



Server Administration and Users Guide

Enterprise Connect™ Data Access Option for Oracle

15.0

[Microsoft Windows, Linux, and UNIX]

DOCUMENT ID: DC34201-01-1500-01

LAST REVISED: August 2007

Copyright © 1991-2007 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.

To order additional documents, U.S. and Canadian customers should call Customer Fulfillment at (800) 685-8225, fax (617) 229-9845.

Customers in other countries with a U.S. license agreement may contact Customer Fulfillment via the above fax number. All other international customers should contact their Sybase subsidiary or local distributor. 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.

Java and all Java-based marks are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries.

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

All other company and product names used herein may be trademarks or registered trademarks of their respective companies.

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

About This Book	v	
CHAPTER 1	Introduction	1
	Overview	1
	Architecture	2
CHAPTER 2	Configuring and Operating ECDA Option for Oracle	5
	Configuration file	5
	Changing the configuration	6
	Data compatibility	7
	Setting environment variables	7
	Oracle connectivity	8
	interfaces file (UNIX only).....	10
	sql.ini file (Windows only)	10
	Adding a service.....	10
	Removing a service.....	11
	Starting ECDA Option for Oracle on UNIX	12
	Creating the runserver file	12
	Starting ECDA Option for Oracle on Windows	12
	Installing a DirectConnect server as a Windows service.....	13
	Shutting down ECDA Option for Oracle	15
	Configuring ASE for ECDA Option for Oracle	16
	Oracle server definition	16
	Displaying the version	16
	Accessing ECDA Option for Oracle using isql.....	17
CHAPTER 3	Executing Remote Procedure Calls	19
	RPCs passed to Oracle.....	19
	Using stored procedures with named parameters.....	20
	Create stored procedures.....	20
	RPCs processed by ECDA Option for Oracle	24

CHAPTER 4	Secure Sockets Layer (SSL)	41
	Secure Sockets Layer	41
	SSL filter.....	41
	Server certificate	43
	Configuring SSL between ECDA Option for Oracle	45
	Connecting SSL between ECDA Option for Oracle and the client ..	45
	Using certreq and certauth utilities.....	45
APPENDIX A	Configuration Parameters	49
	Types of configuration parameters.....	49
	Global configuration parameters	49
	Service-specific configuration parameters	54
APPENDIX B	Reference Topics for Oracle	63
	Connect handler for ECDA Option for Oracle	63
	SQL command handling.....	63
	Support for text and image datatypes	64
	Selecting text and image	64
	Updating data using writetext	64
	Localization and internationalization	65
	Language and character set settings	65
	Default character set	65
	Blank-padded comparisons.....	66
	Passthrough mode	66
	Datatype conversion	68
	Special datatype considerations.....	69
	Global variables	73
	Set commands	74
	set autocommit.....	74
	set rowcount.....	74
	set textsize	74
	set sql_transform.....	74
	set traceon.....	75
	set traceoff.....	75
	Two-phase commit.....	75
	Lightweight Directory Access Protocol (LDAP)	76
	Setting up LDAP for UNIX.....	76
	Setting up LDAP for Windows.....	77
	Glossary	79
	Index	95

About This Book

This book describes the features and functionality of the Enterprise Connect Data Access™ (ECDA) Option for Oracle and how to use it with Sybase® Adaptive Server® Enterprise.

Audience

This book is for System Administrators who are configuring the ECDA Option for Oracle and for users who are accessing remote data.

How to use this book

This book includes the following chapters:

- Chapter 1, “Introduction,” provides an overview of ECDA Option for Oracle.
- Chapter 2, “Configuring and Operating ECDA Option for Oracle,” describes the components and describes how to use ECDA Option for Oracle.
- Chapter 3, “Executing Remote Procedure Calls,” describes remote procedure calls unique to ECDA Option for Oracle.
- Chapter 4, “Secure Sockets Layer (SSL),” describes the steps required for SSL connectivity to ECDA Option for Oracle from the isql client.
- Appendix A, “Configuration Parameters,” lists the global and service-specific parameters.
- Appendix B, “Reference Topics for Oracle,” contains reference material for ECDA Option for Oracle.

Related documents

For information regarding the installation of ECDA Option for Oracle, use the following guides:

- Enterprise Connect Data Access *Installation Guide* for Microsoft Windows
- Enterprise Connect Data Access *Installation Guide* for UNIX

Other sources of information

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

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

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

Refer to the *SyBooks Installation Guide* on the Getting Started CD, or the *README.txt* file on the SyBooks CD for instructions on installing and starting SyBooks.

- The Sybase Product Manuals Web site is an online version of the SyBooks CD that you can access using a standard Web browser. In addition to product manuals, you will find links to EBFs/Maintenance, Technical Documents, Case Management, Solved Cases, newsgroups, and the Sybase Developer Network.

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

Sybase certifications on the Web

Technical documentation at the Sybase Web site is updated frequently.

❖ Finding the latest information on product certifications

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

❖ Finding the latest information on component certifications

- 1 Point your Web browser to Availability and Certification Reports at <http://certification.sybase.com/>.
- 2 Either select the product family and product under Search by Base Product; or select the platform and product under Search by Platform.

- 3 Select Search to display the availability and certification report for the selection.

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

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

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

Sybase EBFs and software maintenance

❖ **Finding the latest information on EBFs and software maintenance**

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

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

- 5 Click the Info icon to display the EBF/Maintenance report, or click the product description to download the software.

Conventions

For readability, directory paths in this manual are in UNIX format. On Windows, substitute */SYBASE* with *<install_dir>*, and replace slashes (/) with back slashes (\). For example, replace this user input:

```
$SYBASE/$SYBASE_DCO/install
```

with:

```
C:\<install_dir>\DCO-15_0\install
```

SQL is a free-form language: There are no rules about the number of words you can put on a line or where you must break a line. However, for readability, all examples and syntax statements in this manual are formatted so that each clause of a statement begins on a new line. Clauses that have more than one part extend to additional lines, which are indented.

The conventions for syntax statements in this manual are as follows:

Table 1: Syntax statement conventions

Key	Definition
command	Command names, command option names, utility names, utility flags, and other keywords are in bold Courier in syntax statements and in bold Helvetica in paragraph text.
<i>variable</i>	Variables, or words that stand for values that you fill in, are in <i>italics</i> .
{ }	Curly braces indicate that you choose at least one of the enclosed options. Do not include braces in your option.
[]	Brackets mean choosing one or more of the enclosed options is optional. Do not include brackets in your option.
()	Parentheses are to be typed as part of the command.
	The vertical bar means you may select only one of the options shown.
,	The comma means you may choose as many of the options shown as you like, separating your choices with commas to be typed as part of the command.

Syntax statements (displaying the syntax and all options for a command) are printed like this:

```
sp_dropdevice [device_name]
```

or, for a command with more options:

```
select column_name
from table_name
where search_conditions
```

In syntax statements, keywords (commands) are in normal font and identifiers are in lowercase: normal font for keywords, italics for user-supplied words.

Examples showing the use of Transact-SQL™ commands are printed like this:

```
select * from publishers
```

In this manual, most of the examples are in lowercase. However, you can disregard case when typing Transact-SQL keywords. For example, SELECT, Select, and select are the same.

Accessibility features

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

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

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

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

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

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

If you need help

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



Topic	Page
Overview	1
Architecture	2

Overview

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

When used with ASE, ECDA Option for Oracle transforms the Transact-SQL generated by ASE to Oracle's 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 in combination with ASE. This combination enables location transparency, distributed query optimization, copy transparency, transaction transparency, and distributed joins.

When used with ASE, you can join Oracle tables with ASE, DB2, or other tables. Access to these objects through ASE 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.

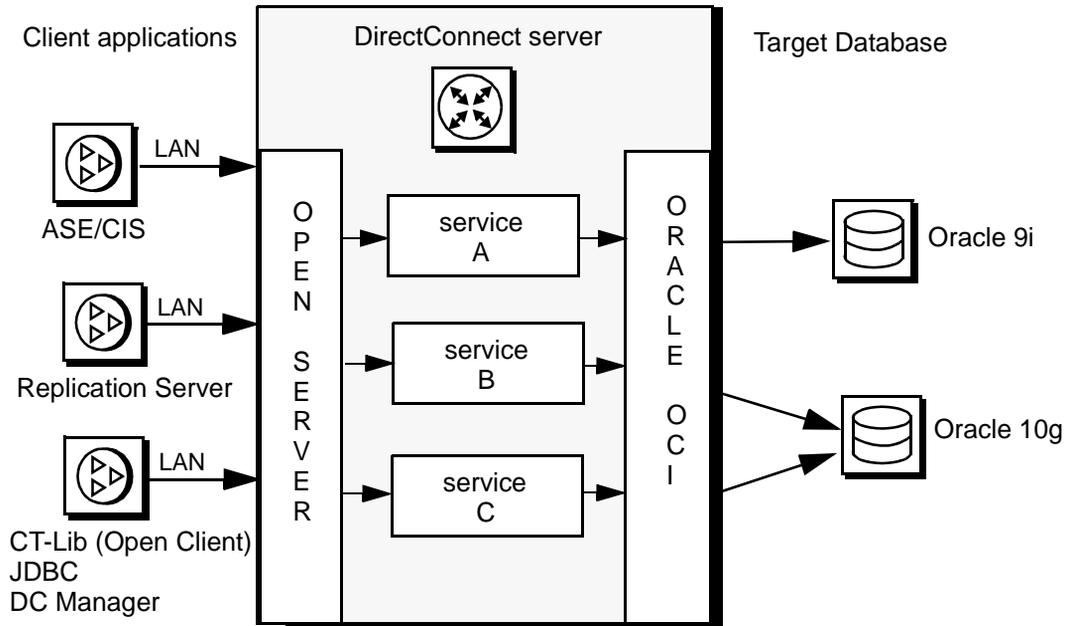
Architecture

Before you configure ECDA Option for Oracle, you should have a basic understanding of how it works:

- ECDA Option for Oracle is a Symmetric Multiprocessing (SMP) Sybase Open Server application that uses an Open Server thread for each incoming client connection.
- It uses the DirectConnect server to route the incoming client connection to the correct service based on the service name and has one Oracle process for each DirectConnect server.
- It does not have a service library but can have multiple services and each service is configured to connect to a different Oracle database. In addition, the DirectConnect server performs SQL transformation and some data translation based on the configuration of the service.
- It accesses Oracle databases using the Oracle Call Interface (OCI) application programming interface (API). The OCI allows the ECDA Option for Oracle the ability to manipulate data in an Oracle database.
- It communicates with ASE using the server class *direct_connect*.

Figure 1-1 shows how ECDA Option for Oracle works with ASE, Replication Server and other APIs.

Figure 1-1: ECDA Option for Oracle



As shown in Figure 1-1, the request from a client uses a TCP/IP LAN connection to the DirectConnect server. Based on the incoming server name, the DirectConnect server routes the connection using OCI to the proper Oracle database depending on the configuration properties.

Configuring and Operating ECDA Option for Oracle

Topic	Page
Configuration file	5
Adding a service	10
Removing a service	11
Starting ECDA Option for Oracle on UNIX	12
Starting ECDA Option for Oracle on Windows	12
Installing a DirectConnect server as a Windows service	13
Shutting down ECDA Option for Oracle	15
Configuring ASE for ECDA Option for Oracle	16
Displaying the version	16
Accessing ECDA Option for Oracle using isql	17

Configuration file

When ECDA Option for Oracle is installed for the first time, a configuration file is automatically created for it. The configuration file is a plain text file that contains definitions for ECDA Option for Oracle. It is located in `<install_dir>/DCO-15_0/install` for UNIX, or `C:\<install_dir>\DCO-15_0\install` for Windows.

The configuration file must have the same name as the ECDA Option for Oracle server name as defined in the *interfaces* file for UNIX, or the *sql.ini* file for Windows. For example, if the server name of the ECDA Option for Oracle in the *interfaces* file is defined as ORACLEDC, the configuration file name will be *ORACLEDC.cfg*. The name of the configuration file is case sensitive.

Description of configuration file contents

The ECDA Option for Oracle configuration file contains:

- All of the configuration information for a particular server.

Configuration file characteristics

- An instance of a ECDA Option for Oracle that can be configured to respond to multiple service names. A service name represents a connection route with a specific target database and a set of attributes defined with service-specific parameters. This allows a single instance of the ECDA Option for Oracle to have different attributes or targets.
- A single configuration file for an instance of ECDA Option for Oracle can define one or more services.
- One of the service names must be the same as the server name you selected. All global parameters are defined under the server name heading which is also the name of your configuration file.
- In the configuration file, lines beginning with “;” are comments and are ignored by the server.
- A service name is enclosed in square brackets.

Changing the configuration

You can change the configuration of the ECDA Option for Oracle by modifying the configuration file. You can modify the configuration file using DirectConnect Manager or by using the command line.

Using DirectConnect Manager

For instructions on how to use DirectConnect Manager to modify the configuration file go to the DirectConnect Manager online Help, and select Managing Access Service | Modifying ECDA Option for Oracle Service Configuration Properties.

Note When a ECDA Option for Oracle server is highlighted, the details panel *Oracle Target* field may show "Unable to make connection" for a service, which indicates that DirectConnect Manager could not connect to the target Oracle database. This may be due to the ECDA Option for Oracle administrator account not being valid on the target Oracle database or possibly a different password. If multiple services are configured in one ECDA Option for Oracle server and there are multiple target Oracle databases involved, Sybase recommends you verify that the ECDA Option for Oracle administrator is a valid user on each of the target Oracle databases and they all have the same password.

Using the command line

A configuration file is created automatically at the time of installation. If you need to display or edit the configuration options after installation, see “sp_configure” on page 27.

Global and service-specific configuration parameters

The global parameters that apply to all services are defined in the Appendix, “Configuration Parameters.”

Service-specific parameters

The parameters that are not global can be specified for any service. If a service-specific keyword is not defined, the default value applies. The service-specific parameters are defined in the Appendix, “Configuration Parameters.” These parameters are used with “sp_configure” on page 27.

Data compatibility

For optimum compatibility with ASE, use the default value of 2 for *number_mode*.

Other combinations of these parameters can be used to change the operation of conversions of these datatypes.

Setting environment variables

To run ECDA Option for Oracle, you must set the following environment variables:

- SYBASE
- SYBASE_OCS
- SYBASE_DCO

These Sybase-specific environment variables are set in the *SYBASE.csh* file in UNIX. You must source this file. In the Windows environment, variables have already been defined in the environment, or you can use the *C:\<install_dir>\DCO-15_0\DCO_SYBASE.bat* file.

Environment variables can be set in an MS-DOS window before starting ECDA Option for Oracle. They can also be set from the System menu in the Windows Control Panel. If ECDA Option for Oracle is running as a service, set these variables as System Environment Variables.

ORA_NLS33 environment variable issue

The ORA_NLS33 environment variable is used by Oracle client programs to locate localization files. When running ECDA Option for Oracle, the ORA_NLS33 environment variable should *not* be set.

If it is set and it points to an invalid directory, users might receive the following error message when connecting to ECDA Option for Oracle:

```
Msg 12705, Level 16, State 0:
Server 'DCRMFDCP_TEST':
ORA-12705: invalid or unknown NLS parameter value specified
CT-LIBRARY error:
ct_connect(): protocol specific layer: external error: The
attempt to connect to the server failed.
```

Oracle connectivity

Oracle connectivity requires special attention to the settings in three different files: *listener.ora*, *sqlnet.ora*, and *tnsnames.ora*.

listener.ora file

the *listener.ora* file is the server-side definition file for the Oracle database listener. This file contains instructions for listening protocols. SID descriptions can be set using two different methods:

- An SID setting
- A GLOBAL_DBNAME setting

This example illustrates entries in a typical *listener.ora* file:

```
SID_LIST_LISTENER =
  (SID_LIST =
    (SID_DESC =
      (SID_NAME = PLSExtProc)
      (ORACLE_HOME = /work/oracle901)
      (PROGRAM = extproc)
    )
  )
  (SID_DESC =
    (GLOBAL_DBNAME = ORA9i.foo.com)
```

```
(ORACLE_HOME = /work/oracle901)
(SID_NAME = ORA9i))
```

sqlnet.ora file The *sqlnet.ora* file defines Oracle network configuration properties. It controls logging and tracing as well as whether domain names are required on the connection.

tnsnames.ora file The *tnsnames.ora* file is equivalent to the Sybase interfaces file. It contains network configuration information for ECDA Option for Oracle.

These three files are related, and entries in *tnsnames.ora* file will depend on the entries in *listener.ora* and *sqlnet.ora* files. The following examples illustrate entries in the *tnsnames.ora* file:

Example 1 If the *listener.ora* file is configured for TCP and is defined with only a SID setting, the *tnsnames.ora* file might take the following form:

```
ORA9I =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP) (HOST =
        oradb_box) (PORT = 1521))
    )
    (CONNECT_DATA =
      (SID = ORA9ISID)
    )
  )
```

Example 2 If the *listener.ora* file is configured for TCP and the listener is defined with a GLOBAL_DBNAME setting, you can use the previous example or the following example, which uses GLOBAL_DBNAME:

```
ORA9I =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP) (HOST =
        oradb_box) (PORT = 1521))
    )
    (CONNECT_DATA =
      (SERVICE_NAME = ORA9ISID.foo.com)
    )
  )
```

Example 3 If the *sqlnet.ora* file has the entry NAMES.DEFAULT_DOMAIN=foo.com, you will need to include the domain suffix in the connection name as shown:

```
ORA9I.FOO.COM =
  (DESCRIPTION =
```

```
(ADDRESS_LIST =
  (ADDRESS = (PROTOCOL = TCP) (HOST =
    oradb_box) (PORT = 1521))
)
(CONNECT_DATA =
  (SERVICE_NAME = ORA9I.foo.com)
)
)
```

interfaces file (UNIX only)

A default *interfaces* file is automatically created in the *<install_dir>/* directory at the time of installation. To edit the *interfaces* file, use *dsedit* found in *<install_dir>/DCO-15_0/bin*.

sql.ini file (Windows only)

A file name *sql.ini* must be present in the *C:\<install_dir>\DCO-15_0* directory. This file contains server entries for each server in the network known to your system.

To modify the *sql.ini* file for a new server, use the *dsedit* utility that is supplied with Open Client and Open Server. Using *dsedit*, enter the new server name, the TCP/IP host and port number.

For more information about *dsedit*, see the Open Client and Open Server documentation.

Note For the server to start, the server name and the configuration file name must be the same.

Adding a service

To use an additional Oracle database, you must add a new service. To add a new service, you can use DirectConnect Manager or you can use *sp_addservice*.

Using DirectConnect Manager For instructions on how to use DirectConnect Manager to add a new service, go to the DirectConnect Manager online Help, and select Managing Access Service | Creating a new ECDA Option for Oracle Service.

Using `sp_addservice` If you are connected directly to ECDA Option for Oracle, use the following syntax:

```
exec sp_addservice service_name
```

or, if you are connected through Adaptive Server:

```
DCServer... sp_addservice service_name
```

These methods add a new service to the ECDA Option for Oracle and write the service entry into the configuration file.

Note The *interfaces* file must contain both the `service_name` and the correct port number. Each service requires an entry in the *interfaces* file. The port numbers must be the same.

Removing a service

To remove a service, you can use DirectConnect Manager or you can use `sp_dropservice`.

Using DirectConnect Manager For instructions on how to use DirectConnect Manager to remove a service, go to the DirectConnect Manager online Help, and select Managing Access Service | Removing a Service.

Using `sp_dropservice` If you are connected directly to ECDA Option for Oracle, use the following syntax:

```
exec sp_dropservice service_name
```

or, if you are connected through Adaptive Server:

```
DCServer... sp_dropservice service_name
```

These commands remove an existing service from the ECDA Option for Oracle and delete the service entry from the configuration file.

Starting ECDA Option for Oracle on UNIX

Use the `startserver` utility to start ECDA Option for Oracle on UNIX. The `startserver` utility is included in the `install` directory of Adaptive Server. For example, to start a ECDA Option for Oracle named `DCSERVER`, enter:

```
startserver -f RUN_DCSERVER
```

where the `-f` flag specifies the relative path to the `runserver` file. After you issue the command, the ECDA Option for Oracle issues a series of messages describing the settings of the configuration properties.

Creating the runserver file

The `runserver` file is created automatically by the installer. It contains start-up commands for ECDA Option for Oracle. The `runserver` file can include the flags shown in the following table:

Table 2-1: Definition of flags in the runserver file

Flag	Definition
<code>-Sserver_name</code>	Specifies the name of the ECDA Option for Oracle. It is used to locate the configuration file and the network connection information in the <code>interfaces</code> file.
<code>-t</code>	Causes ECDA Option for Oracle to write start-up messages to standard error.
<code>-lerrorlog_path</code>	Specifies the path to the <code>error log</code> file.
<code>-iinterfaces_file_path</code>	Specifies the path to the <code>interfaces</code> file.

Starting ECDA Option for Oracle on Windows

You can start the ECDA Option for Oracle as a service, or from the command line:

- As a service – see “Installing a DirectConnect server as a Windows service” on page 13.
- From the command line – use the following syntax:

```
C:\<install_dir>\DCO-15_0\bin\dcoracle.exe -  
Sserver_name
```

```
[-t] [-i%SYBASE%path_to_sql.ini_file]
[-l%SYBASE%path_to_errorlog]
```

where:

- -S is the name of ECDA Option for Oracle you are starting.
- -t directs start-up messages to standard error.
- -i is the path to the *sql.ini* file.
- -l is the path to the error log.

For example, to start a ECDA Option for Oracle named “DCSERVER” on Windows using the default *sql.ini* and error log files, and using -t to trace start-up messages, enter:

```
C:\<install_dir>\DCO-15_0\bin
\dcoracle.exe -SDCSERVER -t
```

The ECDA Option for Oracle is up and running when the “start-up complete” message is displayed.

Installing a DirectConnect server as a Windows service

ECDA Option for Oracle installation does not automatically create the DirectConnect server as a Windows service. This section describes how to add, start, stop, and remove the DirectConnect server as a Windows service.

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

Adding the
DirectConnect server
as a Windows service

To add the DirectConnect server as a Windows service use the ServiceWrapper utility.

❖ **To add the DirectConnect server as a Windows service**

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

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

- 2 Execute the ServiceWrapper utility:

```
ServiceWrapper.exe --install <service_name> --user=<userid>
--password<password> C:\<install_dir>\DCO-15_0\INSTALL
\RUN_<DCO_NAME>.cmd
```

where:

- *service_name* is the Windows service name
- *userid* is the user name for the service that will run
- *password* is the password for the *userid*
- *DCO_NAME* is the name of the DirectConnect server

For example:

```
ServiceWrapper.exe
--install dcw2ksrv
--password=password
C:\<install_dir>\DCO-15_0\INSTALL\RUN_dcw2ksrvid.cmd
```

Starting the
DirectConnect server
as a Windows service

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

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

❖ To start DirectConnect server as a Windows service

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

Stopping
DirectConnect server
as a Windows service

❖ **To stop the DirectConnect server as a Windows service**

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

Removing the
DirectConnect server
as a Windows service

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

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

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

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

- 2 Execute the ServiceWrapper utility:

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

where *service_name* is the Windows service name.

Shutting down ECDA Option for Oracle

Use the following command to shut down the ECDA Option for Oracle from ASE:

```
server_name...sp_shutdown
```

where *server_name* is the name of the ECDA Option for Oracle you are shutting down.

For example, to shut down a ECDA Option for Oracle named DCSERVER, enter:

```
DCSERVER...sp_shutdown
```

Configuring ASE for ECDA Option for Oracle

Before using ECDA Option for Oracle and ASE to access remote data, you must define the remote server to ASE. For more information on configuring ASE, refer to ASE/CIS documentation.

Oracle server definition

Use the system stored procedure `sp_addserver` to define the service, server class, and network access information for ECDA Option for Oracle. The server class is `direct_connect`. Use `sp_addserver` to add an entry for each service.

Parameters to `sp_addserver` are defined as follows:

- *server_name* – the name of the server as it is known to ASE in the `syssservers` table.
- *server_class* – the class to use when defining a ECDA Option for Oracle server is `direct_connect`.
- *network_name* – the value of this parameter represents the name in the *interfaces* file associated with this server. The `network_name` parameter defaults to the `server_name`.

See the following example:

```
sp_addserver ORACLEDC, direct_connect, ORACLEDC
```

Displaying the version

To locate the version of ECDA Option for Oracle, change directories to `<install_dir>\DCO-15_0\bin` for UNIX, or `C:\<install_dir>\DCO-15_0\bin` for Windows. Then execute one of the following commands:

For UNIX:

```
dcoracle -v
```

For Windows:

```
dcoracle.exe -v
```

This prints the version string and exits.

Accessing ECDA Option for Oracle using isql

When accessing ECDA Option for Oracle with isql you must identify the service name with the -S option using the following syntax:

```
isql -User_name -Ppassword -Sservice_name
```

Note The user must be a valid Oracle user.

Executing Remote Procedure Calls

Topic	Page
RPCs passed to Oracle	19
RPCs processed by ECDA Option for Oracle	24

Remote procedure calls (RPCs) are accepted by the ECDA Option for Oracle RPC event handler. The RPC is either interpreted locally or passed to Oracle for processing. RPCs specific to ECDA Option for Oracle and RPCs passed to Oracle are documented in this chapter. The examples in this chapter assume that stored procedures are executed from ASE as RPCs.

RPCs passed to Oracle

RPCs that are not interpreted by ECDA Option for Oracle are passed directly to Oracle for execution.

When ECDA Option for Oracle executes a stored program, it associates the actual and formal parameters by position. Therefore, when calling a subprogram using ECDA Option for Oracle, the parameters must be passed using positional notation or named notation.

Note Mixed notations (both positional notation and named notation) are not supported in a single Oracle procedure call.

Using stored procedures with named parameters

When an Oracle stored procedure is called with named parameters, the name must match the parameter name of the of the Oracle procedure. If the names do not match, a procedure call is made without the named parameter being used and an Oracle error occurs.

Note If the parameter name does not match the input parameter name, and the parameter name has a default coded value, the procedure executes using the default value and no error is issued.

Create stored procedures

The syntax for the Oracle create procedure SQL command can be found in Oracle SQL language reference documentation.

Stored procedures within a package

The following examples illustrate the use of stored procedures. The examples are based on the following Oracle table and data:

```
example_table
(id_num int,
name varchar(30),
phone varchar(20) null,
birthdate date null)
```

id_num	name	phone	birthdate
1	Carl Finklestein	203-231-0123	NULL
2	Fred P. Body	NULL	NULL
3	Carl Winkerbean	603-231-4123	AUG 09 1947
4	Wanda Finklestein	978-245-6789	JUL 12 1972
5	John Smith	607-789-0123	NOV 09 1963

Oracle package and procedure definitions

```
CREATE or REPLACE PACKAGE dco_rpc_pkg
AS

TYPE rpc_ex_cur IS REF CURSOR RETURN
```

```

example_table%ROWTYPE;
PROCEDURE rset_rpc_ex (c1 IN date, c2 IN date, a IN OUT
rpc_ex_cur);
PROCEDURE input_rpc_ex (in_id IN number);
PROCEDURE output_rpc_ex (c1 IN OUT number, c2 OUT
varchar);

END dco_rpc_pkg;

CREATE or REPLACE PACKAGE BODY dco_rpc_pkg
AS

PROCEDURE rset_rpc_ex (c1 IN date, c2 IN date, a IN OUT
rpc_ex_cur)
IS
BEGIN
OPEN a FOR select * from example_table where birthdate
between c1 and c2;
END;

PROCEDURE input_rpc_ex (in_id IN number)
IS
BEGIN
delete example_table where id_num = in_id;
END;

PROCEDURE output_rpc_ex (c1 IN OUT number, c2 OUT
varchar)
IS
BEGIN
select name into c2 from example_table where id_num =
c1;
END;

END dco_rpc_pkg;

```

To map an RPC that returns a result set as an ASE table

The following example maps the Oracle procedure `rset_rpc_ex` as an ASE table with its input parameters mapped as columns, as described in the ASE documentation:

```

create existing table rset_rpc_tab
(id_num int,
name varchar(30),
phone varchar(20) null,

```

```

birthdate smalldatetime null,
_c1 smalldatetime null,
_c2 smalldatetime null)
external procedure
at 'DCSERVER.dco_rpc_pkg..rset_rpc_ex'

```

```

select id_num, name, phone, birthdate from rset_rpc_tab
where _c1 = 'jan 01 1940' and _c2 = 'jan 01 1970'
id_num      name                               phone
birthdate
-----
-----
-----
3 Carl Winkerbean                             603-231-4123
Aug  9 1947 12:00AM
5 John Smith                                   607-789-0123
Nov  9 1963 12:00AM
(2 rows affected)

```

To execute an RPC that returns a result set

The following example executes an RPC, from ASE through ECDA Option for Oracle to Oracle. This RPC takes two input parameters and returns a result set:

```

DCServer.dco_rpc_pkg..rset_rpc_ex 'jan 01 1940', 'jan
01 1970'

```

ID_NUM	NAME	PHONE	BIRTHDATE
3	Carl Winkerbean	603-231-4123	Aug 9 1947 12:00AM
5	John Smith	607-789-0123	Nov 9 1963 12:00AM

```

(2 rows affected)
(return status = 0)
(0 rows affected)

```

To execute an RPC with input and output parameters

Following is an example of an RPC from ASE through ECDA Option for Oracle. Input and output parameters are mapped as ASE variables:

```

declare @io_id int
declare @o_name varchar(30)
select @io_id = 3

```

```

exec DCSERVER.dco_rpc_pkg..output_rpc_ex
    @io_id, @o_name output

select @io_id, @o_name
(1 row affected)
(return status = 0)
(0 rows affected)
(0 rows affected)

-----

3 Carl Winkerbean
(1 row affected)

```

To execute an RPC with an input parameter

The following command executes an RPC from ASE through ECDA Option for Oracle. It passes an input parameter, deleting one row:

```

DCSERVER.dco_rpc_pkg..input_rpc_ex 2
(return status = 0)
(0 rows affected)

select * from example_table
id_num          name
phone          birthdate
-----
-----
1              Carl Finklestein
603-231-0123   NULL
3              Carl Winkerbean
603-231-4123   Aug  9 1947 12:00AM
4              Wanda Finklestein
978-245-6789   Jul 12 1972 12:00AM
5              John Smith
607-789-0123   Nov  9 1963 12:00AM
(4 rows affected)

```

RPC restrictions

When handling RPC parameters of datatype tinyint, only valid values (between 0 and 255) can be passed back. If the back-end database returns a negative value, that value is passed to the user as a NULL value. Sybase recommends the smallint datatype because it can handle negative numbers correctly.

Oracle stored procedures with long and long raw datatypes are not supported with output parameters. Also, long raw input parameters are not supported.

RPCs processed by ECDA Option for Oracle

The following RPCs are processed by ECDA Option for Oracle:

- sp_addservice
- sp_columns
- sp_configure
- sp_dropservice
- sp_ps
- sp_shutdown
- sp_statistics
- sp_stored_procedures
- sp_tables
- sp_terminate
- sp_traceon
- sp_traceoff
- sp_who

Note

- When executing stored procedures from within ASE, use the following syntax:

```
server_name... sp_proc-name
```

- When executing stored procedures from the ECDA Option for Oracle, the stored procedure must be executed with the exec syntax as follows:

```
exec sp_proc-name
```

sp_addservice

Description	Adds a new service to the ECDA Option for Oracle and writes the service entry into the configuration file.
Syntax	<code>server_name... sp_addservice service_name</code>
Usage	To use an additional Oracle database, you must add a new service. Use <code>sp_addservice</code> to add an additional service. Then, use <code>sp_configure</code> to configure the new service.

sp_columns

Description	Returns information about the type of data that can be stored in one or more columns.
Syntax	<code>server_name... sp_columns table_name [, table_owner] [, table_qualifier] [, column_name]</code>
Parameters	<p><i>table_name</i> is the name of the table or view. Use wildcard characters to request information about more than one table.</p> <p><i>table_owner</i> is the owner of the table or view. Use wildcard characters to request information about tables owned by more than one user. If you do not specify a table owner, <code>sp_columns</code> looks for tables owned by the current user.</p> <p><i>table_qualifier</i> is ignored. Null should be specified.</p> <p><i>column_name</i> is the name of the column for which you want information. Use wildcard characters to request information about more than one column.</p>
Examples	<p>The following example displays information about all columns in the <code>publishers</code> table that begin with “p”:</p> <pre> 1> DCSERVER...sp_columns "publishers", null, null, "p%" table_qualifier table_owner table_name column_name data_type type_name precision length scale radix nullable remarks ss_data_type colid remote_data_type </pre>

Column	Datatype	Description
data_type	smallint	Integer code for ODBC datatype. If this is a datatype that cannot be mapped into an ODBC type, it is NULL.
type_name	varchar(30)	String representing a datatype. The underlying DBMS presents this datatype name.
precision	int	Number of significant digits.
length	int	Length in bytes of a datatype.
scale	smallint	Number of digits to the right of the decimal point.
radix	smallint	Base for numeric datatypes.
nullable	smallint	The value 1 means NULL is possible; 0 means NOT NULL.
remarks	varchar(254)	
ss_data_type	smallint	An Adaptive Server datatype.
colid	tinyint	The column ID.
remote_data_type	The Oracle datatype	

Permissions Any user can execute sp_columns.

sp_configure

Description Displays or changes ECDA Option for Oracle configuration properties.

Syntax `server_name... sp_configure [service_name][, config_name [, config_value]]`

Parameters

service_name
is the name of the ECDA Option for Oracle service.

config_name
is the name of the configuration parameter to be displayed or modified.

config_value
is the value you assign to the configuration parameter.

Examples

```
DCSERVER...sp_configure DCO_S1, connect_string,
"ORA817.world"
```

Modifies the *connect_string* for service DCO_S1.

Usage

- If no parameters are supplied, sp_configure will list all the services defined:

```
exec DCSERVER...sp_configure
```

sp_configure, with the *service_name* without a value, will list all of the configuration options for that service.

```
exec DCSERVER...sp_configure, DCSERVER0
```

sp_configure with the *service_name* (DCSRERVER0), and the option will display the current value of the option:

```
exec DCSERVER...sp_configure, DCSERVER0, max_users
```

If a value is supplied, such as “100” in the following example, sp_configure changes the option:

```
exec DCSERVER...sp_configure, DCSERVER0, max_users,  
100
```

Note Languages and charsets cannot be displayed or edited with sp_configure. To change these properties, the configuration file must be edited manually.

- When you execute sp_configure to modify a dynamic parameter:
 - The configuration and run values are updated.
 - The configuration file is updated.
 - The change takes effect immediately for all new connections. Current connections are unchanged.
- When you execute sp_configure to modify a static parameter:
 - The configuration value is updated.
 - The configuration file is updated.
 - The change takes effect only when you restart ECDA Option for Oracle.
- When issued with no parameters, sp_configure displays all the service names.
- When issued with only the *service_name* parameter, sp_configure displays all the configuration information for the service.

- If the *config_name* parameter is specified, but the *config_value* parameter is omitted, `sp_configure` displays the report for the configuration parameter specified.
- For information on the individual configuration properties, see “Using the command line” on page 7.

Permissions Only the ECDA Option for Oracle administrator can execute `sp_configure`.

sp_dropservice

Description Removes an existing service from the ECDA Option for Oracle and deletes the service entry from the configuration file.

Syntax `server_name... sp_dropservice service_name`

Usage To drop a service, use `sp_dropservice`.

sp_ps

Description Returns detailed status information on specified Open Server threads.

Syntax `server_name... sp_ps [loginname | 'spid']`

Parameters *loginname*
is the user's login name.

spid
is the internal identification number of the thread to report on. You can obtain the *spid* from the output of a previous `sp_who` or `sp_ps` call. By default, all threads are listed.

Examples `1>DCSERVER...sp_ps`
`2>go`

spid	Login Name	Host Name	Program Name	Task Type	...
----	-----	-----	-----	-----	...
1				SERVER TASK	...
2				SERVER TASK	...
3				SERVER TASK	...
4				SERVICE TASK	...
11		hiram		SITE HANDLER TASK	...

```

14 bud      sonoma      isql      CHILD TASK      ...
... Status  Sleep Event Sleep Label    Current Command ...

... -----
... runnable 369448      MSG AVAILABLE NETWORK HANDLER ...
... sleeping 369544      MSG AVAILABLE CONNECT HANDLER ...
... sleeping 369640      MSG AVAILABLE DEFERRED HANDLER ...
... runnable 0           MSG AVAILABLE SCHEDULER ...
... sleeping 369736      MSG AVAILABLE ...
... running  416480      ...

... Blocked Run      Current      Stack      Net      Net
...      By Ticks      Priority      Origin      Writes      Reads
... -----
...      0      0           8           2794336      0           0
...      0      0           8           2810792      0           0
...      0      0           8           2827184      0           0
...      0      0           15          2843576      0           0
...      0      0           8           2859968      2           7
...      0      0           8           2909208      3           0

```

This example shows isql output from the sp_ps procedure. For printing purposes, the report was split where indicated by ellipses.

Usage

- sp_ps reports the detailed status of a specified server thread or all current Open Server threads. The information is useful for debugging during application development.
- *loginame* and *spid* are character string parameters. When using isql to execute sp_ps as a remote procedure call from an Adaptive Server, surround the *spid* in quotes to avoid a syntax error.
- If you do not specify *loginame* or *spid*, sp_ps lists all current threads.

Table 3-2 summarizes the information sp_ps returns:

Table 3-2: Information returned (sp_ps)

Type of information	Meaning
SPID	The internal thread number of the thread.
Login Name	The name of the logged in user. Applies only to client threads.
Host Name	For a client task, this is the name of the client's machine. For site handlers and server-to-server RPC connections, this is the name of the remote Adaptive Server.
Program Name	The name of the client application program.
Task Type	The type of thread. The legal values are NETWORK, CLIENT, SERVER, SITE HANDLER, CHILD, SERVICE, and UNKNOWN.
Status	The current status of the thread. The legal values for this column are running, RUNNABLE, SLEEPING, SICK, FREE, STOPPED, SPAWNED, TERMINAL, and UNKNOWN. The one "running" task is the thread that is executing sp_ps.
Sleep Event	The event that will cause a sleeping thread to become runnable.
Sleep Label	A character string label that describes the sleep event.
Current Command	A character string that describes the state of the thread. The contents of this column are set by the srv_thread_props routine.
Blocked By	(Not currently used.)
Run Ticks	(Not currently used.)
Current Priority	The priority at which the thread is running.
Stack Origin	The address in memory where the thread's stack begins.
Net Writes	The number of network writes since the thread started. This number applies only to site handler and client threads.
Net Reads	The number of network reads since the thread started. This number applies only to site handler and client threads.

Table 3-3 summarizes the results returned as rows with these columns:

Table 3-3: Format of information returned (sp_ps)

Column name	Datatype	Length
SPID	CS_INT_TYPE	4
Login Name	CS_CHAR_TYPE	SRV_MAXNAME
Host Name	CS_CHAR_TYPE	SRV_MAXNAME
Program Name	CS_CHAR_TYPE	SRV_MAXNAME
Task Type	CS_CHAR_TYPE	SRV_MAXNAME
Status	CS_CHAR_TYPE	SRV_MAXNAME
Sleep Event	CS_INT_TYPE	4
Sleep Label	CS_CHAR_TYPE	SRV_MAXNAME
Current Command	CS_CHAR_TYPE	SRV_MAXNAME
Blocked By	CS_INT_TYPE	4
Run Ticks	CS_INT_TYPE	4
Current Priority	CS_INT_TYPE	4
Stack Origin	CS_INT_TYPE	4
Net Writes	CS_INT_TYPE	4
Net Reads	CS_INT_TYPE	4

See also

sp_terminate, sp_who

sp_shutdown

Description	Shuts down ECDA Option for Oracle.
Syntax	server_name... sp_shutdown
Permissions	Only the administrator can execute sp_shutdown.

sp_statistics

Description	Returns a list of indexes on a single table.
Syntax	server_name... sp_statistics <i>table_name</i> [, <i>table_owner</i>] [, <i>table_qualifier</i>] [, <i>index_name</i>] [, <i>is_unique</i>]
Parameters	<p><i>table_name</i> is the name of the table. The use of wildcard character pattern matching is not supported.</p> <p><i>table_owner</i> is the owner of the table. The use of wildcard character pattern matching is not supported. If <i>table_owner</i> is not specified, sp_statistics looks for a table owned by the current user.</p> <p><i>table_qualifier</i> is ignored. Null should be specified.</p> <p><i>index_name</i> is the index name. The use of wildcard character pattern matching is not supported.</p> <p><i>is_unique</i> is Y to return only unique indexes; otherwise, is N to return both unique and non-unique indexes.</p>

Examples DCSERVER...sp_statistics publishers

```

table_qualifier          table_owner
  table_name              non_unique
  index_qualifier         index_name
  type    seq_in_index  column_name              collation
  cardinality pages
-----
pubs2
  publishers              NULL
  NULL                    NULL
      0          NULL NULL              NULL
      3          1
pubs2
  publishers              0
  publishers              pubind
      1          1 pub_id              A
      3          1

```

Usage

- Table 3-4 describes the results set:

Table 3-4: Results set for sp_statistics

Column	Datatype	Description
table_qualifier	varchar(32)	NULL.
table_owner	varchar(32)	
table_name	varchar(32)	NOT NULL.
non_unique	smallint	NOT NULL. The value 0 means unique, and 1 means not unique.
index_qualifier	varchar(32)	
index_name	varchar(32)	
type	smallint	NOT NULL. The value 0 means clustered, 2 means hashed, and 3 means other.
seq_in_index	smallint	NOT NULL.
column_name	varchar(32)	NOT NULL.
collation	char(1)	The value A means ascending; D means descending; and NULL means not applicable.
cardinality	int	Number of rows in the table or unique values in the index.
pages	int	Number of pages to store the index or table.

- The indexes in the results set appear in ascending order, ordered by the non-unique, type, index_name, and seq_in_index columns.

Permissions

Any user can execute sp_statistics.

sp_stored_procedures

Description	Returns information about one or more stored procedures.																								
Syntax	server_name... sp_stored_procedures [sp_name [, sp_owner [, sp_qualifier]]]																								
Parameters	<p><i>sp_name</i> is the name of the stored procedure. Use wildcard characters to request information about more than one stored procedure.</p> <p><i>sp_owner</i> is the owner of the stored procedure. Use wildcard characters to request information about procedures that are owned by more than one user.</p> <p><i>sp_qualifier</i> is the name of the package. This can be a package name or NULL.</p>																								
Usage	<ul style="list-style-type: none"> sp_stored_procedures returns information about stored procedures in all packages unless sp_qualifier is specified. Table 3-5 shows the results set: <p>Table 3-5: Results set for sp_stored_procedures</p> <table border="1"> <thead> <tr> <th>Column</th> <th>Datatype</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>procedure_qualifier</td> <td>varchar(30)</td> <td>The name of the package.</td> </tr> <tr> <td>procedure_owner</td> <td>varchar(30)</td> <td></td> </tr> <tr> <td>procedure_name</td> <td>varchar(41)</td> <td>NOT NULL.</td> </tr> <tr> <td>num_input_params</td> <td>int</td> <td>NOT NULL. The value ≥ 0 shows the number of parameters; -1 means the number of parameters is indeterminate.</td> </tr> <tr> <td>num_output_params</td> <td>int</td> <td>NOT NULL. The value ≥ 0 shows the number of parameters; -1 means the number of parameters is indeterminate.</td> </tr> <tr> <td>num_result_sets</td> <td>int</td> <td>NOT NULL. The value ≥ 0 shows the number of parameters; -1 means the number of parameters is indeterminate.</td> </tr> <tr> <td>remarks</td> <td>varchar(254)</td> <td>NULL.</td> </tr> </tbody> </table> <ul style="list-style-type: none"> sp_stored_procedures can return the name of stored procedures for which the current user does not have execute permission. 	Column	Datatype	Description	procedure_qualifier	varchar(30)	The name of the package.	procedure_owner	varchar(30)		procedure_name	varchar(41)	NOT NULL.	num_input_params	int	NOT NULL. The value ≥ 0 shows the number of parameters; -1 means the number of parameters is indeterminate.	num_output_params	int	NOT NULL. The value ≥ 0 shows the number of parameters; -1 means the number of parameters is indeterminate.	num_result_sets	int	NOT NULL. The value ≥ 0 shows the number of parameters; -1 means the number of parameters is indeterminate.	remarks	varchar(254)	NULL.
Column	Datatype	Description																							
procedure_qualifier	varchar(30)	The name of the package.																							
procedure_owner	varchar(30)																								
procedure_name	varchar(41)	NOT NULL.																							
num_input_params	int	NOT NULL. The value ≥ 0 shows the number of parameters; -1 means the number of parameters is indeterminate.																							
num_output_params	int	NOT NULL. The value ≥ 0 shows the number of parameters; -1 means the number of parameters is indeterminate.																							
num_result_sets	int	NOT NULL. The value ≥ 0 shows the number of parameters; -1 means the number of parameters is indeterminate.																							
remarks	varchar(254)	NULL.																							
Permissions	Any user can execute sp_stored_procedures.																								

sp_tables

Description	Returns a list of objects that can appear in a from clause.
Syntax	<code>server_name... sp_tables [table_name] [, table_owner] [, table_qualifier][, table_type]</code>
Parameters	<p><i>table_name</i> is the name of the table. Use wildcard characters to request information about more than one table.</p> <p><i>table_owner</i> is the owner of the table. Use wildcards to request information about tables that are owned by more than one user.</p> <p><i>table_qualifier</i> is ignored. Use NULL.</p> <p><i>table_type</i> is a list of values, separated by commas, giving information about all tables of the table type(s) specified, including the following:</p> <pre>" 'TABLE', 'SYSTEM TABLE', 'VIEW' "</pre>

Note Enclose each table type with single quotation marks, and enclose the entire parameter with double quotation marks. Enter table types in uppercase.

Examples	<pre>DCSERVER...sp_tables @table_type = 'TABLE', 'VIEW'</pre>
----------	---

This procedure returns information about all tables of the type TABLE and VIEW and excludes information about system tables.

Usage	<ul style="list-style-type: none">• ECDA Option for Oracle does not necessarily check the read and write permissions on <i>table_name</i>. Access to the table is not guaranteed, even if you can display information about it.• The results set includes tables, views, synonyms, and aliases.• Table 3-6 describes the results set:
-------	---

Table 3-6: Results set for `sp_tables`

Column	Datatype	Description
<code>table_qualifier</code>	<code>varchar(30)</code>	NULL.
<code>table_owner</code>	<code>varchar(30)</code>	Table owner
<code>table_name</code>	<code>varchar(30)</code>	NOT NULL. The table name.
<code>table_type</code>	<code>varchar(32)</code>	NOT NULL. One of the following: 'TABLE', 'VIEW', 'SYSTEM TABLE'.
<code>remarks</code>	<code>varchar(254)</code>	NULL

Permissions Any user can execute `sp_tables`.

sp_terminate

Description Terminates an Open Server thread.

Syntax `server_name... sp_terminate spid [, options]`

Parameters `spid`

is the thread ID. This can be obtained with the `sp_who` procedure or by calling `srv_thread_props`.

`options`

Determines whether the thread is terminated immediately or by a queued disconnect event:

- Specify “deferred” to queue a disconnect event that occurs after previous events are handled. This is the default action.
- Specify “immediate” to terminate the thread immediately, ignoring current or queued events for the thread.

Examples

```
1> DCSERVER...sp_who
2> go
```

spid	status	loginame	hostname	blk	cmd
1	runnable			0	NETWORK HANDLER
2	sleeping			0	CONNECT HANDLER
3	sleeping			0	DEFERRED HANDLER
4	runnable			0	SCHEDULER
12	runnable	ned	sonoma	0	PRINT TASK
24	running	bud	sonoma	0	

(0 rows affected)

This example shows how to use `isql` to locate and terminate an errant server thread. The thread terminates immediately.

```
1> DCSERVER...sp_terminate 12, "immediate"
2> go

spid = 12;
dbrpcinit(dbproc, "sp_terminate", (DBUSMALLINT) 0);
dbrpcparam(dbproc, "@spid", 0, SYBINT4, -1,
           -1, &spid);
dbrpcparam(dbproc, "@options", 0, SYBCHAR, 9,
           9, "deferred");
dbrpcsend(dbproc);
```

This DB-Library example queues a `SRV_DISCONNECT` event for the thread. The next time the thread can run, it receives the disconnect event and terminates.

Usage	Use <code>sp_who</code> or <code>sp_ps</code> to find the <i>spid</i> for the thread to be terminated.
Messages	<i>spid</i> terminated. <i>spid</i> scheduled for termination. <i>spid</i> not currently in use.
See also	<code>sp_who</code>

sp_traceon

Description	Turns on traceflags for the current connection.
Syntax	<code>server_name... sp_traceon traceflags</code>
Examples	<code>DCSERVER...sp_traceon 2,3,4,5</code>
Usage	Use with traceflags described in “traceflags” on page 61

sp_traceoff

Description	Turns off traceflags for the current connection.
Syntax	<code>server_name... sp_traceoff [traceflags]</code>

Examples DCSERVER...sp_traceoff 2,3,4,5

Usage If no traceflags are specified, all traceflags are turned off.

sp_who

Description Return status information for specified Open Server threads.

Syntax server_name... sp_who [*loginame* | "*spid*"]

Parameters

loginame
The user's login name.

spid
The internal identification number of the thread to report on. The *spid* can be obtained from the output of a previous sp_ps or sp_who call. If no *spid* is specified, all threads are listed.

Examples This example shows output from the sp_who procedure.

```

1>DCSERVER...sp_who
2>go

```

spid	status	loginame	hostname	blk	cmd
1	runnable			0	NETWORK HANDLER
2	sleeping			0	CONNECT HANDLER
3	sleeping			0	DEFERRED HANDLER
4	runnable			0	SCHEDULER
11	sleeping		hiram	0	
14	running	bud	sonoma	0	

In this example, the output columns are as follows:

loginame – the name of the logged in user. Applies only to client threads.

hostname – for a client task, this is the name of the client's machine. For a site handler thread, it is the name of the remote Adaptive Server.

blk – this field is unused and is always set to 0.

cmd – a character string that describes the state of the thread. The contents of this column are set by the srv_thread_props routine.

Usage

- sp_who reports status information about a specified server thread or all current Open Server threads.

- The output from the sp_who system registered procedure matches the output from the Adaptive Server sp_who system procedure.
- sp_who returns a subset of the information that sp_ps returns.
- *loginame* and *spid* are character string parameters. When using isql to execute sp_who as a remote procedure call from an Adaptive Server, surround the *spid* in quotes to avoid a syntax error.
- If you do not specify *loginame* or *spid*, sp_who lists all current threads.
- sp_who returns the following information:

spid – the internal thread number of the thread.

status – the current status of the thread. The values for this column are:

- running
- runnable
- sleeping
- sick
- free
- stopped
- spawned
- terminal
- unknown

The one “running” task is the thread that is executing sp_who.

Table 3-7 summarizes the results returned as rows with these columns:

Table 3-7: Format of information returned (sp_who)

Column name	Datatype	Length
spid	CS_INT_TYPE	4
status	CS_CHAR_TYPE	10
loginame	CS_CHAR_TYPE	12
hostname	CS_CHAR_TYPE	10
blk	CS_INT_TYPE	3
cmd	CS_CHAR_TYPE	16

See also

sp_ps, sp_terminate

Secure Sockets Layer (SSL)

It contains the following topics:

Topic	Page
Secure Sockets Layer	41
Configuring SSL between ECDA Option for Oracle	45
Connecting SSL between ECDA Option for Oracle and the client	45

Secure Sockets Layer

ECDA Option for Oracle supports Secure Sockets Layer (SSL) session-based security. SSL is the standard for securing the transmission of sensitive information, (such as credit card numbers, stock trades, and banking transactions), over the Internet.

The implementation of ECDA Option for Oracle SSL features assume that you have a knowledgeable System Security Officer who is familiar with the security policies and needs of your site, and who has general understanding of SSL and public-key cryptography.

ECDA Option for Oracle uses the SSL Plus™ library API from Certicom Corp.

SSL filter

When establishing a connection to an SSL-enabled ECDA Option for Oracle, the SSL security mechanism is specified as a filter on the master and query lines in the *interfaces* file for UNIX, or the *sql.ini* file on Windows. SSL is used as an Open Client and Open Server protocol layer that sits on top of the TCP/IP connection.

For example, a typical *interfaces* file on a UNIX machine using transport layer interface (tli) and SSL looks like this:

```
SERVER <retries><time-outs>
```


In these examples, the SSL security service is specified on port number 2748(0x0abc). On SYBSRV1, DCO listens for clear text on port number 2749(0x0abd), which is without any security mechanism or security filter.

Server certificate

Each ECDA Option for Oracle must have its own server certificate file that is loaded at start-up. The location of the server certificate is specified in the `certificate_file` configuration option.

The server certificate file consists of encoded data, including the server's certificate and the encrypted private key for the server certificate.

To make a successful client connection, the common name in the certificate must match the ECDA Option for Oracle name in the *interfaces* file.

CA trusted roots certificate

At start-up, ECDA Option for Oracle loads the list of trusted CAs from the trusted roots file. The trusted roots file is similar in format to a certificate file, except that it contains certificates for CAs known to ECDA Option for Oracle. A trusted roots file is accessible by the ECDA Option for Oracle in:

- UNIX - `<install_dir>/DCO-15_0/connectivity/bin/config/trusted.txt`
- Windows - `C:\<install_dir>\DCO-15_0\connectivity\bin\ini\trusted.txt`

Alternatively, you can specify the location of the trusted roots file in the `trusted_roots_file` configuration option.

The System Security Officer adds and deletes CAs that are to be accepted by ECDA Option for Oracle, using a standard ASCII-text editor.

Client login to ECDA Option for Oracle

Open Client applications establish a socket connection to the ECDA Option for Oracle similar to the way that existing client connections are established. Before any user data is transmitted, an SSL handshake occurs on the socket when the network transport-level connect call completes on the client side and the accept call completes on the server side.

When the SSL session is established, user name and password are transmitted over a secure, encrypted connection.

Enabling SSL

The DirectConnect server determines which security service it will use for a port based on the *interface* file on UNIX and *sql.ini* file on Windows.

❖ To enable SSL

- 1 Generate a *certificate* and *trusted roots* file for the server.
- 2 Use `sp_configure` to specify the *certificate* file. From a command prompt, enter:

```
exec sp_configure <servername>, "certificate_file",  
    <certificate file spec>
```
- 3 Use `sp_configure` to specify the *trusted roots* file, unless you are using the default location:

```
exec sp_configure <servername>,  
    "trusted_roots_file", <trusted roots file spec>
```
- 4 Use `sp_configure` to specify the *certificate password* file. From the command prompt, enter:

```
exec sp_configure <servername>,"certificate  
password", <certificate password file spec>
```
- 5 Add the SSL filter to the *interfaces* file.
- 6 Shut down and restart the server.

For more information on SSL, refer to the *ASE System Administration Guide*.

Performance affected by SSL

There is additional overhead required to establish a secure session, because data increases in size when it is encrypted, and it requires additional computation to encrypt or decrypt information. Typically, the additional I/O accrued during the SSL handshake may make user login 10 to 20 times slower.

Configuring SSL between ECDA Option for Oracle

To configure SSL connections from ECDA Option for Oracle to an existing SSL enabled Oracle database instance, see the “How to configure SSL between the ECDA Option for Oracle and an Oracle database” document at <http://www.sybase.com/detail?id=1042098>. This type of connection is available for both UNIX and Windows even though this document shows only the Windows configuration.

Connecting SSL between ECDA Option for Oracle and the client

The following sections describe the steps required for Secure Sockets Layer (SSL) connectivity to ECDA Option for Oracle from the isql client.

Using *certreq* and *certauth* utilities

The following example uses the *certreq* and *certauth* utilities packaged with connectivity to create a *certificate* and *trusted roots* file. These are for testing purposes only and should not be used to replace authentic certificates in a production environment. These utilities can be found in the `C:\<install_dir>\DCO-15_0\bin` directory.

DirectConnect server name

In the example, you must substitute the name of your DirectConnect server for the name "dcssl."

Note In the example, the values of the "Organizational Unit" that occurs twice, must be different. If equal, you will not be able to connect and an error message will be created that will not identify the problem.

Example to set up connectivity

The following example provides all the steps required to set up connectivity to the isql client.

❖ **To copy the certificate and the *trusted roots* file**

- 1 Add the following to the path of the environment variable:

- C:\<install_dir>\OCS-15_0\lib3p
- 2 Set the environment by running the following from a command window:
C:\<install_dir>\DCO-15_0\DCO_SYBASE.bat
 - 3 Enter the following to go to the *certreq* directory:
cd C:\<install_dir>\DCO-15_0\bin
 - 4 Execute the *setsslreq* utility, one time only, on Windows to set SSL registry key information for Open Server.
 - 5 Execute the following command to begin the certification process:
C:\<install_dir>\DCO-15_0\bin>certreq
 - 6 Choose the certificate request type from the following:
S - Server certificate request.
C - Client certificate request.
Q - Quit.
Enter *S*.
 - 7 Choose the key type from the following:
R - RSA key pair.
D - DSA with ephemeral Diffie-Hellman key exchange.
Q - Quit.
Enter *R*.
 - 8 Enter the information in each of the following fields as it appears:
Key length (512,768,1024 for DSA; 512-2048 for RSA)
: 512
Country: *US*
State: *California*
Locality: *Dublin*
Organization: *Sybase*
Organizational Unit: *ecda*
Common Name: *dcossl* (Must be the same name as the DCO server)
 - 9 A key pair is being generated which takes a period of time. When completed, enter the information in the following fields as it appears:
Password for private key (max 64 chars): *sybase*
File path to save request: *ca_req.txt*
File path to save private key: *ca_pkey.txt*
 - 10 Enter the following:

```
C:\<install_dir>\DCO-15_0\bin>certauth -r -C
ca_req.txt -Qca_req.txt -Kca_pkey.txt -O trusted.txt
-P sybase
```

- 11 The following appears:

```
-- Sybase Test Certificate Authority
certauth/15.0/EBF 11798/P/NT(IX86)/OS 4.0/ase1252
/1831/32-bit/OPT/Fri Apr 09 04:35:35 2004 --
Certificate Validity:
startDate = Mon Feb 07 15:18:00 2005
endDate = Tue Feb 07 15:18:00 2006
CA sign certificate SUCCEED (0)
```

- 12 Enter the following to go to the *certreq* directory:

```
C:\<install_dir>\DCO-15_0\bin>certreq
```

- 13 Choose the certificate request type from the following:

```
S - Server certificate request.
C - Client certificate request.
Q - Quit.
```

Enter *S*.

- 14 Choose the key type from the following:

```
R - RSA key pair.
D - DSA with ephemeral Diffie-Hellman key exchange.
Q - Quit.
```

Enter *R*.

- 15 Enter the information in each of the following fields as it appears:

```
Key length (512,768,1024 for DSA; 512-2048 for RSA)
: 512
Country: US
State: California
Locality: Dublin
Organization: Sybase
Organizational Unit: ecda
Common Name: dcossl (Must be the same name as the DCO
server)
```

- 16 A key pair is being generated which takes a period of time. When completed, enter the information in the following fields as it appears:

```
Password for private key (max 64 chars): sybase

File path to save request: dcossl_req.txt (Must
```

begin with the same name as the DCO server)

File path to save private key: *dcossl_pkey.txt*
(Must begin with the same name as the DCO server)

- 17 Enter the following:

```
C:\install_dir\DCO-15_0\bin>certauth -Ctrusted.txt
-Qdcossl_req.txt -Kca_pkey.txt -Odcossl.crt
-Psybase
```

- 18 The following appears:

```
-- Sybase Test Certificate Authority certauth/15/EBF
11798/P/NT(IX86)/OS 4.0/ase1252 /1831/32-
bit/OPT/Fri Apr 09 04:35:35 2004 --
Certificate Validity:
startDate = Mon Feb 07 15:18:00 2005
endDate = Tue Feb 07 15:18:00 2006
CA sign certificate SUCCEED (0)
```

- 19 Copy the certificate created to the directory specified in the DirectConnect server configuration file by entering the following: (This directory is created and determined by the ECDA Option for Oracle administrator.)

```
C:\<install_dir>\DCO-15_0\bin>copy dcossl.crt
C:\<install_dir>\DCO-15_0\certificates
```

- 20 Copy the *trusted roots* file to the directory specified by the DirectConnect server configuration file. If the isql client is in a different location, the *trusted.txt* file created by the steps above, can be concatenated to the *trusted.txt* file contained in the *ini* or *config* directory. Enter the following:

```
C:\<install_dir>\DCO-15_0\bin>type trusted.txt >>
C:\<install_dir>\DCO-15_0\bin\trusted.txt
```

- 21 Alter the *sql.ini* or *interfaces* file and append the master and query entries for the DirectConnect server with the ssl filter tag.

```
[dcossl]

master=NLWNSCK,winserver,12700,ssl
query=NLWNSCK,winserver,12700,ssl
```

- 22 Edit the ECDA Option for Oracle configuration file as described in “Enabling SSL” on page 44.

Configuration Parameters

Topic	Page
Types of configuration parameters	49
Global configuration parameters	49
Service-specific configuration parameters	54

Types of configuration parameters

There are two types of configuration parameters:

- global parameters that apply to all services
- service-specific parameters that apply to a specific service

Global configuration parameters

Define all global configuration parameters as shown in the service entry that has the same name as the configuration file itself. This primary service name is also the server name. Global parameters apply to all services. These parameters are used with “sp_configure” on page 27.

Table 4-1: Global configuration parameters

Keyword	Description	Default
administrator	Defines who can execute sp_shutdown, sp_configure, sp_addservice, sp_dropservice	
certificate_file	The name of the certificate file used for SSL	
certificate_password	The password associated with the certificate file	
charset	The default charset used by Open Server	iso_1, Roman8 (HP), cp850 (Windows)

Keyword	Description	Default
errorlog	Full path name to the error log file	<install_dir> /DCO-15_0/servername.log
interfaces	Full path to the directory of the interfaces file	<install_dir>/interfaces
language	Available languages	us_english
max_packetsize	Sets the maximum packet size that can be negotiated for a client connection	2048
max_pagesize	Sets the maximum page size for transfers using the bulk copy interface	4096
max_stacksize	Sets stack size	100352
max_textsize	Sets the maximum size transferred when a text/image column is queried	32768
max_threads	Set the maximum number of Open Server threads	max_users * 4
max_users	Defines the total number of concurrent users that can be connected to ECDA Option for Oracle	25
network_tracing	Enables packet tracing between the client and ECDA Option for Oracle	0
srv_traceflags	String containing numeric flag identifiers used to generate Open Server diagnostic information	0
text_chunksize	Sets the size of each text/image chunk when retrieving text/image data	32768
trusted_roots_file	The name of the trusted roots file	

administrator

This configuration parameter defines who can execute the `sp_shutdown`, `sp_configure`, `sp_addservice`, and `sp_dropservice` commands. If you want the user “ozzie” to be the authority to shut down DirectConnect, the value of the administrator configuration parameter must be “ozzie”. This user must also be a valid Oracle user. If this parameter is not defined, no one will be able to shut down ECDA Option for Oracle. Only one user can have administrator privileges. This parameter is case sensitive.

```
administrator = ozzie
```

This parameter is dynamic.

certificate_file

This is the name of the certificate file used for SSL.

certificate_password

This is the password associated with the certificate file.

charset

This is the default character set for ECDA Option for Oracle is iso_1. On HP, the default is Roman8, and on Windows the default is cp850.

The character set is requested when a connection is made to Oracle. Results are converted into the client's character set when they are sent to the client. The default character set for ECDA Option for Oracle is:

```
charset = cp850
```

This parameter is static.

errorlog

This parameter sets the full path name to the error log file. It defaults to:

```
<install_dir>/DCO-15_0/install/servername.log.
```

This parameter is static.

interfaces

This sets the full path to the directory of the interfaces file.

This parameter is static.

language

This sets the default language used by ECDA Option for Oracle. However, when a connection is made to Oracle, the language requested is the client language, which changes the language of messages in the log file. The language used to connect to Oracle is passed on from the DirectConnect client. The syntax is:

```
language = us_english
```

This parameter is static.

max_packet_size

This sets the maximum packet size that can be negotiated for a client connection. Without this parameter, the maximum packet size defaults to 2048. If you increase the packet size used in ASE through `sp_configure` (using “`ctlib packet size`”) to a value greater than 2048, you need to set this value to match.

Do not set this parameter larger than necessary, because this amount of memory is allocated whether used or not. The maximum packet size supported is 32,256 bytes. The syntax is:

```
max_packet_size = 4096
```

This parameter is static.

max_pagesize

This sets the maximum page size for transfers using the bulk copy interface. The default is 4096 bytes.

This parameter is static.

max_stacksize

This sets the maximum stack size that is allocated for each user.

This parameter is dynamic.

max_textsize

This sets the maximum size transferred when a text or image column is queried. The default is 32,768 bytes.

This parameter is dynamic.

max_threads

This sets the maximum number of Open Server threads. The default is $max_users * 4$.

This parameter is static.

max_users

This defines the total number of users that can be connected to a DirectConnect concurrently. As the number of users increase, additional resources are required. *max_users* cannot be set to a value less than 5.

To support 50 concurrent connections, set *max_users* as shown below:

```
max_users = 50
```

This parameter is static.

network_tracing

This enables packet tracing between the client and ECDA for Oracle. It also enables tracing for internal communications between threads.

Trace entries between the client and ECDA for Oracle are recorded in *<install_dir>/DCO-15_0/SERVERNAME.log*.

Internal trace entries are recorded in *<install_dir>/DCO-15_0/SERVERNAME.tds*.

```
network_tracing = 1
```

This feature should not be used unless required, because it will cause log files to fill up quickly.

srv_traceflags

This enables tracing of Open Server events. A string containing numeric flag identifiers is used to generate Open Server diagnostic information.

This parameter is static.

Open Server trace flags

Use the *srv_traceflags* parameter to turn on trace flags to log Open Server diagnostic information.

Table A-1: Open Server trace flags

Trace flag	Description
1	Traces TDS headers
2	Traces TDS data
3	Traces attention events
4	Traces message queues
5	Traces TDS tokens
6	Traces Open Server events
7	Traces deferred event queues
8	Traces network requests

For example:

```
sp_configure DCSERVER, 'srv_traceflags', '3'
```

This parameter is dynamic.

text_chunksize

This sets the size of each text or image chunk when retrieving text and image data.

This parameter is dynamic.

trusted_roots_file

This is the name of the *trusted roots* file.

Service-specific configuration parameters

The parameters that are not global can be specified for any service. However, if a service-specific keyword is not defined, the default value applies. Additional services after the primary service do not inherit service-specific parameters from the primary service. These parameters are used with “sp_configure” on page 27.

Table A-2 lists the service-specific parameters:

Table A-2: Service-specific configuration parameters

Keyword	Description	Default
autocommit	Gives transactional control in passthrough mode.	1
array_size	Specifies the number of rows retrieved with a single fetch.	50
connect_string	Defines a connection to the target Oracle database.	
csp_uppercase	Assigns uppercase object names when executing catalog stored procedures.	1
date_format	Applies the syntax for inserting dates in passthrough mode.	MON DD YYYY HH:MI:SSAM
default_precision	Specifies the Sybase precision when an Oracle number datatype is defined without precision or scale.	38
default_scale	Specifies the Sybase scale when an Oracle number datatype is defined without precision or scale.	0
insert_array_size	Specifies the number of rows to batch before issuing an insert.	50
number_mode	Determines the behavior for converting Oracle data with a datatype number.	2
session_time_limit	Specifies the maximum length of time a 2pc transaction can be inactive before it is automatically aborted by the Oracle system. The unit for this time limit is in seconds.	60
timestamp_format	Specifies the timestamp_format used in passthrough mode.	MON DD YYYY HH:MI:SSAM
timestamp_ms_support	Specifies whether the timestamp_format is used.	0
timestamp_tz_format	Specifies the timestamp_format used in passthrough mode.	MON DD YYYY HH:MI:SS.FFAM TZH:TZM
traceflags	Controls the types of messages written to the log file.	
triggers	For internal use only.	0
two_phase_commit	Enables the two-phase commit feature.	0

autocommit

This configuration parameter gives the client application transactional control in passthrough mode. If `autocommit = 0` in the configuration file, the client application has transactional control in passthrough mode. If `autocommit = 1` then Direct Connect for Oracle autocommits each SQL statement from the client application:

```
autocommit = 1
```

This parameter is dynamic.

array_size

This sets the number of rows retrieved with a single fetch. It defaults to 50.

This parameter is dynamic.

connect_string

The `connect_string` parameter defines a connection to the target Oracle database. Each service defines a `connect_string`.

Oracle provides a way of connecting to a target Oracle database using SQLNET8. It searches for the `tnsnames.ora` file in the `<install_dir>/DCO-15_0/network/admin` directory.

For this, you need to have an entry in the `tnsnames.ora` file describing the instance to which you will connect. In the following form:

```
DCSERVER.world =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL=TCP) (HOST = dixville)
        (PORT = 1521))
    (CONNECT_DATA =
      (SERVICE_NAME = DCSERVER.world)
    )
  )
)
```

where:

- `ORA_HOST` is the name of the host running Oracle RDBMS.
`ORACLE_SRV` is the SID of the RDBMS you intend to connect to. It is analogous to the subnetwork name; `PORT` is the socket number on which the TNS Listener is listening for connection requests.

- *ORACLE_SERVER* is the alias used to connect to the Oracle RDBMS. The alias name can be any logical name. It is the same as the server name in Sybase terminology. You can have multiple entries for different servers in *tnsnames.ora*, just as in the Sybase *sql_ini* file.

In this example, the connect string required by ECDA Option for Oracle is *ORACLE_SERVER*.

This is used when defining the *connect_string* parameter in the configuration file:

```
connect_string = ORACLE_SERVER
```

This parameter is dynamic.

csp_uppercase

Before querying Oracle, this converts names to uppercase in the catalog stored procedures *sp_tables*, *sp_columns*, *sp_statistics*, and *sp_stored_procedures*.

This parameter is dynamic.

date_format

Specifies the date format used in SQL statements. This setting is ignored for connections from ASE. The default date format is *MON DD YYYY HH:MI:SSAM*. See your Oracle documentation for more information on date format strings.

The syntax for *date_format* is:

```
date_format=Oracle_Date_Format_String
```

This parameter is dynamic.

Note The configuration parameter setting *date_format* is reserved for all ECDA Option for Oracle client connections in passthrough mode, for example ISQL, which excludes ASE/CIS clients.

default_precision

Specifies the Sybase precision when an Oracle number datatype is defined without precision or scale. This parameter affects proxy table datatype mappings and *sp_columns* results.

A distinct `default_precision` setting can be defined for each service. Valid Sybase precision range is 1-38, with the default of 0.

This parameter is dynamic.

default_scale

Specifies the Sybase scale when an Oracle number datatype is defined without precision or scale. This parameter affects proxy table datatype mappings and `sp_columns` results.

A distinct `default_precision` setting can be defined for each service. Valid Sybase scale range is 0-38. The default value for this configuration parameter is 0.

This parameter is dynamic.

insert_array_size

This represents the number of rows that ECDA Option for Oracle will buffer before calling Oracle to insert them. Array inserts are performed when a bulk load event is sent to the ECDA Option for Oracle.

This parameter is dynamic.

number_mode

This determines the behavior for converting Oracle data with datatype number. A distinct `number_mode` setting can be defined for each service.

With a setting of 1, ECDA Option for Oracle attempts to return data as `tinyint`, `smallint`, `int`, `float` or `numeric`. Overflows are possible.

A setting of 0 was used to support pre-system 10 servers. However, because pre-system 10 servers are no longer supported by ECDA Option for Oracle, a setting of 2 causes all numbers to be returned as `numeric`. This setting gives the best performance and is the default.

The following table describes how data is returned, based on the `number_mode` parameter and scale value of the Oracle number datatype.

Table 4-2: Rules for Oracle datatype number conversion

Scale	number_mode = 0	number_mode = 1	number_mode = 2
scale = 0	If 0 <= precision <=9 Send back as shown in Table 4-3, else send back data as float.	If 0 <= precision <=9, send back as shown else if precision is valid, send back data as numeric, else send back data as float.	If precision is valid, send back data as numeric, else send back data as float.
scale !=0	Send back data as float.	If valid precision and scale, send back data as numeric, else send back data as float.	If valid precision and scale, send back data as numeric.

Table 4-3: Conversion for Oracle datatype number

Oracle datatype	Sybase datatype
number(1,0)	tinyint
number(2,0)	tinyint
number(3,0)	smallint
number(4,0)	smallint
number(5,0)	int
number(p,0) where 5 <= p <= 9	int

This parameter is dynamic.

session_time_limit

Specifies the maximum length of time that a two-phase commit transaction can be inactive before it is automatically aborted by the Oracle system. The unit for this time limit is in seconds. The value of 0 indicates no limit. The default is 60 seconds.

The syntax for session_time_limit is:

```
session_time_limit=value
```

This parameter is dynamic.

timestamp_format

Specifies the timestamp format used in SQL statements. This setting is ignored for connections from ASE. The timestamp format depends on the value of the timestamp_ms_support:

- If the value = 0, the default timestamp format is: MON DD YYYY
HH:MI:SSAM
- If the value = 1, the timestamp format is: MON DD YYYY
HH:MI:SS.FFAM

Note The configuration parameter setting `timestamp_format` is reserved for all ECDA Option for Oracle client connections in passthrough mode, for example ISQL, which excludes ASE/CIS clients.

timestamp_ms_support

Specifies whether the `timestamp_format` is used in SQL statements. A value of 1 indicates that the `timestamp_format` is used, and a value of 0 (default) indicates it is not used.

This parameter is dynamic.

timestamp_tz_format

The timestamp with timezone format (`timestamp_tz_format`), normalizes the data value returned to the timezone of ECDA Option for Oracle. The `timestamp_tz_format` default value is "MON DD YYYY HH:MI:SS.FFAM TZH:TZM" and is independent of the timestamp millisecond support configuration parameter.

The Oracle `timestamp_tz_format` is sent back to the client as a character string, and is formatted depending on the following:

- If the client is a non-ASE/CIS client and the `timestamp_tz_format` was changed, the new format is used.
- If the client is a non-ASE/CIS client and the `timestamp_tz_format` was not changed, the default format is used.
- If the client is an ASE/CIS client, the character string is returned using the default format.

In CIS, to map a `proxy_table` to an existing Oracle table that contains a column with the `timestamp_tz_format` datatype, the `create proxy_table` command will map the column to a char (64) and if the `create existing table` command is used, a char (64) must be specified for the column representing the `timestamp_tz_format`.

traceflags

Controls the types of messages written to the log file.

These traceflags values are recognized:

- 1 - logs startup information
- 2 - logs connection information
- 3 - logs language event processing
- 4 - logs rpc event processing
- 5 - logs cursor event processing
- 6 - logs dynamic event processing
- 7 - logs bulk events
- 8 - logs messages sent to client
- 9 - logs interaction with Oracle RDBMS
- 10 - logs information contained in TDS login record
- 11 - logs all DONE packet processing
- 12 - forces shutdown
- 13 - traces XA (ASTC, DTM) RPCs
- 14 - enables XA LIB tracing. The output is written to `<install_dir>/DCO-15_0/rdbmd/log/xa_servicenameDate.trc`
- 15 - logs parser errors

Multiple traceflags can be specified at once if separated by commas:

```
traceflags = 1,2,3,4,5,6
```

This parameter is dynamic.

triggers

For internal use only.

two_phase_commit

Enables the two-phase commit feature.

This parameter is static.

Reference Topics for Oracle

This chapter describes reference topics for ECDA Option for Oracle.

Topic	Page
Connect handler for ECDA Option for Oracle	63
SQL command handling	63
Support for text and image datatypes	64
Localization and internationalization	65
Passthrough mode	66
Global variables	73
Set commands	74
Two-phase commit	75
Lightweight Directory Access Protocol (LDAP)	76

Connect handler for ECDA Option for Oracle

The ECDA Option for Oracle's connect handler creates Open Server threads and connects to Oracle on behalf of the ASE user.

ASE supplies the user name and password, and it also passes a server name for each login request. This server name must appear in the configuration file to ensure that the correct connect string is used. The connect string defines the target database location. The ECDA Option for Oracle logs into the Oracle database and remains connected.

SQL command handling

ECDA Option for Oracle operates in passthrough mode unless it is used with ASE. In passthrough mode the ECDA Option for Oracle passes SQL directly to the Oracle database with no transformation.

Support for text and image datatypes

ECDA Option for Oracle supports columns of datatype text and image when used with ASE. ECDA Option for Oracle provides text pointers, which handles that reference text or image values in `readtext`, `writetext`, and `select` commands.

Sybase text and image columns are similar to the Oracle CLOB and BLOB columns. When ASE generates a `select textptr()` statement for a column, ECDA Option for Oracle creates a `textptr` value based on the Oracle ROWID. The text pointer is the actual ROWID of the given row.

When ASE requests a text or image column's text pointer using a `select textptr()` function, ECDA Option for Oracle will return a 16-byte binary column.

When inserting and updating text or image columns using ASE, use `readtext` and `writetext` commands for data lengths exceeding 16KB.

Selecting *text* and *image*

To select text and image datatypes, use `select` or `readtext`.

When ASE issues a `select` on a text or image column, it sends out a `select textptr()` statement to get the text pointer of the required row. Using the text pointer, ASE issues a `readtext` command.

Updating data using `writetext`

For writing text or image columns of more than 16KB, use the `writetext` command. The data is read by ECDA Option for Oracle in chunks and written to the Oracle database when all the chunks are received. There must be enough memory to store all the chunks together before an Oracle update statement is executed.

Localization and internationalization

When ASE communicates with ECDA Option for Oracle, client language and the default character set are passed as thread properties. ECDA Option for Oracle interacts with Oracle, conveying the client language and Open Server character set. As a result, ASE receives the character data and error messages in the correct language and character set.

Language and character set settings

ECDA Option for Oracle has multiple settings that correspond to the character set and language setting, which can be set in many ways. However, you must designate the character set settings that is consistent with the charset settings.

If the configuration parameter `charset` is set to *Sybase iso_1*, the mapping coded in the “charsets” stanza must be mapped to Oracle’s equivalent character set, which is *we8iso8859p1*. In addition, this setting must also be reflected in the “languages” stanza by including the *we8iso8859p1* in the Oracle charset setting. If these are not set consistently and a character is used that is contained in either the Sybase or Oracle character set and not in the other, you may receive an “ORA-00911: invalid character” message.

ECDA Option for Oracle converts the incoming client characters to the charset setting, which should match that of the Oracle setting. When characters are returned to the ECDA Option for Oracle client, they are then converted back to the character set setting of the client.

The language setting corresponds to the language in which the local ECDA Option for Oracle error messages are returned. The setting for Oracle Language and Territory coded in the languages stanza is the language in which the Oracle defined error messages will be sent back to the client. All attempts must be made to make sure the language settings match.

Default character set

If the default character set for ECDA Option for Oracle does not match that of ASE/CIS and a `writetext` is issued to insert text, the text field is not converted as expected. For the `us_english` language, this should not be a problem, because the normal printing characters are the same in the supported character sets. However, for other languages, this can be a problem.

The workaround is to make sure that ASE/CIS has the same default character set as ECDA Option for Oracle. For better performance, it is always best to use the same character set.

Blank-padded comparisons

If a column *c1* of type `char(5)` has a value “a” inserted into it, the following SQL statement will not return any rows if the table is on an Oracle database:

```
select...where c1 like "a"
```

Oracle does not make blank-padded comparison for arguments to like clauses. However, when executed against a table on ASE, the same SQL statement fetches the row.

Given the same setup, the following SQL statement returns the row when the table is on either ASE or Oracle:

```
select...where c1 = "a"
```

When `c1 = "a"` is used, Oracle performs blank-padded comparisons.

Passthrough mode

Passthrough mode allows a ECDA Option for Oracle client to interact directly with an Oracle database using native Oracle SQL. When ECDA Option for Oracle is operating as a standalone gateway, clients are automatically put into passthrough mode. When used with ASE, ECDA Option for Oracle performs Transact-SQL parsing.

In passthrough mode, the client program issues Oracle SQL statements directly to the Oracle database, and the results are converted into a form that the Open Client interface understands.

When in passthrough mode, Oracle datatypes are converted to the Open Client datatypes shown in Table B-1:

Table B-1: Oracle-to-Open Client datatype conversions for numeric data with create proxy_table

Oracle datatype	Open Client datatype number_mode=0	Open Client datatype number_mode=1	Open Client datatype number_mode=2
number(1,0)	tinyint	tinyint	numeric (1,0)
number(2,0)	tinyint	tinyint	numeric (2,0)
number(3,0)	smallint	smallint	numeric (3,0)
number(4,0)	smallint	smallint	numeric (4,0)
number(5,0)	int	int	numeric (5,0)
number(p,0) where 6 <= p <= 9	int	int	numeric (p,0)
number(p,0) where 10 <= p <= 38	numeric (p,0)	numeric (p,0)	numeric (p,0)
number(p,s) where 1 <= p <= 38 and 1 <= s <= 38	numeric (p,s)	numeric (p,s)	numeric (p,s)

When in passthrough mode, Oracle datatypes are converted to the ASE datatypes shown in Table 4-4:

Table 4-4: Oracle-to-ASE datatype conversions for non-numeric data with create proxy_table

Oracle datatype	Adaptive Server datatype
nchar(n)	varchar(n)
float	float
char(n)	char(n)
varchar(n)	varchar(n)
varchar2(n)	varchar(n)
raw(n)	binary(n)
clob	text
nclob	text
blob	image
date	datetime
bfile	image
long	text
long raw	image
timestamp	datetime

Datatype conversion

Table 4-5 illustrates datatype conversion that is performed when a create table or alter table statement is processed using ECDA Option for Oracle with ASE. ECDA Option for Oracle constructs syntax for the Oracle datatypes as shown:

Table 4-5: ASE-to-Oracle datatype conversion

ASE datatype	Oracle datatype
bit	char(1)
tinyint	number(3,0)
smallint	number(4,0)
int	number(10,0)
smallmoney	number(10,4)
money	number(19,4)
float	float
real	float
decimal(p,s)	number(p,s)
numeric(p,s)	number(p,s)
char(n)	char(n)
varchar(n)	varchar2(n)
unichar(n)	nchar(n)
nvarchar(n)	nvarchar2(n)
binary(n)	raw(n)
varbinary(n)	raw(n)
timestamp	raw(16)
text	CLOB
image	BLOB
datetime	date
smalldatetime	date
date	date
time	timestamp

When a create existing table command is processed, the datatype for each column specifies the type of conversion to be performed from the Oracle columns to ASE columns during query processing. Table B-2 describes the datatypes that can be used for existing Oracle datatypes.

Table B-2: ASE-to-Oracle datatype conversions

ASE datatype	Oracle datatype
int, smallint, tinyint, float, double, numeric, decimal	number
datetime	date
date	date
time	timestamp
datetime	timestamp
char(n)	char(n)
varchar(n)	varchar2(n) up to 4000 bytes
text	CLOB
binary(n) or varbinary(n)	raw(n) up to 4000 bytes
image	BLOB

Special datatype considerations

The following datatype considerations should be noted.

Joins with *char* and *varchar*

Joins between *char* columns and *varchar* columns may not return any rows. In addition, a query returns no rows when run directly against Oracle. However, when you run the query against ASE with the same data, it returns rows. The difference occurs because Sybase and Oracle have different comparison rules when the columns are not all fixed length. The workaround is to alter the table definitions so the column definitions match.

Padding *raw(n)*

Oracle pads *raw(n)* datatypes with blanks when users insert values less than *n*. SQL Server pads *binary(n)* datatypes with 0s when users insert values less than *n*.

An error may occur if an ASE user expects 0-padded data yet accesses blank-padded data. Likewise, there may be problems if native Oracle applications expect to see blank-padded data, yet access 0-padded data.

Precision and scale with number datatypes

Oracle allows number datatypes to have a scale larger than the precision. ASE numeric datatypes do not allow this.

Oracle allows number datatypes to have negative scales. ASE numeric datatypes do not allow this. If Oracle precision and scale are not specified, precision defaults to 38.

Minimum dates

If existing Oracle tables have datetime values with dates prior to the ASE minimum date (January 1, 1753 12:00:00:000AM), ECDA Option for Oracle will convert these values to the ASE minimum.

Unicode support

ASE unichar and univarchar datatypes are supported with Oracle 9i databases. Oracle errors will result if these datatypes are used with previous versions of Oracle.

To support the ASE unichar and univarchar datatypes the ECDA Option for Oracle must be configured to use the utf8 charset for both Sybase and Oracle as follows:

- The configuration option `charset =` must be set to `utf8`.
- The Sybase language mapping in the `[languages]` section of the configuration file must specify `utf8` as the Oracle charset.

The following example shows the configuration values needed to support Unicode for a server named “unidco” which uses `us_english` as its language.

```
[unidco]
charset = utf8
language = us_english
.
.
.
[languages]
;
; Maps a Sybase language to an Oracle Language,
Territory, and Charset
;
us_english          american          america
utf8
```

Millisecond support for Timestamp parameter for ECDA Option for Oracle

Two configuration properties, `timestamp_ms_support` and `timestamp_format`, are available to allow the ECDA Option for Oracle to insert, update, delete, and select Sybase DATETIME datatypes with the millisecond portion mapped to an Oracle TIMESTAMP column. The following sections describe the properties and their use.

Timestamp_ms_support parameter

Specifies the value of the `timestamp_ms_support` parameter.

Configuration parameter	<code>timestamp_ms_support</code>
Value	0 is off (default). 1 is on.

Timestamp_format parameter

Specifies the Oracle timestamp format to be used by non-ASE clients.

Configuration parameter	<code>timestamp_format</code>
Value	<i>MON DD YYYY HH:MM:SSAM</i> (default) provides the date and time in seconds, and the milliseconds portion contains <i>000</i> (zeros). <i>MON DD YYYY HH:MS:SS.FFAM</i> provides the date and time in milliseconds.

Using the timestamp_ms_support parameter

When you use the configuration parameter `timestamp_ms_support`, consider the following conditions:

- When `timestamp_ms_support` equals 0, Sybase TIME and DATETIME datatype is accessible, but the millisecond portion of an Oracle TIMESTAMP contains *000* (zeros).
- When `timestamp_ms_support` equals 1, Sybase TIME and DATETIME datatype is able to insert, update, delete, or select the millisecond portion of an Oracle TIMESTAMP.

ECDA Option for Oracle 15.0 interoperates with ASE/CIS

When ECDA Option for Oracle 15.0 interoperates with ASE/CIS, the version must be 12.5.1 ESD#2 or later to use the `timestamp_ms_support` configuration option. This is due to changes that were made in ASE/CIS 12.5.1.

ECDA Option for Oracle 15.0 interoperates with other applications

When ECDA Option for Oracle 15.0 interoperates with an application other than ASE/CIS, the `timestamp_format` configuration setting must include the milliseconds (...SS.FF) if the millisecond portion of the time is desired, and the `timestamp_ms_support` parameter must be equal to 1.

Date and time datatype support for ECDA Option for Oracle

Sybase date and time datatypes are supported in ECDA Option for Oracle. The following sections describe how ECDA Option for Oracle supports the new datatypes:

- Datatype mapping
- Incoming data
- Outgoing data

Datatype mapping

Oracle currently provides two date/time datatypes, Oracle DATE and Oracle TIMESTAMP:

- The Oracle DATE datatype can have both a date portion and a time portion, in which the precision of the time portion is at the seconds level. The Sybase DATE datatype maps to the existing Oracle DATE datatype.
- The Oracle TIMESTAMP datatype can have both a date portion and a time portion in which the precision of the time portion includes fractional seconds at the macro level. The Sybase TIME datatype maps to the existing Oracle TIMESTAMP datatype.

Incoming data

Cursor and dynamic statement parameter data for Sybase DATE and TIME datatypes are converted to the proper Oracle datatype. The Sybase TIME data destined for an Oracle TIMESTAMP column is converted based on the setting of the `timestamp_ms_support` parameter.

Incoming language statements with literal string data that is destined for either an Oracle DATE or TIMESTAMP column must conform to the format defined by the `date_format` and `timestamp_format` configuration settings. This implies that column data coming from a Sybase DATE column must include a default time of 12:00:00AM, in addition to the date value. The column data coming from a Sybase TIME column must include a default date of Jan 1 1900 in addition to the time value.

Incoming RPC statements with Sybase DATE or TIME literal string parameters must format the literal string.

Outgoing data

Data being retrieved from both Oracle DATE and TIMESTAMP is returned to the application as a date/time datatype.

Global variables

Global variables are available by using the command:

```
select @@variable_name
```

For example:

```
> select @@servername, @@version
> go
```

Table 4-6 describes the available global variables:

Table 4-6: Description of available global variables

Global variable	Description
@@autocommit	Returns the autocommit setting.
@@compatibility_level	Returns the current configured value for number_mode and truncate_text. truncate_text exists for compatibility only.
@@child_logfile	Returns the file specification of the errorlog. This variable has been replaced and is here for compatibility only. New applications should use @@errorlog instead.
@@client_csname	Returns the character set in use by the current connection.
@@connect_info	Returns the configured value for 'connect_string'. For compatibility only.
@@connect_string	Returns the connect_string that the service is using.
@@default_charset	Returns the charset name used when the server was started.
@@errorlog	Returns the file specification of the errorlog.
@@max_users	Returns the max_users run value.
@@maxpagesize	Returns the maxpagesize run value.
@@language	Returns the language in use by the current connection.
@@ncharsize	Returns the size of an nchar character.
@@nls_lang	Returns the language, territory and character set in use by the Oracle connection.
@@oracle_version	Returns the version of Oracle to which a connection has been established.
@@servername	Returns the current server name to which the client has connected.
@@spid	Returns the process ID of the current connection.
@@sql_transform	Returns the current sql_transform setting.
@@textsize	Returns the current textsize value.

Global variable	Description
@@version	Returns the version string of the DirectConnect server.

Set commands

Set commands only affect the connection on which they are issued. Other connections and the configuration file remain unchanged.

set autocommit

The command `set autocommit on | off | 0 | 1` turns autocommit on or off.

set rowcount

The command `set rowcount value` limits the number of rows returned in a result set. The default is to return all rows.

set textsize

The command `set textsize value` limits the size of a text or image column that is returned through a `select` statement. The default is 32KB.

set sql_transform

The command `set sql_transform on | off | 0 | 1` enables or disables Transact SQL transformation. The default is `off`. When enabled, the ECDA Option for Oracle parser can parse a limited subset of Transact SQL statements and recognize Transact SQL datatype names.

set traceon

The command `set traceon [traceflag]` will turn on specific trace flags in the program. The trace flags are described in the Configuration file section, Chapter 2, “Configuring and Operating ECDA Option for Oracle.” These are the available traceflags:

Table B-3: Trace flag descriptions

Trace flag number	Description
1	Initialization event logging
2	Connect/disconnect/attention
3	Language event logging
4	RPC event logging
5	Cursor event logging
6	Dynamic event logging
7	Bulk event logging
8	DBMS message logging
9	RDBMS interaction logging
10	Dump login record
11	Trace done packets
12	Force exit on shutdown
13	Trace XA (ASTC, DTM) RPCs
14	Enable XA LIB tracing. The output is written to <code><install_dir>/DC-15_0/rdbms/xa_servicenameDate.trc</code>
15	Log parser errors
16	License check message logging

set traceoff

The command `set traceoff [traceflag]` turns off specific trace flags. If no trace flag is provided, all trace flags are cleared.

Two-phase commit

❖ To use two-phase commit with ASE servers

- 1 Connect to an ASE.

- 2 Add an external login entry for the DirectConnect server for the “probe” login which is used to recover two-phase commit transactions. The Oracle login that is used must have the necessary privileges to recover transactions for any user.

For example:

```
sp_addexternlogin dco_server, probe, system, manager
```

- 3 Configure the DirectConnect server to use two phase commit.

For example:

```
dco_server...sp_configure dco-server,  
two_phase_commit, 1
```

- 4 Shut down and restart the DirectConnect server.
- 5 Shut down and restart the ASE.

Lightweight Directory Access Protocol (LDAP)

Lightweight Directory Access Protocol (LDAP) is the Internet directory protocol. LDAP is a protocol for accessing and managing directory services. The LDAP support replaces the *interfaces* file for UNIX and the *sql.ini* file for Windows. This means that the connection information normally stored in an *interfaces* or *sql.ini* file can now be stored in an LDAP server.

You must have an LDAP server running. Also, you need to verify that the port number of the agent on your server machine is added to the LDAP server.

Setting up LDAP for UNIX

In the `<install_dir>/DC-15_0/connectivity/OCS-15_0/config` directory, create a *libtcl.cfg* file and add the following lines under [DIRECTORY]:

```
; This code shows how to specify the different types  
; of drivers that will be used by Client-Library and  
; Server-Library applications.  
;  
; This is the sample for SUN Solaris platform.  
;  
[DRIVERS]  
;libtli.so=tcp unused ; This is the non-
```

```

threaded tli driver.

;libtli_r.so=tcp unused           ; This is the threaded
tli driver.

[DIRECTORY]
ldap=libldldap.so
ldap://<LDAP server location >

```

Then, set up your Sybase environment variables:

```

setenv SYBASE <release path>
setenv SYBASE_OCS OCS-15_0
setenv SYBASE_TCL_CFG <your libtcl.cfg pathname>
setenv LD_LIBRARY_PATH
<install_dir>/OCS_15_0/lib:$SYBASE/$SYBASE_OCS/lib3p

setenv PATH <install_dir>/OCS_15_0/bin:$SYBASE:$PATH

```

Use `<install_dir>/DCO_15_0/connectivity/OCS-15_0/bin/dscp` to add an entry for your server into the LDAP server the same way you would for an ASE server. Check the *Adaptive Server Enterprise 15.0 Utility Guide* for more information on using `dscp`.

To test the new setup, `isql` into the server *without* using the *interfaces* file.

Setting up LDAP for Windows

This section describes how to set up LDAP for the Microsoft Windows platform.

❖ To set up LDAP for Windows

- 1 In the `C:\install_dir\DCO-15_0\connectivity\OCS-15_0\ini` directory, create a `libtcl` file and add the following lines under `[DIRECTORY]`:

```

; This is the sample Net-Library configuration file
; that shows how to specify the different types of
; drivers that will be used by Client-Library and
; Server-Library applications.
;
; This is the sample for Windows platforms.

[DRIVERS]

NLWNSCK=TCP Sockets
NLWNSCK=TCP/IP Sockets

```

```
[NT_DIRECTORY]

ldap=libldap ldap://

[SECURITY]

;NTLM=LIBSMSSP

[FILTERS]

;SSL=LIBFSSL
```

- 2 Set the following Sybase environment variables (assuming that the ECDA Option for Oracle is installed on the “d” drive in the Sybase directory):

```
set SYBASE=d:\Sybase
set SYBASE_TCL_CFG=d:\as150_IR\OCS-15_0\ini\libtcl.cfg
set SYBASE_OCS=OCS-15_0
set PATH=%SYBASE%;%SYBASE%\%SYBASE_OCS%\dll;
%SYBASE%\%SYBASE_OCS%\lib3p;%SYBASE%\%SYBASE_OCS%\lib;
%SYBASE%\%SYBASE_OCS%\ini;%SYBASE%\%SYBASE_OCS%\bin;
%PATH%
```

- 3 Add your server entry to the LDAP server:

```
%SYBASE%\DC-15_0\connectivity\OCS-15_0\bin\dsedit
```
- 4 Add an entry for your server into the LDAP server, following the same steps that you would for an ASE server. For more information about using dsedit, refer to the Adaptive Server Enterprise *Utility Guide*.
- 5 To test the new setup, isql into the server *without* using the *sql.ini* file.

Glossary

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

AIX	See Advanced Interactive Executive .
AMD2	The component of the Mainframe Connect DB2 UDB Option that allows clients to submit SQL statements to DB2 UDB. It is a CICS transaction that receives SQL statements sent from Mainframe Connect DirectConnect for z/OS Option and submits them to DB2 UDB, using the DB2 UDB dynamic SQL facility. It also receives the results and messages from DB2 UDB and returns them to Mainframe Connect DirectConnect for z/OS Option.
American Standard Code for Information Interchange	The standard code used for information interchange among data processing systems, data communication systems, and associated equipment. The code uses a coded character set consisting of 7-bit coded characters (including a parity check, 8 bits).
API	See application program interface .
APPC	See advanced program-to-program communication .
application program interface	The programming language interface between the user and Mainframe Connect Client Option or Mainframe Connect Server Option. The API for Mainframe Connect Client Option is Client-Library. The API for Mainframe Connect Server Option is Gateway-Library.
ASCII	See American Standard Code for Information Interchange .
ASE	See Adaptive Server Enterprise .
ASE/CIS	See Adaptive Server Enterprise/Component Integration Services .
batch	A group of records or data processing jobs brought together for processing or transmission.
bind	In the Sybase environment, this term has different meanings depending on the context: <ul style="list-style-type: none">• In CICS, it is an SNA command used to establish a connection between LUs, or a TCP/IP call that connects an application to a port on its system.• In DB2 UDB, it compiles the Database Request Module, the precompiler product that contains SQL statements in the incoming request, and produces an access plan, a machine code version of the SQL statements that specifies the optimal access strategy for each statement.• In the mainframe access product set, it establishes a connection between a TRS port and a CICS or IMS region.

bulk copy transfer	A transfer method in which multiple rows of data are inserted into a table in the target database. Compare with destination-template transfer and express transfer .
call level interface	A programming style that calls database functions directly from the top level of the code. Contrast with embedded SQL .
catalog	A system table that contains information about objects in a database, such as tables, views, columns, and authorizations.
catalog RPC	A component of the Mainframe Connect DB2 UDB Option that allows clients to access DB2 UDB system catalogs. It uses an interface compatible with the catalog interface for the ODBC API.
catalog stored procedure	A procedure used in SQL generation and application development that provides information about tables, columns, and authorizations.
character set	A set of specific (usually standardized) characters with an encoding scheme that uniquely defines each character. ASCII is a common character set.
CICS	See Customer Information Control System .
CICS region	The instance of CICS.
client	In client/server systems, the part of the system that sends requests to servers and processes the results of those requests. See also client/server . Compare with server .
client application	Software responsible for the user interface that sends requests to applications acting as servers. See also client/server .
Client-Library	A library of routines that is part of Mainframe Connect Client Option.
client request	An RPC or language request sent by a client to a server.
client/server	An architecture in which the client is an application that handles the user interface and local data manipulation functions, and the server is an application providing data processing access and management. See also client application .
Client Services Application	A customer-written CICS program initiated on the host that uses the API to invoke the Mainframe Connect Client Option as a client to the ECDA Option for Oracle server or to ASE. See also application program interface , Client Services for CICS .

Client Services for CICS	A Sybase host API that invokes the Mainframe Connect Server Option as a client to an access service for DB2 UDB or ASE. See also application program interface, Customer Information Control System, Client Services Application, Mainframe Connect Server Option.
clustered index	An index in which the physical order and the logical (indexed) order is the same. Compare with nonclustered index.
code page	An assignment of graphic characters and control function meanings to all code points.
commit	A process that makes permanent all changes made to one or more database files since the initiation of the application program, the start of an interactive session, or the last commit or rollback operation. Compare with rollback.
Common Programming Interface	Specifies the languages and services used to develop applications across SAA environments. The elements of the CPI specification are divided into two parts: processing logic and services.
configuration file	A file that specifies the characteristics of a system or subsystem.
configuration set	A section into which service library configuration files are divided.
conversion	The transformation between values that represent the same data item but which belong to different datatypes. Information can be lost due to conversion, because accuracy of data representation varies among different datatypes.
connection	A network path between two systems. For SNA, the path connects a logical unit (LU) on one machine to an LU on a separate machine. For TCP/IP, the path connects TCP modules on separate machines.
connection router	A program provided with Mainframe Connect Client Option that directs requests to particular remote servers. Mainframe system programmers use the connection router to define remote servers and server connections to Mainframe Connect Client Option.
Connection Router Table	A memory-resident table maintained by a Mainframe Connect Client Option system programmer that lists servers and the connections that a Client-Library transaction can use to access them.
control section	The part of a program specified by the programmer to be a relocatable unit, all elements of which are to be loaded into adjoining main storage locations.
control statement	In programming languages, a statement that is used to alter the continuous sequential execution of statements. A control statement can be a conditional statement or an imperative statement.

conversation-level security	The passing of client login information to the mainframe by TRS when it allocates a conversation.
CSA	See Client Services Application .
CSP	See catalog stored procedure .
cursor	In SQL, a named control structure used by an application program to point to a row of data.
Customer Information Control System	An IBM licensed program that enables transactions entered at remote terminals to be processed concurrently by user-written application programs.
DASD	See direct access storage device .
data definition statement	An IBM mainframe statement used to relate a name with a file.
data definition language	A language for describing data and data relationships in a database.
data set name	The term or phrase used to identify a data set.
database management system	The term or phrase to identify a data set. A computer-based system for defining, creating, manipulating, controlling, managing, and using databases.
database operation	A single action against the database. For Mainframe Connect DirectConnect for z/OS Option, a database operation is usually a single SQL statement. One or more database actions can be grouped together to form a request. See also request .
Database 2	An IBM relational database management system.
datatype	A keyword that identifies the characteristics of stored information on a computer.
DB-Library	A Sybase and Microsoft API that allows client applications to interact with ODS applications. See also application program interface .
DBMS	See database management system .
DB2 UDB	See Database 2 .
DDL	See data definition language .
DD statement	See data definition statement .
default language	The language that displays a user's prompts and messages.

destination-template transfer	A transfer method in which source data is briefly put into a template where the user can specify that some action be performed on it before execution against a target database. See also transfer . Compare with bulk copy transfer and express transfer .
direct access storage device	A device in which access time is effectively independent of the location of the data.
direct request	A request sent directly from a client workstation through Transaction Router Service to the DirectConnect server without going through ASE. 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 ASE, it provides many of the features of a distributed database system, such as location transparency, copy transparency, and distributed joins.

embedded SQL	SQL statements that are embedded within a program and are prepared in the process before the program runs. After it is prepared, the statement itself does not change, although values of host variables specified within the statement might change.
end user	A person who connects to a DirectConnect server using an application to access databases and perform transfers. See also transfer .
Enterprise Connect Data Access	An integrated set of software applications and connectivity tools that allow access to data within a heterogeneous database environment, such as a variety of LAN-based, non-Sybase data sources, as well as mainframe data sources.
environment variable	A variable that describes how an operating system runs and the devices it recognizes.
exit routine	A user-written routine that receives control at predefined user exit points.
express transfer	A form of bulk copy transfer that uses ODBC bulk APIs to improve performance when transferring bulk data between data sources. Because it uses the same syntax as bulk copy transfer, no modification of applications is required.
external call interface	A CICS client facility that allows a program to call a CICS application as if the calling program had been linked synchronously from a previous program instead of started from a terminal.
External Security Manager	An add-on security package for the z/OS mainframe, licensed by Computer Associates.
FCT	See forms control table .
forms control table	An object that contains the special processing requirements for output data streams received from a host system by a remote session.
gateway	Connectivity software that allows two or more computer systems with different network architectures to communicate.
Gateway-Library	A library of communication, conversion, tracing, and accounting functions supplied with Mainframe Connect Server Option.
globalization	The combination of internationalization and localization. See internationalization , localization .
global variable	A variable defined in one portion of a computer program and used in at least one other portion of the computer program. Contrast with local variable .

handler	A routine that controls a program's reaction to specific external events, for example, an interrupt handler.
host	The mainframe or other machine on which a database, an application, or a program resides. In TCP/IP, this is any system that is associated with at least one Internet address. See also Transmission Control Protocol/Internet Protocol .
host ID	In Mainframe Connect Server Option, the ID that the TRS passes to the mainframe with a client request. The host ID is part of the client login definition at the TRS.
host password	In Mainframe Connect Server Option, the password that the client passes to the mainframe with a client request.
host request library	A DB2 UDB table that contains host-resident SQL statements that can be executed dynamically. See also host-resident request .
host-resident request	A SQL request that resides in a DB2 UDB table called the host request library. See also host request library .
IMS	See Information Management System .
indirect request	A client request that is routed through a stored procedure on a SQL Server, which forwards the request to TRS as an RPC. Compare with direct request .
Information Management System	A database/data communication system that can manage complex databases and networks.
interfaces file	An operating system file that determines how the host client software connects to a Sybase product. An <i>interfaces</i> 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 SQL Server application that maintains replicated data and processes data transactions received from a data source.
request	One or more database operations an application sends as a unit to the database. Depending upon the response, the application commits or rolls back the request. See also commit, rollback, unit of work .
resource table	A main storage table that associates each resource identifier with an external logical unit (LU) or application program.
rollback	An instruction to a database to back out of changes requested in a unit of work. Compare with commit .
router	An attaching device that connects two LAN segments, which use similar or different architectures, at the Open System Interconnection (OSI) reference model network layer. Contrast with gateway .
RPC	See remote procedure call .
RSP	See remote stored procedure .
SAA	See System Application Architecture .
secondary connection	The connection specified in the transfer statement. It represents anything that can be accessed using Mainframe Connect Client Option, such as ASE or another access service.
secondary database	In transfer processing, the supported database that is specified in the transfer statement. Compare with primary database .
server	A functional unit that provides shared services to workstations over a network. See also client/server . Compare with client .
server process ID	A positive integer that uniquely identifies a client connection to the server.
service	A functionality available in Mainframe Connect DirectConnect for z/OS Option. It is the pairing of a service library and a set of specific configuration properties.

service library	In Mainframe Connect DirectConnect for z/OS Option, a set of configuration properties that determine service functionality. See also access service library , administrative service library , Transaction Router Service library , transfer service library .
service name redirection	A type of service name resolution that allows a system administrator to create an alternative mechanism to map connections with services. See also service name resolution . Compare with direct resolution .
service name redirection file	The default name of the file used for the service name redirection feature. See service name redirection .
service name resolution	The DirectConnect server mapping of an incoming service name to an actual service. See also direct resolution , service name redirection .
session	A connection between two programs or processes. In APPC communications, sessions allow transaction programs to have conversations between the partner LUs. See also advanced program-to-program communication .
short transaction	A mainframe transaction that ends the communication when it finishes returning results to the client. Compare with long-running transaction .
SNA	See Systems Network Architecture .
SNRF	See service name redirection file .
SPID	See server process ID .
SQL	See structured query language .
SQLDA	See SQL descriptor area .
sqledit	A utility for creating and editing <i>sql.ini</i> 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 ASE.
SQL descriptor area	A set of variables used in the processing of SQL statements.
SQL stored procedure	A single SQL statement that is statically bound to the database. See also stored procedure .
static SQL	SQL statements that are embedded within a program and prepared during the program preparation process before the program runs. Compare with dynamic SQL .

stored procedure	A collection of SQL statements and optional control-of-flow statements stored under a particular name. Adaptive Server stored procedures are called “system procedures.” See also remote stored procedure, system procedures.
structured query language	An IBM industry-standard language for processing data in a relational database.
stub	A program module that transfers remote procedure calls (RPCs) and responses between a client and a server.
SYRT	The component of Mainframe Connect DB2 UDB for IMS that allows clients to submit SQL language requests to DB2 through IMS.
System Administrator	The person in charge of server system administration, including installing and maintaining DirectConnect servers and service libraries.
System Application Architecture	An IBM proprietary plan for the logical structure, formats, protocols, and operational sequences for transmitting information units through networks and controlling network configuration and operation. See also advanced program-to-program communication.
system procedures	A stored procedure that ASE supplies for use in system administration. System procedures serve as shortcuts for retrieving information from system tables, or a mechanism for accomplishing database administration. See also stored procedure.
Systems Network Architecture	An IBM proprietary plan for the structure, formats, protocols, and operational sequences for transmitting information units through networks. See also advanced program-to-program communication.
table	An array of data or a named data object that contains a specific number of unordered rows. Each item in a row can be unambiguously identified by means of one or more arguments.
Tabular Data Stream	A Sybase application-level protocol that defines the form and content of relational database requests and replies.
target	A system, program, or device that interprets, rejects, satisfies, or replies to requests received from a source.
target database	The database to which the DirectConnect server transfers data or performs operations on specific data.
TCP/IP	See Transmission Control Protocol/Internet Protocol.
TDS	See Tabular Data Stream.

transaction	A unit of processing initiated by a single request. A transaction consists of one or more application programs that, when executed, accomplish a particular action. In Mainframe Connect Server Option, a client request (RPC or language request) invokes a mainframe transaction. In Mainframe Connect Client Option, a mainframe transaction executes a stored procedure on a remote server.
transaction processing	A sequence of operations on a database that is viewed by the user as a single, individual operation.
Transaction Router Service	A Mainframe Connect DirectConnect for z/OS Option program used when the mainframe acts as a transaction server to route requests from remote clients to the Mainframe Connect Server Option and return results to the clients.
Transaction Router Service library	A service library that facilitates access to remote transactions, allowing customers to execute transactions from virtually any mainframe data source. See also service library .
Transact-SQL	A Sybase-enhanced version of the SQL database language used to communicate with ASE.
transfer	A Mainframe Connect DirectConnect for z/OS Option feature that allows users to move data or copies of data from one database to another.
transfer service library	A service library that provides copy management functionality. See also service library .
Transmission Control Protocol/Internet Protocol	A set of communication protocols that supports peer-to-peer connectivity functions for both local and wide area networks.
trigger	A form of stored procedure that automatically executes when a user issues a change statement to a specified table.
TRS	See Transaction Router Service .
TRS library	See Transaction Router Service library .
T-SQL	See Transact-SQL .
unit of work	One or more database operations grouped under a commit or rollback. A unit of work ends when the application commits or rolls back a series of requests, or when the application terminates. See also commit , rollback , transaction .
user ID	User identification. The ID number by which a user is known in a specific database or system.

variable	An entity that is assigned a value. Mainframe Connect ECDA Option for Oracle for z/OS Option has two kinds of variables: <i>local</i> and <i>global</i> .
view	An alternate representation of data from one or more tables. A view can include all or some of the columns contained the table or tables on which it is defined.
Virtual Storage Access Method	An IBM-licensed program that controls communication and the flow of data in an SNA network.
Virtual Telecommunications Access Method	IBM mainframe software that allows communication on an SNA network between mainframes and allows the mainframe to have multiple sessions per connection.
VSAM	See Virtual Storage Access Method .
VTAM	See Virtual Telecommunications Access Method .
wildcard	A special character that represents a range of characters in a search pattern.

Index

Symbols

- , (comma)
 - in SQL statements viii
- { } (curly braces)
 - in SQL statements viii
- () (parentheses)
 - in SQL statements viii
- [] (square brackets)
 - in SQL statements viii

A

- accessing ECDA Option for Oracle
 - using isql 16
- adding a service 10
 - using DirectConnect Manager 10
 - using sp_addservice 10, 25
- administrator 50
- architecture 2
- array_size 56
- ASE
 - configuring 16
- attention events, tracing
 - Open Server 54
- autocommit 56

B

- brackets
 - See square brackets [] and angle brackets < >

C

- CA trusted roots certificate file 43
- certauth utility 45
- certificate file

- copying 45
- certificate_file 51
- certificate_password 51
- certificates
 - trusted roots file 43
- certreq utility 45
- changing configuration parameters
 - using sp_configure 27
- charset 51
- comma (,)
 - in SQL statements viii
- compatibility with Adaptive Server data 7
- configuration file 5, 49
 - contents 5
- configuration parameters
 - displaying values 27
 - global 49
 - modifying values 27
 - service-specific 54
- configuration properties
 - global 7
 - service-specific 7
- configuring
 - ASE for DirectConnect 16
 - using DirectConnect Manager 6
 - using sp_configure 6
- connect_string 56
- connection handling 63
- connectivity
 - Secure Sockets Layer (SSL) 45
 - setting up 45
 - setting up for Oracle 8
- connectivity files
 - listener.ora 8
 - sqlnet.ora 9
 - tnsnames.ora 9
- conventions
 - See also syntax
- copying
 - certificate file 45

Index

- trusted roots file 45
- creating
 - runserver file 12
 - stored procedures 19
- csp_uppercase 57
- curly braces ({})
 - in SQL statements viii

D

- data compatibility 7
- database objects
 - sp_tables list of 36
- datatype conversion 68
 - special considerations 69
- datatypes
 - number 70
 - text and image 64
- date_format 57
- dcoracle -v command 16
- dcoracle.exe -v command 16
- default_precision 57
- default_scale 58
- defining the Oracle server 16
- direct_connect server class 2
- displaying version 16
- dropping a service 11
 - using sp_dropservice 29
- dsedit utility 10

E

- environment variables 7
 - ORA_NLS33 8
 - setting 7
- error log file
 - specifying in the runserver file 12
- errorlog 51
- events
 - logging 54
- examples
 - listener.ora file 9
 - setting up connectivity 45
 - sqlnet.ora file 9

- tnsnames.ora file 9

F

- files
 - runserver 12

G

- global parameter
 - administrator 50
 - certificate_file 51
 - certificate_password 51
 - charset 51
 - errorlog 51
 - interfaces 51
 - language 51
 - max_packetsize 52
 - max_pagesize 52
 - max_stacksize 52
 - max_textsize 52
 - max_threads 52
 - max_users 53
 - network_tracing 53
 - srv_traceflags 53
 - text_chunksize 54
 - trusted_roots_file 54
- global parameters 49
- global properties 7
- global variables 73

I

- indexes
 - sp_statistics information on 33
- insert_array_size 58
- interfaces 51
- interfaces file
 - specifying in the runserver file 12
- interfaces file (UNIX) 10
- internationalization 65
- isql utility
 - using to access ECDA Option for Oracle 16

L

language 51
 Lightweight Directory Access Protocol (LDAP) 76
 listener.ora file 8
 localization 65
 logging events using trace flags 54

M

max_packetsize 52
 max_pagesize 52
 max_stacksize 52
 max_textsize 52
 max_threads 52
 max_users 53
 minimum dates 70

N

network requests, tracing 54
 network_tracing 53
 number datatypes
 precision and scale 70
 number_mode 58

O

Open Server
 tracing attention events 54
 tracing events 54
 Open Server threads
 retrieving status information 29
 terminating 37
 Open Server trace flags 53
 ORA_NLS33 environment variable 8
 Oracle Call Interface (OCI) 2
 Oracle connectivity 8
 Oracle server definition 16
 Oracle version
 commands to display 16

P

package and procedure definitions 20
 padding raw(n) 69
 parameters
 global 49
 service-specific 54
 parentheses ()
 in SQL statements viii
 passthrough mode 63, 66
 precision and scale with number datatypes 70
 properties
 global 7
 service-specific 7

Q

queries
 sp_tables and 36

R

raw(n)
 padding 69
 remote procedure calls (RPCs)
 executing with an input parameter 22
 executing with input and output parameters 22
 mapping to return an Adaptive Server table 21
 passed to Oracle 19
 processed by ECDA Option for Oracle 23
 restrictions 23
 removing a service
 using DirectConnect Manager 11
 using sp_dropservice 11
 retrieving status information on Open Server threads
 using sp_ps 29
 runserver file 12
 flags for 12

S

Secure Sockets Layer (SSL) 41
 connectivity 45
 enabling 44

Index

- filter 41
- handshake 43
- performance considerations 44
- security 41
- server certificate file
 - location 43
- services
 - adding 10
 - dropping 11
- service-specific parameter
 - array_size 56
 - autocommit 56
 - connect_string 56
 - csp_uppercase 57
 - date_format 57
 - default_precision 57
 - default_scale 58
 - insert_array_size 58
 - number_mode 58
 - session_time_limit 59
 - timestamp_format 59
 - timestamp_ms_support 60
 - timestamp_tz_format 60
 - traceflags 61
 - triggers 61
 - two_phase_commit 61
- service-specific parameters 54
- service-specific properties 7
- session_time_limit 59
- set autocommit 74
- set commands 74
- set rowcount 74
- set sql_transform 74
- set textsize 74
- set traceoff 75
- set traceon 75
- setsslreq utility 46
- setting environment variables 7
- setting up
 - connectivity 45
 - Oracle connectivity 8
- shutting down ECDA Option for Oracle 15, 32
- sp_addserver
 - using to define the Oracle server 16
- sp_addservice 25
- sp_columns 25
- sp_configure 7
- sp_dropservice 29
- sp_ps 29, 32
- sp_shutdown 15, 32
- sp_statistics catalog stored procedure 33
- sp_stored_procedures catalog stored procedure 35
- sp_tables catalog stored procedure 36
- sp_terminate 37
- sp_text_configure 27
- sp_text_configure system procedure 29
- sp_traceoff 38
- sp_traceon 38
- sp_who 40
 - finding information about Open Server threads 39
- SPID parameter 30
- SQL command handling 63
- sql.ini file (Windows) 10
- sqlnet.ora connectivity file 9
- square brackets []
 - in SQL statements viii
- srv_traceflags 53
- SSL Plus library API 41
- starting ECDA Option for Oracle
 - on UNIX 12
 - on Windows 12
- startserver utility 12
- start-up commands
 - Windows 13
- stored procedures
 - creating 19
 - sp_stored_procedures information on 35
 - within a package 20
- symbols in SQL statements viii
- system procedures
 - sp_text_configure 29

T

- tables
 - sp_tables 36
- TDS
 - data, tracing 54
 - headers, tracing 54
 - tokens, tracing 54
- terminating an Open Server thread 37

- text and image datatypes 64
- text_chunksize 54
- thread information 31
- Timestamp property
 - millisecond support 71
 - parameters 71
- timestamp_format 59
- timestamp_ms_support 60
- timestamp_tz_format 60
- tnsnames.ora connectivity file 9
- trace flags
 - Open Server 53
- traceflags 61
 - turning off 38
 - turning on 38
- triggers 61
- trusted roots certificate 43
- trusted roots file
 - copying 45
- trusted_roots_file 54
- two-phase commit 61
- two-phase commit feature
 - using with ASE servers 75

U

- Unicode support 70
- utilities
 - certauth 45
 - certreq 45
 - dsedit 10
 - isql 16
 - setsslreq 46
 - startserver 12

V

- variables
 - global 73
- version of Oracle
 - commands to display 16
- viewing column information
 - using sp_columns 25

W

- writetext command 64

