Installation Guide

Enterprise Connect™ Data Access

15.0

[Linux and UNIX]

DOCUMENT ID: DC39555-01-1500-02

LAST REVISED: January 2008

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

Contents

About This Book

This book describes how to install Sybase® Enterprise Connect™ Data Access (ECDA) Options on the following platforms:

- HP 9000(800) and HP Itanium
- IBM RISC System/6000 AIX (RS/6000 AIX)
- Sun Solaris
- Linux

Audience

This book is for System Administrators or other qualified persons familiar with their system environment, resources, and devices.

How to use this book

See these chapters for information about installing, setting up connectivity, and configuring the ECDA Options.

See	To
Chapter 1, "Understanding Enterprise Connect Data Access"	Understand ECDA Option for ODBC, ECDA Option for Oracle, and DirectConnect™ Manager.
Chapter 2, "Preparing to Install"	Identify the tasks required to get ready for installing ECDA.
Chapter 3, "Installing ECDA"	Install and uninstall ECDA Option for ODBC, ECDA Option for Oracle, and DirectConnect Manager.
Chapter 4, "Creating DirectConnect Servers Using the Create Server Wizard."	Create and start a DirectConnect server using a GUI.
Chapter 5, "Managing DirectConnect Servers and Services Using ECDA Utilities"	Create, start, and stop a server, and to create, start, and stop a service.
Chapter 6, "Managing DirectConnect Servers and Services Using DirectConnect Manager"	Create and start a server, and create and start an access service.
Chapter 7, "Managing Servers Using ECDA Option for ODBC Utilities"	Create and start a DCDirector server and a DirectConnect server using ECDA Option for ODBC utilities.

Chapter 8, “Configuring ECDA”	Configure client connectivity and to configure the DirectConnect server and access services.
Chapter 9, “Setting Up the ODBC Back-end Driver and Driver Manager”	Set up the ODBC driver manager.
Chapter 10, “Troubleshooting Installation”	Identify and resolve problems that can occur when you install an ECDA product.
Appendix A, “Connectivity Tasks for AS/400”	Set up the AS/400 to communicate with ECDA products.
Appendix B, “Installation Worksheets”	Access worksheets and instructions for recording installation information for ECDA products.
Appendix C, “Configuring Data Sources to Connect to Targets Using DataDirect”	Configure data sources for various targets.
Glossary	Find definitions of technical terms used in this guide.

Related documents

For more information about configuring and administering ECDA, use these guides:

- *Release Bulletin* Enterprise Connect Data Access 15.0 for Linux and UNIX.
- Enterprise Connect Data Access Option for ODBC *Users Guide for Access Services*
- Enterprise Connect Data Access and Mainframe Connect *Server Administration Guide*
- Enterprise Connect Data Access Option for Oracle *Server Administration and Users Guide*
- For ODBC information, use *Microsoft ODBC 3.5 Programmer’s Reference and SDK Guide*

For additional references, use these documents:

- Open Client™ *Client-Library/C Reference Manual*
- Open Server™ *Server-Library/C Reference Manual*
- Mainframe Connect™ Client Option and Server Option *Messages and Codes*
- *Component Integrated Services User’s Guide* for Adaptive Server® Enterprise

- Sybase Adaptive Server® *Reference Manual*, volumes 1 and 2

To administer the ECDA DirectConnect server using DirectConnect Manager, use the DirectConnect Manager online help.

Other sources of information

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

- The Getting Started CD contains release bulletins and installation guides in PDF format, and may also contain other documents or updated information not included on the SyBooks CD. It is included with your software. To read or print documents on the Getting Started CD, you need Adobe Acrobat Reader, which you can download at no charge from the Adobe Web site using a link provided on the CD.
- The SyBooks CD contains product manuals and is included with your software. The Eclipse-based SyBooks browser allows you to access the manuals in an easy-to-use, HTML-based format.

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

  Refer to the *SyBooks Installation Guide* on the Getting Started CD, or the *README.txt* file on the SyBooks CD for instructions on installing and starting SyBooks.
- The Sybase Product Manuals Web site is an online version of the SyBooks CD that you can access using a standard Web browser. In addition to product manuals, you will find links to EBFs/Maintenance, Technical Documents, Case Management, Solved Cases, newsgroups, and the Sybase Developer Network.

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

Sybase certifications on the Web

Technical documentation at the Sybase Web site is updated frequently.

❖ **Finding the latest information on product certifications**

1. Point your Web browser to Technical Documents at http://www.sybase.com/support/techdocs/.
2. Click Certification Report.
3. In the Certification Report filter select a product, platform, and time frame and then click Go.
4. Click a Certification Report title to display the report.

❖ Finding the latest information on component certifications

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

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

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

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

Sybase EBFs and software maintenance

❖ Finding the latest information on EBFs and software maintenance

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

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

Syntax conventions

Syntax statements that display options for a command look like this:

COMMAND [*object_name*, [{TRUE | FALSE}]]

Table 1 explains the syntax conventions used in this guide.

Table 1: Syntax conventions

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

Style conventions

Table 2 explains the style conventions used in this guide.

Table 2: Style conventions

This type of information	Looks like this
Gateway-Library function names	TDINIT, TDCANCEL
Client-Library function names	CTBINIT, CTBCANCEL
Other executables (DB-Library™ routines, SQL commands) in text	the dbrpcparam routine, a select statement
Directory names, path names, and file names	*/usr/bin directory, interfaces* file
Variables	*n* bytes
Adaptive Server® Enterprise (ASE) datatypes	datetime, float
Sample code	`01 BUFFER   PIC S9(9) COMP SYNC`
User input	*01 BUFFER PIC X(n)*
Client-Library and Gateway-Library function argument names	*BUFFER, RETCODE*
Names of objects stored on the mainframe	SYCTSAA5
Symbolic values used with function arguments, properties, and structure fields	CS_UNUSED, FMT_NAME, CS_SV_FATAL
Client-Library property names	CS_PASSWORD, CS_USERNAME
Client-Library and Gateway-Library datatypes	CS_CHAR_TYPE

Accessibility features

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

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

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

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

See Section 508 compliance statement for DirectConnect Manager for Voluntary Product Assessment Templates at http://www.sybase.com/detail_list?id=52484.

If you need help

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

CHAPTER 1

Understanding Enterprise Connect Data Access

ECDA Options

This section describes the two options that are available for ECDA:

- ECDA Option for ODBC
- ECDA Option for Oracle

ECDA Option for ODBC

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

ECDA Option for ODBC provides basic connectivity to DB2 UDB, Microsoft SQL Server, and ODBC-accessible databases.

Note The ODBC driver for ECDA Option for ODBC (the back-end driver connecting to the target) is not provided by Sybase; you must obtain, install, and configure it.

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

Note Be sure to verify that your ODBC driver is compatible with Sybase driver manager software or that it contains a driver manager.

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.

ECDA Option for Oracle

ECDA Option for Oracle provides Open Client access to Oracle databases. It operates in conjunction with the Adaptive Server Enterprise/Component Integration Services feature (ASE/CIS) and as a standalone gateway. For more information, see the ECDA Option for Oracle *Server Administration and Users Guide*.

Using Adaptive Server, you can join Oracle tables with tables in Adaptive Server, DB2 UDB, or other database servers. Access to these objects through Adaptive Server is transparent to the application.

In standalone mode, ECDA Option for Oracle provides client applications with an Open Client interface to Oracle databases. To the client, ECDA Option for Oracle appears as an Open Server application that understands Oracle SQL.

DirectConnect Manager

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

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

You can configure properties using either DirectConnect Manager or a text editor. However, Sybase recommends using DirectConnect Manager for these reasons:

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

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

CHAPTER 2 Preparing to Install

Review the installation process

This section lists the pre-installation, installation, and post-installation steps, including the chapter or appendix where you can find the detailed instructions.

Note To resolve problems that occur when you install ECDA, see Chapter 10, “Troubleshooting Installation.”

Table 2-1: Installation steps for ECDA

Step		Resource
Pre-installation		
1	Set up connectivity to the target database.	• Appendix A, “Connectivity Tasks for AS/400.” • Appendix C, “Configuring Data Sources to Connect to Targets Using DataDirect.”
2	Fill out the installation worksheets.	Appendix B, “Installation Worksheets.”
3	Gather your information and prepare the environment.	Chapter 2, “Preparing to Install.”
Installation		
1	Obtain a SySAM license.	Chapter 3, “Installing ECDA.”

Step		Resource
2	Install ECDA Option for ODBC, ECDA Option for Oracle, and DirectConnect Manager.	Chapter 3, “Installing ECDA.”
3	Create and start a DirectConnect server (using the Create Server Wizard).	Chapter 4, “Creating DirectConnect Servers Using the Create Server Wizard.”
4	Start, configure, and stop DirectConnect servers and access services (using Sybase utilities).	• Chapter 5, “Managing DirectConnect Servers and Services Using ECDA Utilities.” • Chapter 6, “Managing DirectConnect Servers and Services Using DirectConnect Manager.” • Chapter 7, “Managing Servers Using ECDA Option for ODBC Utilities.”
Post-installation		
1	Configure client connectivity to ECDA.	Chapter 8, “Configuring ECDA.”
2	Configure the DirectConnect server and access services.	Chapter 8, “Configuring ECDA.”
3	Set up back-end drivers and configure the Driver Manager.	Chapter 9, “Setting Up the ODBC Back-end Driver and Driver Manager.”

Gather your installation team

To install ECDA, you need a team with specific skills and experience—as well as the authority to carry out the variety of tasks in the installation process.

At your site, identify the people who have the skill sets shown in Table 2-2 and keep them informed of any changes.

Table 2-2: ECDA team skill requirements

Role	Skill set
Operating system administrator	• Understanding of the operating systems for your site's platforms • Knowledge of standards and conventions at the installation site
Communications administrator	• Understanding of connectivity products used at your site • Ability to design, establish, test, and troubleshoot remote physical communications between ECDA and the mainframe host for DB2 UDB access • Understanding of your network configuration
ECDA administrator	• Understanding of the ECDA environment • DirectConnect server Administrator privileges
Target database administrator	• Knowledge of the target database • Knowledge of target environment, including security operations • Database administrator privileges
LAN administrator	• Understanding of LAN communications at your site • Ability to design, establish, test, and troubleshoot remote physical communications between the client and the DirectConnect server • Understanding of the site's network configuration

Prepare the environment

Before you begin, you must build the foundation that will support your ECDA installation and the subsequent access services that you create. If you complete these tasks first, the installation process should be successful.

This section describes these tasks:

- Connectivity tasks
- Critical administrative tasks

Connectivity tasks

Before you begin installation, you must have some preliminary network connections in place and operational. Perform these tasks before you set up connectivity:

- Set up connectivity to the target database
- Check the system requirements
- Research connectivity parameters

Note This is not a complete list. Your team should prepare a list that is pertinent to your site and connectivity setup.

Set up connectivity to the target database

To set up connectivity to the target database, refer to vendor documentation for connectivity protocols that your site supports.

Check the system requirements

Verify that you have the platform and operating system components listed in the ECDA 15.0 *Release Bulletin* for Linux and UNIX. Also, be sure to determine whether any new release levels or bug fixes of Sybase or vendor connectivity protocol products are required.

Research connectivity parameters

You need the data source name (DSN) values for each ODBC target when you set up connectivity for ECDA.

Be sure to record the DSN values on your installation worksheet.

Sybase does not ship ODBC drivers with ECDA to provide connectivity to non-Sybase target databases. You must obtain a separate driver from IBM, Microsoft, DataDirect, or other vendors, for the ODBC driver that is needed on the particular platform where ECDA is running.

Note In some cases, you may already be licensed for ODBC drivers for the databases you are accessing. Refer to the vendor contract and documentation.

Do not continue installation until connectivity is running successfully between the machine that will host the DirectConnect server and the target database.

Critical administrative tasks

Complete these administrative tasks:

- Read the Enterprise Connect Data Access 15.0 *Release Bulletin*. This document provides product information that might not be included in this ECDA guide, as well as information about known issues or problems.
- Make a backup copy of your current Sybase software, particularly if you plan to keep ECDA versions that were previously installed.
- Make a copy of the installation worksheet for your platform for each server that you will install.
- Verify that you have authority to sign in as user sybase or another user that has administrative privileges on the machine on which ECDA is being installed.
- Verify that the host server is connected to the LAN and that network protocols are configured correctly. For instructions, see your LAN administrator documentation.
- Verify previously installed Sybase products. When you install ECDA into an existing directory structure, be aware of any previously installed Sybase software and the version of that software.

CHAPTER 3 Installing ECDA

Installing ECDA

This section describes:

- SySAM licensing requirements
- Using the installation program
- Installing ECDA Options
- Installing DirectConnect Manager

SySAM licensing requirements

ECDA version 15.0 includes a Sybase Software Asset Management System (SySAM) implementation to perform license administration and asset management tasks. Sybase recommends that you get your license files from the Sybase Product Download Center (SPDC) at https://sybase.subscribenet.com before you begin installation. Accessing SPDC requires a separate login and password information through e-mail.

Warning! If a product license cannot be obtained during the grace period (usually 30 days), ECDA will cease to operate.

For information about SySAM 2.0, see the Sybase Software Asset Management *Users Guide*.

Using the installation program

The installation program creates the target directory (if necessary) and loads all of the selected components into that directory.

Installation options

Install ECDA using one of the options:

- GUI (graphical user interface) mode, which allows you to install the components using the installation program described in“Installing ECDA in GUI mode” on page 13.
- Console mode, which allows you to install components using a command line interface described in “Installing ECDA in console mode” on page 16.
- Response file mode, which allows you to record or create a response file described in “Installing ECDA using a response file” on page 17. Using a response file, you can install ECDA two different ways:
  - Silent, which lets you install the product without any interaction required on your part. This is convenient if you are performing identical installations on multiple machines.
  - Interactive installation using response file, which lets you install interactively but with all the responses already filled in, so that you can accept or change the default values and install ECDA according to the responses in the response file. This can be convenient if several sites are installing ECDA and must conform to a standard installation.

Each of these options is described in the following sections.

Note Sybase recommends that you use the GUI mode for installing ECDA Options.

Installing ECDA Options

This section describes how to install the ECDA Option for ODBC and the ECDA Option for Oracle in GUI mode.

Installing ECDA in GUI mode

❖ **To install ECDA Options**

1 Verify that the drive on which you will install the products has enough free disk space for the products being installed, and at least 10MB extra disk space for the installation program. The program frees this extra space after the installation is complete.

2 Insert the ECDA CD-ROM into the CD-ROM drive.

If auto-mount is not available on your machine, follow local procedures for mounting the CD-ROM.

For IBM AIX, HP-UX, HP Itanium, Linux, and Sun Solaris platforms, enter:

```
./setup
```

The Welcome window appears.

3 Click Next to proceed with the installation.

You may see the following error message:

```
Error writing file =  There may not be enough
temporary disk space. Try using -is:tempdir to use a
temporary directory on a partition with more disk
space.
```

If so, set the temporary directory to another directory that has more disk space by entering the following at the command line:

```
./setup -is:tempdir <directory_name>
```

where *directory_name* is the name of the temporary directory to which the installation program will write its temporary files.

4 Read the License and Copyright Agreement. Using the drop-down list at the top of the window, select the geographic location where you are installing ECDA to display the agreement appropriate to your region. Select “I agree” and click Next.

You must agree to the license and copyright agreement before you can continue.

5 In the installation directory window, click Next to accept the default directory for the installation, or enter a different directory where you want to install ECDA. One of the following occurs:

- If the installation directory you chose does not exist, the installation program prompts:

  ```
  The directory does not exist. Do you want to
  create it?
  ```

  Click Yes.

- If the installation directory exists and contains files, the software prompts:

  ```
  You have chosen to install into an existing
  directory. Any older versions of the products you
  choose to install that are detected in this
  directory will be replaced.
  ```

  Note You will not see this message if the directory exists but is empty.

  If you are prompted to overwrite any files, select Yes *only* if the version of the new files is later than the one you are attempting to overwrite.

  Note In certain cases when ECDA is installed with other Sybase products, you may see warnings about overwriting newer versions of files. In these cases, simply select Yes to All to instruct the installation program to overwrite these files and proceed with the installation.

  ---

6 Select one of the two installation types:

- Custom, which allows you to select the ECDA options and components to install
- Full, which installs all ECDA products and components from the CD

Click Next.

7 Based on your selection:

- If you selected Custom, the next window displays all the ECDA options and all the components “checked” or selected. You must “uncheck” or deselect the components that you do *not* want to install.

  Note Be aware that if you deselect a component that is required, it is automatically installed if it is needed to run other selected components.

After you have selected either the ECDA Option for ODBC or ECDA Option for Oracle, and selected the desired components, click Next.

- If you selected Full, the next window displays all the ECDA options including ECDA Option for ODBC and ECDA Option for Oracle, and all of the components "checked" (selected). Click Next.

Before proceeding to the next window, the installation program verifies the selections, and checks for dependencies and available disk space. The Product Summary window displays every component that is to be installed and the total disk space required for all the selected components.

Note If the target directory does not have enough free space, the space-required and the space-available information is displayed. If you click Next without sufficient hard disk space, an error occurs that stops the installation.

8 Verify that you have selected the correct type of installation, and that you have enough disk space to complete the process. Click Next to continue the installation.

The installation program unloads all the components from the CD and displays a progress indicator. When the installation is complete, a message appears indicating that the installation program wizard has completed installing your product.

9 Click Next. The SySAM License Server window opens and displays this prompt:

```
Will licenses be obtained from the License Server?
```

- Select Yes if you have a pre-existing SySAM network license server installed or will install a new license server. See "SySAM licensing requirements" on page 11.

  Enter the host name and the port number of the license server.

- Select No if you do not have a SySAM license server installed and are going to use a local license file. A message appears telling you to download the license file.

If you select Yes and the license server is not found, or if you select No and the local license file is not installed on your local machine, this message appears:

```
Installer can’t check out a license. Do you want to
continue without a license?
```

10 Click Yes to complete the installation. If you select No, this message appears:

```
Please remember to download and install the license
file after this installation.
```

Click OK.

The SySAM Notification window asks you to configure your server for e-mail notification. When configuration is enabled, you will receive information about license management events requiring attention.

11 Enter values for these fields:

- SMTP server host nam
- SMTP server port number
- E-mail Return Address
- Recipient e-mail addresses
- Message severity that triggers e-mail messages

Click Next.

12 A window displays a message indicating that the installation was successful and advising you to check for software updates. Click Finish.

Installing ECDA in console mode

If you want to run the installation program without the graphical user interface (GUI), you can launch the installation program in console mode. In cases where the installation program launches automatically, click Cancel to cancel the GUI installation and then launch the setup program from a terminal or console.

❖ To install in console mode

The steps for installing components in console mode are the same as those described in “Installing ECDA Options” on page 12, except that you execute the installation program from the command line using the setup -console command, and you enter text to select the installation options, as follows:

1 For IBM AIX, HP-UX, HP-Itanium, Linux, and Sun Solaris, enter this at the command line:

```
./setup -console
```

The installation program starts and displays a Welcome message.

2 The flow of the installation is identical to a regular GUI installation, except that the display is written to a terminal window, and you enter the responses using the keyboard. Follow the remaining prompts to install ECDA.

Installing ECDA using a response file

A silent installation (sometimes referred to as an unattended install) is done by running the installation program and providing a response file that contains answers to all of the installation program questions.

Creating a response file

There are two methods of generating a response file for the installation program:

- *Record* mode: In this mode, the installation program performs an installation of the product, and records all of your responses and selections in the specified response file. You must complete the installation to generate a response file. To create a response file, enter the following:

```
./setup -options-record <responseFileName>
```

Note There should be no space between *-options* and *-record*.

where *responseFileName* is a name you choose for the response file.

These are the results:

  - An installation of ECDA on your computer
  - A response file containing all of your responses from the installation

    If this response file is used for a silent installation, the resulting installation is identical to the one from which the response file was created: the same installation location, same feature selection, and all ofthe same remaining information. The response file is a text file that you can edit to change any responses prior to using it in any subsequent installations.

- *Template* mode: In this mode, the installation program creates a response file containing commented-out values for all required responses and selections. However, you do not need to install the product, and you can cancel the installation after the response file has been created. To create this template file, enter the following:

```
./setup -console -options-template <responseFileName>
```

where *responseFileName* is the absolute file name you chose for the response file, for example:

```
/sybase/DC/OptionsTemplate.txt
```

If run in console mode, as shown in the previous example, the installation program provides a message indicating that the template creation was successful. If run in GUI mode, no message is provided.

If you use this response file for a silent installation, the default values for all responses are used. Edit the template with the values you want to use during installation.

Warning! If you created a response file using a Custom installation, you need to edit the response file to allow the custom selections to be chosen correctly during installation. Following is a workaround for an installer issue when using a response file.

Use a text editor and delete the word "Custom" in the setup type in this line:

```
-W setupTypes.selectedSetupTypeId=Custom
```

The resulting line should look similar to this:

```
-W setupTypes.selectedSetupTypeId=
```

Failure to delete the “Custom” setup type results in a Full installation of the product.

Interactive installation using a response file An interactive installation using a response file allows you to accept the default values obtained from the response file that you have set up, or to change any of those values for the specific installation. This is useful when you want multiple similar installations but with some minor differences that you want to change at installation time.

At the command line, execute the following command. (Enter the command all on one line.)

```
./setup -console -options <responseFileName>
-W SybaseLicense.agreeToLicense=true
```

Installing in silent mode A silent mode installation, sometimes referred to as an unattended installation, allows you to install the product with all responses being taken from the response file that you have set up. There is no user interaction. This is useful when you want multiple identical installations or when you want to automate the installation process.

At the command line, execute the following command (enter the command all on one line):

```
./setup -silent -options /<responseFileName>
-W SybaseLicense.agreeToLicense=true
```

where *responseFileName* is the name of the file containing the installation options you chose. The -W option specifies that you agree with the Sybase License Agreement text.

Except for the absence of the GUI screens, all actions of the installation program are the same, and the result of an installation in silent mode is exactly the same as one done in GUI mode with the same responses.

Installing DirectConnect Manager

This section describes the installation process for DirectConnect Manager.

❖ **To install DirectConnect Manager**

1 Insert the PC Client CD into the CD-ROM drive.

2 Go to the DirectConnect Manager directory.

3 Execute the following, depending on the platform:

- For AIX:

  ```
  setupaix.bin
  ```

- For Solaris:

  ```
  setupsolarisSparc.bin
  ```

- For HP-Itanium:

  ```
  setupHPIA64.bin
  ```

- For HP-UX:

  ```
  setuphp11x.bin
  ```

- For Linux:

  ```
  setuplinux.bin
  ```

4. The Welcome window appears. Click Next.
5. The License and Copyright Agreement window appears.
6. Read the License and Copyright Agreement. Using the drop-down list at the top of the window, select the geographic location where you are installing to display the agreement appropriate to your region. Select “I agree” and click Next. The Choose Directory dialog box appears.

   Note You must agree to the license and copyright agreement before you can continue.

7. Enter the directory path for the installation and click Next.
8. Select one of the two types of installation:
   - Install a new version of Sybase Central™ and render the other installation invalid. Click Next.

     This type of installation will install DirectConnect Manager as well as all the other necessary components and create a new set of icons and Registry entries.

     Note This option will make your current Sybase Central installation invalid.

   - Register with your existing Sybase Central. Click Next.

     This type of installation will install the DirectConnect Manager plug-in and register it with the existing Sybase Central application.
9. A Summary window appears and lists all the features that you are going to install:.
   - If the information is correct, click Next
   - If the information is not correct, click Back and make the corrections.
10. The installation process begins and a Setup window appears, which shows the percentage (%) of installation that is complete and monitors the decompressing and installing of the DirectConnect Manager files. The installation will register the plug-in with Sybase Central. Click Next.
11. A status window appears, showing the Sybase Central registration process. Click Next.
12. When installation is completed, a DirectConnect Manager *README* file appears. After reading the file, click Finish to clear the window.

Uninstalling ECDA

Note Before uninstalling Sybase software, shut down all Sybase applications and processes.

To uninstall ECDA, use the installation program uninstall feature. This removes all servers, all common files, and all required components. After the uninstall procedure runs, you may need to delete any remaining files and directories in the directory where ECDA was installed.

You can invoke the uninstall procedure using either the GUI or the console method. However, Sybase recommends the GUI method.

Log in to your machine using an account with “administrator” privileges, then shut down all other processes for the components you are uninstalling.

Note The uninstall procedure removes only those files that were loaded from the installation media. Some Sybase files, such as log and configuration files, are left intact for administrative purposes.

❖ To uninstall in GUI mode

1. Verify that the ECDA server is shut down.
2. Proceed to the *uninstall* directory:
   - Go to the directory where your application is installed.
   - Go to the *uninstall* directory.
   - Go to the directory for your application that you want to uninstall, *ECDA150Suite*.
   - Enter uninstall from the command line. The wizard appears.
3. Click Next. All of the products and components that were installed are displayed and already checked to allow you to remove the total installation. If you do not want to remove a product or component, deselect or “un-check” that product or component.
4. Click Next. A summary of all the products and components appears.
5. Click Next. A message indicating that an uninstall of ECDA is in progress. When this is completed, a message is displayed indicating a successful uninstall.
6. Click Finish to complete the uninstall process.

❖ **To uninstall in console mode**

1. Execute the following at the command line:

```
<install_dir>/uninstall/ECDA150Suite/uninstall -console
```

   The uninstall program starts.

2. Choose the ECDA software product you want to uninstall.
3. The ECDA software product you chose is uninstalled.

CHAPTER 4

Creating DirectConnect Servers Using the Create Server Wizard

Create Server wizard overview

The Create Server wizard is a Java program that allows you to create a DirectConnect server and complete the connection to the target database based on the driver and driver manager you want to use. It supports:

- ECDA Option for ODBC
- ECDA Option for Oracle

After you create the DirectConnect server, you can use DirectConnect Manager to configure a service to connect to a specific target database and to test the configuration. For more information, see Chapter 6, “Managing DirectConnect Servers and Services Using DirectConnect Manager.”

Creating a server in ECDA Option for ODBC

This section describes how to create a server for the target databases for ECDA Option for ODBC for these drivers:

- Sybase-provided unixODBC driver manager
- DataDirect driver

- IBM DB2 driver

Note For ECDA Option for ODBC, the Create Server wizard is located in the *<install_dir>/DC-15_0/DCWizard* directory.

Sybase-provided unixODBC driver manager

❖ **To create a server that will use a unixODBC driver manager**

1 Start the Create Server wizard by entering:

```
DCWizard.sh
```

2 When the Welcome Create Server Wizard window appears, Click Next. The Create DirectConnect Server Options window appears.

3 Select the ECDA Option for ODBC option and click Next.
The Server Name and Port Information window appears.

4 Enter the Server Name for the new DirectConnect server and the Port Number that you want the server to “listen” on. Click Next.
The ECDA - ODBC Driver Manager Options window appears.

5 Select the unixODBC Driver Manager option and click Next.
The DirectConnect Service Name window appears.

6 Enter the ECDA access service name that you want to use for this server and click Next. The ECDA Server Summary and Build window appears.

- If correct, select Create Server. The Start DirectConnect Server (Optional) window appears.
- If incorrect, click Back to change the information.

7 Optionally, you can start the DirectConnect server that you created. Click Finish.

Note The service requires additional configuration to connect to the target database. This additional configuration can be added using DirectConnect Manager. For basic connectivity to the target database, the next step is to start DirectConnect Manager and set ConnectionSpec1 to a data source name (DSN) contained in the *odbc.ini* file located at:

```
<dc_install_dir>/DC-15_0/odbc.ini.
```

8 Set the configuration property called SQLODBCCursors to driver in the *dcany.cfg* file.

DataDirect driver

Before you create a DirectConnect server:

- Install the DataDirect ODBC software on the same machine that you are installing the ODBC option.
- Configure your DataDirect *odbc.ini* file with a data source name (DSN) for your target database. Verify connectivity using the DataDirect tools.

❖ To create a server using a DataDirect driver

1 Start the Create Server wizard by entering:

```
DCWizard.sh
```

2 When the Welcome Create Server Wizard window appears, click Next. The Create DirectConnect Server Options window appears.

3 Select the ECDA Option for ODBC option and click Next.
The Server Name and Port Information window appears.

4 Enter the Server Name for the new DirectConnect server and the Port Number that you want the server to “listen” on. Click Next.
The ECDA - ODBC Driver Manager Options window appears.

5 Select the DataDirect Driver Manager option to use for this server. Click Next. The Driver Manager Information window appears.

6 Specify the following from the browse files and directories:

- The path to the driver manager within the DataDirect installation, for example:

  ```
  <datadirect_ install_dir>/lib/libodbc.so
  ```

- The path to the Driver Manager support libraries, for example:

  ```
  <datadirect_install_dir>/lib/
  ```

- The path to the Driver Manager *odbc.ini* file you want to use, for example:

```
<datadirect_install_dir>/odbc.ini/
```

Click Next. The DirectConnect Service Name window appears.

7 Enter the ECDA access service name that you want to use for this server. Click Next. The ECDA Server Summary and Build window appears.

8 Verify the ECDA Server information that appears.

- If correct, select Create Server. The Start DirectConnect Server (Optional) window appears.
- If incorrect, click Back to change the information.

9 Optionally, you can start the DirectConnect server that you created to validate the server configuration. If you exit from this wizard, the DirectConnect server you have started will continue to execute. To skip this step, click Finish.

Note The service requires additional configuration to connect to the target database. To add this configuration, use DirectConnect Manager. For basic connectivity to the target database, the next step is to start DirectConnect Manager and set ConnectionSpec1 to a data source name (DSN) contained in the *odbc.ini* file .

10 Set the configuration property called SQLODBCCursors to driver in the *dcany.cfg* file.

IBM DB2 driver

Before you create a DirectConnect server:

- Install the IBM DB2 client software on the same machine that you are installing the ODBC option.
- Configure your IBM DB2 client software with a defined database alias. Verify connectivity using IBM tools.

❖ To create a server using an IBM driver

1 Start the Create Server wizard by entering:

```
DCWizard.sh
```

2 When the Welcome Create Server Wizard window appears, click Next. The Create DirectConnect Server Options window appears.

3 Select the ECDA Option for ODBC option and click Next.
The Server Name and Port Information window appears.

4 Enter the Server Name for the new DirectConnect server and the Port Number that you want the server to "listen" on. Click Next.
The ECDA - ODBC Driver Manager Options window appears.

5 Select the unixODBC Driver Manager without a Driver Manager Option and click Next. The Driver Manager Information window appears.

6 Select the following from the browse files:

- The path to the Sun Solaris IBM ODBC Driver file, for example:

  ```
  /software/IBM/db2_08_01/lib/libdb2o.so
  ```

  Note Version 9.1 of the IBM ODBC 64-bit driver has a new name ("libdb2o.so"), as shown in the directory path.

- The *DB2 script* file that is sourced before DirectConnect is started, for example:

  ```
  /export/home/nedb2i7/sqllib/db2cshrc
  ```

Click Next. The DirectConnect Service Name window appears.

7 Enter the ECDA access service name that you want to use for this server and click Next. The ECDA Server Summary and Build window appears.

8 Verify the ECDA Server information that appears:

- If correct, select Create Server. The Start DirectConnect Server (Optional) window appears.
- If incorrect, click Back to change the information.

9 Optionally, you can start the DirectConnect server that you created to validate the server configuration. If you exit from this wizard, the DirectConnect server you have started will continue to execute. If you want to skip this step, click Finish.

Note The service requires additional configuration to connect to the target database. This additional configuration can be added using DirectConnect Manager. For basic connectivity to DB2, start DirectConnect Manager and set ConnectionSpec1 to a DB2 alias. The DB2 aliases can be found by executing “list database directory” in the db2 command shell.

Creating a server in ECDA Option for Oracle

Note For ECDA Option for Oracle, the Create Server wizard is located in the *<install_dir>/DCO-15_0/DCWizard* directory.

❖ To create a server for Oracle Option

1 Start the Create Server wizard by entering:

```
DCWizard.sh
```

2 When the Welcome Create Server Wizard window appears, click Next. The Create DirectConnect Server Options window appears.

3 Select the ECDA Option for Oracle option. Click Next.
The Server Name and Port Information window appears.

4 Enter the Server Name for the new DirectConnect server and the Port Number that you want the server to “listen” on. Click Next.
The DirectConnect Server Admin Account and Password window appears.

5 Enter the Admin Account Name. Click Next The Oracle Target Information window appears.

6 Enter the following information:

- Oracle Connect String
- Path to the *tnsnames.ora* file, including the file name

7 Click Next. The Oracle DirectConnect Server Summary and Build window appears.

8 Verify the ECDA Server information that appears:

- If correct, select Create Server. The Oracle DirectConnect Server Summary and Build window appears
- If incorrect, click Back to change the information.

9 Verify the ECDA Config information that appears:

- If correct, select Create Server. The Start DirectConnect Server (Optional) window appears.
- If incorrect, click Back button to change the information.

10 Optionally, you can start the DirectConnect server that you created to validate the server configuration. If you exit from this wizard, the DirectConnect server you have started will continue to execute. If you want to skip this step, click Finish.

Note You can use DirectConnect Manager to change configuration settings.

CHAPTER 5

Managing DirectConnect Servers and Services Using ECDA Utilities

ECDA utilities overview

ECDA utilities allow you to create, start, and stop a server, and create, start and stop a service. You can also use DirectConnect Manager to manage servers and services. For details, see Chapter 6, “Managing DirectConnect Servers and Services Using DirectConnect Manager.”

Managing ECDA Option for ODBC

This section describes how to create and start a server and service using the ECDA utilities and the command line for ECDA Option for ODBC. For more ECDA utility information, see Chapter 7, “Managing Servers Using ECDA Option for ODBC Utilities.”

Creating a new DirectConnect server

At this point, the installation program has automatically created the subdirectories and files needed for ECDA. You must perform a number of tasks to create a new DirectConnect server and access services.The first step is to create a new server by running the AddServer utility from *<install_dir>/DC-15_0/bin* using the syntax in this example:

```
cd <install_dir>/DC-15_0/bin
AddServer srvname 1234
```

where:

- *srvname* is the name of the new server.
- *1234* is the port that it will listen on.
- AddServer is the utility that will:
  - Set the environment
  - Create the server
  - Start the server
  - Insert the server name in the Sybase *interfaces* file

Set up and verify your environment

Note ECDA has created utilities that will allow you to create and start a DirectConnect server. Refer to Chapter 7, “Managing Servers Using ECDA Option for ODBC Utilities.”

Set the environment variables from a command line prompt by running one of the following environment scripts, as appropriate for your platform:

- *DC_SYBASE.csh*
- *DC_SYBASE.sh*

Sybase provides scripts to set the following variables, which ECDA requires in order to run:

- echo $SYBASE
  response: *<install_dir>/DC-15_0/connectivity*
- echo $SYBASE_ECON
  response: .. *(parent directory)*

- echo $SYBASE_OCS
  response: *OCS-15_0*
- SHLIB_PATH (for HP-UX), LIBPATH (for AIX), LD_LIBRARY_PATH (for Sun Solaris, Linux, HP Itanium)

The response should match the home directory on your worksheet:

- For HP:

  ```
  echo $SHLIB_PATH
  ```

- For AIX:

  ```
  echo $LIB_PATH
  ```

- For Solaris, Linux, and HP Itanium:

  ```
  echo $LD_LIBRARY_PATH
  ```

This variable (shown for HP) should equal the following:
<install_dir>/DC-15_0/lib
<install_dir>/DC-15_0/connectivity/OCS-15_0/lib

Creating a new access service

You can create and configure an access service by using a text editor to edit the service library configuration file, which resides in the *cfg* subdirectory under the *ServerName* directory.

❖ To configure the access service using a text editor

1. Navigate to the *<install_dir>/DC-15_0/servers/<server name>/cfg* subdirectory of the *server name* directory.
2. At the end of the configuration file, enter a name for the access service in brackets, for example, for Microsoft SQL Server:

   ```
   [mss_acs]
   ```

3. Enter the required parameters and their values from your worksheet, as well as other values for parameters you might need.

For more information about access service configuration properties, see the Enterprise Connect Data Access Options for ODBC *Users Guide for Access Services*.

Adding a new service to the interfaces file

To allow an Open Client application, such as isql, to connect to the Open Client installation that will be used by your client application. Add the new service to the *interfaces* file as a Server Object using the dsedit utility, specifying the same connectivity information for this Server Object as specified in the *interfaces* file for the DirectConnect server.

To preserve a unique directory structure and environment for Sybase applications, each Sybase product must have a unique user or start-up script for each application. The key file for maintaining this uniqueness is the Open Server network address file. On UNIX-based systems, this is the *interfaces* file, which resides in the <*install_dir*> directory.

The network address file configures the Open Server application listening network address. On UNIX systems, it configures the network type and TCP/IP values, plus the operating system type and network.

❖ To modify the *interfaces* file, using the *dsedit* utility

1. Right-click the server object.
2. From the menu, click Add to display the Input Server Name window.
3. Enter the server object and modify its attributes.
4. On the Network Address window that appears, enter protocol information for your site.
5. Enter the machine name on which you installed the DirectConnect server and the connectivity parameter.
6. Click OK twice and exit dsedit.

Defaults are provided for the server configuration parameters. For more information, see the Enterprise Connect Data Access and Mainframe Connect *Server Administration Guide*.

Starting an access service

The only way you can start the new access service without stopping and restarting ECDA is by using DirectConnect Manager.

To start your new service *without* using DirectConnect Manager, be sure that the EnableAtStartup=Yes property is set in the service configuration (*.cfg*) file, which starts the service automatically when the server is started, and then restarts the server.

Verifying an access service

Use one of the SQL utilities to verify that the access service is connecting to its target data source and working properly.

Using isql from a command line

Note If you are using isql on the same machine on which you installed ECDA, you must add an access service entry to the *interfaces* file that points to the access service you are testing.

❖ To verify the access service configuration using *isql*

1. On the client machine, use dsedit to create an *interfaces* file entry for the access service.

   Be sure to enter the access service name exactly as you defined it in the configuration file or by using DirectConnect Manager.

2. Set the environment variables from a command line prompt by running one of the following environment scripts, as appropriate for your platform:

   - *DC_SYBASE.csh*
   - *DC_SYBASE.sh*

3. Run isql from the command line by entering the following:

   ```
   isql -SServiceName -Uuserid -Ppassword
   ```

   where:

   - *ServiceName* is the name of the access service exactly as you defined it in the *interfaces* file.
   - *userid* is a valid user ID for the target database.
   - *password* is a valid password for the user ID in the target database.

If the connection is successful, a 1> prompt appears.

Note If you cannot connect, the access service might not be running. Start the access service with DirectConnect Manager or edit the configuration file, and then verify that the Enable at Startup parameter is set to Yes. This starts the access service automatically when the server is started.

---

4 At the 1> prompt, query a table in the target database by entering a select statement and pressing Return.

5 At the 2> prompt, enter:

```
go
```

Press Return.

The query should run and return a result set, followed by a >1 prompt.

6 To exit isql, enter this at the 1> prompt:

```
exit
```

Press Return.

A regular operating system command appears.

Note At this point, you can stop the DirectConnect server using the stopsvr utility, which shuts down the server and terminates all client connections. However, if you plan to perform post-installation tasks for the client and server, leave both the DirectConnect server and access service running.

Stopping an access service

You can stop the access service only through DirectConnect Manager. The procedure is described in Chapter 6, “Managing DirectConnect Servers and Services Using DirectConnect Manager.”

Stopping a DirectConnect server

At this point, you can stop the DirectConnect server using the stopsrvr utility, which shuts down the server and ends all client connections.

Set the environment variables from a command line prompt by running one of the following environment scripts, as appropriate for your platform:

- *DC_SYBASE.csh*
- *DC_SYBASE.sh*

The stopsrvr format is:

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

where:

- -v displays the program version only.
- -? or -h displays the Help screen.
- -S defines the name of the server to be shut down.
- -d is the delay, in seconds, before client connections are terminated. The default is 3.

Managing ECDA Option for Oracle

When ECDA Option for Oracle is created for the first time, a configuration file is automatically created for it.

You can change the configuration of the ECDA Option for Oracle by modifying the configuration file using the command line. If you need to display or edit the configuration options after installation, refer to the command called sp_configure in Chapter 2, “Configuring and Operating ECDA Option for Oracle,” in the ECDA Option for Oracle *Administration and Users Guide*.

Using the DCOConfig utility

To configure and start the server, you can use the *DCOConfig* script, which is located at *<install_dir>/DCO-15_0/install/DCOConfig*. Before starting the configuration, you must have the following available:

- A copy of the existing *tnsnames.ora* file currently being used to connect to Oracle. You will copy this file to a temporary file location.
- The Oracle connect string.
- The name for a valid Oracle account, which is the administrator for ECDA Option for Oracle.
- The number of an unused port to be used by ECDA Option for Oracle.

❖ **To invoke the configuration utility**

1. Change directories to *<install_dir>/DCO-15_0/install*.
2. Execute DCOConfig.
3. Enter the server name for ECDA Option for Oracle.
4. Enter an unused port on the machine that ECDA Option for Oracle will be listening on.
5. Enter a valid ECDA Option for Oracle Administrator Name.
6. Enter the Oracle connection string as previously defined in the *tnsnames.ora* file.
7. Enter the path (including the file name) to the previously defined temporary location of the *tnsnames.ora* file.
8. ECDA Option for Oracle starts in a separate window.

Note In the window where the ECDA Option for Oracle is started, the following informational message appears. It appears when the ECDA Option for Oracle is started from DCOConfig and can be ignored.

```
Option [traceflags] not found in configuration file
'<path to configuration file>'. Generating new
configuration file.
```

Adding and removing a service

The ECDA Option for Oracle utility to add a service is sp_addservice; the utility to remove a service is sp_dropservice. The commands and syntax are described in Chapter 2, “Configuring and Operating ECDA Option for Oracle,” in the Enterprise Connect Data Access Option for Oracle *Administration and Users Guide*.

Shutting down ECDA Option for Oracle

The utility to shut down ECDA Option for Oracle is sp_shutdown. The command and syntax is described in Chapter 2, “Configuring and Operating ECDA Option for Oracle’ in the Enterprise Connect Data Access Option for Oracle *Administration and Users Guide*.

CHAPTER 6

Managing DirectConnect Servers and Services Using DirectConnect Manager

DirectConnect Manager overview

DirectConnect Manager allows you to:

- Create, start, and stop a server
- Create, start, and stop an access service

Before you can use DirectConnect Manager, you must identify and establish a connection between the server and DirectConnect Manager.

Creating, starting, and stopping a new server

Note To create, start, and stop a *new* server, you must first have a DirectConnect Director (DCDirector) server running in the environment of the server you want to administer.

After you have a DirectConnect Director server up and running, refer to the following sections for how to create, start, and stop a new server.

Creating a new server

For instructions on how to use DirectConnect Manager to create a server, go to the DirectConnect Manager online Help and select Server Administration | Creating New Servers.

Starting a server

The only way you can start the server without stopping and restarting ECDA is by using DirectConnect Manager.

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

Stopping a server

For instructions on using DirectConnect Manager to stop a server, go to the DirectConnect Manager online Help and select Server Administration | Stopping a Server.

Creating, starting, and stopping a new access service

This section describes how to create, start, and stop an access service.

Creating a new access service

For instructions on how to use DirectConnect Manager to create a service, go to the DirectConnect Manager online Help and select Managing Access Services | Creating a New Service or Copying a Service.

Note If you use service name redirection, the connectivity parameters must match the connectivity parameters or the default, and the access service name must map through the redirection file to the DirectConnect server.

For more information about service name redirection, see Enterprise Connect Data Access and Mainframe Connect *System Administration Guide*.

Starting an access service

The only way you can start the new access service without stopping and restarting ECDA is by using DirectConnect Manager. For instructions on how to use DirectConnect Manager to start a service, go to the DirectConnect Manager online Help and select Managing Access Services | Starting a Service.

Stopping an access service

For instructions on using DirectConnect Manager to stop a service, go to the DirectConnect Manager online Help and select Managing Access Services | Stopping a Service.

CHAPTER 7

Managing Servers Using ECDA Option for ODBC Utilities

ECDA Option for ODBC utilities overview

To simplify the execution of ECDA, Sybase provides utilities to start a server, create and start a new server, and migrate files from one server to another.

These utilities are *cshell* scripts that can be found in the *<install_dir>/DC-15_0/bin* directory, where they should be kept to run properly. It is from this directory that the utilities can derive the paths to the other files they need to perform their tasks.

Note These utilities cannot be used for ECDA Option for Oracle.

Creating and starting a DCDirector server

ECDA 15.0 provides a feature that allows DirectConnect Manager to connect to a “directing” server, called a “DCDirector,” that is capable of creating, starting, and stopping DirectConnect servers. To create a new DCDirector for an installation, use the *DCDirector* script.

This script creates a new server with the default name "DCDirector" and sets the port number it listens on to a default port of 7711. The script then starts the server. No parameters are required or allowed, just enter:

```
DCDirector
```

You may change the default server name and default port number. If you need to use different values to identify the directors, you can use a text editor to modify the DCDirector script by assigning different values.

DCDirector utility

This utility creates a default DCDirector server in the installation area. This script does not accept any parameters and uses the server name "DCDirector," and the port 7711 by default. You can change these values by using a text editor, if you prefer to use different values.

Usage DCDirector

Example

```
DCDirector
```

When created, DCDirector will be able to start, stop, and provide other server functions.

Creating a new DirectConnect server

The AddServer utility creates the necessary entries in the *interfaces* file before starting the DirectConnect server. AddServer requires two parameters to identify the name of the new server and to provide the port number for the server to listen on. One important limitation of AddServer: It does not check the *interfaces* file for duplicate server names or ports in the same way that DCDirector does.

AddServer utility

This utility provides a way to create a server entry in the *interfaces* file. It makes changes to the *interfaces* file, but it does not verify that the *servername* or the *port number* are already being used.

Usage AddServer [*servername*] [*port number*]

Example

```
AddServer srvname 1133
```

Starting a DirectConnect server

In the past, you started a DirectConnect server using a command that used the direct executable itself. For example, the command, direct –Ssrvname, was the standard way to start an existing DirectConnect server. This method is still supported, but it has some limitations: For this command to work, all of the appropriate environment variables need to be set properly before the command is executed. Also, if multiple installations of ECDA exist on a single machine, each installation will need its own environment.

To ease the use of this process, Sybase provides a shell script with ECDA version 15.0 that sets the installation-specific variables *before* executing the direct –Ssrvname command. This script, called DCStart, requires that all the non-Sybase variables be set properly and ensures that the environment variables that are specific to a ECDA installation are all correctly set.

DCStart utility

This utility is similar to using the direct executable. DCStart will automatically “source” the appropriate *DC_SYBASE.csh* file to ensure that all the appropriate Sybase-specific environment variables are set properly. This is the required method to start a DirectConnect server, for one or multiple versions, since ECDA no longer sets the system environment.

Usage

DCStart [direct executable *Parameters*]

Syntax

```
DCStart -Ssrvname
```

CHAPTER 8 Configuring ECDA

Note Be sure that both the DirectConnect server and the access services are running.

Configuring client connectivity to ECDA

Note If you verified the ECDA access service in the previous section using isql, you have already created an entry in the *interfaces* file on that machine.

The following procedure ensures that the client environment is properly connected to the LAN and to the DirectConnect server.

❖ To add, edit, or delete entries in the *interfaces* and *libtcl.cfg* files on client machines

1 Using dsedit, verify that the ECDA service name and port number match the entry for the access service.

Note Be sure to use the access *service* name, not the DirectConnect server name.

If you use service name redirection for the server name, refer to the Enterprise Connect Data Access and Mainframe Connect *Server Administration Guide*.

2 Configure the *libtcl.cfg* configuration file that contains information for each installed Net-Library driver.

Use dsedit to view a list of the drivers installed on your machine. You can view a description of any driver by selecting it from the list.

Client applications use the information in this file, along with information in the *interfaces* file, to connect to a DirectConnect server using the correct file.

Configuring servers and access services

After you create a DirectConnect server or a DCDirector server, you can configure the server through DirectConnect Manager. For a description of all of the configuration properties, refer to the following:

- For the DirectConnect server:
  - For the ECDA Option for ODBC, refer to the Enterprise Connect Data Access and Mainframe Connect *Server Administration Guide*
  - For the ECDA Option for Oracle, refer to the “Using the DCOConfig utility” section in Chapter 5, “Managing DirectConnect Servers and Services Using ECDA Utilities,” and the Enterprise Connect Data Access Option for Oracle *Server Administration and Users Guide*
- For the access service:
  - For the ECDA Option for ODBC, refer to the ECDA *Users Guide for Access Services*.
  - For the ECDA Option for Oracle, refer to the ECDA Option for Oracle *Server Administration and Users Guide*

CHAPTER 9 Setting Up the ODBC Back-end Driver and Driver Manager

Note This chapter does not apply to ECDA Option for Oracle.

ECDA support for back-end drivers

Because Sybase no longer provides ODBC drivers for back-end access to the target database, you need to obtain your own driver from your database vendor. However, Sybase does certify with and support these back-end drivers:

- IBM DB2 CLI ODBC driver
- DataDirect Driver for DB2 and Microsoft SQL Server

ODBC Driver Manager architecture

ECDA Option for ODBC architecture depends on an ODBC driver to provide back-end (server-side) connectivity to the target database and an ODBC driver manager.

Starting with version 15.0, the ECDA Option for ODBC includes a back-end, open-source unixODBC driver manager.

Note The front-end (client-side) ODBC driver that was used to set up connectivity between ECDA and the client is no longer available with version 15.0.

When to use the unixODBC driver manager

Use the unixODBC driver manager for one of the following conditions:

- ECDA Option for ODBC configuration must load a variety of ODBC drivers connecting to many different targets.
- Express Transfer requires an ODBC driver manager to manage the ODBC drivers.

Using the unixODBC driver manager

If you choose to use the unixODBC driver manager supplied by Sybase, you must verify that the driver selected is compatible with the unixODBC driver manager.

ECDA Option for ODBC version 15.0 ships with an open-source ODBC driver manager located in the *<install_dir>/DC-15_0/drv_mgr/unixODBC-2.2.12* directory. To use the unixODBC driver manager, ECDA provides the folloiwng two options.

❖ To use the unixODBC driver manager - Option 1

- In the *dcany.cfg* file, under the [Service Library] stanza, add this configuration value:

  ```
  ODBCDriverManager=<install_dir>/DC-15_0/drv_mgr
  /unixODBC-2.2.12/libodbc.yy.1.x.x
  ```

  where:

  - *yy* represents the library extension *so* or *sl.*
  - *x.x* represents the numeric string appended to the library name.

❖ **To use the unixODBC driver manager - Option 2**

- Move the library found in the *<install_dir>/DC-15_0/drv_mgr /unixODBC-2.2.12* directory found in the ECDA runtime library path.

  In the *dcany.cfg* file, under the [Service Library] stanza, add this configuration value:

  ```
  ODBCDriverManager=libodbc.yy.1.x.x
  ```

  where:

  - *yy* represents the library extension *so* or *sl*.
  - *x.x* represents the numeric string appended to the library name.
  - *<install_dir>/DC-15_0* must be expanded to the complete path name from the root directory. ECDA will not expand the variable.

  Note This driver manager uses the $ODBCSYSINI environment variable to locate the *odbc.ini* file.

ODBC Driver Manager configuration properties

Starting with version 15.0, the ECDA Option for ODBC provides the capability to specify the ODBC driver manager at runtime. The ODBC driver manager that is loaded at runtime is determined by the file defined in the Service Library configuration property called ODBCDriverManager. The library extension can be *.sl* (for HP) or *.so* (for Solaris, AIX, Linux, and HP Itanium). The default for this property is the Sybase-provided unixODBC driver manager.

In addition, when using the unixODBC Driver Manager with the IBM ODBC Driver, you must set the SQLODBCCursors property to driver. If set to the default, an error message appears:

```
Cannot load library libodbccr
```

For more information, see the Enterprise Connect Data Access Option for ODBC *Users Guide for Access Services*.

Note If you purchase a driver manager other than the one that Sybase provides, you must follow the installation and configuration vendor documentation.

Environment variables

Library path environment variables

ECDA relies on the library path environment variable to locate necessary libraries, including the *libodbc* library and its library dependencies. During the build process, libraries become linked with other libraries that provide functions that the loading library uses, thus creating a runtime dependency; in other words, one library depends on the presence of another. As a result, the runtime loader must and load these other libraries to satisfy runtime dependencies. If it does not, the initial library load fails.

The library path environment variables are:

- *Linux, Solaris, HP Itanium*: LD_LIBRARY_PATH
- *AIX*: LIBPATH
- *HP UX*: SHLIB_PATH

When it uses the default ODBCDriverManager value, *libodbc.xx*, ECDA searches the library path and loads the first library found with the name matching the configuration property string, *libodbc.xx*. If the *ODBCDriverManager* library does not find its dependent libraries in its library path, the driver load fails and the ECDA start-up fails.

ODBC Driver Manager environment variables

To find and load drivers, the OCBC Driver Manager uses configuration files, which contain information about target databases and driver location. All of the configuration files have an *.ini* extension (for example, *odbc.ini*) that can be identified by name and extension.

The environment variable identifier for the unixODBC driver manager is ODBCSYSINI, which points to the directory containing the *odbc.ini* and *odncinstl.ini* files, for example:

```
setenv ODBCSYSINI pathname
```

where *pathname* points to the *odbc.ini* file.

Configuration files that the ODBC Driver Manager uses have target database-specific information:

- *odbc.ini* contains the data source name (DSN) information, which includes connection information, driver identifiers, and other driver-specific attributes.
- *odbcinst.ini* contains driver-specific information.

CHAPTER 10 Troubleshooting Installation

Troubleshooting overview

This chapter tells you how to identify and resolve problems that occur when you install an ECDA product. Also, it describes several diagnostic tools that you can use.

If you try the suggested methods described in this chapter and still have problems, notify your company’s contact person for Sybase. Each Sybase installation that has purchased a support contract has one or more designated people who are authorized to contact Sybase Technical Support or the Sybase subsidiary in your area.

In general, there are several sources from which error messages can be generated, including:

- ECDA
- Open Client and Open Server
- Network
- Target database
- ODBC drivers (not supplied by ECDA)

ODBC-specific problems

This section describes the following types of problems that may occur:

- ODBC driver errors
- ODBC data source errors
- ODBC driver manager errors

ODBC driver errors

This is an example of error reported on an ODBC driver supplied by DataDirect:

[vendor] [ODBC_component] message

where:

- *vendor* is the name of the ODBC vendor.
- *ODBC_component* is the component in which the error occurred.
- *message* is the content of the error message.

For example, an error message from the DataDirect SQL Server driver looks like this:

```
[DataDirect] [ODBC SQL Server driver] Invalid
precision specified.
```

If you receive this type of error, check the last ODBC call made by your application for possible problems, or contact your ODBC application vendor.

ODBC data source errors

This is an example of the type of error that occurs in the data source as defined in DataDirect and includes the data store name:

[vendor] [ODBC_component] [data_store] message

where:

- *vendor* is the name of the ODBC vendor.
- *ODBC_component* is the component that received the message from the data store indicated.

- *data_store* is the name of the location in the data source where the error occurred.
- *message* is the content of the error message.

For example, you might receive the following message from an Oracle data store:

```
[DataDirect] [ODBC Oracle driver] [Oracle] ORA-0919:
specified length too long for CHAR column.
```

If you receive this type of error, check your database system documentation for more information, or consult your database administrator. In this example, you need to check your Oracle documentation.

ODBC driver manager errors

The driver manager is a DLL that establishes connections with drivers, submits requests to drivers, and returns results to applications. An error that occurs in the driver manager has this format:

[*vendor*] [*ODBC XXX*] *message*

where:

- *vendor* is the name of the ODBC vendor.
- *ODBC XXX* is the driver manager and its version number.
- *message* is the content of the error message.

For example, an error from the Microsoft driver manager might look like this:

```
[Microsoft] [ODBC Driver Manager] Driver does not
support this function.
```

If you receive this type of error, consult the *Programmers Reference* for the Microsoft ODBC Software Development Kit, available from Microsoft.

ECDA problems

If your system does not work properly after you install ECDA, and if you already performed the connection steps in Chapter 2, "Preparing to Install," try the following:

- Confirm the release number of Open Server software. ECDA is compatible with the Open Server release and Adaptive Server release identified in the release bulletin.
- Check the log and trace files in the *log* file of the *ServerName* for more information. This is an example of an error message that can occur:

  ```
  Error: 16029 Severity: 20 State: 0 OS Error: -1:
  Failed to start any network listeners OS Error Text:
  <srv-lib>
  ```

  One of these causes is possible:

  - One of the specified port numbers is in use. Change the port number to one that is not in use and try again.
  - The wrong machine name or IP address was specified. Enter the machine name or the IP address running the DirectConnect server.

  If the server fails before the log files initialize, error messages can be written to either the console or the Windows event log. If this occurs, see the Enterprise Connect Data Access and Mainframe Connect *Server Administration Guide* for explanations of these "pre-log" messages.

Connectivity problems

This section describes some tools you can use to diagnose connectivity problems.

Using the odbct tool for ODBC problems

To help you diagnose ODBC connectivity problems, the odbct tool verifies your system configuration by attempting to make a simple connection to the target.

Note Be sure to run the odbct tool in the same environment in which you plan to use ECDA targets by executing *DC_SYBASE.csh*.

The odbct tool has three parameters: a data source name (DSN), user ID, and password. This example shows a connection to the *dcmssql* DSN using the user ID ”joe” and password “joe21”.

```
odbct <DSN> <USERID> <PASSWORD>
odbct dcmssql joe joe21
Would you also like the output saved to a file?(Y/N):
y
Attempting Connection--------------------
> Allocating Environment
> Allocating Connection
> Attempting Connection
    DSN: dcmssql
    USER: joe
    PASSWORD: joe21
> CONNECTION SUCCEEDED
> Allocating Statement
> Connection succeeded, would you like to proceed
with datasource and datatype reporting? (Y/N):
n
Connection Cleanup-----------------------
> Free Statement
> Attempting Disconnection
+ Disconnection succeeded
> Free Connection
> Free Environment
belford [ 23 ]
```

UNIX configuration problems

This section describes the ivtestlib tool (not supplied with ECDA), which you can use to find and diagnose configuration problems in the UNIX environment. This command attempts to load a specific ODBC driver and prints out all error information if the load fails.

For example, if a driver is installed on HP-UX in the *<DataDirect installation directory>/lib*, this command attempts to load the driver:

```
ivtestlib /<DataDirect installation directory>
/lib/lib/MV<driver library>20.sl
```

If the driver cannot be loaded, ivtestlib returns an error message with an explanation.

Note Unlike HP-UX, the Solaris and AIX versions of ivtestlib do not require you to specify a full path.

APPENDIX A Connectivity Tasks for AS/400

The ECDA product that accesses DB2 UDB for AS/400 is ECDA for ODBC, using TCP/IP for connectivity. To set up the AS/400 to communicate with these ECDA products, you must perform the connectivity tasks described in this appendix after logging in to the target AS/400 server as the System Administrator.

Note SNA/APPC connectivity is not supported.

Enable DRDA capability of AS/400

To use an AS/400 service, you must enable the Distributed Relational Database Architecture (DRDA) capability of the AS/400. The service communicates with the AS/400 through a TCP/IP connection to DB2 UDB.

To confirm that your AS/400 has DRDA capability:

- Verify that you are using OS/400 V5R2 or higher.
- Be sure that you define the local relational database using the add relational database directory entry command, ADDRDBDIRE.

Enter this name on the worksheet for the appropriate platform as the Database Name. The Database Name is requested during installation and is the DatabaseName property in the service library configuration file called *dcany.cfg*, which is located in the *ServerName\cfg* subdirectory.

Set security levels

The AS/400 has four security levels, each of which is described in Table A-1.

Note Sybase recommends that you use AS/400 security level 20 or higher.

Table A-1: AS/400 security levels

Level	Description
10	Requires only a user ID. If you enter a user ID that does not exist, AS/400 automatically creates it. You can access all system resources.
20	Requires a user ID and password. The user ID must already exist in AS/400. When you sign in to AS/400, you have access to all system resources.
30	Requires a user ID and password and includes further measures to secure objects on the system. By default, objects are secured by a user’s class (*SECOFR, *SECADM, *PGMR, *SYSOPR, AND *USER).
40	Requires a user ID and password, and includes further measures to secure objects on the system. By default, objects are secured by a user’s class (*SECOFR, *SECADM, *PGMR, *SYSOPR, AND *USER). Unsupported interfaces cannot access the system.

The access service enforces security only through level 20. It returns an error only if the user ID and password do not exist on the AS/400.

Error messages relating to higher security levels on the AS/400 are returned from the AS/400. If your AS/400 security is set at a level higher than 20 and you experience problems with the access service, please have the designated person at your site contact Sybase Technical Support or the Sybase subsidiary in your area.

❖ To check the AS/400 security settings

1 Log in to the AS/400 from the main console, a 5250 terminal, or a 5250 terminal emulator.

2 At the Main menu, enter:

```
WRKSYSVAL QSECURITY
```

The Work With System Value window appears.

3 Enter the following to display your security settings:

```
DSPSYSVAL QSECURITY
```

❖ To change the security settings

1 Be sure you have System Security Officer (QSECOFR) authority.

2 From the Work with System Values menu, enter:

```
CHGSYSVAL QSECURITY
```

3 Enter the value.

4 Restart the AS/400 to make the change effective.

Change the CCSID

The CCSID (coded character set ID) designates the binary code page in which the AS/400 returns data. The default is 65535. If you do not change the CCSID, the access service returns character data from the AS/400 in binary form. When mapped to ASCII format, this data appears as hexadecimal characters.

Changing the CCSID does not affect how the AS/400 stores data. AS/400 programs set the CCSID to a value appropriate to its function. The ECDA access service does not require a specific CCSID. For English installations, use code page 37 or 500.

Changing the CCSID at the User Profile level

To change the CCSID at the user profile level, your user ID must have QSECOFR authority.

❖ **To change the CCSID at the user profile level**

1 Log in to the AS/400 from the main console, a 5250 terminal, or a 5250 emulator.

2 To display the Work with User Profile window, enter:

```
WRKUSRPRF USER_ID
```

where *USER_ID* is a valid user ID for the AS/400 with QSECOFR authority.

3 Select option 2.

4 Press F10 to view additional parameters. Page down until you find the CCSID setting.

5 Change the CCSID and press Enter.

Any physical file that is created with this user profile has the new CCSID assigned to the file character fields.

Create the Sybase collection on AS/400 (for DB2 UDB)

The access service requires certain files on the AS/400. ECDA for ODBC access service looks for these files in a collection called “SYBASE.” You must create this collection and name it “SYBASE” before installing the access service.

Note The owner of the SYBASE collection must have QSECOFR privileges.

The SYBASE collection contains a package for the DC DB2 UDB driver that is created when the drivers are installed. After you create the packages, you need to grant permissions to PUBLIC for the end users to access the Sybase collection located on the target AS/400. This can be done with isql (which comes with the ECDA installation), or entered directly on the AS/400 machine.

Note The ECDA service used to issue the GRANT EXECUTE command must be in sybase mode for SQL transformation. (SQL transformation is an access service property with one of two settings, passthrough or sybase.) To ensure this, enter the following command in the isql session before you issue the GRANT EXECUTE command:

```
set sqltransformation sybase
```

❖ **To grant permissions from isql**

- Issue this command:

```
GRANT EXECUTE ON PACKAGE SYBASE.DEF000x TO PUBLIC
```

Because the AS/400 allows you to change security levels and ownership of objects, you can use another user ID to create the SYBASE collection. For example, you can use one ID to create the SYBASE collection, then grant other users create and execute authorization for the collection and its objects.

You can create the SYBASE collection one of two ways:

- If you have DB2 (or DB2/400) Query Manager and SQL Development Kit (SDK) installed on the AS/400, you can run the SQL utility using a 5250 session.
- If not, you need to use the AS/400 SEU utility program to create the SYBASE collection.

❖ To grant permissions directly on the AS/400

1. Log in to the AS/400 from the main console, a 5250 terminal, or a 5250 terminal emulator, using the user ID that will be the owner of the SYBASE collection.
2. To display the Sybase packages, on the command line, enter:

   ```
   WRKOBJOWN SYBASE
   ```

3. On the Work with Object by Owner window, locate and select a package from the attribute column and in the corresponding OPT column, enter:

   ```
   2
   ```

4. On the Edit Object Authority window, in the User column, locate *PUBLIC, and in the corresponding Object Authority column, enter:

   ```
   ALL
   ```

5. Press F12 to return to the Work with Object by Owner window, and repeat steps 2 and 3 until all packages are completed. To exit, press F3 repeatedly, followed by *90*.

Creating the SYBASE collection using the AS/400 SEU utility

The user ID for creating the SYBASE collection must have a valid CCSID for your language installation.

❖ To create the SYBASE collection through the AS/400 SEU utility

1. Log in to the AS/400 from the main console, a 5250 terminal, or a 5250 terminal emulator, using the user ID that is to be the owner of the SYBASE collection.
2. Start the SEU program by entering:

   ```
   STRSEU
   ```

3. Press F4. The Work With Members Using SEU window appears.
4. Provide the following information:

- In the Source File field, enter:

  ```
  QQMQRYSRC
  ```

- In the Source Member field, enter:

  ```
  *SELECT
  ```

- In the Library File field, enter:

  ```
  QGPL
  ```

  If the source file does not exist, you must create it using the CRTSRCPF command.

5 Press Enter. The members of the source file are listed.

6 Build the query that creates the collection by creating a source member for the query:

- In the New Member field, enter:

  ```
  SYBASE
  ```

- In the Type for New Member field, enter:

  ```
  TXT
  ```

7 Press Enter. The Edit window appears.

8 Create the query member by entering the following in the first line of the source file:

```
CREATE COLLECTION SYBASE
```

9 Press F3. Verify that the Change/Create Member field is set to Y, and the Member field is SYBASE.

10 Press Enter to exit the editor. Now, you can work with the query you created.

11 Press F3 again to exit and return to the Main menu.

❖ **To build the query that creates the SYBASE collection**

1 On the command line, enter:

```
CRTQMQRY QMQRY(QGPL/SYBASE) SRCFILE(QGPL/QQMQRYSRC)
```

Press Enter. The AS/400 utility creates the query.

Note You might receive a warning message that the source length exceeds 79 characters. Disregard this message, as it does not affect the access service.

2 To run the query that creates the SYBASE collection, enter the following on the command line:

```
STRQMQRY QMQRY(QGPL/SYBASE)
```

Press Enter. The SYBASE collection is created.

Building the collection might take a few minutes. During this time, you will receive messages indicating the status of the operation.

Creating the SYBASE collection using SQL/400

The user ID for creating the SYBASE collection must have a valid CCSID for your language installation.

❖ To create the SYBASE collection through SQL/400

1 Log in to the AS/400 from the main console, a 5250 terminal, or a 5250 terminal emulator, using the ID that you want to own the collection (usually SYBASE).

2 On the command line, enter:

```
STRSQL
```

The STRSQL program begins.

3 On the SQL command line, enter:

```
CREATE COLLECTION SYBASE
```

The collection process requires a few minutes to complete. You can create other collections by using this same syntax and substituting the name of the collection for "SYBASE."

Authorizing users for the SYBASE collection

You must have the following minimum authorizations to access the AS/400 through the access service:

- USE authorization to the service package in the SYBASE collection
- CHANGE authorization to:
    - Journal object QSQJRN
    - Journal receivers QSQJRN0001, QSQJRN002, and any other additional journal receivers

❖ **To confirm that authorizations are set properly**

1. Verify that the *SQLPKG objects in SYBASE are set to *PUBLIC, *USE, or higher authorization level.
2. Verify that the journal objects are set to *PUBLIC, *CHANGE, or higher authorization level.

APPENDIX B Installation Worksheets

How to use the worksheets

Fill out one of these worksheets for each DirectConnect server that you plan to install. Keep it with you for each step of the installation process.

The worksheets cover:

- Information from Sybase and non-Sybase products that must be installed before and after installing ECDA.
- Information that is requested during installation, connectivity setup, and configuration of ECDA access service configuration files.

The worksheets present the list of parameters you will be prompted for during installation, give you a place for you to write down the parameter values to use, and provide a short description of the parameter.

ECDA Option for ODBC worksheet

Use the following worksheet to identify information that you should designate and record prior to installing ECDA Option for ODBC. You are prompted for this information during installation.

Table B-1: ECDA Option for ODBC worksheet

Your installation information	Description
DSN(s):	*DSN:* ODBC data source name.
CONNECTIVITY PARAMETERS:	*Connectivity*: Record the parameters for the connection protocol at your site. For example, if TCP/IP is the protocol, you record the IP address and port number. Check with your administrator for these parameter values

ECDA Option for Oracle worksheet

Use the following worksheet to identify information that you should designate and record prior to installing ECDA Option for Oracle. You are prompted for this information during installation.

Table B-2: ECDA Option for Oracle worksheet

Your installation information	Description
SERVER NAME:	*DirectConnect Server Name:* Name of the DirectConnect server you want to create or update.
PORT NUMBER:	*ECDA Port Number:* Name of the ECDA port number you are using.
DCO ADMIN ACCOUNT NAME:	*DCO Admin Account Name:* The name of a valid Oracle account that can be used to administer ECDA Option for Oracle. It is usually the system account name.
ORACLE CONNECT STRING:	*Oracle connect string:* The entry for the Oracle instance of the *tnsnames.ora* file.
ENTER LOCATION OF TNSNAMES.ORA FILE:	*Location of the tnsnames.ora file:* Full path (including file name) of the existing *tnsnames.ora* file, usually located in the network *admin* directory under the *ORACLE_HOME* directory. The *tnsnames.ora* identified must be accessible to this machine. If the file is not available, then it must be copied on to a local drive before configuring ECDA Option for Oracle. ECDA copies the existing *tnsnames.ora* into its directory.

APPENDIX C

Configuring Data Sources to Connect to Targets Using DataDirect

This appendix describes how to create and configure the data source name (DSN) for each ECDA component for ECDA Option for ODBC using DataDirect drivers to connect to the following targets:

- DB2 UDB
- Microsoft SQL Server or an ODBC-accessible target

Note The DataDirect driver is no longer provided by Sybase, but the information in this appendix can help you connect to the various targets.

Use the attributes listed in the section for each ECDA product and a text editor to define the data source entries. The *odbc.ini* file is installed in the *<install_dir>/DC-15_0* directory.

Note The DSN templates provided by the database vendor contains additional parameters that you should remove.

Connecting to DB2 UDB

In the UNIX environment, the ODBC driver allows you to establish a centralized ODBC system information file that you can control. To do so, set the environment variable ODBCINI by executing the ECDA environment script to point to the qualified path name of the centralized file, provided by the ECDA installation. For example:

For C shell:

```
DC_SYBASE.csh
```

For Bourne or Korn shell:

```
DC_SYBASE.sh
```

Executing this script sets ODBCINI to *<install_dir>/DC-15_0/connectivity/odbcini*.

Remember, there must be an [ODBC] section in the system information file that includes the InstallDir keyword. Be sure that the keyword value is in the path to the directory *<install_dir>/DC-15_0/connectivity*.

To define data source entries

To create and configure data source entries for DRDA, edit the *odbc.ini* file using the attributes in each of these categories, or sections:

- *General*, which defines connectivity parameters
- *Advanced*, which contains optional fields that affect performance and use of resources
- *Bind*, which defines parameters for package creation in the DB2 system

Each of these sections is delimited with comments.

As you edit the *odbc.ini* file, be sure you have the following parameters available on your worksheet. For all others, you can accept the default or provide a value appropriate for your site.

General parameters

Table C-1 contains the General parameters; the required parameters are indicated by an asterisk (*).

Table C-1: General parameters for DB2 UDB

Parameter	Comments
**Driver*	The fully qualified path to the driver.
**Ip address*	The IP (Internet Protocol) address of the machine where the catalog tables are stored. Specify the address using the machine's numeric address (for example, 123.456.78.90) or its host name. **Warning!** For Solaris systems, use the numeric IP address, not the host name. If you enter a host name, the driver must find this name (with the correct address assignment) in the *HOSTS* file on the workstation or in a DSN server.
**TcpPort*	The port number that is assigned to the DB2 server on the machine where the catalog tables are stored. Specify either this port's numeric address or its service name. The default numeric port address varies, depending on the OS of the DB2 server machine. For DB2, the default may be 50000. For AS/400, the default is 446. If you specify a service name, the driver must find this name (with the correct port assignment) in the *SERVICES* file on the workstation.
**Location* (Required for DB2 UDB on AS/400 and z/OS only)	This field is valid only if you are connecting to a DB2 database running on z/OS or AS/400. Type the DB2 location name, using the name defined during the local DB2 installation. **Note** This parameter is not applicable if the Database Name field is populated.
**Collection* (Required for DB2 UDB on AS/400 and z/OS only)	A field that is valid only if you are connecting to a DB2 database running on z/OS or AS/400. Enter the name that identifies a group of packages that include the ECDA DC DB2 DRDA driver packages. The default for DB2 and AS/400 is the user ID. **Note** This field is not applicable if the Database Name field is populated.
**Database* (Required for connecting to DB2 on Windows and UNIX)	The name of the database to which you want to connect to, by default. **Note** This parameter is not used for AS/400 and z/OS.
**IANAAppCodePage* (Required for UNIX platforms)	A driver code page configuration property for the DB2 UDB driver, for example, 3, 4, 2004, and 2250. Find a complete list of the values and their descriptions, see IANA at http://www.iana.org/assignments/character-sets.
WorkArounds2	The name of the string key to allow the driver to pad the DB2 Char for Bit Data field with spaces. The value is *131072*.

After entering these parameters, use the odbct tool (see Chapter 10, “Troubleshooting Installation”) to test connectivity to the DB2 UDB source.

Advanced parameters

Table C-2 shows the default values for the Advanced parameters that should be sufficient for most DB2 UDB installations.

Table C-2: Advanced parameters for DB2 UDB

Parameter	Comments
AlternateID	A value that is substituted at connect time for the current schema. This sets the default qualifier for unqualified object names in SQL statements. If the attempt to change the current schema fails, the connection fails with "Invalid value for Alternate ID." DB2 permissions should be set to SYSADM. (Not valid for AS/400 V5R1.)
WithHold	The cursor behavior for the application used with this data source—either DB2 closes all open cursors (Delete cursors) after a commit or rollback, or leaves them open (Preserve cursors). • When this option is *1* (the default), the cursor behavior is Preserve. Sybase recommends this option. • When this option is *0*, the cursor behavior is Delete.
AddStringToCreateTable	Use this parameter if you want to append the in tablespace clause to create table commands.

Defining data source entries

Table C-3 contains the Bind parameters, with the required parameters, indicated by an asterisk (*).

Table C-3: Bind parameters for DB2 UDB

Parameters	Comments
**GrantExecute*	An indicator to grant privileges on the package that you are creating.
**GrantAuthid*	The ID of the person to whom the GrantExecute privileges were assigned. The default value is Grant Execute privileges on the package to PUBLIC.

Parameters	Comments
**IsolationLevel*	The Isolation Level method by which locks are acquired and released by the system. Valid values are: • ALL – prevents any other process from accessing data that your application has read or modified. All read or modified data is locked until the end of the transaction. • CHANGE – allows other processes to read from the database. Only modified data is locked until the end of the transaction • Cursor_Stability (the default) – allows other processes to change a row that your application has read if the cursor is not on the row you want to change. Prevents other processes from changing records that your application has changed until your program commits them or terminates. It prevents your program from reading a modified record that has not been committed by another process. • No_Commit – allows your program to read modified records even if they have not been committed by another person. • Repeatable_Read – prevents other processes from changing records that are read or changed by your application (including phantom records) until your program commits them or terminates. It prevents the application from reading modified records that have not been committed by another process. If your program opens the same query during a single unit of work under this isolation level, the results table will be identical to the previous table; however, it can contain updates made by your program.
**DynamicSections*	The number of statements that the DB2 Wire Protocol driver package can prepare for a single user. The default is 64. This value determines the maximum cursors or dynamic statements that a single connection can have open simultaneously.

Note For version 15.0, packages are created automatically on the initial connection to the DB2 UDB server. The Bind utility is only needed to modify the package settings.

Use the bind20 utility included in the *bin* subdirectory to create packages. The bind20 utility will use the bind options defined in your DRDA DSN to configure the package and grant proper ownership. The bind20 syntax is:

```
bind20 db2-dsn
```

This utility prompts for a DB2 user ID and password, creates the packages, and then reports a final status.

A separate package is not needed for each DRDA access service; however, if an access service has different characteristics in its parameters that suit it for a specific solution, you can create other packages for other solutions.

The examples in the next sections identify the DSN parameters required and used for DRDA to the following targets:

- DB2 UDB on AS/400
- DB2 UDB on Windows
- DB2 UDB on UNIX

Example of an *odbc.ini* file for DB2 UDB on AS/400 or z/OS

This is an example of a DSN to access DB2 UDB for AS/400 or z/OS:

```
;;DB2 UDB AS/400 or z/OS DSN template
;;
[dcdb2udbas400]
Driver=/<ODBC_install_dir>/lib/<driver_library>
IPAddress=
TcpPort=
Location=
Collection=
SecurityMechanism=0
WorkArounds2=131072
;;
;; Advanced options
;;
AddStringToCreateTable=
AlternateID=
CatalogSchema=
IsolationLevel=CURSOR_STABILITY
CharsetFor65535=0
ReportCodePageConversionErrors=0
ApplicationUsingThreads=1
UseCurrentSchema=1
WithHold=1
;;
;; Bind options - for DSNs used with bind20 utility
;;
GrantExecute=1
GrantAuthid=PUBLIC
DynamicSections=64
;;
;; Optional Failover options
;;
LoadBalancing=0
AlternateServers=
ConnectionRetryCount=0
ConnectionRetryDelay=3
```

Example of an *odbc.ini* file for DB2 UDB on Windows or UNIX

This is an example of a DSN to access DB2 UDB on Windows or UNIX:

```
;;UDB Windows or UNIX template
;;
[dcdb2udbwin]
Driver=/<ODBC_install_dir>/lib/<driver_library>
IPAddress=
TcpPort=
Database=
SecurityMechanism=0
WorkArounds2=131072
;;
;; Advanced options
;;
AddStringToCreateTable=
AlternateID=
CatalogSchema=
IsolationLevel=CURSOR_STABILITY
CharsetFor65535=0
ReportCodePageConversionErrors=0
ApplicationUsingThreads=1
UseCurrentSchema=1
WithHold=1
;;
;; Bind options - for DSNs used with bind20 utility
;;
GrantExecute=1
GrantAuthid=PUBLIC
DynamicSections=64
;;
;; Optional Failover options
;;
LoadBalancing=0
AlternateServers=
ConnectionRetryCount=0
ConnectionRetryDelay=3
```

Connecting to Microsoft SQL Server

In the UNIX environment, the ODBC driver allows you to establish a centralized ODBC system information file that you can control. To do so, set the environment variable ODBCINI by executing the ECDA environment script to point to the qualified path name of the centralized file, provided by the ECDA install. For example:

For C shell:

```
DC_SYBASE.csh
```

For Bourne or Korn shell:

```
DC_SYBASE.sh
```

Executing the script sets the ODBCINI environment variable to *<install_dir>/DC-15_0/connectivity/odbcini*.

Remember, there must be an [ODBC] section in the system information file that includes the InstallDir keyword. Be sure that the keyword value is the path to the directory *<install_dir>/DC-15_0/connectivity*.

To define data source entries

To define data source entries, edit the *odbc.ini* file, setting the attributes listed in Table C-4.

As you edit the *odbc.ini* file, be sure you have the necessary information available. For all others, you can accept the default or provide a value appropriate for your site.

Table C-4: Descriptions of data source parameters

Parameter	Description
**Driver*	The full path to the driver.
Description	An optional long description of the data source name.
**QuotedID*	Parameter that allows quoted identifiers, that is, identifiers in the SQL Server that you can quote using a quoting character.
**Database*	Name of the database to which you want to connect by default.
**Address*	Parameter that contains a 4-part IP address and the port number that it is listening on.

Example of an *odbc.ini* file

Following is an example of the DSN template for Microsoft SQL Server. You must edit the section of the file that applies to your site.

```
; MS SQLServer DSN template
;;
[dcmssql]
Driver=/<ODBC_install_dir>/lib/<driver_manager>
QuotedId=Yes
Database=
;; Address: Enter numbers from your install worksheet
;;   In the form: AAA.BBB.CCC.DDD,PPPP
;;   Where AAA.BBB.CCC.DDD is the 4 part IP Address
;;   and PPPP is the port number it is listening on.
Address=
;;
;; Optional Failover options
;;
LoadBalancing=0
AlternateServers=
ConnectionRetryCount=0
ConnectionRetryDelay=3
```

Glossary

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 service library

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

ACSLIB

See **access service library**.

Adaptive Server Enterprise

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

Adaptive Server Enterprise/Component Integration Services

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

administrative service library

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

AIX

See **Advanced Interactive Executive.**

API

See **application program interface**.

APPC

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

application program interface

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

ASCII

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

ASE

See **Adaptive Server Enterprise**.

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

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

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

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

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

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

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**.

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**.

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.

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

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

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

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

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**.

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.

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

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

dll See **dynamic link library**.

DSN See **data set name**.

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

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

ECDA See **Enterprise Connect Data Access**.

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

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

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

end user

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

Enterprise Connect Data Access

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

environment variable

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

exit routine

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

express transfer

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

external call interface

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

External Security Manager

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

gateway

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

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**.

interfaces file

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

internationalization

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

keyword

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

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.

LU 6.2

See **logical unit 6.2**.

Mainframe Connect DB2 UDB Option

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

Mainframe Connect DirectConnect for z/OS Option

A Sybase Open Server application that provides access management for non-Sybase databases, copy management (transfer), and remote systems management.

Mainframe Connect Server Option

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

network protocol

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

nonclustered index

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

null

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

ODBC See **Open Database Connectivity**.

ODS See **Open Data Services**.

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

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

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

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

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

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

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).

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

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

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

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

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

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

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

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

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

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

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

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

SNRF See **service name redirection file**.

SPID See **server process ID**.

SQL See **structured query language**.

SQLDA See **SQL descriptor area**.

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

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

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.

System Administrator

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

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**.

Transact-SQL

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

transfer

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

transfer service library

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

Transmission Control Protocol/Internet Protocol

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

T-SQL

See **Transact-SQL**.

unit of work

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

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

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

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

Index