



Installation Guide

**Enterprise Connect™ Data
Access 15.7**

Linux and UNIX

DOCUMENT ID: DC39555-01-1570-01

LAST REVISED: November 2012

Copyright © 2012 by Sybase, Inc. All rights reserved.

This publication pertains to Sybase software and to any subsequent release until otherwise indicated in new editions or technical notes. Information in this document is subject to change without notice. The software described herein is furnished under a license agreement, and it may be used or copied only in accordance with the terms of that agreement.

Upgrades are provided only at regularly scheduled software release dates. No part of this publication may be reproduced, transmitted, or translated in any form or by any means, electronic, mechanical, manual, optical, or otherwise, without the prior written permission of Sybase, Inc.

Sybase trademarks can be viewed at the Sybase trademarks page at <http://www.sybase.com/detail?id=1011207>. Sybase and the marks listed are trademarks of Sybase, Inc. ® indicates registration in the United States of America.

SAP and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP AG in Germany and in several other countries all over the world.

Java and all Java-based marks are trademarks or registered trademarks of Oracle and/or its affiliates in the U.S. and other countries.

Unicode and the Unicode Logo are registered trademarks of Unicode, Inc.

All other company and product names mentioned may be trademarks of the respective companies with which they are associated.

Use, duplication, or disclosure by the government is subject to the restrictions set forth in subparagraph (c)(1)(ii) of DFARS 52.227-7013 for the DOD and as set forth in FAR 52.227-19(a)-(d) for civilian agencies.

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

Contents

- Conventions1**
- Introduction3**
 - ECDA Option for ODBC3
 - ODBC Driver3
 - ECDA Option for Oracle4
 - DirectConnect Manager4
- Installation Task Flows7**
- Planning Your Installation9**
 - Role and Skill Requirements9
 - Prepare the Environment10
 - Connectivity Tasks10
 - System Requirements10
 - SySAM Licensing11
 - Installation Mode11
 - Performing Critical Administrative Tasks12
- Installing ECDA13**
 - Installing ECDA in GUI Mode13
 - Installing ECDA in Console Mode15
 - Installing ECDA Using a Response File16
 - Creating a Response File16
 - Installing in GUI Mode Using a Response File17
 - Installing in Silent Mode18
- Installing DirectConnect Manager19**
- Upgrading ECDA for Oracle or ECDA for ODBC21**
- Uninstalling ECDA23**
 - Uninstalling in GUI Mode23
 - Uninstalling in Console Mode24
- Create DirectConnect Server25**
 - Create a Server in ECDA Option for ODBC25
 - Using unixODBC Driver Manager25
 - Using DataDirect Driver26

- Using IBM DB2 Driver27
- Creating a Server in ECDA Option for Oracle28
- ECDA Utilities31**
 - Manage ECDA Option for ODBC31
 - Creating a New DirectConnect Server31
 - Creating a New Access Service32
 - Add a New Service to the interfaces File33
 - Starting an Access Service33
 - Verify an Access Service34
 - Stopping an Access Service35
 - Stopping a DirectConnect Server35
 - Manage ECDA Option for Oracle35
 - Using the DCOConfig Utility36
 - Add and Remove a Service36
 - Shut Down ECDA Option for Oracle37
- DirectConnect Manager39**
 - Manage a New Server39
 - Manage a New Access Service39
 - Configure Servers and Access Services40
- ECDA Option for ODBC Utilities41**
 - Create and Start a DirectConnect Director Server41
 - DCDirector Utility41
 - Create a DirectConnect Server42
 - AddServer Utility42
 - Start a DirectConnect Server42
 - DCStart Utility42
- Configuring Client Connectivity to ECDA45**
- Set Up the ODBC Back-End Driver and Driver Manager47**
 - Supported ECDA Back-End Drivers47
 - ODBC Driver Manager Architecture47
 - Using the unixODBC Driver Manager47
 - ODBC Driver Manager Configuration Properties48
 - Environment Variables49
 - Library Path Environment Variables49

ODBC Driver Manager Environment Variables	49
Troubleshoot Installation	51
ODBC-Specific Problems	51
ODBC Driver Errors	51
ODBC Data Source Errors	52
ODBC Driver Manager Errors	52
ECDA Problems	53
Connectivity Problems	53
UNIX Configuration Problems	54
Connectivity Tasks for AS/400	55
Enabling DRDA Capability of AS/400	55
Set Security Levels	55
Verifying the AS/400 Security Settings	56
Changing the Security Settings	56
CCSID	57
Changing the CCSID at the User Profile Level	57
SYBASE Collection on AS/400 (for DB2 UDB)	57
Setting Permissions from isql	58
Granting Permissions Directly on the AS/400	58
Creating the SYBASE Collection Using the AS/ 400 SEU Utility	59
Creating the SYBASE Collection Through SQL/ 400	60
Authorize Users for the SYBASE Collection	60
Installation Worksheets	61
Worksheets Instructions	61
ECDA Option for ODBC Worksheet	61
ECDA Option for Oracle Worksheet	61
Configure Data Sources for ECDA Option for ODBC	63
Connect to DB2 UDB	63
Define Datasource Entries	64
Example of an odbc.ini File for DB2 UDB on AS/ 400 or z/OS	67
Example of an odbc.ini File for DB2 UDB on Windows or UNIX	68

Contents

Connect to Microsoft SQL Server on UNIX	69
Datasource Entries	69
Example of an odbc.ini File	70
Glossary	71
Obtaining Help and Additional Information	85
Technical Support	85
Downloading Sybase EBFs and Maintenance Reports	85
Sybase Product and Component Certifications	86
Creating a MySybase Profile	86
Accessibility Features	86
Index	89

Conventions

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

Style Conventions

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

Syntax Conventions

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

Conventions

Introduction

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

ECDA Option for ODBC

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

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

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

- The DirectConnect™ server, which provides management and support functions for DirectConnect service libraries.
- An access service library, which accesses data from ODBC-accessible target databases, such as DB2 UDB and Microsoft SQL Server.
- Access services, which contain specific sets of configuration properties relating to the target to be accessed and define how each access service behaves.

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

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

ODBC Driver

ECDA Option for ODBC provides basic connectivity to non Sybase datasources, using the ODBC back-end (server-side) driver that you purchase for your target database, such as IBM or Microsoft SQL.

Following the vendor's instructions, you install the ODBC driver on the same server as ECDA Option for ODBC and then configure ECDA Option for ODBC to use that ODBC driver for access to your database.

Note: Verify that your ODBC driver is compatible with Sybase driver manager software.

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

ECDA Option for Oracle

ECDA Option for Oracle provides Open Client™ access to Oracle databases. It works with the Component Integration Services feature of Adaptive Server® Enterprise (Adaptive Server/CIS) or as a standalone gateway.

When used with Adaptive Server, ECDA Option for Oracle transforms the Transact-SQL™ generated by Adaptive Server to Oracle native SQL. ECDA Option for Oracle also handles datatype mapping between Sybase datatypes and Oracle datatypes. ECDA Option for Oracle provides many of the features of a distributed database system when used with Adaptive Server. This combination enables location transparency, distributed query optimization, copy transparency, transaction transparency, and distributed joins.

When used with Adaptive Server, you can join Oracle tables with Adaptive Server, DB2, or other tables. Access to these objects through Adaptive Server is transparent to the application. ECDA Option for Oracle supports full, two-phase commit transaction management.

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

DirectConnect Manager

DirectConnect Manager graphically represents each DirectConnect object on a tree list or an “icon map,” which is a customizable workspace where you can add or remove objects.

When you add a DirectConnect server to DirectConnect Manager, its server name, access service library, and any access services appear on the tree list or the icon map.

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

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

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

- DirectConnect Manager can perform all of its management functions remotely. With DirectConnect Manager, you do not need physical access to the DirectConnect server machine or directory.
- DirectConnect Manager provides management services to multiple servers at the same time, including the ability to copy access service configurations from one server to another.

For more information about DirectConnect Manager features, use the online help menu option.

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

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

Installation Task Flows

Task flows define a complete path for planning, installing, configuring, and uninstalling.

Choose the path that best describes your scenario.

Note: Print this topic and use it as a checklist.

Preparing for Installation

1. Setting up connectivity to the target database. See *Connectivity Tasks for AS/400* on page 55 and *Configure Data Sources for ECDA Option for ODBC* on page 63.
2. Filling up the *Installation Worksheets* on page 61.
3. Gathering your information and preparing the environment. See *Planning Your Installation* on page 9

Installing the Product for the First Time

1. *Installing ECDA* on page 13.
2. *Installing DirectConnect Manager* on page 19.

Setting Up Servers and Access Services

1. Creating and starting a DirectConnect server. See *Create DirectConnect Server* on page 25.
2. Managing DirectConnect servers and access services. See:
 - *ECDA Utilities* on page 31.
 - *DirectConnect Manager* on page 39.
 - *ECDA Option for ODBC Utilities* on page 41.

Performing Postinstallation

1. *Configuring client connectivity to ECDA* on page 45.
2. Setting up the back-end drivers and configure the driver manager. See *Set Up the ODBC Back-End Driver and Driver Manager* on page 47.

Uninstalling ECDA

Uninstalling ECDA on page 23.

See also

- *Troubleshoot Installation* on page 51

Planning Your Installation

Before you install, prepare your environment.

- Set up connectivity tasks.
- Obtain a license.
- Make sure your systems meet all requirements for your installation scenario and planned use.
- Perform administrative tasks.

Role and Skill Requirements

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.

Table 1. ECDA Team Skill Requirements

Role	Skill Set
Operating system administrator	<ul style="list-style-type: none"> • Understanding of the operating systems for your site's platforms. • Knowledge of standards and conventions at the installation site.
Communications administrator	<ul style="list-style-type: none"> • 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	<ul style="list-style-type: none"> • Understanding of the ECDA environment. • DirectConnect server administrator privileges.
Target database administrator	<ul style="list-style-type: none"> • Knowledge of the target database. • Knowledge of target environment, including security operations. • Database administrator privileges.
LAN administrator	<ul style="list-style-type: none"> • 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 prepare the environment that supports your ECDA installation and the subsequent access services that you create.

- Set up connectivity.
- Performing critical administrative tasks.

Connectivity Tasks

Before you begin installation, you must have some preliminary network connections in place and operational.

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

This is not a complete list. Prepare a list that is pertinent to your site and connectivity setup.

Set Up Target Database Connectivity

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

Connectivity Parameters

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

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. See the vendor contract and documentation.

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

System Requirements

Verify that you have the platform and operating system components listed in the ECDA release bulletin for your platform.

Also, determine whether your operating systems require any new release levels or CR fixes of Sybase or vendor connectivity protocol products.

SySAM Licensing

ECDA includes a Sybase Software Asset Management System (SySAM) implementation to perform license administration and asset management tasks.

Before installing your product, choose a SySAM license model, determine license server information, and obtain license files.

When you purchase SySAM 2–enabled Sybase products, you must generate, download, and deploy SySAM product licenses.

- If you purchased your product from Sybase or an authorized Sybase reseller, go to the secure Sybase Product Download Center (SPDC) at <https://sybase.subscribenet.com> and log in to generate license keys. The license generation process may vary slightly, depending on whether you ordered directly from Sybase or from a Sybase reseller.
- If you ordered your product under an SAP® contract and were directed to download from SAP Service Marketplace (SMP), you can use SMP at <http://service.sap.com/licensekeys> to generate license keys for Sybase products that use SySAM 2-based licenses.

You can install and use the ECDA components without a license for 30 days. To continue using these components after the grace period, obtain valid licenses from the SPDC or from SMP. See the *SySAM Users Guide*.

Installation Mode

Determine the installation mode to use. Sybase recommends that you use the GUI mode for installing ECDA Options.

You can install ECDA using:

- Graphical user interface (GUI) – lets you install the components in an interactive user mode.
- Console – lets you install components in a command line environment.
- Response file – lets you record or create a response file. You can install ECDA in two different ways using a response file:
 - Silent – save the installation settings in a response file and 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 – install interactively, but with all the responses already filled in, so you can accept all the defaults and install ECDA according to the responses in the response file. This can be convenient if several sites are installing ECDA in a nongraphical user interface environment and must conform to a standard installation.

See also

- *Installing ECDA in GUI Mode* on page 13
- *Installing ECDA in Console Mode* on page 15

- *Installing ECDA Using a Response File* on page 16

Performing Critical Administrative Tasks

Complete the administrative tasks before you begin your installation.

1. Read the *Enterprise Connect Data Access Release Bulletin* for your platform. This document provides product information that may not be included in this guide, as well as information about known issues or problems.
2. Make a backup copy of your current Sybase software, particularly if you plan to keep ECDA versions that were previously installed.
3. Make a copy of the installation worksheet for your platform for each server that you are installing.
4. Verify that you have authority to log in as "sybase" user account or another user that has administrative privileges on the machine on which ECDA is being installed.
5. Verify that the host server is connected to the LAN and that network protocols are configured correctly. For instructions, see your LAN administrator documentation.
6. 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.

Installing ECDA

Install ECDA using your chosen method.

Prerequisites

Complete installation planning tasks.

Task

Read this section for step-by-step installation instructions.

Installing ECDA in GUI Mode

The installer creates the target directory and installs the selected components into that directory.

Prerequisites

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

Task

1. Insert the ECDA installation media in the appropriate drive, or download and extract the ECDA installation media from the Sybase Product Download Center (SPDC) or the SAP Service Marketplace (SMP).
2. Start the installer.

If automount is not available on your machine, follow local procedures for mounting the CD.

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

```
./setup
```

3. On the Welcome screen, click **Next**.

You may see this 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. In the command prompt, enter:

Installing ECDA

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

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

4. Select the geographic location, agree to the license agreement, then click **Next**.
5. Specify where to install ECDA:

Option	Description
Click Next.	To accept the default directory for the installation.
Enter a different directory.	To install in the specified directory.

- If the directory you chose does not exist, click **Yes** to create it.
- If it does exist, and already contains an ECDA installation, you are warned that you will overwrite the older version.
Select **Yes** only if the version of the new files is later than the one you are attempting to overwrite.

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

6. Select the type of installation:

Option	Description
Custom	Lets you select the ECDA options and components to install. Some components are automatically installed if they are required to run your selected components. If you select this option, the next window displays all the ECDA options and all the selected components. You must unselect the components that you do not want to install. After you have selected either the ECDA Option for ODBC or ECDA Option for Oracle, and the specific components, click Next .
Full	Installs all ECDA products and components. If you selected this option, the next window displays all the ECDA options including ECDA Option for ODBC and ECDA Option for Oracle, and all of the selected components. Click Next .

Before proceeding to the next window, the installation program verifies the selections, and checks for dependencies and available disk space.

Note: If you click **Next** without sufficient hard disk space, an error occurs that stops the installation.

7. The Product Summary window displays the selections and the total disk space required for all the selected components. Review the information and click **Next**.
8. On the Sybase Software Asset Management License Server window:

Option	Description
Yes	If you have a preexisting SySAM network license server installed, or to install a new license server. Enter the host name and the port number of the license server.
No	If you do not have a SySAM license server installed, and to use a local 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, you see:

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

- Click **Yes** to complete the installation.
 - Click **No** and a message prompts you to download the license file. Click **OK**.
9. (Optional) Set SySAM e-mail configuration. Click **Next**.
 10. 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

Choose command line installation if you prefer a nonwindowing interface, or for developing custom installation scripts.

Prerequisites

Launch the installer in console mode. If the installer launches automatically, click **Cancel** to cancel the GUI installation, then launch the **setup** program from a terminal or console.

Task

The steps for installing components in console mode are the same as those for GUI mode, except you use **setup -console** to execute the installer from the command line, and you specify options by entering text.

1. At the command prompt, enter:

```
./setup -console
```

Installing ECDA

The installation program starts and displays a Welcome message.

2. Follow the remaining prompts to install ECDA. Output is written to a terminal window, and you must use the keyboard to enter responses.

See also

- *Installing ECDA in GUI Mode* on page 13

Installing ECDA Using a Response File

To perform a silent or unattended installation, run the installer and provide a response file that contains your installation configuration information.

Unattended or silent installations are typically used to update multiple systems across the enterprise.

Creating a Response File

Record installation responses into a response file, which is a text file you can edit to change any responses before using it in subsequent installations.

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

- Record mode – the installation program records all of your responses and selections in the specified response file. You must complete the installation to generate a response file.
- Template mode – the installation program creates a response file containing commented-out values for all required responses and selections. When using a template, you do not need to install the product, and you can cancel the installation after the response file has been created.

Warning! If you created a response file using a custom installation, there is an issue requiring you to edit the response file to allow the custom selections to be chosen correctly during installation.

Use a text editor to 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=
```

Without this change, a full installation is performed.

- To create a response file, enter:

```
./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 of the same remaining information.
- To create a template file, enter:

```
./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.

Installing in GUI Mode Using a Response File

An interactive installation using a response file lets you either accept the default values supplied by the response file or enter different values. This is useful when installing multiple installations that you want to change at installation time.

Prerequisites

Create an installation response file.

Task

At the command prompt, execute this command all on one line:

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

See also

- *Installing ECDA in GUI Mode* on page 13
- *Creating a Response File* on page 16

Installing in Silent Mode

A silent or unattended installation requires no user interaction; all installation configuration information is taken from the response file. This is useful when you want multiple identical installations, or to fully automate installation.

Prerequisites

Create an installation response file.

Task

At the command prompt, execute this 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 installer are the same, and the result of an installation in silent mode is exactly the same as one performed in GUI mode with the same responses.

See also

- *Installing ECDA in GUI Mode* on page 13
- *Creating a Response File* on page 16

Installing DirectConnect Manager

The installer creates the target directory and installs the selected components into that directory.

1. Insert the PC-Client CD into the CD drive.
2. Go to the DirectConnect Manager directory.
3. Execute this command depending on the platform:
 - On IBM AIX:
`setupaix.bin`
 - On Solaris:
`setupsolarisSparc.bin`
 - On HP-UX Itanium:
`setupHPIA64.bin`
 - On HP-UX:
`setuphp11x.bin`
 - On Linux:
`setuplinux.bin`
4. On the Welcome window, click **Next**.
5. Select the geographic location, agree to the license agreement, then click **Next**.
6. Enter the directory path for the installation and click **Next**.
7. Select one of:
 - Install a new version of Sybase Central™ and render the other installation invalid. Click **Next**.
This option installs DirectConnect Manager and all the required components, and create a new set of icons and registry entries.
 - Register with your existing Sybase Central. Click **Next**.
This option installs the DirectConnect Manager plug-in and registers it with the existing Sybase Central application.
8. The installation summary window displays the selections you have made. Review the information, and click **Next**.
9. The installation process begins and a Setup window appears. Click **Next**.
The Setup window shows the percentage (%) of installation that is complete, and monitors the decompressing and installing of the DirectConnect Manager files. The installation registers the plug-in with Sybase Central.
10. In the status window showing the Sybase Central registration process, click **Next**.

Installing DirectConnect Manager

11. When installation is completed, a DirectConnect Manager README file appears. After reading the file, click **Finish** to exit.

Upgrading ECDA for Oracle or ECDA for ODBC

Upgrade ECDA for Oracle or ECDA for ODBC server from 15.0 and later.

Prerequisites

- Back up the current ECDA directory tree from the install directory, including all sub-directories as downgrades are not supported.

Note: Any ECDA or DC for z/OS product that is version 15.0 or later can be upgraded with a newer version of the product.

- Take the services inventory and configuration properties currently running on the server using DCManger or copy the configuration files to a safe location and refer to those later.

Task

1. Stop ECDA for Oracle or ECDA for ODBC server.
2. Use the relevant version of the *Enterprise Connect Data Access Installation Guide* (this document) to install the newer version of ECDA in to the same based directory. For example, \$SYBASE directory.

For the ECDA for Oracle server, follow any instructions in the installation guide about downloading instant client libraries. This is necessary as the requirements for these libraries may have changed from the server you are upgrading.
3. Restart the ECDA server.
4. Run the DCManger GUI tool and connect to the new server instance to validate that the configuration properties and services are running as expected and match it to the inventory you created in the prerequisite section.
5. Use **isql** to connect to the target databases through each of the services and issue a simple **select** command to validate the connection.

Uninstalling ECDA

Uninstall the products and components.

Prerequisites

- Log in to your machine using an account with “administrator” privileges.
- Shut down all Sybase applications and processes.

Note: The uninstaller 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. You may need to delete any remaining files and directories in the directory where ECDA was installed only if this is the only product installed.

Task

1. Choose an uninstallation method:
 - GUI mode (recommended)
 - Console mode
2. Follow the instructions for your chosen method.

Uninstalling in GUI Mode

Uninstall ECDA in GUI mode.

1. Verify that the ECDA server is shut down.
2. Enter:

```
$SYBASE/uninstall/ECDA150Suite/uninstall
```

3. Click **Next**.

All installed products and components appear as selected, to allow you to remove the entire installation. Unselect any products or components you do not want to remove.

4. Click **Next**.
5. The uninstallation summary window displays the selections you have made. Review the information, and click **Next**.
6. Click **Finish** to complete the uninstallation program.

Uninstalling in Console Mode

Uninstall ECDA in console mode.

1. At the command prompt, navigate to the installation directory and enter:

```
$SYBASE/uninstall/ECDA150Suite/uninstall -console
```

The uninstallation program starts.

2. Follow the remaining prompts to uninstall ECDA. Output is written to a terminal window, and you must use the keyboard to enter responses.

See also

- *Uninstalling in GUI Mode* on page 23

Create DirectConnect Server

Use the Create Server wizard, which is a Java program, to create a DirectConnect server and complete the connection to the target database based on the driver and driver manager you want to use.

The wizard 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.

See also

- *DirectConnect Manager* on page 39

Create a Server in ECDA Option for ODBC

Create a server for the target databases for ECDA Option for ODBC using the Create Server wizard with the back-end drivers.

The supported back-end drivers are:

- Sybase-provided unixODBC driver manager
- DataDirect driver
- IBM DB2 driver

The Create Server wizard is located in `install_dir/DC-15_0/DCWizard`.

Using unixODBC Driver Manager

Create a server using a unixODBC driver manager.

1. Start the Create Server wizard. Enter:
`DCWizard.sh`
2. On the Welcome Create Server Wizard window, click **Next**.
3. Select **ECDA Option for ODBC** and click **Next**.
4. In the Server Name and Port Information window, enter:

Field	Description
Server name	Name of the DirectConnect server

Create DirectConnect Server

Field	Description
Port number	Listener port number of the server

Click **Next**.

5. Select **unixODBC Driver Manager** and click **Next**.
6. Enter the ECDA access service name to use for this server and click **Next**.
7. The ECDA Server Summary and Build window displays the selections you have made. Review the information and click **Create Server**.
8. (Optional) Start the DirectConnect server that you created and 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, start DirectConnect Manager and set **ConnectionSpec1** to a data source name (DSN) in the `odbc.ini` file located in `dc_install_dir/DC-15_0`.

9. Set the configuration property, **SQLODBCursors to driver** in the `dcany.cfg` file.

Using DataDirect Driver

Create a server using the DataDirect driver.

Prerequisites

- Install the DataDirect ODBC software on the same machine where 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.

Task

1. Start the Create Server wizard. Enter:

```
DCWizard.sh
```
2. On the Welcome Create Server Wizard window, click **Next**.
3. Select **ECDA Option for ODBC** and click **Next**.
4. In the Server Name and Port Information window, enter:

Field	Description
Server name	Name of new DirectConnect server
Port number	Listener port number of the server

Click **Next**.

5. Select the **DataDirect Driver Manager** option to use for this server and click **Next**.

6. Specify the path to the driver manager:

- Within the DataDirect installation, for example:

```
datadirect_install_dir/lib/libodbc.so
```

- Supported libraries, for example:

```
datadirect_install_dir/lib/
```

- `odbc.ini` file to use, for example:

```
datadirect_install_dir/odbc.ini/
```

Click **Next**.

7. Enter the ECDA access service name to use for this server and click **Next**.

8. The ECDA Server Summary and Build window displays the selections you have made. Review the information and click **Create Server**.

9. (Optional) Start the DirectConnect server that you created to validate the server configuration.

If you exit from this wizard, the DirectConnect server you have started continues 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, start DirectConnect Manager and set **ConnectionSpec1** to a data source name (DSN) in the `odbc.ini` file.

10. Set the configuration property, **SQLODBCursors** to **driver** in the `dcany.cfg` file.

Using IBM DB2 Driver

Create a server using the IBM DB2 driver.

Prerequisites

- 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.

Task

1. Start the Create Server wizard. Enter:

```
DCWizard.sh
```

2. On the Welcome Create Server Wizard window, click **Next**.

3. Select **ECDA Option for ODBC** and click **Next**.

Create DirectConnect Server

- In the Server Name and Port Information window, enter:

Field	Description
Server name	Name of new DirectConnect server
Port number	Listener port number of the server

Click **Next**.

- Select **unixODBC Driver Manager without a Driver Manager** and click **Next**.
- Specify the path to:

- The 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 is named `libdb2o.so`, as shown in the directory path.

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

```
/export/home/nedb2i7/sqlllib/db2cshrc
```

Click **Next**.

- Enter the ECDA access service name to use for this server and click **Next**.
- The ECDA Server Summary and Build window displays the selections you have made. Review the information and click **Create Server**.
- (Optional) Start the DirectConnect server that you created to validate the server configuration.

If you exit from this wizard, the DirectConnect server you have started continues 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 DB2, start DirectConnect Manager and set **ConnectionSpec1** to a DB2 alias by executing “list database directory” in the **db2** command shell.

Creating a Server in ECDA Option for Oracle

Use the Create Server wizard to create a server for an Oracle-accessible target database.

The Create Server wizard is located in `install_dir/DCO-15_0/DCWizard`.

- Start the Create Server wizard. Enter:

```
DCWizard.sh
```

- On the Welcome Create Server Wizard window, click **Next**.

3. Select **ECDA Option for Oracle** and click **Next**.
4. In the Server Name and Port Information window, enter:

Field	Description
Server name	Name of new DirectConnect server
Port number	Listener port number of the server

Click **Next**.

5. Enter the Admin Account Name and click **Next**.
6. In the Oracle Target Information window, enter:

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

Click **Next**.

7. The Oracle DirectConnect Server Summary and Build window displays the selections you have made. Review:

Option	Description
ECDA Server information	If correct, click Create Server .
ECDA Config information	(Optional) If correct, click Create Server and the Start DirectConnect Server window appears.

8. (Optional) Start the DirectConnect server that you created to validate the server configuration.
 - If you exit from this wizard, the DirectConnect server you have started continues to execute.
 - If you want to skip this step, click **Finish**.

Note: You can use DirectConnect Manager to change the configuration settings.

Create DirectConnect Server

ECDA Utilities

After installing ECDA Option for ODBC or ECDA Option for Oracle, use the ECDA utilities to manage a server, and a service.

Use either:

- ECDA Option for ODBC utility, or,
- ECDA Option for Oracle utility.

You can also use DirectConnect Manager to manage servers and services.

See also

- *DirectConnect Manager* on page 39

Manage ECDA Option for ODBC

Use the ECDA utilities and command line to manage a server and service for the ECDA Option for ODBC.

See also

- *ECDA Option for ODBC Utilities* on page 41

Creating a New DirectConnect Server

The installer automatically creates the subdirectories and files for the ECDA installation. Use the **AddServer** utility to create a new DirectConnect server.

1. Create a new server:

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

where:

- *srvname* is the name of the new server.
- *1234* is the listener port.
- **AddServer** is the utility that:
 - Sets the environment variables
 - Creates the server
 - Starts the server
 - Inserts the server name in the Sybase interfaces file

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

Setting Up the Environment Variables

Set up the environment variables for your installation.

Note: ECDA includes utilities that allow you to create and start a DirectConnect server.

Set the environment variables from a command prompt by executing the appropriate script for your platform:

- `DC_SYBASE.csh`
- `DC_SYBASE.sh`

Use these Sybase-provided scripts to set the environment variables scripts for executing ECDA:

- `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 IBM AIX), `LD_LIBRARY_PATH` (for Solaris, Linux, HP-UX Itanium)

The response should match the home directory on your worksheet:

- For HP-UX:
`echo $$SHLIB_PATH`
- For IBM AIX:
`echo $LIB_PATH`
- For Solaris, Linux, and HP-UX Itanium:
`echo $LD_LIBRARY_PATH`

The HP environment variable is:

```
install_dir/DC-15_0/lib
install_dir/DC-15_0/connectivity/OCS-15_0/lib
```

Creating a New Access Service

Use a text editor to edit the service library configuration file, which resides in the `cfg` subdirectory under the `ServerName` directory to create and configure an access service.

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 need.

Next

See *Enterprise Connect Data Access Options for ODBC Users Guide for Access Services*.

Add a New Service to the interfaces File

Use the **dsedit** utility to add the new service to the `interfaces` file as a server object.

To allow an Open Client application, such as **isql**, to connect to the Open Client installation that is used by your client application, 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.

Modifying the interfaces File

Use **dsedit** to modify the `interfaces` file.

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. In the Network Address window, 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. See the *Enterprise Connect Data Access and Mainframe Connect Server Administration Guide*.

Starting an Access Service

Use the DirectConnect Manager to start the new access service without stopping and restarting ECDA.

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

Verify 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 the Command Prompt

Use the **isql** to verify the access service configuration.

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.

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

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 prompt by executing the appropriate script for your platform:

- `DC_SYBASE.csh`
- `DC_SYBASE.sh`

3. From the command prompt, enter:

```
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, you see a `1>` prompt.

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

4. At the `1>` prompt, enter a **select** statement and press **Return** to query a table in the target database.
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:

1> prompt: `exit`

Press **Return**.

You see a regular operating system command.

Note: You can stop the DirectConnect server using the **stopsrvr** utility, which shuts down the server and terminates all client connections. However, if you plan to perform postinstallation 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.

See also

- *DirectConnect Manager* on page 39

Stopping a DirectConnect Server

Use the **stopsrvr** utility to stop the DirectConnect server, which shuts down the server and ends all client connections.

Set the environment variables from a command prompt by executing the appropriate script for your platform:

- `DC_SYBASE.csh`
- `DC_SYBASE.sh`

The **stopsrvr** format is:

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

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.

Manage ECDA Option for Oracle

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

Use the **sp_configure** in the command prompt to modify the ECDA Option for Oracle configuration.

See **sp_configure** in the *Enterprise Connect Data Access Option for Oracle Administration and Users Guide*.

Using the DCOConfig Utility

Use the **DCOConfig** script to configure and start a server.

Prerequisites

You must have:

- A copy of the existing `tnsnames.ora` file that is currently being used to connect to Oracle 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

Task

The `DCOConfig` script is located in `install_dir/DCO-15_0/install` directory.

1. Change 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 listens 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.

ECDA Option for Oracle starts in a separate window.

In the window where the ECDA Option for Oracle is started, you see this message, which you can safely ignore:

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

Add and Remove a Service

Use the ECDA Option for Oracle utility **sp_addservice** to add a service and **sp_dropservice** to remove a service.

See the commands and syntaxes in *Adding a Service* and *Removing a Service* in the *Enterprise Connect Data Access Option for Oracle Administration and Users Guide*.

Shut Down ECDA Option for Oracle

Use **sp_shutdown** to shut down ECDA Option for Oracle.

See the command and syntax in *Shutting Down ECDA Option for Oracle* in the *Enterprise Connect Data Access Option for Oracle Administration and Users Guide*.

DirectConnect Manager

DirectConnect Manager allows you to create and start a server, and create and start an access service.

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

Manage a New Server

You can use the DirectConnect Manager to create a new server, or to start and stop a server.

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, see topics in the DirectConnect Manager online help:

- *Server Administration > Creating New Servers.*
- *Server Administration > Starting a Server.*
- *Server Administration > Stopping a Server.*

Manage a New Access Service

Use the DirectConnect Manager to create a new access service, or to start and stop an access service.

See topics in the DirectConnect Manager online help:

- *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.

See *Service Name Redirection* in the *Enterprise Connect Data Access and Mainframe Connect Server Administration Guide*.

- *Managing Access Services > Starting a Service.*
- *Managing Access Services > Stopping a Service.*

Configure Servers and Access Services

Use the DirectConnect Manager to configure the server after you have created a DirectConnect server or a DCDirector.

For a description of all the configuration properties, see:

- For the DirectConnect server:
 - (ECDA Option for ODBC) See *Enterprise Connect Data Access and Mainframe Connect Server Administration Guide*.
 - (ECDA Option for Oracle) See *Enterprise Connect Data Access Option for Oracle Server Administration and Users Guide*.
- For the access service:
 - See *Enterprise Connect Data Access Option for ODBC Users Guide for Access Services*.
 - See *Enterprise Connect Data Access Option for Oracle Server Administration and Users Guide*.

See also

- *Using the DCOConfig Utility* on page 36

ECDA Option for ODBC Utilities

To simplify the execution of ECDA on multiple platforms, Sybase provides utilities for starting server, creating a server, and migrating files from one server to another.

These utilities are `cshell` scripts that are located in `install_dir/DC-15_0/bin`. Keep the utilities in this directory; it is from this directory that the utilities can derive the paths to the other files they need to perform their tasks.

Note: You cannot use these utilities for ECDA Option for Oracle.

Create and Start a DirectConnect Director Server

Use the **DCDirector** script to create a new DCDirector for an installation.

ECDA allows DirectConnect Manager to connect to a directing server, called a DCDirector, that is capable of creating, starting, and stopping DirectConnect servers.

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. Enter:

```
DCDirector
```

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

DCDirector Utility

DCDirector 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 using a text editor, if you prefer to use different values.

Syntax

```
DCDirector
```

Example

```
DCDirector
```

When created, DCDirector can start, stop, and provide other server functions.

Create a DirectConnect Server

Use **AddServer** to create 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** is that it does not check the `interfaces` file for duplicate server names or ports in the same way that **DCDirector** does.

AddServer Utility

Use **AddServer** 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.

Syntax

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

Example

```
AddServer srvname 1133
```

Start a DirectConnect Server

In the past, you started a DirectConnect server using a command that used the **direct** executable itself. Although this method is still supported, it has some limitations.

For example, the command, **direct -Ssrvname**, was the standard way to start an existing DirectConnect server. For this command to work, you must properly set all of the appropriate environment variables before the command is executed. Also, if multiple installations of ECDA exist on a single machine, each installation needs its own environment.

To ease the use of this process, Sybase provides a shell script with ECDA 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

DCStart is similar to using the **direct** executable.

DCStart automatically sources the appropriate `DC_SYBASE.csh` file to ensure that all the appropriate Sybase-specific environment variables are set. This is the required method to start

a DirectConnect server, for one or multiple versions, since ECDA no longer sets the system environment.

Syntax

DCStart [**direct** executable **Parameters**]

Example

DCStart -*Ssrvname*

Configuring Client Connectivity to ECDA

Use **dsedit** to verify that the client environment is connected correctly to the local area network (LAN) and to the DirectConnect server.

Prerequisites

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

Task

Tip: If you verified the ECDA access service using **isql**, you have already created an entry in the `interfaces` file on that machine.

1. Use **dsedit** to verify that the ECDA service name and port number match the entry for the access service.

Note: Use the access service name, not the DirectConnect server name.

If you use service name redirection for the server name, see 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.
3. 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.

See also

- *Using isql from the Command Prompt* on page 34

Set Up the ODBC Back-End Driver and Driver Manager

Set up the supported ODBC back-end drivers and driver managers, and the environment variables for ECDA Option for ODBC.

Supported ECDA Back-End Drivers

Sybase does not provide ODBC drivers for back-end access to the target database. You must 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
- 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.

As of 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.

Using the unixODBC Driver Manager

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

Either when:

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

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 two options.

Set Up the ODBC Back-End Driver and Driver Manager

Option 1

Use the unixODBC driver manager to specify the configuration value in the `dcany.cfg` set in the full path.

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.

Option 2

Use the unixODBC driver manager to specify the configuration value in the `dcany.cfg` set in the relative path.

Move the library found in the `install_dir/DC-15_0/drv_mgr/unixODBC-2.2.12` directory located 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 does not expand the variable.

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

ODBC Driver Manager Configuration Properties

The ODBC driver manager that is loaded at runtime is determined by the file defined in the Service Library configuration property, **ODBCDriverManager**.

As of version 15.0, the ECDA Option for ODBC provides the capability to specify the ODBC driver manager at runtime. The library extension can be `.sl` (for HP) or `.so` (for Solaris, IBM AIX, Linux, and HP-UX 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 **SQLODBCursors** property to **driver**. If set to the default, you see:

```
Cannot load library libodbccr
```

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 vendor's installation and configuration documentation.

Environment Variables

Review the required environment variables for the back-end driver and driver manager setup.

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 load these other libraries to satisfy runtime dependencies. If it does not, the initial library load fails. The library path environment variables are:

- On Linux, Solaris, HP Itanium – `LD_LIBRARY_PATH`
- On AIX – `LIBPATH`
- On 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

The unixODBC driver manager uses the `ODBCSYSINI` environment variable to find and load drivers.

The ODBC driver manager uses the configuration files that 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 points to the directory containing the `odbc.ini` and `odbcinst.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:

Set Up the ODBC Back-End Driver and Driver Manager

- `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.

Troubleshoot Installation

Identify and resolve problems that occur when you install an ECDA product. You can use the diagnostic tools provided to resolve problems.

If you try the suggested methods 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 included with ECDA)

ODBC-Specific Problems

ODBC-specific problems that may occur including driver errors, data source errors, and driver manager errors.

ODBC Driver Errors

The ODBC driver error format and its solution.

The format of an error reported on an ODBC driver is:

```
[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 similar to:

```
[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

The ODBC data source error format and its solution.

This type of error occurs in the datasource and includes the data store name, as shown in this format:

```
[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 datasource where the error occurred.
- *message* is the content of the error message.

For example, you may receive this 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 ODBC driver manager error format and its solution.

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 looks similar to:

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

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

ECDA Problems

Address postinstallation ECDA problems.

- Confirm the release number of Open Server software. ECDA is compatible with Open Server and Adaptive Server versions identified in the release bulletin.
- Check the log and trace files in the `log` file of the `ServerName`. Error messages look similar to:

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

Possible causes include:

- 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 prelog messages.

See also

- *Planning Your Installation* on page 9

Connectivity Problems

Use the **odbct** tool to diagnose the connectivity 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
```

Troubleshoot Installation

```
> 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

Use **ivtestlib** to find and resolve the configuration problems.

You can use **ivtestlib**, which is not included with ECDA, 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 IBM AIX versions of **ivtestlib** do not require you to specify a full path.

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 after logging in to the target AS/400 server as the system administrator.

Note: Systems Network Architecture (SNA) /Advanced Program-to-Program Communications (APPC) connectivity is not supported.

Enabling 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.

Prerequisites

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 **ADDRDBDIRE** command.

Task

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, `dcany.cfg`, which is located in the `ServerName\cfg` subdirectory.

Set Security Levels

Set the security levels for AS/400.

There are four security levels. Sybase recommends that you use AS/400 security level 20 or higher.

Table 2. 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.

Verifying the AS/400 Security Settings

Use DSPSYSVAL QSECURITY to check 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
```

3. On the Work with System Value window, to show your security settings, enter:

```
DSPSYSVAL QSECURITY
```

Changing the Security Settings

Use CHGSYSVAL QSECURITY to change the security settings.

Prerequisites

You must have system security officer (QSECOFR) authority.

Task

1. From the Work with System Values menu, enter:

```
CHGSYSVAL QSECURITY
```

2. Enter the value.
3. Restart the AS/400.

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.

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. Scroll 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.

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 includes with the ECDA installation), or entered directly on the AS/400 machine.

Setting Permissions from isql

Use the **isql** to grant permission.

The ECDA service used to issue **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 this command before you issue **GRANT EXECUTE** command:

```
set sqltransformation sybase
```

Issue:

```
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 in one of:

- 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, use the AS/400 **SEU** utility program to create the SYBASE collection.

Granting Permissions Directly on the AS/400

Grant permission on the package for SYBASE collection.

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 the owner of the SYBASE collection.
2. To display the Sybase packages, enter this from the command prompt:

```
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

Use the SEU utility to create SYBASE collection. The user ID for creating the SYBASE collection must have a valid CCSID for your language installation.

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. Enter:

```
STRSEU
```

3. Press **F4** and provide this information:

- Source File – QQMQRYSRC
- Source Member – *SELECT
- Library File – QGPL

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

4. Press **Enter**. The members of the source file are listed.
5. Build the query that creates the collection by creating a source member for the query; enter:
 - New Member – SYBASE
 - Type for New Member – TXT

6. Press **Enter**.

7. To create the query member, enter this in the first line of the source file:

```
CREATE COLLECTION SYBASE
```

8. Press **F3**. Verify that the Change/Create Member field is set to Y, and the Member field is SYBASE.
9. Press **Enter** to exit the editor.
You can now work with this query.
10. Press **F3** again to return to the main menu.

Building the Query for Sybase Collection

Use the AS/400 utility to create build query for Sybase collection.

1. From the command prompt, enter:

```
CRTQMORY QMORY (QGPL/SYBASE) SRCFILE (QGPL/QQMQRYSRC)
```

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

Note: You may disregard a warning message that the source length exceeds 79 characters; it does not affect the access service.

2. To run the query that creates the SYBASE collection, enter:

```
STRQMORY QMORY (QGPL/SYBASE)
```

Press **Enter**. The SYBASE collection is created.

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

Creating the SYBASE Collection Through SQL/400

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

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. From the command prompt, enter:

```
STRSQL
```

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.

Authorize Users for the SYBASE Collection

You must have the 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 QSQJRN001, QSQJRN002, and any other additional journal receivers

Verifying Authorization Level Setting

Make sure that the authorization level for *SQLPKG and journal objects is set correctly.

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.

Installation Worksheets

Record installation information on the worksheets to complete installation and connectivity tasks.

Worksheets Instructions

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 are prompted for during installation, provide 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

Information you should designate and record before installing ECDA Option for ODBC. You are prompted for this information during installation.

Your Installation Information	Description
<i>DSN(s):</i>	ODBC data source name.
<i>CONNECTIVITY PARAMETERS:</i>	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

Information you should designate and record before installing ECDA Option for Oracle. You are prompted for this information during installation.

Your Installation Information	Description
<i>SERVER NAME:</i>	Name of the DirectConnect server you want to create or update.
<i>PORT NUMBER:</i>	Name of the ECDA port number you are using.
<i>DCO ADMIN ACCOUNT NAME:</i>	The name of a valid Oracle account to administer ECDA Option for Oracle. Usually the system account name.
<i>ORACLE CONNECT STRING:</i>	The entry for the Oracle instance of the <code>tnsnames.ora</code> file.
<i>ENTER LOCATION OF TNSNAMES.ORA FILE:</i>	Full path (including file name) of the existing <code>tnsnames.ora</code> file, usually located in the network <code>admin</code> directory under the <code>ORACLE_HOME</code> directory. The <code>tnsnames.ora</code> identified must be accessible to this machine. If the file is unavailable, then it must be copied on to a local drive before configuring ECDA Option for Oracle. ECDA copies the existing <code>tnsnames.ora</code> into its directory.

Configure Data Sources for ECDA Option for ODBC

You must create and configure the data source name (DSN) for each ECDA component for ECDA Option for ODBC to connect to target databases.

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

Note: The DataDirect driver is not included with the ECD Option for ODBC, but the information in this topic 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.

Connect to DB2 UDB

Execute the ECDA environment script to point to `odbc.ini` file to connect 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`.

Define Datasource Entries

To create and configure datasource entries for DRDA, edit the `odbc.ini` in the required parameters settings.

- 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 these parameters available on your worksheet. For all others, you can accept the default or provide a value appropriate for your site.

General Parameters

Edit the general parameters for the connectivity settings in the `odbc.ini` file. The required parameters are indicated by an asterisk (*).

Parameter	Comments
*Driver	The fully qualified path to the driver.
*Ip address	<p>The Internet Protocol (IP) 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.</p> <hr/> <p>Warning! For Solaris systems, use the numeric IP address, not the host name.</p> <p>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 data source name (DSN) server.</p>
*TcpPort	<p>Enter the port number that is assigned to the DB2 server on the machine where the catalog tables are stored.</p> <p>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. The default for:</p> <ul style="list-style-type: none"> • DB2 may be 50000 • AS/400 is 446 <p>If you specify a service name, the driver must find this name (with the correct port assignment) in the SERVICES file on the workstation.</p>
*Location (Required for DB2 UDB on AS/400 and z/OS only)	<p>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.</p> <hr/> <p>Note: This parameter is not applicable if the Database Name field is populated.</p>

Parameter	Comments
* Collection (Required for DB2 UDB on AS/400 and z/OS only)	<p>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.</p> <p>The default for DB2 and AS/400 is the user ID.</p> <hr/> <p>Note: This field is not applicable if the Database Name field is populated.</p>
* Database (Required for connecting to DB2 on Windows and UNIX)	<p>The name of the database to which you want to connect to, by default.</p> <hr/> <p>Note: This parameter is not used for AS/400 and z/OS.</p>
* IANAAppCodePage (Required for UNIX platforms)	<p>A driver code page configuration property for the DB2 UDB driver, for example, 3, 4, 2004, and 2250.</p> <p>For a complete list of the values and their descriptions, see <i>IANA</i>.</p>
WorkAroun2	<p>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.</p>

After entering these parameters, use the **odbc** tool to test connectivity to the DB2 UDB source.

See also

- *Troubleshoot Installation* on page 51

Advanced Parameters

Edit the Advanced parameters optional fields that affect performance and use of resources in the `odbc.ini` file.

These default values for the Advanced parameters are sufficient for most DB2 UDB installations.

Parameter	Comments
AlternateID	<p>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 current schema fails, the connection fails with <code>Invalid value for Alternate ID</code>. DB2 permissions should be set to <code>SYSADM</code>. (Not valid for AS/400 V5R1.)</p>

Configure Data Sources for ECDA Option for ODBC

Parameter	Comments
WithHold	<p>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).</p> <ul style="list-style-type: none"> • 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.
AddStringToCreate Table	Use this parameter to append the in tablespace clause to create table commands.

Bind Parameters for DB2 UDB

Edit the bind parameters in the `odbc.ini` file to specify package creation in the DB2 system.

The required parameters are indicated by an asterisk (*).

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.
*IsolationLevel	<p>Select the isolation level method by which locks are acquired and released by the system. Valid values are:</p> <ul style="list-style-type: none"> • 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 is identical to the previous table; however, it can contain updates made by your program.

Parameters	Comments
*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.

Bind Utility

Use the **bind20** utility included in the `bin` subdirectory to create packages.

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

The **bind20** utility uses the bind options defined in your Distributed Relational Database Architecture (DRDA) DSN to configure the package and grant proper ownership and the 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 identify the DSN parameters required and used for DRDA to these 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

An example of the DSN template for DB2 UDB. You must edit the section of the file that applies to your site.

```
;;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=
```

Configure Data Sources for ECDA Option for ODBC

```
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

An example of the DSN template for DB2 UDB on Windows or UNIX. You must edit the section of the file that applies to your site.

```
;;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
```

Connect to Microsoft SQL Server on UNIX

Execute the ECDA environment script to point to `odbc.ini` file to connect 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 installation. 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`.

Datasource Entries

Edit the datasource parameters in the `odbc.ini` file.

When 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 3. Datasource Parameters

Parameter	Description
*Driver	The full path to the driver.
Description	An optional long description of the datasource name.
*QuotedID	Parameter that allows quoted identifiers, that is, identifiers in the Microsoft SQL Server that you can quote using a quoting character.
*Database	Name of the database to which you want to connect by default.

Configure Data Sources for ECDA Option for ODBC

Parameter	Description
*Address	Parameter that contains a 4-part IP address and the port number that it is listening on.

Example of an odbc.ini File

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
;;
[dcmsql]
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

Glossary of terms used in Enterprise Connect™ Data Access.

- **accept** – establishment of a SNA or TCP/IP connection between Mainframe Connect™ Server Option and Mainframe Connect DirectConnect for z/OS Option.
- **access service** – the named set of properties, used with an access service library, to which clients connect. Each DirectConnect server can have multiple services.
- **access code** – a number or binary code assigned to programs, documents, or folders that allows authorized users to access them.
- **access service library** – a service library that provides access to non-Sybase data contained in a database management system or other type of repository. Each such repository is called a “target.” Each access service library interacts with exactly one target and is named accordingly. See also *service library*.
- **ACSLIB** – see *access service library*.
- **Adaptive Server Enterprise** – the server in the Sybase client/server architecture. Adaptive Server manages multiple databases and multiple users, tracks the actual location of data on disks, maintains mapping of logical data description to physical data storage, and maintains data and procedure caches in memory.
- **Adaptive Server Enterprise/Component Integration Services** – includes a variation of Adaptive Server that provides a Transact-SQL interface to various sources of external data. Component Integration Services allows Adaptive Server to present a uniform view of enterprise data to client applications.
- **administrative service library** – a service library that provides remote management capabilities and server-side support. It supports a number of remote procedures, invoked as RPC requests, that enable remote DirectConnect server management. See also *remote procedure call*, *service library*.
- **ADMLIB** – see *administrative service library*.
- **Advanced Interactive Executive** – the IBM implementation of the UNIX operating system. The RISC System/6000, among other workstations, runs the AIX operating system.
- **advanced program-to-program communication** – hardware and software that characterizes the LU 6.2 architecture and its implementations in products. See also *logical unit 6.2*.
- **AIX** – see *Advanced Interactive Executive*.
- **AMD2** – the component of the Mainframe Connect DB2 UDB Option that allows clients to submit SQL statements to DB2 UDB. It is a CICS transaction that receives SQL statements sent from Mainframe Connect DirectConnect for z/OS Option and submits them to DB2 UDB, using the DB2 UDB dynamic SQL facility. It also receives the results

and messages from DB2 UDB and returns them to Mainframe Connect DirectConnect for z/OS Option.

- **American Standard Code for Information Interchange** – the standard code used for information interchange among data processing systems, data communication systems, and associated equipment. The code uses a coded character set consisting of 7-bit coded characters (including a parity check, 8 bits).
- **API** – see *application program interface*.
- **APPC** – see *advanced program-to-program communication*.
- **application program interface** – the programming language interface between the user and Mainframe Connect Client Option or Mainframe Connect Server Option. The API for Mainframe Connect Client Option is Client-Library. The API for Mainframe Connect Server Option is Gateway-Library.
- **ASCII** – see *American Standard Code for Information Interchange*.
- **Adaptive Server Enterprise** – see *Adaptive Server Enterprise*.
- **Adaptive Server Enterprise /CIS** – see *Adaptive Server Enterprise/Component Integration Services*.
- **batch** – a group of records or data processing jobs brought together for processing or transmission.
- **bind** – in the Sybase environment, this term has different meanings depending on the context:
 - In CICS, bind is an SNA command that establishes a connection between LUs, or a TCP/IP call that connects an application to a port on its system.
 - In DB2 UDB, bind compiles the Database Request Module, the precompiler product that contains SQL statements in the incoming request, and produces an access plan, a machine code version of the SQL statements that specifies the optimal access strategy for each statement.
 - In the mainframe access product set, bind establishes a connection between a TRS port and a CICS or IMS region.
- **bulk copy transfer** – a transfer method in which multiple rows of data are inserted into a table in the target database. Compare with *destination-template transfer* and *express transfer*.
- **call level interface** – a programming style that calls database functions directly from the top level of the code. Contrast with *embedded SQL*.
- **catalog** – a system table that contains information about objects in a database, such as tables, views, columns, and authorizations.
- **catalog RPC** – a component of the Mainframe Connect DB2 UDB Option that allows clients to access DB2 UDB system catalogs. It uses an interface compatible with the catalog interface for the ODBC API.
- **catalog stored procedure** – a procedure used in SQL generation and application development that provides information about tables, columns, and authorizations.

- **character set** – a set of specific (usually standardized) characters with an encoding scheme that uniquely defines each character. ASCII is a common character set.
- **CICS** – see *Customer Information Control System*.
- **CICS region** – the instance of CICS.
- **client** – in client/server systems, the part of the system that sends requests to servers and processes the results of those requests. See also *client/server*. Compare with *server*.
- **client application** – software responsible for the user interface that sends requests to applications acting as servers. See also *client/server*.
- **Client-Library** – a library of routines that is part of Mainframe Connect Client Option.
- **client request** – an RPC or language request sent by a client to a server.
- **client/server** – an architecture in which the client is an application that handles the user interface and local data manipulation functions, and the server is an application providing data processing access and management. See also *client application*.
- **Client Services Application** – a customer-written CICS program initiated on the host that uses the API to invoke the Mainframe Connect Client Option as a client to the ECDA Option for Oracle server or to Adaptive Server. See also *application program interface*, *Client Services for CICS*.
- **Client Services for CICS** – a Sybase host API that invokes the Mainframe Connect Server Option as a client to an access service for DB2 UDB or Adaptive Server. See also *application program interface*, *Customer Information Control System*, *Client Services Application*, *Mainframe Connect Server Option*.
- **clustered index** – an index in which the physical order and the logical (indexed) order is the same. Compare with *nonclustered index*.
- **code page** – an assignment of graphic characters and control function meanings to all code points.
- **commit** – a process that makes permanent all changes made to one or more database files since the initiation of the application program, the start of an interactive session, or the last **commit** or **rollback** operation. Compare with *rollback*.
- **Common Programming Interface** – specifies the languages and services used to develop applications across SAA environments. The elements of the CPI specification are divided into two parts: processing logic and services.
- **configuration file** – a file that specifies the characteristics of a system or subsystem.
- **configuration set** – a section into which service library configuration files are divided.
- **conversion** – the transformation between values that represent the same data item but which belong to different datatypes. Information can be lost due to conversion, because accuracy of data representation varies among different datatypes.
- **connection** – a network path between two systems. For SNA, the path connects a logical unit (LU) on one machine to an LU on a separate machine. For TCP/IP, the path connects TCP modules on separate machines.
- **connection router** – a program provided with Mainframe Connect Client Option that directs requests to particular remote servers. Mainframe system programmers use the

connection router to define remote servers and server connections to Mainframe Connect Client Option.

- **Connection Router Table** – a memory-resident table maintained by a Mainframe Connect Client Option system programmer that lists servers and the connections that a Client-Library transaction can use to access them.
- **control section** – the part of a program specified by the programmer to be a relocatable unit, all elements of which are to be loaded into adjoining main storage locations.
- **control statement** – in programming languages, a statement that is used to alter the continuous sequential execution of statements. A control statement can be a conditional statement or an imperative statement.
- **conversation-level security** – the passing of client login information to the mainframe by TRS when it allocates a conversation.
- **CSA** – see *Client Services Application*.
- **CSP** – see *catalog stored procedure*.
- **cursor** – in SQL, a named control structure used by an application program to point to a row of data.
- **Customer Information Control System** – an IBM licensed program that enables transactions entered at remote terminals to be processed concurrently by user-written application programs.
- **DASD** – see *direct access storage device*.
- **data definition statement** – an IBM mainframe statement that relates a name to a file.
- **data definition language** – a language for describing data and data relationships in a database.
- **data set name** – the term or phrase used to identify a data set.
- **database management system** – a computer-based system for defining, creating, manipulating, controlling, managing, and using databases.
- **database operation** – a single action against the database. For Mainframe Connect DirectConnect for z/OS Option, a database operation is usually a single SQL statement. One or more database actions can be grouped together to form a request. See also *request*.
- **Database 2** – an IBM relational database management system.
- **datatype** – a keyword that identifies the characteristics of stored information on a computer.
- **DB-Library™** – a Sybase and Microsoft API that allows client applications to interact with ODS applications. See also *application program interface*.
- **DBMS** – see *database management system*.
- **DB2 UDB** – see *Database 2*.
- **DDL** – See *data definition language*.
- **DD statement** – see *data definition statement*.
- **default language** – the language that displays a user's prompts and messages.
- **destination-template transfer** – a transfer method in which source data is briefly put into a template where the user can specify that some action be performed on it before execution

against a target database. See also *transfer*. Compare with *bulk copy transfer* and *express transfer*.

- **direct access storage device** – a device in which access time is effectively independent of the location of the data.
- **direct request** – a request sent directly from a client workstation through Transaction Router Service to the DirectConnect server without going through Adaptive Server. Contrast with *indirect request*.
- **direct resolution** – a type of service name resolution that relies upon a client application specifying the exact name of the service to be used. See also *service name resolution*. Compare with *service name redirection*.
- **DirectConnect Manager** – a Java application from Sybase that can be used in Windows and UNIX environments. It provides remote management capabilities for DirectConnect products, including starting, stopping, creating, and copying services.
- **ECDA Option for Oracle server** – the component of Mainframe Connect DirectConnect for z/OS Option that provides general management and support functions to service libraries.
- **dll** – see *dynamic link library*.
- **DSN** – see *data set name*.
- **dynamic link library** – a file containing executable code and data bound to a program at load time or runtime, rather than during linking.
- **dynamic SQL** – the preparation and processing of SQL source statements within a program while the program runs. The SQL source statements are contained in host-language variables rather than being coded directly into the application program. Contrast with *static SQL*.
- **ECDA** – see *Enterprise Connect Data Access*.
- **ECDA Option for ODBC** – a Sybase solution that allows client applications to access ODBC data. It combines the functionality of the ECDA Option for ODBC architecture with ODBC to provide dynamic SQL access to target data, as well as the ability to support stored procedures and text and image pointers.
- **ECDA Option for Oracle** – a Sybase solution that provides Open Client access to Oracle databases. When used in combination with Adaptive Server, it provides many of the features of a distributed database system, such as location transparency, copy transparency, and distributed joins.
- **Embedded SQL™** – SQL statements that are embedded within a program and are prepared in the process before the program runs. After it is prepared, the statement itself does not change, although values of host variables specified within the statement might change.
- **end user** – a person who connects to a DirectConnect server using an application to access databases and perform transfers. See also *transfer*.
- **Enterprise Connect Data Access** – an integrated set of software applications and connectivity tools that allow access to data within a heterogeneous database environment,

such as a variety of LAN-based, non-Sybase datasources, as well as mainframe data sources.

- **environment variable** – a variable that describes how an operating system runs and the devices it recognizes.
- **exit routine** – a user-written routine that receives control at predefined user exit points.
- **express transfer** – a form of bulk copy transfer that uses ODBC bulk APIs to improve performance when transferring bulk data between datasources. Because it uses the same syntax as bulk copy transfer, no modification of applications is required.
- **external call interface** – a CICS client facility that allows a program to call a CICS application as if the calling program had been linked synchronously from a previous program instead of started from a terminal.
- **External Security Manager** – an add-on security package for the z/OS mainframe, licensed by Computer Associates.
- **FCT** – see *forms control table*.
- **forms control table** – an object that contains the special processing requirements for output data streams received from a host system by a remote session.
- **gateway** – connectivity software that allows two or more computer systems with different network architectures to communicate.
- **Gateway-Library** – a library of communication, conversion, tracing, and accounting functions supplied with Mainframe Connect Server Option.
- **globalization** – the combination of internationalization and localization. See *internationalization, localization*.
- **global variable** – a variable defined in one portion of a computer program and used in at least one other portion of the computer program. Contrast with *local variable*.
- **handler** – a routine that controls a program's reaction to specific external events, for example, an interrupt handler.
- **host** – the mainframe or other machine on which a database, an application, or a program resides. In TCP/IP, this is any system that is associated with at least one Internet address. See also *Transmission Control Protocol/Internet Protocol*.
- **host ID** – in Mainframe Connect Server Option, the ID that the TRS passes to the mainframe with a client request. The host ID is part of the client login definition at the TRS.
- **host password** – in Mainframe Connect Server Option, the password that the client passes to the mainframe with a client request.
- **host request library** – a DB2 UDB table that contains host-resident SQL statements that can be executed dynamically. See also *host-resident request*.
- **host-resident request** – a SQL request that resides in a DB2 UDB table called the host request library. See also *host request library*.
- **IMS** – see *Information Management System*.
- **indirect request** – a client request that is routed through a stored procedure on a SQL Server, which forwards the request to TRS as an RPC. Compare with *direct request*.

- **Information Management System** – a database/data communication system that can manage complex databases and networks.
- **interfaces file** – an operating system file that determines how the host client software connects to a Sybase product. An `interfaces` file entry contains the name of any ECDA Option for Oracle server and a list of services provided by that server.
- **internationalization** – the process of extracting locale-specific components from the source code and moving them into one or more separate modules, making the code culturally neutral so it can be localized for a specific culture. See also *globalization*. Compare with *localization*.
- **keyword** – a word or phrase reserved for exclusive use by Transact-SQL.
- **language RPC** – the name TRS uses to represent a client’s language request. TRS treats a language request as a remote procedure call (RPC) and maps it to a language transaction at the remote server.
- **language transaction** – the server transaction that processes client language requests. The Mainframe Connect DB2 UDB Option language transaction for CICS is **AMD2**, which uses the DB2 UDB dynamic SQL facilities to process incoming SQL strings. The Mainframe Connect DB2 UDB Option for IMS uses **SYRT** by default.
- **linkage** – in computer security, combining data or information from one information system with data or information from another system with the intention to derive additional information; for example, the combination of computer files from two or more sources.
- **linkage editor** – a computer program that creates load modules from one or more object modules or creates load modules by resolving cross references among the modules, and if necessary, adjusts those addresses.
- **link-edit** – to create a loadable computer program by using a linkage editor. See also *linkage editor*.
- **localization** – the process of preparing an extracted module for a target environment, in which messages are displayed and logged in the user’s language. Numbers, money, dates, and time are represented using the user’s cultural convention, and documents are displayed in the user’s language. See also *globalization*.
- **local variable** – a variable that is defined and used only in one specified portion of a computer program. Contrast with *global variable*.
- **logical unit** – a type of network addressable unit that enables a network user to gain access to network facilities and communicate remotely. A connection between a TRS and a CICS region is a connection between logical units.
- **logical unit 6.2** – a type of logical unit that supports general communication between programs in a distributed processing environment. See also *advanced program-to-program communication*.
- **login ID** – in Mainframe Connect Server Option, the ID that a client user uses to log in to the system.

- **login packet** – client information made available to Mainframe Connect Server Option. The client program sets this information in a login packet and sends it to TRS, which forwards it to the mainframe.
- **long-running transaction** – a transaction that accepts more than one client request. Whereas short transactions end the communication after returning results to a client, a long-running transaction can await and process another request. Compare with *short transaction*.
- **LU 6.2** – see *logical unit 6.2*.
- **mainframe access products** – Sybase products that enable client applications to communicate with mainframes in a client/server environment. See *client/server*.
- **Mainframe Connect** – the Sybase product set that provides access to mainframe data.
- **Mainframe Connect Client Option** – a Sybase product that, using Client-Library, allows mainframe clients to send requests to SQL Server, Open Server, the Mainframe Connect DB2 UDB Option and Mainframe Connect Server Option. Mainframe Connect Client Option provides capability for the mainframe to act as a client to LAN-based resources in the CICS or the IMS and MVS environment.
- **Mainframe Connect DB2 UDB Option** – a Sybase mainframe solution that provides dynamic access to DB2 UDB data. It is available in the CICS or IMS environment. See also *Customer Information Control System, Database 2, Multiple Virtual Storage*.
- **Mainframe Connect ECDA Option for Oracle for z/OS Option** – a Sybase Open Server application that provides access management for non-Sybase databases, copy management (transfer), and remote systems management.
- **Mainframe Connect Server Option** – a Sybase product that provides capability for programmatic access to mainframe data. It allows workstation-based clients to execute customer-written mainframe transactions remotely. It is available for the CICS and the IMS and MVS environments
- **Multiple Virtual Storage** – an IBM operating system that runs on most System/370 and System/390 mainframes. It supports 24-bit addressing up to 16 megabytes.
- **network protocol** – a set of rules governing the way computers communicate on a network.
- **nonclustered index** – an index that stores key values and pointers to data. Compare with *clustered index*.
- **null** – having no explicitly assigned value. NULL is not equivalent to 0 or to blank.
- **ODBC** – see *Open Database Connectivity*.
- **ODS** – see *Open Data Services*.
- **Open Client** – a Sybase product that provides customer applications, third-party products, and other Sybase products with the interfaces required to communicate with Open Client and Open Server applications.
- **Open Data Services** – a product that provides a framework for creating server applications that respond to DB-Library clients.
- **Open Database Connectivity** – a Microsoft API that allows access to both relational and non-relational databases. See also *application program interface*.

- **Open Server** – a Sybase product that provides the tools and interfaces required to create a custom server. Clients can route requests to the ECDA Option for Oracle server through an Open Server configured to meet specific needs, such as the preprocessing of SQL statements.
- **parameter** – a variable that is given a constant value for a specified application and can denote the application. Compare with *property*.
- **Partner Certification Reports** – Sybase publications that certify third-party or Sybase products to work with other Sybase products.
- **Password Expiration Management** – an IBM password management program with CICS Version 3.3 through an optional program temporary fix, and as an integral part of CICS with version 4.1 and higher.
- **PEM** – see *Password Expiration Management*.
- **PL/1** – see *Programming Language /1*.
- **primary database** – the database management system that the DirectConnect server is always connected to. It is implied in the **transfer** statement.
- **Programming Language/1** – a programming language designed for use in a wide range of commercial and scientific computer applications.
- **property** – a setting for a server or service that defines the characteristics of the service, such as how events are logged. Compare with *parameter*.
- **protocol** – the rules for requests and responses used to manage a network, transfer data, and synchronize the states of network components.
- **query** – a request for data from a database, based upon specified conditions.
- **Registry** – the part of the Windows operating system that holds configuration information for a particular machine.
- **relational database** – a database in which data is viewed as being stored in tables consisting of columns (data items) and rows (units of information).
- **relational operators** – operators supported in search conditions.
- **relops** – see *relational operators*.
- **remote procedure call** – a call to execute a stored procedure on a remote server. For Mainframe Connect Server Option, an RPC is a direct request from a client to TRS. For Mainframe Connect Client Option, a Client-Library transaction that calls a procedure on a remote server acts like an RPC.
- **remote stored procedure** – a customer-written CICS program using an API that resides on the mainframe and communicates with Mainframe Connect DB2 UDB Option. See also *Customer Information Control System, stored procedure*. Compare with *Client Services Application*.
- **remote systems management** – a feature that allows a system administrator to manage multiple DirectConnect servers and multiple services from a client.
- **Replication Server**[®] – a Sybase Adaptive Server application that maintains replicated data and processes data transactions received from a datasource.

- **request** – one or more database operations an application sends as a unit to the database. Depending upon the response, the application commits or rolls back the request. See also *commit*, *rollback*, *unit of work*.
- **resource table** – a main storage table that associates each resource identifier with an external logical unit (LU) or application program.
- **rollback** – an instruction to a database to back out of changes requested in a unit of work. Compare with *commit*.
- **router** – an attaching device that connects two LAN segments, which use similar or different architectures, at the Open System Interconnection (OSI) reference model network layer. Contrast with *gateway*.
- **RPC** – see *remote procedure call*.
- **RSP** – see *remote stored procedure*.
- **SAA** – see *System Application Architecture*.
- **secondary connection** – The connection specified in the **transfer** statement. It represents anything that can be accessed using Mainframe Connect Client Option, such as Adaptive Server or another access service.
- **secondary database** – in transfer processing, the supported database that is specified in the **transfer** statement. Compare with *primary database*.
- **server** – a functional unit that provides shared services to workstations over a network. See also *client/server*. Compare with *client*.
- **server process ID** – a positive integer that uniquely identifies a client connection to the server.
- **service** – a functionality available in Mainframe Connect DirectConnect for z/OS Option. It is the pairing of a service library and a set of specific configuration properties.
- **service library** – in Mainframe Connect DirectConnect for z/OS Option, a set of configuration properties that determine service functionality. See also *access service library*, *administrative service library*, *Transaction Router Service library*, *transfer service library*.
- **service name redirection** – a type of service name resolution that allows a system administrator to create an alternative mechanism to map connections with services. See also *service name resolution*. Compare with *direct resolution*.
- **service name redirection file** – the default name of the file used for the service name redirection feature. See *service name redirection*.
- **service name resolution** – the DirectConnect server mapping of an incoming service name to an actual service. See also *direct resolution*, *service name redirection*.
- **session** – a connection between two programs or processes. In APPC communications, sessions allow transaction programs to have conversations between the partner LUs. See also *advanced program-to-program communication*.
- **short transaction** – a mainframe transaction that ends the communication when it finishes returning results to the client. Compare with *long-running transaction*.
- **SNA** – see *Systems Network Architecture*.

- **SNRF** – see *service name redirection file*.
- **SPID** – see *server process ID*.
- **SQL** – see *structured query language*.
- **SQLDA** – see *SQL descriptor area*.
- **sqledit** – a utility for creating and editing `sql.ini` files and file entries.
- **sql.ini** – the interfaces file containing definitions for each ECDA Option for Oracle server to which a workstation can connect. The file must reside on every client machine that connects to Adaptive Server.
- **SQL descriptor area** – a set of variables used in the processing of SQL statements.
- **SQL stored procedure** – a single SQL statement that is statically bound to the database. See also *stored procedure*.
- **static SQL** – SQL statements that are embedded within a program and prepared during the program preparation process before the program runs. Compare with *dynamic SQL*.
- **stored procedure** – a collection of SQL statements and optional control-of-flow statements stored under a particular name. Adaptive Server stored procedures are called “system procedures.” See also *remote stored procedure*, *system procedures*.
- **structured query language** – an IBM industry-standard language for processing data in a relational database.
- **stub** – A program module that transfers remote procedure calls (RPCs) and responses between a client and a server.
- **SYRT** – the component of Mainframe Connect DB2 UDB for IMS that allows clients to submit SQL language requests to DB2 through IMS.
- **system administrator** – the person in charge of server system administration, including installing and maintaining DirectConnect servers and service libraries.
- **System Application Architecture** – an IBM proprietary plan for the logical structure, formats, protocols, and operational sequences for transmitting information units through networks and controlling network configuration and operation. See also *advanced program-to-program communication*.
- **system procedures** – a stored procedure that Adaptive Server supplies for use in system administration. System procedures serve as shortcuts for retrieving information from system tables, or a mechanism for accomplishing database administration. See also *stored procedure*.
- **Systems Network Architecture** – an IBM proprietary plan for the structure, formats, protocols, and operational sequences for transmitting information units through networks. See also *advanced program-to-program communication*.
- **table** – an array of data or a named data object that contains a specific number of unordered rows. Each item in a row can be unambiguously identified by means of one or more arguments.
- **Tabular Data Stream**[™] – a Sybase application-level protocol that defines the form and content of relational database requests and replies.

- **target** – a system, program, or device that interprets, rejects, satisfies, or replies to requests received from a source.
- **target database** – the database to which the DirectConnect server transfers data or performs operations on specific data.
- **TCP/IP** – see *Transmission Control Protocol/Internet Protocol*.
- **TDS** – see *Tabular Data Stream*.
- **transaction** – a unit of processing initiated by a single request. A transaction consists of one or more application programs that, when executed, accomplish a particular action. In Mainframe Connect Server Option, a client request (RPC or language request) invokes a mainframe transaction. In Mainframe Connect Client Option, a mainframe transaction executes a stored procedure on a remote server.
- **transaction processing** – a sequence of operations on a database that is viewed by the user as a single, individual operation.
- **Transaction Router Service** – a Mainframe Connect DirectConnect for z/OS Option program used when the mainframe acts as a transaction server to route requests from remote clients to the Mainframe Connect Server Option and return results to the clients.
- **Transaction Router Service library** – a service library that facilitates access to remote transactions, allowing customers to execute transactions from virtually any mainframe datasource. See also *service library*.
- **Transact-SQL[®]** – a Sybase-enhanced version of the SQL database language used to communicate with Adaptive Server.
- **transfer** – a Mainframe Connect DirectConnect for z/OS Option feature that allows users to move data or copies of data from one database to another.
- **transfer service library** – a service library that provides copy management functionality. See also *service library*.
- **Transmission Control Protocol/Internet Protocol** – a set of communication protocols that supports peer-to-peer connectivity functions for both local and wide area networks.
- **trigger** – A form of stored procedure that automatically executes when a user issues a change statement to a specified table.
- **TRS** – see *Transaction Router Service*.
- **TRS library** – see *Transaction Router Service library*.
- **unit of work** – one or more database operations grouped under a commit or rollback. A unit of work ends when the application commits or rolls back a series of requests, or when the application terminates. See also *commit, rollback, transaction*.
- **user ID** – user identification. The ID number by which a user is known in a specific database or system.
- **variable** – an entity that is assigned a value. Mainframe Connect ECDA Option for Oracle for z/OS Option has two kinds of variables: *local* and *global*.
- **view** – an alternate representation of data from one or more tables. A view can include all or some of the columns contained the table or tables on which it is defined.

- **Virtual Storage Access Method** – an IBM-licensed program that controls communication and the flow of data in an SNA network.
- **Virtual Telecommunications Access Method** – IBM mainframe software that allows communication on an SNA network between mainframes and allows the mainframe to have multiple sessions per connection.
- **VSAM** – see *Virtual Storage Access Method*.
- **VTAM** – see *Virtual Telecommunications Access Method*.
- **wildcard** – a special character that represents a range of characters in a search pattern.

Obtaining Help and Additional Information

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

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

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

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

Technical Support

Get support for Sybase products.

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

Downloading Sybase EBFs and Maintenance Reports

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

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

Obtaining Help and Additional Information

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

- f) Click the **Info** icon to display the EBF/Maintenance report, or click the product description to download the software.
- If you ordered your Sybase product under an SAP contract:
 - a) Point your browser to <http://service.sap.com/swdc> and log in if prompted.
 - b) Select **Search for Software Downloads** and enter the name of your product. Click **Search**.

Sybase Product and Component Certifications

Certification reports verify Sybase product performance on a particular platform.

To find the latest information about certifications:

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

Creating a MySybase Profile

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

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

Accessibility Features

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

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

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

Sybase HTML documentation has been tested for compliance with accessibility requirements of Section 508 of the U.S Rehabilitation Act. Documents that comply with Section 508 generally also meet non-U.S. accessibility guidelines, such as the World Wide Web Consortium (W3C) guidelines for Web sites.

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

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

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

Index

A

- Adaptive Server Enterprise/Component Integration
 - Services feature (ASE/CIS) 4
- AddServer utility
 - interfaces, create entries 42
- Advanced Program-to-Program Communications (APPC) 55

B

- bind parameters
 - Bind utility 67
 - bind20 utility 67
 - DSN 67
 - ECDA Option for ODBC 67
- Bind utility
 - create connection 67
 - create packages 67
 - Distributed Relational Database Architecture (DRDA) 67

C

- CCSID (coded character set ID) 57
 - change at user profile level 57
- client applications
 - Open Client 4
 - Open Server 4
 - Oracle SQL 4
- client connectivity
 - configuring 45
 - DirectConnect server 45
 - local area network (LAN) 45
- command shell
 - db2 27
- configuration checklist
 - holding the preliminary planning session 10
- configuration properties
 - ODBCDriverManager 48, 49
 - SQLODBCursors 48
- configure
 - data source name (DSN) 63
- configure data source
 - advanced parameters 65

- bind parameters 66
 - ECDA Option for ODBC 64–66, 69
 - general parameters 64
 - ODBC drivers 64–66
 - configuring
 - client connectivity 45
 - connectivity
 - access service 55
 - change security settings 56
 - connect to Microsoft SQL Server 69
 - ECDA Option for ODBC 55, 63, 69
 - ODBC drivers 63, 69
 - preparing for ECDA installation 10
 - security levels 55, 56
 - TCP/IP 55
 - verify settings 56
 - connectivity parameters
 - data source name 10
 - console mode
 - ECDA 17
 - installing ECDA 15
 - response file, installing ECDA with 17
 - uninstalling ECDA 24
 - conventions
 - style 1
 - syntax 1
 - create server
 - DataDirect driver 26
 - ECDA Option for ODBC 25
 - IBM DB2 driver 27
 - Oracle-accessible target database 28
 - unixODBC driver manager 25
 - Create Server wizard
 - DirectConnect server 25–28
 - ECDA Option for ODBC 25–27
 - ECDA Option for Oracle 25, 28
 - creating
 - response files for ECDA 16
 - CRTSRCPF 59
 - custom installation 13
- ## D
- data source name (DSN) 10, 67
 - ConnectionSpec1 25–28
 - DB2 27

Index

- DCDirector
 - create and start 41
 - utility 41
- diagnostic tools
 - ivtestlib 54
 - odbct 53
- DirectConnect Director (DCDirector) 39
- DirectConnect Manager 33
 - configure servers and access services 40
 - create an access service 39
 - create server 39
 - features 5
 - manage an access service 39
 - manage server 39
 - online help 39
 - start an access service 39
 - start server 39
 - stop an access service 39
- DirectConnect server
 - AddServer utility 31, 32
 - installation 61
 - starting using DCStart 42
- Distributed Relational Database Architecture (DRDA) 3, 55, 67

E

- ECDA
 - configuring client connectivity 45
 - in console mode 11
 - in GUI mode 11, 13
 - in silent mode 11
 - installing using a response file 16
 - uninstalling, prerequisites 23
 - with response files 11
- ECDA Option for ODBC
 - access service library 3
 - basic connectivity 3
 - DataDirect driver 25
 - description 3
 - DirectConnect 3
 - IBM DB2 driver 25
 - installation worksheets 61
 - ODBC drivers 3
 - Sybase-provided unixODBC driver manager 25
- ECDA Option for Oracle
 - installation worksheets 61
 - sp_configure 35
- ECDA Options
 - installation worksheets 61

- ECDA utilities
 - ECDA Option for ODBC 31
 - ECDA Option for Oracle 31
- Enterprise Connect Data Access (ECDA) 3
 - DirectConnect Manager 4
 - ECDA Option for Oracle 4
 - used with Adaptive Server 4
- environment variables
 - \$ODBCSYSINI 48, 49
 - \$SYBASE 32
 - \$SYBASE_ECON 32
 - \$SYBASE_OCS 32
 - DC_SYBASE.csh 32, 34, 35, 53
 - DC_SYBASE.sh 32, 34, 35
 - LD_LIBRARY_PATH 49
 - LIBPATH 49
 - library path 32
 - ODBCINI 63, 69
 - SHLIB_PATH 49
- error messages
 - checking server log and trace files 53
 - confirming the Open Server software release number 53
 - prelog 53
- errors
 - while starting installer 13

F

- files
 - .cfg 33
 - dcany.cfg 25, 26, 48, 55
 - DCOConfig 36
 - interfaces 33, 34, 45
 - libtcl.cfg 45
 - odbc.ini 25, 26, 48, 49, 63–67, 69
 - odbcinst.ini 49
 - sql.ini 45
 - tnsnames.ora 28, 36, 61
- full installation 13

G

- GUI mode
 - installing ECDA 13
 - uninstalling ECDA 23

I

installation

- DirectConnect Manager 19
- GUI mode 13
- preparing the environment 10
- role and responsibility 9
- skills 9
- Sybase Central 19
- worksheets 61

Installation

- task flow 7

installer

- custom installation 13
- errors in starting 13
- full installation 13

installing ECDA

- in console mode 15
- using a response file 16

interactive mode

- response file, installing ECDA with 11

ivtestlib tool 54

L

licenses

- grace period 11
- obtaining 11

log and trace files

- postinstallation troubleshooting 53

M

manage service

- DirectConnect Manager 31
- ECDA Option for ODBC utility 31
- ECDA Option for Oracle utility 31

methods of installation

- ECDA 13

O

ODBC driver

- install and configure 3

ODBC driver manager

- configuration properties 48

ODBC-specific problems

- format 51, 52
- ODBC data source errors 52

ODBC driver manager errors 52

ODBC drivers errors 51

ODBCDriverManager 48, 49

odbct tool 53

operating system requirements

- release levels 10

P

preinstallation

- checklist for HP 9000(8xx) 61
- verify previously installed Sybase products 12

prelog error messages

- explanation 53

R

response files

- ECDA, creating for 16
- installation 16–18
- installing with, ECDA 11

S

SAP Service Marketplace (SMP) 11, 13

server

- DirectConnect 42
- DirectConnect Director (DCDirector) 39, 41

setting up

- connectivity to the target database 10

silent mode installations

- ECDA 11, 18
- response file, installing ECDA with 18

SQLODBCursors 48

starting

- DirectConnect server using DCStart 42

supported back-end drivers

- considerations 47
- DataDirect Driver for DB2 47
- IBM DB2 CLI ODBC 47
- Microsoft SQL Server 47
- unixODBC driver manager 47–49

Sybase Central

- install 19
- register 19

SYBASE collection

- authorize users 60
- building query 59

Index

- CCSID (coded character set ID) 59, 60
 - creating 60
 - creating, use SEU utility 59
 - ECDA for ODBC access service 57
 - grant permission 58
 - set authorization 60
 - set authorization level 60
 - verify authorization level 60
- Sybase Product Download Center (SPDC) 11, 13
- Sybase Software Asset Management System (SySAM) 11
- SySAM 13
 - licensing requirements 11
 - system requirements 10
- Systems Network Architecture (SNA) 55

T

- troubleshooting
 - configuration problems 54
 - connectivity problems 53
 - diagnostic tools 51, 53, 54
 - ECDA problems 53
 - installation 51
 - ODBC data source errors 52
 - ODBC driver manager errors 52
 - ODBC drivers errors 51
 - ODBC-specific problems 51
- types of installations
 - custom 13

- full 13
- standard 13

U

- uninstalling
 - in console mode, ECDA 24
 - in GUI mode, ECDA 23
- upgrading ECDA server 21
- utilities
 - AddServer 31, 32, 42
 - Bind 67
 - bind20 67
 - DCDirector 41, 42
 - DCOConfig 36
 - DCStart 42
 - dsedit 33, 34, 45
 - isql 33, 34, 45, 57, 58
 - odbct 64
 - SEU 58, 59
 - sp_addservice 36
 - sp_configure 35
 - sp_dropservice 36
 - sp_shutdown 37
 - stopsvr 34, 35

V

- verify security settings 56