServiceWrapper** utility.

Prerequisites

Stop the Windows service.

Task

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

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

2. Execute the **ServiceWrapper** utility.

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

   where *service_name* is the Windows service name.

Service Name Redirection

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

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` (Windows 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.

Edit a Service Name Redirection Table Using DirectConnect Manager

Use the DirectConnect Manager to 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 *Managing Service Name Redirection* in the DirectConnect Manager online help and select *Understanding service name redirection* or *Adding an item to the SNRF table*.

Edit a Service Name Redirection Table Using Command Line Syntax

Learn how to edit the `snrf` table using the command line.

Service Name Redirection File Format

Activate the service name redirection file by configuring the `ServiceRedirectionFile` property.

The `ServiceRedirectionFile` property indicates the name of the `Service Redirection` file, which is a text file consisting of four columns separated by tabs, that has as many rows as necessary need to define the redirections.

Using a text editor or DirectConnect Manager, you can change the name of the file. If you use a text editor, be sure that it inserts actual tabs, not just spaces that simulate tabs.

The format of a service name redirection file is:

```
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.

See also

- *Configure the DirectConnect Server* on page 21
- *ServiceRedirectionFile* on page 28

Null Service Names

Requests from the DB-Library™ applications contain null service names if the DB-Library versions do not provide a remote server name for service routing.

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 4. Sample of Null Service Name Format

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

Precedence Rules

The system uses precedence rules to resolve the problems if you inadvertently create a service name redirection file in which an assigned access service name is not uniquely specified.

The first rule defines the highest precedence, the eighth one the lowest.

Table 5. 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.

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

Table 6. 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

- 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.”

Service Name Redirection File Format Validation

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

Validate Service Name Redirection File Format Using Basic Command

The **snrfck** basic command requires only the **-i** option to read the redirection file, validate each line, and flag 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 the examples.

Next, assume the redirection file contains duplicate entries.

Table 7. Example of a Redirection File with a Duplicate Entry

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

In this example, **snrfck** returns:

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

If the file does not contain errors, the rows are sorted in the order used in the redirection operation and printed to the current window. The **snrfck** utility adds line numbers for clarity.

Table 8. 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

Validate Service Name Redirection File Format Using Specified Values

Test the redirection process by supplying values for `requested_service`, `user_id`, and `application_name`.

Testing the redirection is subject to these 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 is directed.

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

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

where:

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

Table 9. Redirection File with an Entry Match

	reques-ted_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

You see:

```
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.

Table 10. Redirection File with Failed Entry Match

	reques-ted_service	user_id	applica-tion_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				

You see:

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

Add Lines to the Redirection File

Add lines to the service name redirection file list by specifying the **-t** option.

When you use the **-t** 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 appear, but are not saved.

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

Table 11. 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

You see:

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

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

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

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

Table 12. 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.

Other snrfck Options

Other **snrfck** options show the version number and help text.

- For example, to show the version number, enter:

  ```
  snrfck -v
  ```

  snrfck returns:

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

- For example, to display help text, enter:

  ```
  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.

- On UNIX systems, use the **-?** argument as follows:

  ```
  snrfck -\?
  ```

Implement 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.

Substituting a Modified File to Implement a Service Name Redirection File

You can implement a new service name redirection file or copy a modified file.

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

Updating a Running Server Using snrfck

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.

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:

```
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.

Log and Trace Files

ECDA log and trace files provide troubleshooting information, but each is intended for a different audience.

Log File Description

The log file is a collection of records intended primarily for the system administrator.

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 at these levels:

- DirectConnect server
- Service library
- Access service

See also

- *Configure the DirectConnect Server* on page 21

Trace File Description

The trace file is a collection of records intended primarily for Sybase Technical Support personnel.

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

Configure the logging and tracing properties by using DirectConnect Manager or by editing server or access service library configuration files.

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 US 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.

DirectConnect Manager	Go to the DirectConnect Manager online help to to make configuration changes dynamically to the logging and tracing properties. Select **Editing server configuration properties** and **Modifying server configuration properties**.

Text editor	Before you edit server or access service library configuration files, stop the server, edit the configuration files, then restart the server for the changes to take effect.

See also

- *Configure the DirectConnect Server* on page 21

Reading the Log and Trace Files

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

Retrieving the Server Log File Using DirectConnect Manager

Access the server log file, retrieve its messages, and view them in a text editor using DirectConnect Manager.

You can retrieve the entire server log file, or set criteria to retrieve only a subset on the log file.

Go to the DirectConnect Manager online help, and select **Managing Server Administration > Filtering and retrieving the log**.

Log and Trace File Location

Log and trace files reside 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 default trace file name is *ServerName*`.trc`, where *ServerName* is the name you assigned to the DirectConnect server during installation.

Log and Trace File Structure

The log and trace files are ASCII text files and each contain 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.

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.

See also

- *Configure the DirectConnect Server* on page 21

Backup Log and Trace Files

Each time the DirectConnect server starts, it creates new log and trace files and the existing files are renamed as backup files.

The existing files are renamed as backup files, using this 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: To conserve disk space, periodically delete or archive the backup files.

Log and Trace Record Format

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

Table 13. 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)

Column	Description
Specific Information	The message text, which may contain embedded tabs to further separate the information in this column

If an access service library logs a message during its start or stop functions, client information is unavailable. 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.

Log Records Example

An example that shows the 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.

```
--------------------------------
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\admin.lcu
--- D:\SYB-15_0\DC-15_0\Connectivity\locales\unicode\econnect
\japanese\admin.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 --------------------
```

Table 14. 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.
19	The system sends log records to the log file but not to the current window.

Line Number	Log Record Entries
28	An indication of 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.
31, 39	Each access service library module installed in the `\DC-15_0\`*Server-Name*`\svclib` subdirectory is loaded in a specific order.
32	The access service library properties. 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. 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.
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.
53	An indication that server initialization is complete. Clients can now connect to any enabled access service.

Trace Records Example

An example showing trace records.

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!
```

ECDA Security

ECDA security uses the user ID and password combination, coupled with the 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 in 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 two entries:

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 must limit access to this file using standard file security techniques.

Administrator Credentials Management in DirectConnect Manager

The administrator user IDs and passwords must be consistent between the DCDirector and its directed DirectConnect servers.

If they are inconsistent, 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 user ID and password 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 user ID and password combinations, but it also reduces security.

Troubleshooting

Learn how to troubleshoot various start-up and file security issues.

Process Exit Codes

Use the process exit codes to diagnose start-up errors.

If the server terminates normally, it returns an exit code of 0 (zero) to the operating system.

Table 15. 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.

DirectConnect Server Error Messages During Start-Up

The DirectConnect server stops if it encounters server or access service library configuration errors during start-up.

If the server encounters errors in an access service configuration, start-up continues. In addition, if the server is configured for `snrf.tbl` and the table does not exist, the server creates one and populates it with "****Service A.**" However, the access service name redirection functionality does not work until you replace the prepopulated `snrf` table with valid information.

Displaying Error Messages from Start-Up

The server configuration error message lists the type of error and the line number where the error occurred.

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:

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

Access Service Configuration Error Conditions

Learn about the access service configuration errors.

Typical access service configuration errors include:

- The client user ID and password are used to log in to the database system. If the access service cannot 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, 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.

"Pre-Log" Messages During Start-Up

The server sends messages to the console and to the Windows event log (on Windows systems only) if the server configuration is invalid, or if an early start-up error occurs.

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.

Messages Sent to the Console

Error messages are sent to the console (`stderr`) when the DirectConnect server is started from the command line and encounters an error.

Table 16. Start-up Messages Sent to the Console

Console Messages	Description
`System info is invalid`	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`	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

Error messages are sent to the Windows Event Log when the DirectConnect server is started as a Windows service and encounters an error.

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

Table 17. 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.

Message	Description
`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 *SYBASEDSLISTEN* 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, and are not automatically erased. If you start and stop the server frequently, you should purge your Windows event log periodically.

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.

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 `snrf.tble` and the table does not exist, the server creates a new one and populates it with enough data to allow DirectConnect Manager to create new access services.

Troubleshooting File Security Issues

To fix problems due to corruption or user error in the `user.pwd` security file, delete the file. The system re-creates the file using default passwords. You can do this while the server is still running.

Next

After the file has been re-created, you can use DirectConnect Manager to reenter the user information. Alternatively, copy over the `user.pwd` file with a backup file of known users.

Obtaining Help and Additional Information

Use the Sybase Getting Started CD, Product Documentation site, and online help to learn more about this product release.

- The Getting Started CD (or download) – contains release bulletins and installation guides in PDF format, and may contain other documents or updated information.
- Product Documentation at *http://sybooks.sybase.com/* – is an online version of Sybase documentation that you can access using a standard Web browser. You can browse documents online, or download them as PDFs. In addition to product documentation, the Web site also has links to EBFs/Maintenance, Technical Documents, Case Management, Solved Cases, Community Forums/Newsgroups, and other resources.
- Online help in the product, if available.

To read or print PDF documents, you need Adobe Acrobat Reader, which is available as a free download from the *Adobe* Web site.

Note: A more recent release bulletin, with critical product or document information added after the product release, may be available from the Product Documentation Web site.

Technical Support

Get support for Sybase products.

If your organization has purchased a support contract for this product, then one or more of your colleagues is designated as an authorized support contact. If you have any questions, or if you need assistance during the installation process, ask a designated person to contact Sybase Technical Support or the Sybase subsidiary in your area.

Downloading Sybase EBFs and Maintenance Reports

Get EBFs and maintenance reports from the Sybase Web site or the SAP® Service Marketplace (SMP). The location you use depends on how you purchased the product.

- If you purchased the product directly from Sybase or from an authorized Sybase reseller:
  a) Point your Web browser to *http://www.sybase.com/support*.
  b) Select **Support > EBFs/Maintenance**.
  c) If prompted, enter your MySybase user name and password.
  d) (Optional) Select a filter, a time frame, or both, and click **Go**.
  e) Select a product.

Padlock icons indicate that you do not have download authorization for certain EBF/Maintenance releases because you are not registered as an authorized support contact. If you have not registered, but have valid information provided by your Sybase representative or through your support contract, click **My Account** to add the "Technical Support Contact" role to your MySybase profile.

f) Click the **Info** icon to display the EBF/Maintenance report, or click the product description to download the software.

- If you ordered your Sybase product under an SAP contract:
  a) Point your browser to *http://service.sap.com/swdc* and log in if prompted.
  b) Select **Search for Software Downloads** and enter the name of your product. Click **Search**.

Sybase Product and Component Certifications

Certification reports verify Sybase product performance on a particular platform.

To find the latest information about certifications:

- For partner product certifications, go to *http://www.sybase.com/detail_list?id=9784*
- For platform certifications, go to *http://certification.sybase.com/ucr/search.do*

Creating a MySybase Profile

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

1. Go to *http://www.sybase.com/mysybase*.
2. Click **Register Now**.

Accessibility Features

Accessibility ensures access to electronic information for all users, including those with disabilities.

Documentation for Sybase products is available in an HTML version that is designed for accessibility.

Vision impaired users can navigate through the online document with an adaptive technology such as a screen reader, or view it with a screen enlarger.

Sybase HTML documentation has been tested for compliance with accessibility requirements of Section 508 of the U.S Rehabilitation Act. 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 may 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 the Sybase Accessibility site: *http://www.sybase.com/products/accessibility*. The site includes links to information about Section 508 and W3C standards.

You may find additional information about accessibility features in the product documentation.

Glossary

Glossary of terms used in Enterprise Connect™ Data Access.

- **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. Adaptive Server 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 Adaptive Server that provides a Transact-SQL interface to various sources of external data. Component Integration Services allows Adaptive Server 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 characterizes 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*.
- **Adaptive Server Enterprise –** see *Adaptive Server Enterprise*.
- **Adaptive Server Enterprise /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, bind is an SNA command that establishes a connection between LUs, or a TCP/IP call that connects an application to a port on its system.
  - In DB2 UDB, bind 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, bind 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 ECDA Option for Oracle server or to Adaptive Server. 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 Adaptive Server. 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 that relates a name to 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 –** 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 Adaptive Server. 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.
- **ECDA Option for Oracle 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 Adaptive Server, 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 datasources, 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 datasources. 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 ECDA Option for Oracle 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 ECDA Option for Oracle 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 ECDA Option for Oracle 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 Adaptive Server application that maintains replicated data and processes data transactions received from a datasource.

- **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 Adaptive Server 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 ECDA Option for Oracle server to which a workstation can connect. The file must reside on every client machine that connects to Adaptive Server.
- **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 Adaptive Server 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 datasource. See also *service library*.
- **Transact-SQL –** a Sybase-enhanced version of the SQL database language used to communicate with Adaptive Server.
- **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*.
- **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 ECDA Option for Oracle 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

E

F

G

H

I

L

M

N

T

U

V

W