SYBASE®

Server Administration and User's Guide

Enterprise Connect Data Access Option for Oracle

12.6

[Microsoft Windows, UNIX, and Linux]

DOCUMENT ID: DC34201-01-1260-01

LAST REVISED: February 2005

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

Contents

About This Book

This book describes the features and functionality of the product and how to use the DirectConnect™ for Oracle product with Sybase® Adaptive Server® Enterprise.

Audience

This book is for System Administrators who are configuring a DirectConnect for Oracle and for users who are accessing remote date with the product.

How to use this book

This book includes the following chapters:

Chapter 1, “Introduction,” provides an overview of DirectConnect for Oracle.

Chapter 2, “Configuring and Operating DirectConnect for Oracle,” describes the components of the DirectConnect for Oracle and how to use DirectConnect for Oracle.

Chapter 3, “Reference Topics for Oracle,” contains reference material for DirectConnect for Oracle.

Chapter 4, “Executing Remote Procedure Calls,” describes remote procedure calls unique to DirectConnect for Oracle.

Appendix A, “SSL Connectivity,” describes the steps required for SSLconnectivity to DirectConnect for Oracle (DCO) from the isql client.

Related documents

For information regarding the installation of DirectConnect 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 Sybase Technical Library CD, and the Technical Library 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 Technical Library CD. It is included with your software. To read or print documents on the Getting Started CD, you need Adobe Acrobat Reader (downloadable at no charge from the Adobe Web site, using a link provided on the CD).
- The Technical Library CD contains product manuals and is included with your software. The DynaText reader (included on the Technical Library CD) allows you to access technical information about your product in an easy-to-use format.

  Refer to the *Technical Library Installation Guide* in your documentation package for instructions on installing and starting the Technical Library.
- The Technical Library Product Manuals Web site is an HTML version of the Technical Library 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 Technical Library 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. Select Products from the navigation bar on the left.
3. Select a product name from the product list and click Go.
4. Select the Certification Report filter, specify a time frame, and click Go.
5. Click a Certification Report title to display the report.

❖ **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 *%SYBASE%*, and replace slashes (/) with back slashes (\). For example, replace this user input:

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

with:

```
%SYBASE%\%SYBASE_DCO%\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.
variable	Variables, or words that stand for values that you fill in, are in *italics*.
{ }	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.

If you need help

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

CHAPTER 1

Introduction

This chapter introduces basic DirectConnect for Oracle concepts, and describes the hardware and software requirements.

It contains the following topics:

Before Enterprise Connect Data Access 12.5, its options were sold as individual DirectConnect products named “DirectConnect for [target].” Other than in the installer, you will see the name “DirectConnect” used in the software and in documents. As such, this document uses the old product name DirectConnect for Oracle except for the title page which identifies the product by its new name, Enterprise Connect Data Access Option for Oracle.

Overview

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

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

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

In standalone mode, DirectConnect 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.

Hardware and software requirements

The following hardware and software configurations for Windows and UNIX platforms are compatible with DirectConnect for Oracle release 12.6.

Table 1-1: Hardware and Software Requirements for Windows

Item	Requirements
CPU	Sybase recommends an Intel-compatible processor with a minimum of 500 megahertz.
RAM	512MB of RAM to run Windows 2000 and Windows 2003 for DirectConnect products.
Software	Microsoft Windows 2000 and Windows 2003.
Storage	A minimum of 300MB, plus at least 500KB bytes for each locale you plan to support.

Table 1-2: Hardware and Software Requirements for UNIX

Item	HP 9000(8xx)	RS/6000 AIX	Sun Solaris	Linux
CPU	An HP 9000/800	An IBM RISC System/6000	A Sun Solaris (SPARC) system	Linux Red Hat 2.1
RAM	Minimum of 64MB	Minimum of 64MB	Minimum of 64MB	Minimum of 64MB
Storage	Minimum of 300MB, plus at least 500KB for each *locale* you plan to support.	Minimum of 300MB, plus at least 500KB for each *locale* you plan to support	Minimum of 300MB, plus at least 500KB for each *locale* you plan to support	Minimum of 300MB, plus at least 500KB for each *locale* you plan to support
Platform requirements	HP 9000/800 and 900 HP-UX 11.11, and 11.23	AIX 5.2, and 5.3	Sun Solaris 2.8, and 2.9	Linux Red Hat 2.1, and 3.0

Disk requirements

DirectConnect for Oracle requires approximately 125MB of disk space on Windows and 280MB on UNIX.

Oracle requirements

DirectConnect for Oracle supports Oracle version 9i or 10g.

CHAPTER 2

Configuring and Operating DirectConnect for Oracle

This chapter explains how to configure and run DirectConnect for Oracle.

It contains the following topics:

Understanding the architecture

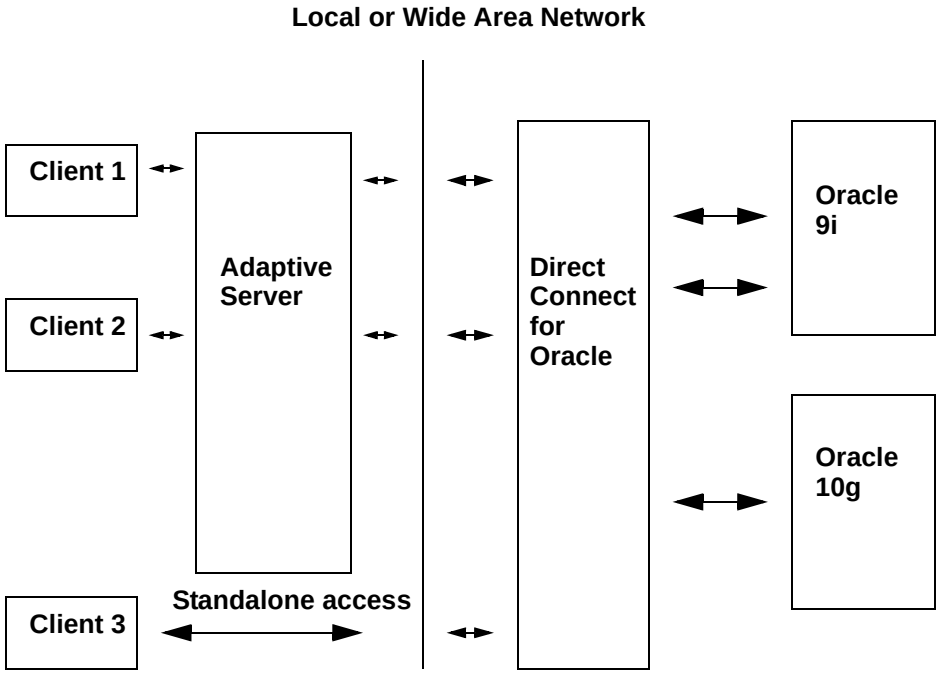
Before you configure DirectConnect for Oracle, you should have a basic understanding of how the product works.

DirectConnect for Oracle accesses Oracle databases using the Oracle Call Interface (OCI) application programming interface (API). The OCI API allows DirectConnect for Oracle to manipulate data in an Oracle database.

DirectConnect for Oracle itself is built using the client/server paradigm. DirectConnect for Oracle is a Sybase Open Server application that provides the flexibility of a number of concurrent client connections that can be configured.

DirectConnect for Oracle communicates with Adaptive Server Enterprise using server class *direct_connect*.

The following diagram shows how DirectConnect for Oracle works with Adaptive Server Enterprise as a standalone gateway:



Configuration file

When DirectConnect for Oracle is installed for the first time, a configuration file for DirectConnect for Oracle is automatically created for it. The configuration file, a plain text file which contains definitions for DirectConnect, is located in *$SYBASE/$SYBASE_DCO/install* for UNIX, or *%SYBASE%\%$SYBASE_DCO%\install* for Windows.

The configuration file must have the same name as the DirectConnect 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 DirectConnect 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 following lists the contents of the DirectConnect for Oracle configuration file

- Contains all of the configuration information for a particular server.
- An instance of a DirectConnect for Oracle 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 DirectConnect for Oracle to have different attributes or targets.
- A single configuration file for an instance of DirectConnect for Oracle defines one or more services.
- One of the service names must be the same as the server name you selected. All global properties 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.
- You can modify the configuration file by using DirectConnect Manager or by using sp_configure.

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 DirectConnect for Oracle (DCO) Service Configuration Properties.

Using sp_configure

A configuration file is created automatically at the time of installation. If you need to modify the configuration after installation, see sp_configure on page 55.

Use sp_configure to display or edit the configuration options. The syntax for sp_configure is:

```
sp_configure [service_name [, option_name [, value]]]
```

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 parameters, the configuration file must be edited manually.

Global parameters for the configuration file

Define all global configuration properties 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 properties apply to all services. These parameters are used with sp_configure on page 55.

Table 2-1: Global parameters for the configuration file

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(NT)
errorlog	Full path name to the error log file	*$SYBASE/$SYBASE_DCO/servername.log*
interfaces	Full path to the directory of the interfaces file	
language	Available languages	us_english
max_packetsize	Sets the maximum packet size that can be negotiated for a client connection	2048

Keyword	Description	Default
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 DirectConnect for Oracle	25
network_tracing	Enables packet tracing between the client and DirectConnect 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 DirectConnect for Oracle. Only one user can have administrator privileges. This parameter is case sensitive.

```
administrator = ozzie
```

This parameter is dynamic.

charset

This is the default character set for DirectConnect 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 DirectConnect is:

```
charset = cp850
```

This parameter is static.

certificate_file

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

certificate_password

This is the password associated with the certificate file.

errorlog

This parameter sets the full path name to the error log file. It defaults to: *$SYBASE/$SYBASE_DCO/DCO-12_5/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 DirectConnect 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's client. The syntax is:

```
language = us_english
```

This parameter is static.

max_packetsize

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 Adaptive Server Enterprise 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_packetsize = 4096
```

This parameter is static.

maxpagesize

This sets the maximum page size for transfers using the bulk copy interface. The defaults 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 DirectConnect for Oracle. It also enables tracing for internal communications between threads.

Trace entries between the client and DirectConnect for Oracle are recorded in *$SYBASE/$SYBASE_DCO/SERVERNAME.log*.

Internal trace entries are recorded in *$SYBASE/$SYBASE_DCO$/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 2-2: 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 parameters

The properties 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 55.

The following table lists the service-specific parameters:

Table 2-3: Service-specific parameters for configuration file

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
timestamp_ms_support	Specifies whether the timestamp_format is used.	0
timestamp_format	Specifies the timestamp_format used in SQL statements.	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

Keyword	Description	Default
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
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 *$SYBASE/$SYBASE_DCO/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 case, the connect string required by DirectConnect 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 Adaptive Server Enterprise. 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.

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_format

Specifies the timestamp format used in SQL statements. This setting is ignored for connections from Adaptive Server Enterprise. 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

This parameter is dynamic.

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 DirectConnect for Oracle will buffer before calling Oracle to insert them. Array inserts are performed when a bulk load event is sent to the DirectConnect 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, DirectConnect 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, pre-system 10 servers are no longer supported by DirectConnect 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 2-4: 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 2-5, 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 2-5: 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.

traceflags

Controls the types of messages written to the log file.

The following 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 *$SYBASE/$SYBASE_DCO/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 is a separately licensed feature.

This parameter is static.

Data compatibility

For optimum compatibility with Adaptive Server, 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 DirectConnect 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 registry, or you can use the *%SYBASE%\DCO-12_6\DCO_SYBASE.bat* file.

Environment variables can be set in an MS-DOS window before starting DirectConnect for Oracle. They can also be set from the System menu in the Windows Control Panel. If DirectConnect 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 DirectConnect 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 DirectConnect 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

The following 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 DirectConnect 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 *$SYBASE* directory at the time of installation. To edit the *interfaces* file, use dsedit found in *$SYBASE/$SYBASE_OCS/bin.*

sql.ini file (Windows only)

A file name *sql.ini* must be present in the *%SYBASE%\ini* 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 documentation on dsedit, see the Open Client and Open Server documentation.

Note In order 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 DirectConnect for Oracle (DCO) Service.

Using *sp_addservice*

If you are connected directly to DirectConnect 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 DirectConnect 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 DirectConnect 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 DirectConnect for Oracle and delete the service entry from the configuration file.

Start DirectConnect for Oracle on UNIX

Use the startserver utility to start DirectConnect for Oracle on UNIX. The startserver utility is included in the *install* directory of DirectConnect for Oracle. For example, to start a DirectConnect 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 DirectConnect for Oracle issues a series of messages describing the settings of the configuration parameters.

Creating the runserver file

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

Table 2-6: Definition of flags in the runserver file

Flag	Definition
-S*server_name*	Specifies the name of the DirectConnect for Oracle. It is used to locate the configuration file and the network connection information in the *interfaces* file.
-t	Causes DirectConnect for Oracle to write start-up messages to standard error.
-l*errorlog_path*	Specifies the path to the *error log* file.
-i*interfaces_file_path*	Specifies the path to the *interfaces* file.

Starting DirectConnect for Oracle on Windows

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

- As a service – see "Starting DirectConnect for Oracle as a service" on page 26.
- From the command line – use the following syntax:

  %SYBASE%\%SYBASE_DCO%\bin\dcoracle.exe -S*server_name*
  [-t] [-i%SYBASE%*path_to_sql.ini_file*]
  [-l%SYBASE%*path_to_errorlog*]

  where:

  - -S is the name of DirectConnect 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 DirectConnect for Oracle named "DCSERVER" on Windows using the default *sql.ini* and error log files, and using -t to trace start-up messages, enter:

```
%SYBASE%\%SYBASE_DCO%\bin\dcoracle.exe -SDCSERVER -t
```

The DirectConnect for Oracle is up and running when you see the "start-up complete" message.

Starting DirectConnect for Oracle as a service

If you did not choose to start DirectConnect for Oracle as a service at the time of installation, you can use the instdco utility in Sybase Central to add DirectConnect for Oracle to the list of items you can start and stop with the Services utility. The utility instdco is located in the *%SYBASE%\%SYBASE_DCO%\bin* directory.

The instdco utility uses the following syntax:

```
instdco.exe service_name
%SYBASE%\%SYBASE_DCO%\bin\dcoracle.exe "startup_parameters"
```

where:

- *service_name* is the name of the DirectConnect for Oracle you are adding as a service.
- *startup_parameters* are any parameters you want used at start-up.

For example, to install a DirectConnect for Oracle named DCSERVER as a service, enter:

```
instdco.exe DCSERVER %SYBASE%\DCO\bin\dcoracle.exe
"-SDCSERVER -t"
```

Note If you need to include more than one parameter (for example, -i), you must include all the parameters in one set of double quotes.

Removing a DirectConnect for Oracle service

The syntax to remove DirectConnect for Oracle as a service is:

```
instdco service_name remove
```

Shutting down DirectConnect for Oracle

Use the following command to shut down the DirectConnect for Oracle from Adaptive Server Enterprise:

```
server_name...sp_shutdown
```

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

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

```
DCSERVER...sp_shutdown
```

Configuring Adaptive Server for DirectConnect

Before using DirectConnect for Oracle and Adaptive Server to access remote data, you must define the remote server to Adaptive Server Enterprise. For more information on configuring Adaptive Server Enterprise, 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 DirectConnect 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 Adaptive Server in the sysservers table.
- *server_class* – the class to use when defining a DirectConnect 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 DirectConnect for Oracle, change directories to *$SYBASE/$SYBASE_DCO/bin* for UNIX, or *%SYBASE%\%SYBASE_DCO%\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 DCO using isql

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

```
isql -Uuser_name -Ppassword -Sservice_name
```

Note The user must be a valid Oracle user.

CHAPTER 3

Reference Topics for Oracle

This chapter describes reference topics for DirectConnect for Oracle (DCO).

It contains the following topics:

Connect handler for DirectConnect for Oracle

The DirectConnect for Oracle's connect handler creates Open Server threads and connects to Oracle on behalf of the Adaptive Server Enterprise 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 DirectConnect logs into the Oracle database and remains connected.

SQL command handling

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

Support for text and image datatypes

DirectConnect for Oracle supports columns of datatype text and image when used with Adaptive Server Enterprise. DirectConnect for Oracle provides text pointers, which are 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 Adaptive Server Enterprise generates a select textptr() statement for a column, DirectConnect for Oracle creates a textptr value based on the Oracle ROWID. The text pointer is the actual ROWID of the given row.

When Adaptive Server Enterprise requests a text or image column's text pointer using a select textptr() function, DirectConnect for Oracle will return a 16-byte binary column.

When inserting and updating text or image columns using Adaptive Server Enterprise, 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 Adaptive Server Enterprise 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, Adaptive Server Enterprise 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 DirectConnect 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 Adaptive Server Enterprise communicates with DirectConnect for Oracle, client language and the default character set are passed as thread properties. DirectConnect for Oracle interacts with Oracle, conveying the client language and Open Server character set. Then, Adaptive Server Enterprise receives character data and error messages in the correct language and character set.

Passthrough mode

Passthrough mode allows a DirectConnect for Oracle client to interact directly with an Oracle database using native Oracle SQL. When DirectConnect for Oracle is operating as a standalone gateway, clients are automatically put into passthrough mode. When used with Adaptive Server Enterprise, DirectConnect 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 following Open Client datatypes:

Table 3-1: Oracle-to-Adaptive Server 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)

Oracle datatype	Open Client datatype number_mode=0	Open Client datatype number_mode=1	Open Client datatype number_mode=2
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)

Table 3-2: Oracle-to-Adaptive Server 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)
nvarchar2(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

The following table illustrates datatype conversion that is performed when a create table or alter table statement is processed using DirectConnect for Oracle with Adaptive Server Enterprise. DirectConnect for Oracle constructs syntax for the Oracle datatypes as shown:

Table 3-3: ASE-to-Oracle datatype conversion for create table

Adaptive Server 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)
univarchar(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 Adaptive Server Enterprise columns during query processing. The following table describes the allowable datatypes that can be used for existing Oracle datatypes.

Table 3-4: Allowable ASE-to-Oracle conversions

Adaptive Server 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.

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 Adaptive Server Enterprise 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. Adaptive Server Enterprise numeric datatypes do not allow this.

Oracle allows number datatypes to have negative scales. Adaptive Server Enterprise 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), DirectConnect 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 DirectConnect 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 property for DirectConnect for Oracle

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

Timestamp_ms_support parameter

Specifies the value of the timestamp_ms_support parameter.

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

Using the new parameters

When you use the new 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.

DCO 12.6 interoperates with ASE/CIS	When DCO 12.6 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.
DCO 12.6 interoperates with other applications	When DCO 12.6 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 DirectConnect for Oracle

Sybase date and time datatypes are now supported in ECDA for Oracle previously known as DirectConnect for Oracle (DCO). The following sections describe how DCO 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
```

Following is a list of available global variables:

@@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.

@@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 DirectConnect 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 section of the previous chapter. Following is a list of available traceflags:

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 *$SYBASE/$SYBASE_DCO/rdbms/xa_servicenameDate.trc*

15 Log parser errors

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

The two-phase commit feature is a separately licensed product.

❖ **To use two-phase commit with ASE servers**

1 Connect to an Adaptive Server Enterprise.

2 Add an external login entry for the DirectConnect for Oracle 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 for Oracle 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 Adaptive Server Enterprise.

Secure Sockets Layer

DirectConnect for Oracle now 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 DirectConnect 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.

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

SSL filter

When establishing a connection to an SSL-enabled DirectConnect for Oracle, the SSL security mechanism is specified as a filter on the master and query lines in the *interfaces* file (*sql.ini* 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>

    query tli tcp /dev/tcp tli_add1 ssl
    master tli tcp /dev/tcp tli_add1 ssl
```

A typical *sql.ini* file on Windows using SSL looks like this:

```
[SERVER]
```

```
query=TCP,hostname,address1, ssl
master=TCP,hostname,address1, ssl
```

where:

hostname is the name of the server to which the client is connecting
address1 is the port number of the host machine.

All connection attempts to a master or query entry in the *interfaces* file with an SSL filter must support the SSL protocol. A server can be configured to accept SSL connections and have other connections that accept plain text (unencrypted data), or use other security mechanisms.

For example, a DirectConnect for Oracle *interfaces* file on UNIX that supports both SSL-based connections and plain-text connections looks like this:

```
SYBSRV1 (tli format)

    master tli tcp /dev/tcp \x00020abc12345678000000000000000 ssl
    query tli tcp /dev/tcp \x00020abc12345678000000000000000 ssl
    master tli tcp /dev/tcp \x00020abd12345678000000000000000
```

Or, the same entry with the text format style of Sybase *interfaces* file on UNIX looks like this:

```
SYBSRV1 (text format)

    master tcp hostname 2748 ssl
    query tcp hostname 2748 ssl
    master tcp hostname 2749
```

An example of a socket-style *interfaces* file looks like this:

```
SYBSRV1

    master tcp ether hostname 2748 ssl
    query tcp ether hostname 2748 ssl
    master tcp ether hostname 2749
```

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.

The server certificate

Each DCO 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 DCO name in the *interfaces* file.

The CA trusted roots certificate

At start-up, DCO 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 DCO. A trusted roots file is accessible by the DCO in:

UNIX - *$SYBASE/config/trusted.txt*
Windows - *%SYBASE%\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 DCO, using a standard ASCII-text editor.

Adaptive Server provides tools to generate a certificate request and to authorize certificates. See "Using Adaptive Server tools to request and authorize certificates" in the Adaptive Server *System Administration Guide*.

Client login to DirectConnect for Oracle

Open Client applications establish a socket connection to the DCO 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.

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

Enabling SSL

The DCO 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. For additional information, refer to Appendix A, “SSL Connectivity.”

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.

Lightweight Directory Access Protocol (LDAP)

Lightweight Directory Access Protocol (LPDAP) is the Internet directory protocol. LDAP is a protocol for accessing and managing directory services. The LDAP support replaces the *interfaces* file. This means that the connection information normally stored in an interfaces 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.

In the $*SYBASE/$SYBASE_OCS/config* directory, create a *libtcl.cfg* file and add the following lines under [DIRECTORY]:

```
;       shows how to specify the different types of drivers which
;       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=libdldap.so
ldap://<LDAP server location >
```

Then set up your Sybase environment variables:

```
setenv SYBASE <release path>
setenv SYBASE_OCS OCS-12_5
setenv SYBASE_ASE ASE-12_5
setenv SYBASE_TCL_CFG <your libtcl.cfg pathname>
setenv LD_LIBRARY_PATH
$SYBASE/$SYBASE_OCS/lib:$SYBASE/$SYBASE_ASE/lib:$SYBASE/
$SYBASE_OCS/lib3p

setenv PATH
$SYBASE/$SYBASE_OCS/bin:$SYBASE/$SYBASE_ASE/bin:$PATH
```

To add your server entry to the LDAP server use the *$SYBASE/$SYBASE_OCS/bin/dscp*. Add an entry for your server into the LDAP server, the same way you would for an Adaptive Server Enterprise server. Check the Adaptive Server Enterprise 12.5 *Utility Guide* for more information on using dscp.

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

CHAPTER 4

Executing Remote Procedure Calls

Remote procedure calls (RPCs) are accepted by the DirectConnect for Oracle RPC event handler. The RPC is either interpreted locally or passed to Oracle for processing. RPCs specific to DirectConnect 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.

This chapter contains the following topics:

RPCs passed to Oracle

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

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

Note Named notation and mixed notation are not supported.

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;
```

Map an RPC that returns a result set as an Adaptive Server table

The following example maps the Oracle procedure rset_rpc_ex as an Adaptive Server 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)
```

Execute an RPC that returns a result set

The following example executes an RPC, from ASE through DirectConnect for Oracle 12.6 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)
```

Execute an RPC with input and output parameters

The following is an example of an RPC from Adaptive Server through DirectConnect for Oracle. Input and output parameters are mapped as Adaptive Server 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)
```

Execute an RPC with an input parameter

The following command executes an RPC from Adaptive Server Enterprise through DirectConnect 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 backend database returns a negative value, it 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. long raw input parameters are not supported.

RPCs processed by DirectConnect for Oracle

The following RPCs are processed by DirectConnect 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 DirectConnect 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 DirectConnect for Oracle and writes the service entry into the configuration file.
Syntax	server_name... sp_addservice service_name
Usage	To use an additional Oracle database, you must add a new service. Use sp_addservice to add an additional service. Then, use sp_configure 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

server_name... sp_columns *table_name* [, *table_owner*]
[, *table_qualifier*] [, *column_name*]

Parameters

table_name
is the name of the table or view. Use wildcard characters to request information about more than one table.

table_owner
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, sp_columns looks for tables owned by the current user.

table_qualifier
is ignored. Null should be specified.

column_name
is the name of the column for which you want information. Use wildcard characters to request information about more than one column.

Examples

The following example displays information about all columns in the publishers table that begin with "p":

```
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
-------------------------------- ----------------------
--------- ------------------------------ -------------
----------------- ----------- ------------------------
-----------------------------------------------------
----------------------------
     ----------- ----------- ----------- ----------- --
---------
     -------------------------------------------------
-----------------------------------------------------
-----------------------------------------------------
-----------------------------------------------------
-----------------------------------------
```

```
     ------------ ----------- ----------------
NULL                          SYSTEM
PUBLISHERS                 PUB_ID
1 CHAR
           NULL          4        NULL          0          0
    NULL

                47            1                96
NULL                          SYSTEM
PUBLISHERS                 PUB_NAME
12 VARCHAR2

           NULL         40        NULL          0          1
    NULL

                39            2                 1

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

Usage

- Table 4-1 shows the results set:

Table 4-1: Results set for sp_columns

Column	Datatype	Description
table_qualifier	varchar(32)	Always NULL.
table_owner	varchar(32)	The table owner. If no value was specified for the *table_owner* parameter, this value is the current owner.
table_name	varchar(32)	The table name.
column_name	varchar(32)	The column name.
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 DirectConnect for Oracle configuration parameters.

Syntax — server_name... sp_configure *[service_name]* [, *config_name* [, *config_value*]]

Parameters

service_name
is the name of the DirectConnect 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

- 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 DirectConnect 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 parameters, see “Using sp_configure” on page 7.

Permissions — Only the DirectConnect for Oracle administrator can execute sp_configure.

sp_dropservice

Description — Removes an existing service from the DirectConnect 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 [*loginame* | '*spid*']

Parameters

loginame
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                          NETWORK HANDLER    ...
 ...   sleeping   369544        MSG AVAILABLE     CONNECT HANDLER    ...
 ...   sleeping   369640        MSG AVAILABLE     DEFERRED HANDLER   ...
 ...   runnable        0                          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.

The following table summarizes the information sp_ps returns:

Table 4-2: 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.

The following table summarizes the results returned as rows with these columns:

Table 4-3: 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 DirectConnect 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 *table_name* [, *table_owner*]
[, *table_qualifier*] [, *index_name*] [, *is_unique*]

Parameters

table_name
is the name of the table. The use of wildcard character pattern matching is not supported.

table_owner
is the owner of the table. The use of wildcard character pattern matching is not supported. If *table_owner* is not specified, sp_statistics looks for a table owned by the current user.

table_qualifier
is ignored. Null should be specified.

index_name
is the index name. The use of wildcard character pattern matching is not supported.

is_unique
is Y to return only unique indexes; otherwise, is N to return both unique and non-unique indexes.

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                           dbo
    publishers                      NULL
    NULL                            NULL
        0         NULL NULL                                 NULL
             3           1
pubs2                           dbo
    publishers                                0
    publishers                      pubind
         1            1 pub_id                          A
             3           1
```

Usage

- Table 4-4 describes the results set:

Table 4-4: 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

sp_name

is the name of the stored procedure. Use wildcard characters to request information about more than one stored procedure.

sp_owner

is the owner of the stored procedure. Use wildcard characters to request information about procedures that are owned by more than one user.

sp_qualifier

is the name of the package. This can be a package name or NULL.

Usage

- sp_stored_procedures returns information about stored procedures in all packages unless sp_qualifier is specified.
- Table 4-5 shows the results set:

Table 4-5: Results set for sp_stored_procedures

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.

- sp_stored_procedures can return the name of stored procedures for which the current user does not have execute permission.

Permissions

Any user can execute sp_stored_procedures.

sp_tables

Description

Returns a list of objects that can appear in a from clause.

Syntax

server_name... sp_tables [*table_name*] [, *table_owner*]
[, *table_qualifier*][, *table_type*]

Parameters

table_name
is the name of the table. Use wildcard characters to request information about more than one table.

table_owner
is the owner of the table. Use wildcards to request information about tables that are owned by more than one user.

table_qualifier
is ignored. Use NULL.

table_type
is a list of values, separated by commas, giving information about all tables of the table type(s) specified, including the following:

```
"'TABLE', 'SYSTEM TABLE', 'VIEW'"
```

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

Examples

```
DCSERVER...sp_tables @table_type = "'TABLE', 'VIEW'"
```

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

Usage

- DirectConnect for Oracle does not necessarily check the read and write permissions on *table_name*. 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 4-6 describes the results set:

Table 4-6: Results set for sp_tables

Column	Datatype	Description
table_qualifier	varchar(30)	NULL.
table_owner	varchar(30)	Table owner
table_name	varchar(30)	NOT NULL. The table name.
table_type	varchar(32)	NOT NULL. One of the following: 'TABLE', 'VIEW', 'SYSTEM TABLE'.
remarks	varchar(254)	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 with the thread. The next time the thread becomes runnable, it receives the disconnect event and terminates.

Usage — Use sp_who or sp_ps to find the *spid* for the thread to be terminated.

Messages — *spid* terminated.

spid scheduled for termination.

spid not currently in use.

See also — sp_who

sp_traceon

Description — Turns on traceflags for the current connection.

Syntax — server_name... sp_traceon traceflags

Examples

```
DCSERVER...sp_traceon 2,3,4,5
```

Usage — Use with traceflags described in “traceflags” on page 18

sp_traceoff

Description — Turns off traceflags for the current connection.

Syntax — server_name... sp_traceoff [traceflags]

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_thead_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.

The following table summarizes the results returned as rows with these columns:

Table 4-7: 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

APPENDIX A

SSL Connectivity

This appendix describes the steps required for Secure Sockets Layer (SSL) connectivity to DirectConnect for Oracle (DCO) from the isql client.

It contains the following topics:

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 *%SYBASE%\OCS-12_5\bin* directory.

DirectConnect for Oracle server name

In the example, you must substitute the name of your DCO server for the name "dcossl."

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 *(%SYBASE%\SYBASE.bat*):

```
%SYBASE%\OCS-12_5\lib3p
```

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

```
%SYBASE%\SYBASE.bat
```

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

```
cd %SYBASE%\OCS-12_5\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:

```
D:\ecda_126\OCS-12_5\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:

```
D:\ecda_126\OCS-12_5\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/12.5.2/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:

```
D:\ecda_126\OCS-12_5\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:

```
D:\ecda_126\OCS-12_5\bin>certauth -Ctrusted.txt -
Qdcossl_req.txt -Kca_pkey.txt -Odcossl.crt -Psybase
```

18 The following appears:

```
-- Sybase Test Certificate Authority
certauth/12.5.2/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 DCO server configuration file by entering the following: (This directory is created and determined by the DCO administrator.)

```
D:\ecda_126\OCS-12_5\bin>copy dcossl.crt
d:\ecda_126\DCO-12_5\certificates
```

20 Copy the *trusted roots* file to the directory specified by the DCO 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:

```
D:\ecda_126\OCS-12_5\bin>type trusted.txt >>
d:\ecda_126\ini\trusted.txt
```

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

```
[dcossl]

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

22 Edit the DirectConnect for Oracle configuration file as described in “Enabling SSL” on page 43.

Index

Symbols

A

B

C

D

E

F

G

H

I

L

M

N

O

P

Q

R

S

T

U

V

W