

## **ORACLE Driver**

### **For All Users**

The following topics discuss the ORACLE driver and how to install it for use by an application.

[Overview](#)

[Hardware and Software Requirements](#)

[Setting Up the ORACLE Driver](#)

[Adding, Modifying, and Deleting ORACLE Data Sources](#)

[Connecting to an ORACLE Data Source](#)

[Troubleshooting](#)

### **For Advanced Users**

The following topics discuss how to use the ORACLE driver directly.

[Connection Strings \(Advanced\)](#)

[SQL Statements \(Advanced\)](#)

[Data Types \(Advanced\)](#)

[Error Messages \(Advanced\)](#)

### **For Programmers**

The following topics discuss how to use the ORACLE driver programmatically. They are intended for application programmers and require knowledge of the Open Database Connectivity (ODBC) application programming interface (API).

[SQLGetInfo Return Values \(Programming\)](#)

[ODBC API Functions \(Programming\)](#)

[Implementation Issues \(Programming\)](#)

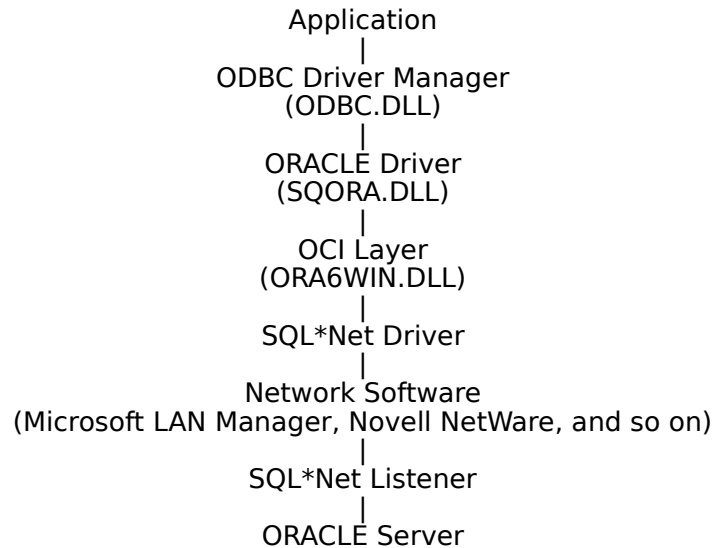
## Overview

See Also

ORACLE Server is a multiuser relational database management system (DBMS) that runs on a variety of workstation and minicomputer operating systems. Structured Query Language (SQL) is used to access data in ORACLE Server. IBM PCs and compatibles running Microsoft Windows communicate with ORACLE database servers across a network such as Microsoft LAN Manager, Novell NetWare, Banyan VINES, DECnet, or any TCP/IP network.

The ORACLE driver enables applications to access data in an ORACLE database through the Open Database Connectivity (ODBC) interface. It communicates with the network through SQL\*Net.

The application/driver architecture is:



**See Also**

For All Users

[Adding, Modifying, and Deleting ORACLE Data Sources](#)

[Connecting to an ORACLE Data Source](#)

[Hardware and Software Requirements](#)

[Setting Up the ORACLE Driver](#)

## Hardware and Software Requirements

See Also

To access ORACLE data, you must have:

- The ORACLE driver.
- An ORACLE Server database.
- A network connecting the computers on which these reside and a SQL\*Net connection across that network.

The following paragraphs describe the hardware and software required by each of these components.

### ORACLE Driver

The ORACLE driver requires the following hardware:

- An Industry Standard Architecture (ISA) computer, such as the IBM PC/AT or compatible, or
- A Micro Channel Architecture (MCA) computer, such as an IBM PS/2 or compatible, or
- An Extended Industry Standard Architecture (EISA) computer with an 80286, 80386, or 80486 microprocessor.
- At least 2 megabytes of random-access memory (RAM); 4 MB of RAM are recommended.
- A hard disk drive and approximately 350 kilobytes of hard disk space for the ORACLE driver.

The ORACLE driver requires the following software:

- MS-DOS version 3.3 or later
- Microsoft Windows version 3.0 or later
- ODBC Driver Manager version 1.0 (ODBC.DLL)
- ORACLE Call Interface (OCI) dynamic-link library (ORA6WIN.DLL)

---

**Note** The ORACLE driver requires ORA6WIN.DLL, regardless of whether a later version of OCI has been installed. ORA6WIN.DLL is shipped with the ORACLE driver.

---

### ORACLE Server

To access data in an ORACLE database with the ORACLE driver, you must have ORACLE Server version 6.0.34 or later. The ORACLE driver is designed for use with ORACLE Server version 6; you might encounter some minor inconsistencies when accessing data in ORACLE Server version 7. For information about the hardware and software required by ORACLE Server, see the ORACLE documentation.

### Network Software

A network is required to connect the platforms on which ORACLE Server and the ORACLE driver reside. For information about the hardware and software required by each network, see that network's documentation.

---

**Note** If you are using Microsoft LAN Manager with the named pipes protocol, it must be version 2.0 or later. If you are using Microsoft LAN Manager with the TCP/IP protocol, it must be version 2.2 or later.

---

### SQL\*Net

The ORACLE driver and ORACLE Server use SQL\*Net as a common network interface; it should be possible to use the ORACLE driver on any SQL\*Net connection from a computer running Windows to an ORACLE database server. The ORACLE driver is only guaranteed to work with SQL\*Net dynamic-link libraries (DLLs).

For complete information about SQL\*Net, see the SQL\*Net documentation.

---

**Note** SQL\*Net components are available only from Oracle Corporation.

---

**See Also**

For All Users

[Setting Up the ORACLE Driver](#)

**SQL\*Net**

SQL\*Net is a network interface package from Oracle Corporation that allows applications to access data in ORACLE Server across a variety of networks.

## Setting Up the ORACLE Driver

See Also

### To set up the ORACLE driver

- 1 If you have not already done so, install the SQL\*Net for Windows DLLs on your computer. If you want to test your SQL\*Net connection, set up your computer as an ORACLE client workstation and run SQL\*DBA or SQL\*Plus. For information on how to do this, see the SQL\*Net documentation.

---

**Important** The ORACLE driver is designed for use with ORACLE Server version 6; SQL\*Net for Windows DLLs are designed for use with ORACLE Server version 7. For information on how to configure your system to use the SQL\*Net for Windows DLLs with the ORACLE driver, see the ORACLE.TXT file shipped with the ORACLE driver.

---

- 2 Add a data source for each copy of ORACLE Server in which you want to access data.

### To set up a new version of the ORACLE driver

- 1 In the Main group in the Program Manager window, double-click the Control Panel icon. In the Control Panel window, double-click the ODBC icon.

---

**Note** For Microsoft Windows version 3.0, start the ODBC Administrator by double-clicking the Microsoft ODBC Administrator icon in the Microsoft ODBC group.

---

The Data Sources dialog box is displayed.

- 2 In the Data Sources dialog box, choose the Drivers button.  
The Drivers dialog box is displayed.
- 3 In the Drivers dialog box, choose the Add button.  
The Add Driver dialog box is displayed.
- 4 In the text box, type the name of the drive and directory containing the ORACLE driver in the text box. Or choose the Browse button to select a drive and directory name.
- 5 In the Add Driver dialog box, choose the OK button.  
The Install Drivers dialog box is displayed.
- 6 In the Available ODBC Drivers list, select ORACLE.
- 7 Choose the OK button.  
The ORACLE driver is installed.

### To delete the ORACLE driver

- 1 In the Main group in the Program Manager window, double-click the Control Panel icon. In the Control Panel window, double-click the ODBC icon.

---

**Note** For Windows version 3.0, start the ODBC Administrator by double-clicking the Microsoft ODBC Administrator icon in the Microsoft ODBC group.

---

The Data Sources dialog box is displayed.

- 2 In the Data Sources dialog box, choose the Drivers button.  
The Drivers dialog is displayed.
- 3 In the Installed ODBC Drivers list, select ORACLE.
- 4 Choose the Delete button.  
A message asks you to confirm that you want to remove the driver and all of the data sources that use the driver.
- 5 Choose the Yes button.

**See Also**

For All Users

[Adding, Modifying, and Deleting ORACLE Data Sources](#)

[Hardware and Software Requirements](#)



**Data Source (ORACLE)**

A data source includes the data a user wants to access and the information needed to get to that data. For the ORACLE driver, a data source is a specific copy of ORACLE Server, the computer on which it resides, the network used to access that computer, and the SQL\*Net components used as an interface to that network.

## Adding, Modifying, and Deleting ORACLE Data Sources

See Also

Before you can access data with the ORACLE driver, you must add a data source for each of your copies of ORACLE Server. The ORACLE driver uses the information you enter when you add the data source to access the data. You can change or delete a data source at any time.

### To add an ORACLE data source

- 1 In the Main group in the Program Manager window, double-click the Control Panel icon. In the Control Panel window, double-click the ODBC icon.

---

**Note** For Microsoft Windows version 3.0, start the ODBC Administrator by double-clicking the Microsoft ODBC Administrator icon in the Microsoft ODBC group.

---

- 2 In the Data Sources dialog box, choose the Add button.  
The Add Data Source dialog box is displayed.
- 3 In the Installed ODBC Drivers list, select ORACLE and choose the OK button.  
The ODBC Oracle Driver dialog box is displayed.
- 4 In the ODBC Oracle Setup dialog box, set the option values as necessary and choose the OK button.

### To modify an ORACLE data source

- 1 In the Main group in the Program Manager window, double-click the Control Panel icon. In the Control Panel window, double-click the ODBC icon.

---

**Note** For Windows version 3.0, start the ODBC Administrator by double-clicking the Microsoft ODBC Administrator icon in the Microsoft ODBC group.

---

- 2 In the Data Sources dialog box, select the data source in the Data Sources list and choose the Setup button.  
The ODBC Oracle Driver dialog box is displayed.
- 3 In the ODBC Oracle Setup dialog box, set the option values as necessary and choose the OK button.

### To delete an ORACLE data source

- 1 In the Main group in the Program Manager window, double-click the Control Panel icon. In the Control Panel window, double-click the ODBC icon.

---

**Note** For Windows version 3.0, start the ODBC Administrator by double-clicking the Microsoft ODBC Administrator icon in the Microsoft ODBC group.

---

- 2 In the Data Sources dialog box, select the data source you want to delete in the Data Sources list.
- 3 Choose the Delete button, and then choose the Yes button to confirm the deletion.

**See Also**

For All Users

[Connecting to an ORACLE Data Source](#)

[Setting Up the ORACLE Driver](#)

## Connecting to an ORACLE Data Source

See Also

To connect to a data source, the ORACLE driver requires that a SQL\*Net driver be loaded on your computer and the corresponding SQL\*Net listener be running on the ORACLE database server. For more information about SQL\*Net, see the SQL\*Net documentation.

As part of the connection process, an application can prompt you for information. If an application prompts you for information about an ORACLE data source, do the following:

### **To connect to an ORACLE data source**

- 1 In the User Name box, type the name you use on ORACLE Server.
- 2 In the Password box, type the password you use on ORACLE Server.
- 3 Choose OK.

**See Also**

For All Users

[Adding, Modifying, and Deleting ORACLE Data Sources](#)

For Advanced Users

[Connection Strings \(Advanced\)](#)

For Programmers

[SQLDriverConnect Implementation \(Programming\)](#)

## Troubleshooting

The following paragraphs discuss how to solve problems you might encounter while using the ORACLE driver.

### **Application doesn't work while using SQL\*Net TSRs**

The ORACLE driver is only guaranteed to work with SQL\*Net dynamic-link libraries (DLLs). For complete information about SQL\*Net, see the SQL\*Net documentation.

---

**Note** SQL\*Net components are available only from Oracle Corporation.

---

### **"Message send failure" error while running Microsoft LAN Manager using TCP/IP**

On Microsoft LAN Manager using the TCP/IP protocol, the number of bytes of data that can be sent to or retrieved from ORACLE Server at one time is controlled by the MAXSENDSIZE keyword in the [sockets] section of TCPUTILS.INI. If you receive a "Message send failure" error, try increasing this value. This error is most likely to occur when SQL\_LONGVARCHAR or SQL\_LONGVARIABLE data is being sent. For more information, see the ORACLE documentation.

## ODBC Oracle Setup Dialog Box

The ODBC Oracle Setup dialog box has the following options.

### Data Source Name

A name by which you will identify the data source. For example, "Personnel Data."

### Description

A description of the data in the data source. For example, "Hire date, salary history, and current review of all employees."

### SQL\*Net Connect String

A SQL\*Net connect string that specifies the location of the copy of ORACLE Server from which the driver will retrieve data. A SQL\*Net connect string uses the format:

*net\_prefix:hostname[:SID]*

The arguments in this format are:

<b>Argument</b>	<b>Meaning</b>
<i>net_prefix</i>	Specifies the SQL*Net driver to use. Some common <i>net_prefixes</i> are: P: Named pipes T: TCP/IP X: SPX/IPX
<i>hostname</i>	The name or alias of the server on which the ORACLE Server resides.
<i>SID</i>	The system ID of the database you want to use on the server (optional).

For more information, see the SQL\*Net documentation.

To access the following fields, click the Options button.

### Translation

The description of the current translator is displayed. To select a new translator, choose the Select button and select a new translator from the list in the Select Translator dialog box.

### Convert OEM to ANSI Characters

If the ORACLE driver and ORACLE Server are using the same non-ANSI character set, select the Convert OEM to ANSI Characters check box.

If the ORACLE driver and ORACLE Server are using different character sets, you must specify a character set translator.

## Connection Strings (Advanced)

See Also

The connection string for the ORACLE driver uses the following keywords.

<b>Keyword</b>	<b>Description</b>
<b>DSN</b>	The name of the data source.
<b>DBQ</b>	An ORACLE connect string of the form: <i>net_prefix:hostname[:SID]</i> ] where <i>net_prefix</i> specifies the SQL*Net driver to use, <i>hostname</i> specifies the name or alias of the server on which ORACLE Server resides, and <i>SID</i> is the optional system ID of the database you want to use on the server. Some common <i>net_prefixes</i> are: P: Named pipes T: TCP/IP X: SPX/IPX For more information, see the SQL*Net documentation.
<b>UID</b>	The user login ID.
<b>PWD</b>	The user-specified password.

For example, to connect to the Human Resources data source on the server HRSRVR using the NetBIOS SQL\*Net driver, the login ID Smith, and the password Sesame, you would use the following connection string:

```
DSN=Human Resources;DBQ=b:HRSRVR;UID=Smith;PWD=Sesame
```



**See Also**

For All Users

[Connecting to an ORACLE Data Source](#)

For Programmers

[SQLDriverConnect Implementation \(Programming\)](#)

## **SQL Statements (Advanced)**

See Also

The ORACLE driver supports the core ODBC SQL grammar. In addition, it supports almost all SQL statements in the extended ODBC SQL grammar. In accordance with the design of ODBC, the ORACLE driver will pass native SQL grammar to ORACLE Server.

The following Help topics describe the SQL grammar implemented by the ORACLE driver.

For Advanced Users

[Limitations to the ODBC SQL Grammar \(Advanced\)](#)

[Unsupported ODBC SQL Grammar \(Advanced\)](#)

For Programmers

[Implementation of the ODBC SQL Grammar \(Programming\)](#)

**See Also**

For Advanced Users

[Data Types \(Advanced\)](#)

For Programmers

[SQLGetInfo Return Values \(Programming\)](#)

## **Limitations to the ODBC SQL Grammar (Advanced)**

The only limitation that the ORACLE driver and ORACLE Server impose on the ODBC SQL grammar is statement length.

Statements are limited to roughly 2000 bytes. The ORACLE driver translates parameter markers to ":cn", where *n* is the position number of the marker, starting at 0. The ORACLE driver also translates ODBC escape clauses to native SQL. The resulting statement must be less than 2000 bytes long. This restriction is imposed by SQL\*Net running on Microsoft Windows.

## Unsupported ODBC SQL Grammar (Advanced)

The ORACLE driver completely supports all SQL statements and clauses in both the core and extended ODBC grammars, including the Integrity Enhancement Facility (IEF), except:

Statement not supported	Description
DELETE	The WHERE CURRENT OF <i>cursor-name</i> clause is not supported (positioned delete statement).
IEF	The CASCADE and RESTRICT clauses in the DROP TABLE, DROP VIEW, and REVOKE statements.
Procedures	Calls to procedures.
UPDATE	The WHERE CURRENT OF <i>cursor-name</i> clause is not supported (positioned update statement).

## Implementation of the ODBC SQL Grammar (Programming)

The only noteworthy part of the implementation of the ODBC SQL grammar is the implementation of comparison predicates.

If a comparison predicate has a parameter marker as the second expression in the comparison, and the value of that parameter is set to SQL\_NULL\_DATA with **SQLSetParam**, the comparison will fail. This is consistent with the null predicate grammar in ODBC SQL.

## Data Types (Advanced)

[See Also](#)

The ORACLE driver maps ORACLE Server data types to ODBC SQL data types. The following table lists all ORACLE Server data types and shows the ODBC SQL data types they are mapped to.

<b>ORACLE Server SQL data type</b>	<b>ODBC SQL data type</b>
CHAR	SQL_VARCHAR*
DATE	SQL_TIMESTAMP
FLOAT	SQL_FLOAT
LONG	SQL_LONGVARCHAR
LONG RAW	SQL_LONGVARBINARY
NUMBER	SQL_NUMERIC
NUMBER(5,0)	SQL_SMALLINT
NUMBER(10,0)	SQL_INTEGER
RAW	SQL_VARBINARY
VARCHAR	Not supported*
VARCHAR2	Not supported*

\* The CHAR data type has a variable length on ORACLE Server version 6.0 and a fixed length on version 7.0. The VARCHAR and VARCHAR2 data types are supported only on ORACLE Server version 7.0. The ORACLE driver implements the ORACLE Server version 6.0 data types.

---

**Note** All conversions in Appendix D of the *Microsoft ODBC SDK Programmer's Reference* are supported for the ODBC SQL data types listed earlier in this topic.

---

The following Help topics describe the data types implemented by the ORACLE driver.

For Advanced Users

[Implementation of Data Types \(Advanced\)](#)

[Limitations to Data Types \(Advanced\)](#)

For Programmers

[Implementation of Data Types \(Programming\)](#)

**See Also**

For Advanced Users

[SQL Statements \(Advanced\)](#)

## **Implementation of Data Types (Advanced)**

For advanced users, the only noteworthy part of the implementation of the data types is the implementation of SQL\_TIMESTAMP.

The DATE data type in ORACLE Server has both date and time parts. In the native SQL used by ORACLE Server, the time part of a DATE column is not returned unless a query specifically requests it. However, because the ORACLE driver maps the DATE data type to the SQL\_TIMESTAMP data type, the driver always returns both the date and time parts of a DATE column.



## Limitations to Data Types (Advanced)

The ORACLE driver and ORACLE Server impose the following limitations on the data types.

<b>Limited data type</b>	<b>Description</b>
Literals	ORACLE Server limits literals in SQL statements to 255 bytes.
<u>Number of bytes of data</u>	On Microsoft LAN Manager using the TCP/IP protocol, the number of bytes of data that can be sent to or retrieved from ORACLE Server is limited.
SQL_LONGVARCHAR	The ORACLE driver limits SQL_LONGVARCHAR data (LONG data) to 32,768 bytes.
SQL_LONGVARCHAR and SQL_LONGVARBINARY	ORACLE Server allows only a single long data column per table. The long data types are SQL_LONGVARCHAR (LONG) and SQL_LONGVARBINARY (LONG RAW).

## Implementation of Data Types (Programming)

For programmers, the only noteworthy part of the implementation of the data types concerns the CHAR, VARCHAR, and VARCHAR2 data types.

The ORACLE driver implements the data types from ORACLE Server version 6.0. Because the character data types are different on versions 6.0 and 7.0, the information returned by **SQLGetTypeInfo** can be misleading if the data source is version 7.0.

The CHAR data type has a variable length in version 6.0 and is therefore mapped by the ORACLE driver to SQL\_VARCHAR. It has a fixed length in version 7.0 but is still mapped to SQL\_VARCHAR. The VARCHAR and VARCHAR2 data types exist only on version 7.0 and are therefore not mapped to any ODBC SQL data types by the ORACLE driver.

For an *fSqlType* value of SQL\_VARCHAR, **SQLGetTypeInfo** returns the ORACLE Server data type CHAR. For an *fSqlType* value of SQL\_CHAR, **SQLGetTypeInfo** doesn't return any ORACLE Server data types. This behavior is consistent with the version 6.0 data types, but misleading for the version 7.0 data types.

## Error Messages (Advanced)

When an error occurs, the ORACLE driver returns the native error number, the SQLSTATE (an ODBC error code), and an error message. The driver derives this information both from errors detected by the driver and errors returned by ORACLE Server.

### Native Error

For errors that occur in the data source, the ORACLE driver returns the native error returned to it by ORACLE Server. When the ORACLE driver or the Driver Manager detects an error, the ORACLE driver returns a native error of zero.

### SQLSTATE

For errors that occur in the data source, the ORACLE driver maps the returned native error to the appropriate SQLSTATE. When the ORACLE driver detects an error, it generates the appropriate SQLSTATE. When the Driver Manager detects an error, it generates the appropriate SQLSTATE.

### Error Message

For errors that occur in the data source, the ORACLE driver returns an error message based on the message returned by ORACLE Server. For errors that occur in the ORACLE driver or the Driver Manager, the ORACLE driver returns an error message based on the text associated with the SQLSTATE.

Error messages have the following format:

*[vendor][ODBC-component][data-source]error-message*

The prefixes in brackets ([ ]) identify the source of the error. The following table shows the values of these prefixes returned by the ORACLE driver. When the error occurs in the data source, the *[vendor]* and *[ODBC-component]* prefixes identify the vendor and name of the ODBC component that received the error from the data source.

Error source	Prefix	Value
Driver Manager	<i>[vendor]</i>	[Microsoft]
	<i>[ODBC-component]</i>	[ODBC DLL]
	<i>[data-source]</i>	N/A
ORACLE Driver	<i>[vendor]</i>	[Microsoft]
	<i>[ODBC-component]</i>	[ODBC Oracle Driver]
	<i>[data-source]</i>	N/A
ORACLE Server	<i>[vendor]</i>	[Microsoft]
	<i>[ODBC-component]</i>	[ODBC Oracle Driver]
	<i>[data-source]</i>	[Oracle OCI]

## SQLGetInfo Return Values (Programming)

The following table lists the C language #defines for the *fInfoType* argument and the corresponding values returned by **SQLGetInfo**. An application can retrieve this information by passing the listed C language #defines to **SQLGetInfo** in the *fInfoType* argument.

<i>fInfoType</i> value (#define)	Returned value
SQL_ACCESSIBLE_TABLES	No.
SQL_ACCESSIBLE_PROCEDURES	No.
SQL_ACTIVE_CONNECTIONS	15
SQL_ACTIVE_STATEMENTS	No Limit.
SQL_CONCAT_NULL_BEHAVIOR	Non-NULL.
SQL_CONVERT_FUNCTIONS	None.
SQL_CONVERT_type, where <i>type</i> is the SQL data type (such as CHAR)	None.
SQL_CURSOR_COMMIT_BEHAVIOR	Preserves cursors.

SQL_CURSOR_ROLLBACK_BEHAVIOR	Preserves cursors.
SQL_DEFAULT_TXN_ISOLATION	Transaction 2 can insert (but not update or delete) rows that match the selection criteria of transaction 1 (SQL_TXN_REPEATABLE_READ).
SQL_DBMS_NAME	Oracle
SQL_DBMS_VER	06.00.0000 or 07.00.0000. (Note that ORACLE Server must be version 6.0.34 or later to be used with the ORACLE driver.)
SQL_DRIVER_NAME	SQORA.DLL
SQL_DRIVER_VER	01.00.nnnn, where nnnn specifies the build date.
SQL_EXPRESSIONS_IN_ORDERBY	Yes.
SQL_FETCH_DIRECTION	Next.
SQL_IDENTIFIER_CASE	Identifiers are not case sensitive.
SQL_IDENTIFIER_QUOTE_CHAR	" (double quote)
SQL_MAX_COLUMN_NAME_LEN	30
SQL_MAX_CURSOR_NAME_LEN	18
SQL_MAX_OWNER_NAME_LEN	30
SQL_MAX_PROCEDURE_NAME_LEN	0
SQL_MAX_QUALIFIER_NAME_LEN	0
SQL_MAX_TABLE_NAME_LEN	30
SQL_MULT_RESULT_SETS	No.
SQL_MULTIPLE_ACTIVE_TXN	Yes.
SQL_NUMERIC_FUNCTIONS	ABS, CEILING, FLOOR, MOD, SIGN, SQRT
SQL_ODBC_API_CONFORMANCE	Level 1.
SQL_ODBC_SAG_CLI_CONFORMANCE	SAG-compliant.
SQL_ODBC_SQL_CONFORMANCE	Core.
SQL_ODBC_SQL_OPT_IEF	No.
SQL_OUTER_JOINS	Yes.
SQL_OWNER_TERM	OWNER
SQL_PROCEDURE_TERM	Not supported.
SQL_PROCEDURES	No.
SQL_QUALIFIER_NAME_SEPARATOR	. (period)
SQL_QUALIFIER_TERM	Not supported.
SQL_ROW_UPDATES	No.
SQL_SCROLL_CONCURRENCY	Read only.
SQL_SCROLL_OPTIONS	Forward only.
SQL_SEARCH_PATTERN_ESCAPE	\ (backslash)
SQL_STRING_FUNCTIONS	ASCII, CHAR, CONCAT, INSERT, LCASE, LEFT, LENGTH, LOCATE, LTRIM, REPEAT, REPLACE, RIGHT, RTRIM, SUBSTRING, UCASE
SQL_SYSTEM_FUNCTIONS	DBNAME, IFNULL, USERNAME
SQL_TABLE_TERM	TABLE
SQL_TIMEDATE_FUNCTIONS	NOW, CURDATE, DAYOFMONTH,

SQL_TXN_CAPABLE	DAYOFWEEK, DAYOFYEAR, MONTH, QUARTER, WEEK, YEAR, CURTIME, HOUR, MINUTE, SECOND Transactions can contain only DML statements (SELECT, INSERT, UPDATE, and DELETE).
SQL_TXN_ISOLATION_OPTION	Transaction 2 can insert (but not update or delete) rows that match the selection criteria of transaction 1 (SQL_TXN_REPEATABLE_READ).

## **ODBC API Functions (Programming)**

[See Also](#)

The ORACLE driver supports all core and Level 1 functions. It supports the following Level 2 functions:

SQLDataSources  
SQLMoreResults  
SQLNativeSql

In addition, the ORACLE driver supports translation DLLs.

The following Help topics describe the ODBC API functions implemented by the ORACLE driver.

For Programmers

[Extensions to ODBC API Functions \(Programming\)](#)

[Implementation of ODBC API Functions \(Programming\)](#)

[Limitations to ODBC API Functions \(Programming\)](#)

**See Also**

For Advanced Users

[Error Messages \(Advanced\)](#)

## **Extensions to ODBC API Functions (Programming)**

The only function in the ORACLE driver that exceeds the specifications in the *Microsoft ODBC SDK Programmer's Reference* is **SQLGetData**.

**SQLGetData** can retrieve data from any column, regardless of whether there are bound columns to the right of that column.

## Implementation of ODBC API Functions (Programming)

The following table describes how the ORACLE driver implements specific functions.

Function	Description
<b>SQLConnect</b>	If the ORACLE workstation configuration file (CONFIG.ORA) contains the LOCAL keyword, <b>SQLConnect</b> requires only a user ID and password.
<b><u>SQLDriverConnect</u></b>	<b>SQLDriverConnect</b> uses the DSN, DBQ, UID, and PWD keywords.
<b>SQLMoreResults</b>	Because ORACLE doesn't support multiple result sets, <b>SQLMoreResults</b> always returns SQL_NO_DATA_FOUND and adjusts the state of the <i>hstmt</i> to allocated (if the SQL statement was executed with <b>SQLExecDirect</b> ) or prepared (if the SQL statement was executed with <b>SQLExecute</b> ).
<b>SQLSpecialColumns</b>	If <b>SQLSpecialColumns</b> is called with the SQL_BEST_ROWID option, it always returns the ROWID column.

## SQLDriverConnect Implementation (Programming)

If the ORACLE workstation configuration file (CONFIG.ORA) contains the LOCAL parameter, **SQLDriverConnect** requires only a user ID and password.

The **SQLDriverConnect** connection string uses the following keywords:

Keyword	Description
<b>DSN</b>	The name of the data source as listed in the ODBC.INI file.
<b>DBQ</b>	An ORACLE connect string of the form: <i>net_prefix:hostname[:SID]</i> where <i>net_prefix</i> specifies the SQL*Net driver to use, <i>hostname</i> specifies the name or alias of the server on which the ORACLE Server resides, and <i>SID</i> is the optional system ID of the database you want to use on the server. Some common <i>net_prefixes</i> are: P: Named pipes T: TCP/IP X: SPX/IPX For more information, see the SQL*Net

<b>UID</b>	documentation. The user login ID.
<b>PWD</b>	The user-specified password.

## Limitations to ODBC API Functions (Programming)

The following functions in the ORACLE driver do not meet the specifications in the *Microsoft ODBC SDK Programmer's Reference*.

Function	Description
<b>SQLCancel</b>	Because the ORACLE driver uses ORACLE Call Interface (OCI) and OCI does not support asynchronous processing, <b>SQLCancel</b> is equivalent to <b>SQLFreeStmt(hstmt,SQL_CLOSE)</b> .
<b>SQLGetConnectOption</b> and <b>SQLSetConnectOption</b>	These functions support only the SQL_ACCESS_MODE, SQL_AUTOCOMMIT, SQL_TRANSLATE_DLL, and SQL_TRANSLATE_OPTION options.
<b>SQLGetCursorName</b> and <b>SQLSetCursorName</b>	These functions get and set the cursor name in accordance with the <i>Microsoft ODBC SDK Programmer's Reference</i> . However, you cannot use the cursor name in an SQL statement because the ORACLE driver doesn't support positioned update and delete statements.
<b>SQLGetStmtOption</b> and <b>SQLSetStmtOption</b>	These functions support only the SQL_MAX_LENGTH, SQL_MAX_ROWS, and SQL_NOSCAN options.

## Implementation Issues (Programming)

The following implementation-specific issues might affect the use of the ORACLE driver.

Issue	Description
ORACLE Call Interface (OCI)	The ORACLE driver accesses data in an ORACLE data source through OCI.
Setup DLL	The ODBC Administrator calls the function <b>ConfigDSN</b> when users configure data sources. For the ORACLE driver, this function is in a setup DLL (SQORASTP.DLL).
Transactions	For ORACLE Server version 6.0, only Data Manipulation Language (DML) statements can be explicitly committed or rolled back when the driver is in manual commit mode. Data Definition Language (DDL) statements are committed automatically, regardless of



the commit mode. In both manual and auto commit modes, DML and DDL statements can be mixed.

For ORACLE Server version 7.0, some DDL statements can be explicitly committed or rolled back. For more information, see the ORACLE documentation.

**API**

Application programming interface. A set of routines that an application, such as Microsoft Access, uses to request and carry out lower-level services.

**character set**

A character set is a set of 256 letters, numbers, and symbols specific to a country or language. Each character set is defined by a table called a code page. An OEM (Original Equipment Manufacturer) character set is any character set except the ANSI character set. The ANSI character set (code page 1007) is the character set used by Microsoft Windows.

**conformance level**

Some applications can use only drivers that support certain levels of functionality, or conformance levels. For example, an application might require that drivers be able to prompt the user for the password for a data source. This ability is part of the Level 1 conformance level for the application programming interface (API).

Every ODBC driver conforms to one of three API levels (Core, Level 1, or Level 2) and one of three SQL grammar levels (Minimum, Core, or Extended). Drivers may support some of the functionality in levels above their stated level.

For detailed information about conformance levels, programmers should see the *Microsoft ODBC SDK Programmer's Reference*.

**data source**

A data source includes the data a user wants to access and the information needed to get to that data. Examples of data sources are:

- A SQL Server database, the server on which it resides, and the network used to access that server.
- A directory containing a set of dBASE files you want to access.

**DBMS**

Database management system. The software used to organize, analyze, search for, update, and retrieve data.

**DDL**

Data definition language. Any SQL statement that can be used to define data objects and their attributes. Examples include CREATE TABLE, DROP VIEW, and GRANT statements.

**DLL**

Dynamic-link library. A set of routines that one or more applications can use to perform common tasks. The ODBC drivers are DLLs.



**DML**

Data manipulation language. Any SQL statement that can be used to manipulate data. Examples include UPDATE, INSERT, and DELETE statements.

**ODBC**

Open Database Connectivity. A Driver Manager and a set of ODBC drivers that enable applications to access data using SQL as a standard language.

**ODBC Driver Manager**

A dynamic-link library (DLL) that provides access to ODBC drivers.

**ODBC driver**

A dynamic-link library (DLL) that an ODBC-enabled application, such as Microsoft Excel, can use to gain access to a particular data source. Each database management system (DBMS), such as Microsoft SQL Server, requires a different driver.

**SQL**

Structured Query Language. A language used for retrieving, updating, and managing data.

**SQL statement**

A command written in Structured Query Language (SQL); also known as a query. An SQL statement specifies an operation to perform, such as SELECT, DELETE, or CREATE TABLE; the tables and columns on which to perform that operation; and any constraints to that operation.

**translation option**

An option that specifies how a translator translates data. For example, a translation option might specify the character sets between which a translator translates character data. It might also provide a key for encryption and decryption.

**translator**

A dynamic-link library (DLL) that translates all data passing between an application, such as Microsoft Access, and a data source. The most common use of a translator is to translate character data between different character sets. A translator can also perform tasks such as encryption and decryption or compression and expansion.



