

Contents

The Visigenic Oracle7 driver conforms to the Open Database Connectivity (ODBC) specification described in the *ODBC Programmer's Reference (Version 2.0)* for your platform. The Oracle7 driver allows you to connect your ODBC-compliant application to an Oracle7 database.

This help file describes how to set up, configure, and use the Oracle7 driver so that it can connect to a data source. It includes these major sections:

- [Setting Up the Oracle7 Driver](#)
- [Adding and Modifying Data Sources](#)
- [Connecting to a Data Source](#)
- [Connection String Attributes](#)
- [ODBC Conformance Levels](#)
- [Mapping Data Types](#)

Setting Up the Oracle7 Driver

See also: [System Requirements](#)

The name of the Oracle7 driver varies by platform:

- VSORAC.DLL for Windows
 - VSORA32.DLL for Windows NT
- Although the Oracle7 driver was installed during the installation procedure, you must confirm that your system has the appropriate software installed.

System Requirements

To use the Oracle7 driver, you must have the following Oracle software installed on your system:

- ORACLE7 Server
- ORACLE SQL*Net (if you want to access remote databases)

To create, query, and call stored subprograms (functions and procedures), you must have the Oracle procedural option installed on the Oracle server.

For information on Oracle products, refer to your Oracle documentation set.

Adding and Modifying Data Sources

See also: [Adding and Modifying Data Sources for Windows](#)

A data source identifies a path to data that may include a network library, server, database, and other attributes. In this case, the data source is the path to an Oracle7 database. In order to connect to a data source, the Driver Manager looks at your ODBC.INI file for specific connection information.

The ODBC.INI file is an initialization file used by the ODBC Driver Manager and ODBC drivers. This file contains information about each data source and its associated driver. Before you can connect to a data source, its connection information must be added to this file. The following sections contain platform-specific information.

Adding and Modifying Data Sources for Windows

See also: [Adding a data source for Windows](#)
[Modifying a data source for Windows](#)

If you're developing on Windows, you add and configure data sources using the ODBC Administrator. The ODBC Administrator then updates your ODBC.INI file, located in the WINDOWS directory, to reflect your data source connection information. As you add data sources, the ODBC Administrator adds the information to this file for you that is, you should never modify the ODBC.INI file directly.

Adding a data source for Windows

- 1 To start the ODBC Administrator, double-click the ODBC icon in the Windows Control Panel. The Data Sources dialog box appears.
- 2 When you see the Data Source dialog box, click the Add button. The Add Data Source dialog box appears.
- 3 Select the Oracle7 driver, and then click OK. The Visigenic ODBC Oracle Setup dialog box appears.
- 4 In the Data Source Name box, enter the name of the data source you want to access. The data source name is defined by you that is, it can be any name that you choose.
- 5 In the Description box, enter the description for the driver. This is an optional field that describes the database driver that the data source connects to. The data source description is defined by you that is, it can be any string that you choose.
- 6 In the User Name box, enter your database user name. The user name is your database user id.
- 7 In the SQL*Net Connect String box, enter the SQL*Net connect string for the Oracle7 Server engine. The SQL*Net connect string identifies the Oracle7 Server engine that you want to access.
- 8 In the Translation Information area, click on the Select button to choose a translation DLL. Only use this field if you must translate data from your data source to your application. When you click on the Select button, the Select Translator dialog appears. This dialog gives you a choice of installed translation libraries. Choose one and click OK.
- 9 In the Buffer Size box, enter the size of the buffer (in bytes) for caching. To determine the buffer size, take the maximum column size and multiply by the number of rows to be returned. Each column of a result set has its own buffer. The default is 40000 bytes---the minimum is 1000 bytes, and the maximum is 60000 bytes.
- 10 In the Row Limit box, enter the maximum number of rows to cache (in the driver) from the data source. Use this option to specify the number of rows returned. The default value is 400 rows---the minimum is 1 row, and the maximum is 500 rows.
- 11 In the Customization area, check the System Table option if you want to expand the definition of a system table with respect to an Oracle DBMS. When set, SYSTEM TABLE is returned by a SQLTables() call for all tables, synonyms, and views owned by SYS or SYSTEM. If you do not set this option, SQLTables() returns values as follows: SYNONYM for all synonyms, VIEW for all views, SYSTEM TABLE for all tables owned by SYS or SYSTEM, and TABLE for all other tables.
- 12 In the Customization area, check the Capitalize catalog arguments option for backward compatibility with the 1.02 DriverSet. When checked, all arguments in catalog functions are converted to uppercase. Otherwise, leave the option unchecked for optimal functionality with Microsoft Windows 95 applications, and arguments will remain case sensitive. The default Oracle7 database behavior is that all arguments are uppercase, but the Visigenic Oracle7 driver puts double quotes around arguments so that case is meaningful. To follow Oracle7 default behavior, check this option.
- 13 In the Customization area, check the TXN_CAPABLE option for backward compatibility with the Visigenic 1.02 DriverSet. Leave the option unchecked for optimal performance with 16-bit, Windows 3.1-compatible applications such as Microsoft Access. When you have this option checked, a call to SQLGetInfo(SQL_TXN_CAPABLE) returns SQL_TC_DML. Otherwise, SQL_TC_DDL_COMMIT is returned.
- 14 In the Customization area, check the Number/Float Mapping option for backward compatibility with the Visigenic 1.02 DriverSet. Leave this option unchecked for optimal mapping with Microsoft Windows 95 applications. When you check this option, the number type maps to SQL_DECIMAL, and the float type maps to SQL_DOUBLE. The default is unchecked.
- 15 In the Customization area, check the SQLColumns Synonyms option to use the Oracle7 driver with Microsoft Access. When checked, SQLColumns() returns the column information for the underlying table when the input table name or table pattern matching sequence matches a synonym for a table. Leave the option unchecked for backward compatibility with the 1.02 DriverSet. See [SQLColumns Synonyms Option--More Information](#) for greater detail.
- 16 In the Customization area, check the SQLColumns, Ignore Comments option to optimize processing of synonyms when the SQLColumns Synonyms option is checked. This option suppresses the retrieval of comment data for synonym entries, and improves catalog access performance.
- 17 Click OK to add this data source. When you click OK, the Data Sources dialog box appears. After you click OK, the ODBC Administrator updates your ODBC.INI file. The User Name and SQL*Net connect string that you enter become the default data source connection values for this data source. That is, when you connect to the data source using either a dialog box or connection string, these values become the default entries for

the data source connection.

18 Click Add to add another data source or click Close to exit.

SQLColumns Synonyms Option--More Information

When checked, SQLColumns() returns the column information for the underlying table when the input table name or table pattern matching sequence matches a synonym for a table. For example, given the following definition, a call to SQLColumns() with f% or foosyn as the table name returns the column information for the foo table:

```
Create synonym foo_syn1 for foo;
```

Where foo is of type TABLE.

The SQLCOLSYN option has the following limitations:

- Only synonyms created for a table will be resolved (i.e. only synonyms that are one level deep). If you create a synonym for another synonym, the SQLColumns() call will not be properly resolved, and no column information will be returned. For example, SQLColumns() will not return any column information for foo_syn2 given the following definition:

```
Create synonym foo_syn2 for foo_syn1;
```

- If you are using Microsoft Access, synonyms are not updatable because SQLStatistics() does not currently support synonyms.

Modifying a data source for Windows

- 1 Invoke the ODBC Administrator. The Data Sources dialog box appears.
- 2 In the Data Sources dialog box, select the Oracle data source you want to modify and then click the Setup button. The Visigenic ODBC Oracle Setup dialog box appears.
- 3 Modify the applicable data source fields, and then click OK.

When you have finished modifying the information in this dialog box, the ODBC Administrator updates your ODBC.INI file. The fields that make up this dialog box are described in "[Adding a data source for Windows](#)."

Connecting to a Data Source

See also: [Connecting to a data source using a dialog box](#)
[Connection String Attributes](#)

An ODBC application can pass connection information in a number of ways. For example, the application may have the driver always prompt the user for connection information. Or the application may expect a connection string that specifies the data source connection. How you connect to a data source depends on the connection method that your ODBC applications uses.

One common way of connecting to a data source is through the Data Source dialog box. If your ODBC application is set up to use a dialog box, that dialog box is displayed and prompts you for the appropriate data source connection information.

Another way is through use of the connection string.

Connecting to a data source using a dialog box

1 When you see the Data Source dialog box, select an Oracle data source and then click OK. A Connect dialog boxes appears.

2 Fill in the appropriate information for the dialog box, and then click OK.

Once the connection information is verified, your application can access the information that the data source contains using the Oracle7 driver.

Connection String Attributes

Some applications may require a connection string that specifies data source connection information instead of using a dialog box to obtain this information. The connection string is made up of a number of attributes that specify how a driver connects to a data source. An attribute identifies a specific piece of information that the driver needs to know before it can make the appropriate data source connection. Each driver may have a different set of attributes but the connection string format is always the same. A connection string has the following format:

```
DSN=data-source-name[;SERVER=value] [;PWD=value] [;UID=value]
```

You must specify the data-source-name. However, all other attributes are optional. If you do not specify an attribute, that attribute defaults to the one that is specified in the data source specifications section (for the data source specified in the connection string) of your ODBC.INI file.

The attributes for the connection string are as follows:

DSN	The data source name. This name is listed in the ODBC Data Sources section and has its own Data Source Specifications section in the .odbc.ini file.
PWD	The password for the Oracle7 server that you want to access.
SERVER	The SQL*Net connect string for the Oracle7 Server that you want to access.
UID	The Oracle7 Server user name. Depending on your system, this attribute may not be optional---that is, certain databases and tables may require this attribute for security purposes.

For example, a connection string that connects to the Employees data source using the employees server, mickey host machine, and TCP/IP network would have the following attributes in the connection string:

```
"DSN=Employees;SERVER=t:mickey:employees;UID=cindy;PWD=secret"
```

ODBC Conformance Levels

See also: [API Conformance Level](#)
[SQL Conformance Level](#)

ODBC defines two types of conformance standards for drivers: the API conformance standard and the SQL grammar conformance standard. API conformance refers to the functions that a driver supports. SQL conformance refers to the SQL grammar that the driver supports. Each conformance standard is made up of levels.

API Conformance Level

See also: [Supported Options](#)

The Oracle7 driver supports the Core and Level 1 API. The driver also supports the following Level 2 functions:

- SQLBrowseConnect()
- SQLDataSources() *
- SQLDrivers() *
- SQLExtendedFetch()
- SQLForeignKeys()
- SQLNativeSql()
- SQLNumParams()
- SQLPrimaryKeys()
- SQLProcedures()
- SQLProcedureColumns()
- SQLSetPos()
- SQLSetScrollOptions()

*Supported through the Cursor Library or Driver Manager.

Supported Options

The driver supports the following options for the **SQLGetConnectOption()** and **SQLSetConnectOption()** Level 1 functions:

- SQL_ACCESS_MODE (**SQLGetConnectOption()** only)
- SQL_AUTOCOMMIT
- SQL_ODBC_CURSORS
- SQL_OPT_TRACE
- SQL_OPT_TRACEFILE
- SQL_TRANSLATE_DLL
- SQL_TRANSLATE_OPTION
- SQL_TXN_ISOLATION

The driver supports the following options for the **SQLGetStmtOption()** and **SQLSetStmtOption()** Level 1 functions:

- SQL_BIND_TYPE
- SQL_CONCURRENCY
- SQL_CURSOR_TYPE
- SQL_KEYSET_SIZE
- SQL_RETRIEVE_DATA
- SQL_ROWSET_SIZE
- SQL_SIMULATE_CURSOR

SQL Conformance Level

The Oracle7 driver supports the Minimum SQL grammar and Core SQL grammar listed in "SQL Conformance Levels" on page 1-9. The Oracle7 driver also supports the following ODBC extensions to SQL:

- Date, time, and timestamp data
- Outer joins
- The following numeric functions:
 - abs floor pi sin
 - ceiling log power sqrt
 - cos log10 round tan
 - exp mod sign truncate
- The following date functions:
 - curdate dayofweek month second
 - curtime dayofyear monthname week
 - dayname hour now year
 - dayofmonth minute quarter
- The following string functions:
 - ascii left right ucase
 - char length rtrim
 - concat ltrim soundex
 - lcase replace substring
- The following system functions:
 - ifnull (UNIX) user

Mapping Data Types

The Oracle7 Server supports a set of data types. The Oracle7 driver maps these data types to their appropriate ODBC SQL data types. The following table lists the Oracle7 Server data type and its corresponding ODBC SQL data type.

Note The following data type mapping is best for using Microsoft Windows 95 applications. For backward compatibility with the 1.02 version of the DriverSet, you might consider setting the Number/Float Mapping option.

Oracle7 Server Data Type	ODBC SQL Data Type
CHAR	SQL_CHAR
DATE	SQL_TIMESTAMP
FLOAT	SQL_FLOAT *
LONG	SQL_LONGVARCHAR
LONG RAW	SQL_LONGVARBINARY
NUMBER	SQL_FLOAT *
NUMBER(P)	SQL_DECIMAL
NUMBER(P,S)	SQL_DECIMAL
RAW	SQL_BINARY
VARCHAR2	SQL_VARCHAR

Note ODBC SQL data types do not support the MLSLABEL Oracle7 data type.

There is no conformance testing (i.e. range checking) done for:

- SQL_C_xTINYINTs (these are interpreted as integers)
 - SQL_C_xLONG (these are interpreted as integers)
 - SQL_C_BIT (these are interpreted as chars)
 - SQL_C_DOUBLE (these are interpreted as floats)
-

Note With SQL_C_DOUBLE, if you insert a value greater than 3.4e38 or less than 1.1e-38, you will get an Oracle range error.

Although all values will be inserted into the data source (except for SQL_C_DOUBLE outside of a float range), no range checking is performed (for example, storing a value of 10 with SQL_C_BIT is not flagged as an error).

