

ODBC Paradox Driver

For All Users

The following topics discuss the ODBC Paradox driver and how to install it.

[Overview](#)

[Hardware and Software Requirements](#)

[Setting Up the ODBC Paradox Driver](#)

[Adding, Modifying, and Deleting a Paradox Data Source](#)

[Connecting to a Paradox Data Source](#)

For Advanced Users

The following topics discuss how to use the ODBC Paradox driver directly.

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

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

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

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

For Programmers

The following topics provide programming information on the ODBC Paradox driver. 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 \(Programming\)](#)

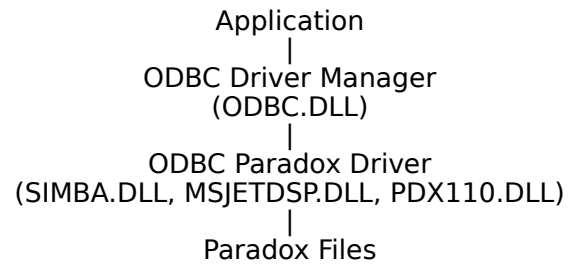
[Implementation Issues \(Programming\)](#)

Overview

See Also

The ODBC Paradox driver enables applications to access data in Paradox files through the Open Database Connectivity (ODBC) interface. Structured Query Language (SQL) is used to access the data.

The application/driver architecture is:



See Also

For All Users

[Adding, Modifying, and Deleting a Paradox Data Source](#)

[Connecting to a Paradox Data Source](#)

[Hardware and Software Requirements](#)

[Setting Up the ODBC Paradox Driver](#)

Hardware and Software Requirements

See Also

To access Paradox data, you must have:

- The ODBC Paradox driver.
- The ODBC Driver Manager 1.0 or later (ODBC.DLL).
- A computer running MS-DOS 3.3 or later.
- Microsoft Windows 3.0a or later.

ODBC Paradox driver

The ODBC Paradox 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
- A hard disk drive and approximately 750 kilobytes of hard disk space for the driver software.

Paradox Data Files

The ODBC Paradox driver can access data only in Paradox version 3.5 files.

See Also

For All Users

[Setting Up the ODBC Paradox Driver](#)

Setting Up the ODBC Paradox Driver

See Also

To set up the ODBC Paradox 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 3.0a, 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 Drivers button.
- 3 In the Drivers dialog box, choose the Add button.
- 4 In the text box, type the name of the drive and directory containing the ODBC Paradox 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.
- 6 In the Available ODBC Drivers list, select Paradox.
- 7 Choose the OK button to install the driver.

Note The ODBC Paradox driver may share some of the same dynamic link libraries (DLLs) with other drivers installed on your computer. If so, you will be asked to overwrite the ODBC Paradox driver, regardless of whether it has been installed. Choose the Yes button to install the driver.

Before using the ODBC Paradox driver, you must add a data source for each directory containing Paradox files you want to access.

To delete the ODBC Paradox 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 3.0a, 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 Drivers button.
- 3 In the Drivers dialog box, select the ODBC Paradox driver from the Installed ODBC Drivers list.
- 4 Choose the Delete button.
- 5 The ODBC Paradox setup program asks if you want to remove the driver and all the data sources that use the driver.
- 6 Choose the Yes button.

See Also

For All Users

[Hardware and Software Requirements](#)

data source (Paradox)

A data source includes the data a user wants to access and the information needed to get to that data. For the ODBC Paradox driver, a data source is a directory containing the Paradox files you want to access.

Adding, Modifying, and Deleting a Paradox Data Source

See Also

Before you can access data with the ODBC Paradox driver, you must add a [data source](#) for it. A Paradox data source specifies the default data directory in which the ODBC Paradox driver searches for Paradox files you want to access, as well as other database and driver information. The ODBC Paradox driver uses the information you enter to access the data. You can change or delete a data source at any time.

To add a Paradox 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 3.0a, 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.
- 3 In the Add Data Source dialog box, select Paradox from the Installed ODBC Drivers list and choose OK.
- 4 In the ODBC Paradox Setup dialog box, enter information to set up the data source.

To modify a Paradox 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 3.0a, 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 from the Data Sources list.
- 3 Choose the Setup button.
- 4 In the ODBC Paradox Setup dialog box, enter information to set up the data source.

To delete a Paradox 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 3.0a, 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 from 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 a Paradox Data Source](#)

[Setting Up the ODBC Paradox Driver](#)

Connecting to a Paradox Data Source

See Also

When you connect to a Paradox data source, an application may prompt you to enter the name of the directory containing the Paradox files.

To connect to a Paradox data source

▶ Enter the name of the directory containing the Paradox files you want to access and choose OK.

See Also

For All Users

[Adding, Modifying, and Deleting a Paradox Data Source](#)

For Advanced Users

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

ODBC Paradox Setup Dialog Box

The ODBC Paradox Setup dialog box has the following fields:

Data Source Name

A name that identifies the data source, such as Payroll or Personnel.

Description

An optional description of the data in the data source; for example, "Hire date, salary history, and current review of all employees."

Directory

The full pathname of the directory containing your Paradox files. Before you add the data source, you must either use the Select Directory button to select a directory, or select the Use Current Directory check box to use the application's current working directory.

Options

Displays the following options:

Caution Except for the Exclusive option, these options apply to all data sources that use the ODBC Paradox driver.

Collating Sequence

The sequence in which characters are sorted.

Page Timeout

The period of time, in tenths of a second, that a page (if not used) remains in the buffer before being removed. The default is 600 tenths of a second (60 seconds).

Exclusive

If the Exclusive box is selected, Paradox files will be opened in Exclusive mode and can be accessed by only one user at a time. If the Exclusive box is cleared, Paradox files will be opened in Shared mode and can be accessed by more than one user at a time. Performance is enhanced when running in Exclusive mode.

Network Directory

The full pathname of the directory containing the PARADOX.NET file. If the directory does not contain a PARADOX.NET file, the Paradox driver creates one. For information about the PARADOX.NET file, see the Paradox documentation.

Before you can select a network directory, you must enter your Paradox user name in the User Name text box. Use the Select Network Directory button to select a network directory.

User Name

Your Paradox user name.

Connection Strings (Advanced)

See Also

The connection string for the ODBC Paradox driver uses the following keywords:

Keyword	Description
DSN	The name of the data source.
UID	The user login ID.
PWD	The user-specified password.
DBQ	The name of the directory containing the Paradox file or files.
FIL	The file type (always PARADOX).

For example, to connect to the Accounting data source in the directory C:\PARADOX\ACCT, use the following connection string:

```
DSN=Accounting;UID=Smith;PWD=Sesame;DBQ=C:\PARADOX\ACCT;FIL=PARADOX
```

See Also

For All Users

[Connecting to a Paradox Data Source](#)

SQL Statements (Advanced)

See Also

The ODBC Paradox driver fully supports the Minimum SQL grammar. In addition, it supports some SQL statements in both the Core and Extended ODBC grammars.

For information about ODBC SQL grammar limitations, and additional and driver-specific grammar supported, see the following topics:

For Advanced Users

[Additional Supported ODBC SQL Grammar \(Advanced\)](#)

[Driver-Specific ODBC SQL Grammar \(Advanced\)](#)

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

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

For Programmers

[Limitations to ODBC SQL Grammar \(Programming\)](#)

See Also

For Advanced Users

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

Additional Supported ODBC SQL Grammar (Advanced)

See Also

The ODBC Paradox driver completely supports the following SQL statements and clauses in the Core and Extended ODBC grammar:

Supported statement	Description
[GROUP BY column-name [,column- name]...]	Supported.
[HAVING search- condition]	Supported.
Literals	Approximate numeric literals, Exact numeric literals, and date literals.
Outer Joins	Outer joins are supported.
Search Criteria	IN and BETWEEN predicates. IN is supported only in a list.
Set Functions	AVG (<i>expression</i>), COUNT (*), MIN (<i>expression</i>), MAX (<i>expression</i>), and SUM (<i>expression</i>).
Table Correlation Names	Table correlation names are supported.

See Also

For Advanced Users

[Driver-specific ODBC SQL Grammar \(Advanced\)](#)

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

Driver-Specific ODBC SQL Grammar (Advanced)

See Also

The ODBC Paradox driver supports the following driver-specific ODBC SQL grammar:

Driver-specific ODBC SQL Grammar	Description
<u>COUNT (expression)</u>	The COUNT set function supports COUNT (expression).
<u>Date Arithmetic</u>	A number of days can be added to or subtracted from a date column. One date column can be subtracted from another.
<u>Grouping by Expressions</u>	A GROUP BY clause can contain expressions.
HAVING clause	Not supported without GROUP BY.
<u>Ordering by Expressions</u>	An ORDER BY clause can contain expressions.
<u>Outer Joins</u>	A SELECT statement can contain a list of OUTER JOIN clauses.
<u>Table Names</u>	You can use the full path name of a Paradox file as a table name in the CREATE TABLE, DELETE, DROP TABLE, INSERT, SELECT, and UPDATE statements.
<u>USE directory</u>	The USE statement changes the directory in which the ODBC Paradox driver searches for Paradox files.

See Also

For Advanced Users

[Additional Supported ODBC SQL Grammar \(Advanced\)](#)

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

Date Arithmetic (Advanced)

To calculate the date of a certain number of days before or after the date in a date column, subtract or add that number of days from the date column. For example, to calculate the date 30 days before the date in EndDate, use the expression:

`EndDate - 30`

To calculate the number of days between the dates in two date columns, subtract one date column from the other. For example, to calculate the number of days between the values in StartDate and EndDate, use the expression:

`EndDate - StartDate`

Grouping by Expressions (Advanced)

The ODBC Paradox driver extends the GROUP BY clause to allow grouping by expressions. The select list of a SELECT statement that has a GROUP BY clause can only have in it:

- Expressions from the GROUP BY clause. Each expression can be a subexpression of a select list expression. For example, SELECT City + ' ' + State FROM Employee GROUP BY City + ' ' + State.
- Set functions.

Ordering by Expressions (Advanced)

The ODBC Paradox driver extends the ORDER BY clause to allow ordering by expressions. The syntax of the ORDER BY clause is:

ORDER BY *sort-specification* [, *sort-specification*]...

where *sort-specification* is the number of a column in the result set, a column name, or an expression, optionally followed by ASC or DESC to specify an ascending or descending order. If the sort specification is an expression, it cannot contain a set function (AVG, COUNT, MAX, MIN, or SUM).

For example, to order companies by net profit in descending order, you might use the clause:

```
ORDER BY Income - Expenses DESC
```


Outer Joins (Advanced)

See Also

The ODBC Paradox driver extends the OUTER JOIN syntax to support nested outer joins. The OUTER JOIN syntax is:

```
left-outer-join ::=  
    table-reference LEFT OUTER JOIN table-reference  
    ON search-condition
```

```
table-reference ::=  
    table-name | [(left-outer-join )]
```

where *table-name* can be a table name or a table name followed by a correlation name. For example, the following statement uses a three-way outer join to create a list of sales orders. For each sales order, all line numbers (if any) are listed, and for each line number, the part and description (if any) are listed.

```
SELECT Order.SONum,  
       Line.LineNum,  
       Part.PartNum,  
       Part.Description  
FROM Order LEFT OUTER JOIN  
    (Line LEFT OUTER JOIN Part  
    ON Line.PartNum=Part.PartNum)  
    ON Order.SONum=Line.SONum
```

Note The rightmost ON corresponds to the leftmost LEFT OUTER JOIN.

See Also

For Advanced Users

[Additional Supported ODBC SQL Grammar \(Advanced\)](#)

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

Table Names (Advanced)

The ODBC Paradox driver treats a Paradox data file as a table. You may use the full MS-DOS path of a Paradox file as a table name in the following SQL statements:

```
CREATE TABLE table-name
DELETE FROM table-name
DROP TABLE table-name
INSERT INTO table-name
SELECT FROM table-name
UPDATE table-name
```

For example, to retrieve the number and description of parts included on a line in a sales order where the sales order line number file and the parts file are in different directories, you might use the statement:

```
SELECT LINE.SONum,
        LINE.LineNum,
        PART.PartNum,
        PART.Description
FROM A:\DATA\BUSINESS\LINE.PDX,
     C:\INVENTORY\PART.PDX
WHERE LINE.PartNum=PART.PartNum
```

USE directory (Advanced)

The USE statement specifies the default data directory in which the ODBC Paradox driver searches for Paradox files. The syntax of the USE statement is:

USE [*drive-letter*]:*directory*

where *directory* is:

\directory-name[\directory-name...]

Note that the directory specification must be absolute, that is, it begins with a backslash (\). Relative directory specifications are not supported.

To use the Paradox files in the \DATA\BUSINESS directory on the A: drive, use the statement:

USE A:\DATA\BUSINESS

Implementation of the ODBC SQL Grammar (Advanced)

The ODBC Paradox driver implements the ODBC SQL grammar as follows:

SQL statement	Description
Comparison Predicate	In keeping with the ODBC SQL grammar, comparisons to NULL must be made in a NULL predicate. The NULL predicate has the syntax: <i>column-name</i> IS [NOT] NULL The predicate <i>column-name</i> = NULL returns a syntax error.
<u>CREATE INDEX</u>	The CREATE INDEX statement creates Paradox primary key files and secondary files.
Delimited Identifiers	If column or table names contain any characters except letters, numbers, and underscores, they must be delimited. To delimit a column or table name, enclose the name in double quotes("").
<u>DROP INDEX</u>	The DROP INDEX statement deletes Paradox secondary index files and does not use the ODBC SQL grammar.
Sort Order	The ODBC Paradox driver supports the following sort orders: ASCII International Norwegian/Danish Swedish/Finnish You can choose a sort order from the Collating Sequence list in the <u>ODBC Paradox Setup dialog box</u> .
<u>Timeliness of Result Set Data</u>	If the ODBC Paradox driver needs to sort data, such as in the case of an ORDER BY clause, it takes a snapshot of the data.

CREATE INDEX (Advanced)

The syntax of the CREATE INDEX statement for the ODBC Paradox driver is:

```
CREATE [UNIQUE] INDEX index-name  
  ON table-name  
  (column-identifier [ASC]  
  [, column-identifier [ASC]....])
```

The ODBC Paradox driver does not support the **DESC** keyword in the ODBC SQL grammar for the CREATE INDEX statement. The *table-name* argument can specify the full path of the table.

If the keyword **UNIQUE** is specified, the ODBC Paradox driver will create a unique index. This is a Paradox primary key file named *table-name.PX*. Unique indexes are subject to the following restrictions:

- The unique index must be created before any rows are added to the table.
- Only one unique index is allowed per table.
- The *index-name* argument for a unique index must be the same as the base name of the table, as required by Paradox.

If the keyword **UNIQUE** is omitted, the ODBC Paradox driver will create a non-unique index. This consists of two Paradox secondary index files named *table-name.Xnn* and *table-name.Ynn*, where *nn* is the number of the column in the table. Non-unique indexes are subject to the following restrictions:

- Before a non-unique index can be created for a table, a unique index must exist for that table.
- The *index-name* argument for a non-unique index must be the same as the column name.
- Only one column can be specified for a non-unique index.

For example, to use the sales order number and line number columns as the unique index on the SO_LINES table, use the statement:

```
CREATE UNIQUE INDEX SO_LINES  
  ON SO_LINES (SONum, LineNum)
```

To use the part number column as a non-unique index on the SO_LINES table, use the statement:

```
CREATE INDEX PartNum  
  ON SO_LINES (PartNum)
```

DROP INDEX (Advanced)

The syntax of the DROP INDEX statement for the ODBC Paradox driver is:

DROP INDEX *index-name* ON *table-name*

The *table-name* argument can specify the full path of the table.

Note This is different than the ODBC SQL grammar for the DROP INDEX statement. The DROP INDEX statement cannot be used to delete the unique index (Paradox primary key file) on a table.

For example, to drop the PartNum index from the SO_LINES table, use the statement:

```
DROP INDEX PartNum ON SO_LINES
```

Timeliness of Result Set Data (Advanced)

When the ODBC Paradox driver returns data in the form of a result set, that data is usually taken directly from the data file. However, to execute DISTINCT, GROUP BY, and ORDER BY clauses in SELECT statements, the ODBC Paradox driver must use or return the data in sorted order. If the ODBC Paradox driver cannot find an index that correctly orders the data, it copies and sorts the data. When the driver returns the copied data as a result set, it is possible that, due to changes in the original data since the copy was taken, the copied data no longer matches the original data.

Limitations to ODBC SQL Grammar (Advanced)

See Also

The ODBC Paradox driver and Paradox impose the following limitations on the ODBC SQL grammar:

Grammar	Limitation
BETWEEN Predicate	The ODBC Paradox driver considers <i>value1</i> to be between <i>value2</i> and <i>value3</i> if <i>value1</i> is greater than or equal to <i>value2</i> and less than or equal to <i>value3</i> . For example, the following expression is true: 'b' BETWEEN 'a' AND 'c' and the following expression is false: 'b' BETWEEN 'c' AND 'a'
DELETE Statement	In order to delete a row from a table, the ODBC Paradox driver requires that the table have a unique index (Paradox primary key).
HAVING Clause	A SELECT statement cannot contain a HAVING clause unless it also contains a GROUP BY clause. An expression in a predicate in a HAVING clause can contain only a literal, a set function, or an expression in the GROUP BY clause. For example: <pre>SELECT City + ' ' + State FROM Employee GROUP BY City + ' ' + State HAVING {fn LENGTH(City + ' ' + State)}>10</pre>
LIKE Predicate	If data in a column is longer than 255 characters, the LIKE comparison will be based only on the first 255 characters.
NOT NULL	Not supported in the CREATE TABLE statement.
ORDER BY Clause	If a SELECT statement contains a GROUP BY clause and an ORDER BY clause, the ORDER BY clause can contain only a column in the result set or an expression in the GROUP BY clause.
Self Joins	In order to perform a self join on a table, the ODBC Paradox driver requires that the table have a unique index (Paradox primary key).
Set Functions	AVG, MAX, MIN, and SUM do not support the DISTINCT keyword.
Sort Keys	The maximum length of a sort key in a GROUP BY clause, ORDER BY clause, SELECT DISTINCT statement, or outer join is 255 bytes; the maximum length of all sort keys in a sort row is 65,500 bytes. If the length of the data in a column is greater than 255 characters, sorting will be based on the first 255 characters.
SQL Statements	The ODBC Paradox driver imposes the following maximums on SQL statements: Column Name Length: 25 characters Token (Word) Length: 1000 characters Number of ANDed Predicates: 300
UPDATE Statement	In order to update a table, the ODBC Paradox driver requires that the table have a unique index (Paradox primary key).

See Also

For Advanced Users

[Additional Supported ODBC SQL Grammar \(Advanced\)](#)

[Driver-specific ODBC SQL Grammar \(Advanced\)](#)

Limitations to ODBC SQL Grammar (Programming)

When the ODBC Paradox driver executes an *hstmt* on a table, it opens that table (file) and holds it open. The ODBC Paradox driver closes the table when:

- The *hstmt* is dropped with **SQLFreeStmt** with the SQL_DROP option.
- The *hstmt* is prepared again (at which time the table is closed and reopened if necessary).

Because of this, a table cannot be dropped with a DROP TABLE statement until all other *hstmts* executed on the table have been dropped or have been reprepared and no longer refer to that table.

Note Each file used by the ODBC Paradox driver requires a file handle. Because tables (files) remain open until **SQLFreeStmt** is called with the SQL_DROP option, reusing an *hstmt* for different tables without dropping it can result in an error caused by attempting to open too many files. Also note that Paradox indexes require separate files, so opening a single table can use several file handles.

Data Types (Advanced)

See Also

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

Paradox data type	ODBC SQL data type
Alphanumeric	SQL_CHAR
Date	SQL_DATE
Number	SQL_DOUBLE
Short	SQL_SMALLINT

Note **SQLGetTypeInfo** returns ODBC SQL data types. 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.

When converting Paradox data to the C data type SQL_C_TINYINT, numbers from 0 to 127 are converted correctly. Numbers from 128 to 255 are converted to numbers from -128 to -1. Numbers less than 0 or greater than 255 cannot be converted.

When converting data from the C data type SQL_C_TINYINT to Paradox data, numbers from 0 to 127 are converted correctly. Numbers from -128 to -1 are converted to numbers from 128 to 255.

This occurs because SQL_C_TINYINT is signed, but the ODBC Paradox driver uses unsigned single-byte integers.

See Also

For Advanced Users

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

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

Implementation of Data Types (Advanced)

The data type of an expression with nonnumeric operands is the data type of those operands. For example, the expression `FirstName + LastName`, where both `FirstName` and `LastName` are of type `SQL_CHAR`, has the data type `SQL_CHAR`.

The data type of an expression with numeric operands is always `SQL_DOUBLE`.

Error Messages (Advanced)

When an error occurs, the ODBC Paradox driver returns the native error number, the SQLSTATE (an ODBC error code), and an error message.

Native Error

For errors that occur in the ISAM layer, the ODBC Paradox driver returns the native error returned to it by the ODBC File Library (that is, the Paradox ISAM). For errors that are detected by the Simba driver, the ODBC Paradox driver returns a native error of zero.

SQLSTATE

The ODBC Paradox driver or the Driver Manager determines the appropriate SQLSTATE for the error.

Error Message

The ODBC Paradox driver or the Driver Manager generates an error message for the error. Error messages have the following format:

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

where the prefixes in brackets ([]) identify the source of the error. When an error occurs in the Driver Manager or the ODBC Paradox driver, the *[data-source]* prefix is omitted. When an error occurs accessing the Paradox file, the *[data-source]* prefix is included. The following table shows the values of the prefixes returned by the ODBC Paradox driver.

Error message	Error location
<i>[Microsoft][ODBC DLL]error-message</i>	Driver Manager (ODBC.DLL)
<i>[Microsoft][ODBC Single-Tier Driver]error-message</i>	Simba Driver (SIMBA.DLL)
<i>[Microsoft][ODBC Single-Tier Driver][ODBC File Library]error-message</i>	ODBC Paradox Driver (PDX110.DLL)

SQLGetInfo Return Values (Programming)

See Also

The following table lists the C language #defines for the *flInfoType* argument and the corresponding values returned by **SQLGetInfo**. An application retrieves this information by passing the listed C language #defines to **SQLGetInfo** in the *flInfoType* argument. Where **SQLGetInfo** returns a 32-bit bitmask, a vertical bar (|) represents a bitwise OR. For more information about the values return by **SQLGetInfo**, see the *Microsoft ODBC SDK Programmer's Reference, Version 1.0*.

<i>flInfoType</i> value (#define)	Returned value
SQL_ACCESSIBLE_PROCEDURES	"N"
SQL_ACCESSIBLE_TABLES	"N"
SQL_ACTIVE_CONNECTIONS	0
SQL_ACTIVE_STATEMENTS	0
SQL_CONCAT_NULL_BEHAVIOR	1
SQL_CONVERT_BIGINT	0
SQL_CONVERT_BINARY	0
SQL_CONVERT_BIT	SQL_CVT_CHAR SQL_CVT_DOUBLE SQL_CVT_SMALLINT
SQL_CONVERT_CHAR	SQL_CVT_CHAR SQL_CVT_DATE SQL_CVT_DOUBLE SQL_CVT_SMALLINT
SQL_CONVERT_DATE	SQL_CVT_CHAR SQL_CVT_DATE
SQL_CONVERT_DECIMAL	0
SQL_CONVERT_DOUBLE	SQL_CVT_CHAR SQL_CVT_DOUBLE SQL_CVT_SMALLINT
SQL_CONVERT_FLOAT	SQL_CVT_CHAR SQL_CVT_DOUBLE SQL_CVT_SMALLINT
SQL_CONVERT_FUNCTIONS	SQL_FN_CVT_CONVERT
SQL_CONVERT_INTEGER	SQL_CVT_CHAR SQL_CVT_DOUBLE SQL_CVT_SMALLINT
SQL_CONVERT_LONGVARBINARY	0
SQL_CONVERT_LONGVARCHAR	SQL_CVT_CHAR
SQL_CONVERT_NUMERIC	SQL_CVT_CHAR SQL_CVT_DOUBLE SQL_CVT_SMALLINT
SQL_CONVERT_REAL	SQL_CVT_CHAR SQL_CVT_DOUBLE SQL_CVT_SMALLINT
SQL_CONVERT_SMALLINT	SQL_CVT_CHAR SQL_CVT_DOUBLE SQL_CVT_SMALLINT
SQL_CONVERT_TIME	SQL_CVT_CHAR
SQL_CONVERT_TIMESTAMP	SQL_CVT_CHAR SQL_CVT_DATE
SQL_CONVERT_TINYINT	SQL_CVT_CHAR SQL_CVT_DOUBLE

	SQL_CVT_SMALLINT
SQL_CONVERT_VARBINARY	0
SQL_CONVERT_VARCHAR	0
SQL_CORRELATION_NAME	2
SQL_CURSOR_COMMIT_BEHAVIOR	2
SQL_CURSOR_ROLLBACK_BEHAVIOR	0
SQL_DATA_SOURCE_READ_ONLY	"N" (The driver does not check to see whether the disk drive is read-only.)
SQL_DBMS_NAME	"PARADOX"
SQL_DBMS_VER	"3.5"
SQL_DEFAULT_TXN_ISOLATION	0
SQL_DRIVER_NAME	"SIMBA.DLL"
SQL_DRIVER_VER	" 1.01.nnnn" (nnnn specifies the build date.)
SQL_EXPRESSIONS_IN_ORDERBY	"Y"
SQL_FETCH_DIRECTION	SQL_FD_FETCH_NEXT
SQL_IDENTIFIER_CASE	4
SQL_IDENTIFIER_QUOTE_CHAR	"" (double quotation mark).
SQL_MAX_COLUMN_NAME_LEN	25
SQL_MAX_CURSOR_NAME_LEN	18
SQL_MAX_OWNER_NAME_LEN	0
SQL_MAX_PROCEDURE_NAME_LEN	0
SQL_MAX_QUALIFIER_NAME_LEN	66
SQL_MAX_TABLE_NAME_LEN	12
SQL_MULT_RESULT_SETS	"N"
SQL_MULTIPLE_ACTIVE_TXN	"N"
SQL_NON_NULLABLE_COLUMNS	0
SQL_NUMERIC_FUNCTIONS	SQL_FN_NUM_MOD
SQL_ODBC_API_CONFORMANCE	1
SQL_ODBC_SAG_CLI_CONFORMANCE	1
SQL_ODBC_SQL_CONFORMANCE	0
SQL_ODBC_SQL_OPT_IEF	"N"
SQL_OUTER_JOINS	"Y"
SQL_OWNER_TERM	""
SQL_PROCEDURE_TERM	""
SQL_PROCEDURES	"N"
SQL_QUALIFIER_NAME_SEPARATOR	"\" (backslash)
SQL_QUALIFIER_TERM	"DIRECTORY"
SQL_ROW_UPDATES	"Y"
SQL_SCROLL_CONCURRENCY	SQL_SCCO_READ_ONLY
SQL_SCROLL_OPTIONS	SQL_SO_FORWARD_ONLY
SQL_SEARCH_PATTERN_ESCAPE	"\" (backslash)
SQL_SERVER_NAME	"PARADOX"
SQL_STRING_FUNCTIONS	SQL_FN_STR_CONCAT SQL_FN_STR_LCASE SQL_FN_STR_LEFT SQL_FN_STR_LENGTH SQL_FN_STR_LOCATE

	SQL_FN_STR_LTRIM
	SQL_FN_STR_RIGHT
	SQL_FN_STR_RTRIM
	SQL_FN_STR_SUBSTRING
	SQL_FN_STR_UCASE
SQL_SYSTEM_FUNCTIONS	SQL_FN_SYS_DBNAME
	SQL_FN_SYS_USERNAME
SQL_TABLE_TERM	"TABLE"
SQL_TIMEDATE_FUNCTIONS	SQL_FN_TD_CURDATE
	SQL_FN_TD_CURTIME
	SQL_FN_TD_DAYOFMONTH
	SQL_FN_TD_DAYOFWEEK
	SQL_FN_TD_MONTH
	SQL_FN_TD_YEAR
SQL_TXN_CAPABLE	0
SQL_TXN_ISOLATION_OPTIONS	0

See Also

For Advanced Users

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

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

ODBC API Functions (Programming)

See Also

The ODBC Paradox driver supports all Core and Level 1 functions and the following Level 2 functions:

- SQLDataSources
- SQLMoreResults

These ODBC API functions have the following implementations with the ODBC Paradox driver.

Function	Description
SQLDriverConnect	The following keywords are supported in the <u>connection string</u> : DSN , UID , PWD , DBQ , and FIL .
SQLGetConnectOption and SQLSetConnectOption	These functions support the SQL_ACCESS_MODE, SQL_CURRENT_QUALIFIER, SQL_OPT_TRACE, and SQL_OPT_TRACEFILE connect options. SQLGetConnectOption also supports the SQL_AUTOCOMMIT option. The Paradox driver also supports an additional connection option (1150). The <i>vParam</i> argument is a 32-bit window handle. If the window handle is a null pointer, the ODBC Paradox driver does not display any dialog boxes (except those used by SQLDriverConnect). If the handle is not a null pointer, it should be the parent window handle of the application; the driver uses this handle to display dialog boxes.
SQLGetData	This function can retrieve data from any column, whether or not there are bound columns after it and regardless of the order in which the columns are retrieved.
SQLGetInfo	SQLGetInfo supports a driver-specific information type, SQL_FILE_USAGE (65002). The returned value is a 16-bit integer that indicates how the driver directly treats files in a data source: 0 (SQL_FILE_NOT_SUPPORTED) = The driver is not a single-tier driver. 1 (SQL_FILE_TABLE) = A single-tier driver treats files in a data source as tables.

3 (SQL_FILE_QUALIFIER) = A single-tier driver treats files in a data source as a qualifier.

The ODBC Paradox driver returns 1, since each Paradox file is a table.

SQLGetStmtOption
and
SQLSetStmtOption

These functions support the SQL_MAX_LENGTH, SQL_MAX_ROWS, and SQL_NOSCAN statement options.

They also support a driver-specific statement option, SQL_LOCK TABLES.

The Paradox driver also supports an additional statement option (1151), with which an application specifies the password for a table. The *vParam* argument is a null-terminated character string that uses the format *tablename,password*.

SQLGetTypeInfo

Only data type names returned by **SQLGetTypeInfo** can be used with CREATE statements.

SQLMoreResults

SQLMoreResults always returns SQL_NO_DATA_FOUND. It cannot return additional results.

SQLTransact

The ODBC Paradox driver does not support transactions; all SQL statements that modify data are always committed. Therefore, **SQLTransact** always returns SQL_SUCCESS when *fType* is SQL_COMMIT and always returns SQL_ERROR when *fType* is SQL_ROLLBACK.

See Also

For Advanced Users

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

For Programmers

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

SQL_LOCK_TABLES Statement Option (Programming)

The ODBC Paradox driver supports a driver-specific statement option, `SQL_LOCK_TABLES`, that supports the values `DEFLOCK` and `XLOCK`. When the value of the option is `DEFLOCK`, tables used by the *hstmt* are subject only to the default locking mechanisms of the ODBC Paradox driver. This is the default setting.

When the value of the option is `XLOCK`, all tables used by the *hstmt* are exclusively locked when a **SELECT**, **UPDATE**, **INSERT**, **DELETE**, or **CREATE INDEX** statement is executed on the *hstmt*. The tables remain locked until the *hstmt* is dropped (by calling **SQLFreeStmt** with the `SQL_DROP` option) or the option is set to `DEFLOCK` and the *hstmt* is reexecuted.

Locked tables can only be used by the locking *hstmt*; they cannot be used by any other *hstmts*. For example, the last function call in the following code generates an access violation because it uses a different *hstmt*:

```
SQLSetStmtOption(hstmt1, SQL_LOCK_TABLES, XLOCK);
SQLExecDirect(hstmt1, "SELECT * FROM EMP", SQL_NTS);
SQLExecDirect(hstmt1, "UPDATE EMP SET DEPT=12", SQL_NTS);
/* This call generates an access violation */
SQLExecDirect(hstmt2, "SELECT * FROM EMP", SQL_NTS);
```

Furthermore, **SQLColumns**, **SQLSpecialColumns**, and **SQLStatistics** cannot retrieve information about a table locked on another *hstmt*.

Caution Be careful to avoid locking more files than necessary. If the same *hstmt* is used to execute statements for many different tables, all the tables will remain locked until the *hstmt* is dropped or locking is turned off.

The `SQL_LOCK_TABLES` statement option uses the following #defines:

```
#define SQL_LOCK_TABLES 1153
#define DEFLOCK 0
#define XLOCK 1
```

The `SQL_LOCK_TABLES` statement option cannot be used with **SQLSetConnectOption**.

Implementation Issues (Programming)

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

Arithmetic Errors

The ODBC Paradox driver evaluates the WHERE clause in a SELECT statement as it fetches each row. If a row contains a value that causes an arithmetic error, such as divide-by-zero or numeric overflow, the driver returns all rows, but returns errors for columns with arithmetic errors. When inserting or updating, however, the ODBC Paradox driver stops inserting or updating data when the first arithmetic error is encountered.

Multiple *hstmt*s

In order to execute more than one *hstmt* on a table, the ODBC Paradox driver requires that the table have a unique index (Paradox primary key).

Setup DLL

The ODBC Administrator calls the function **ConfigDSN** when users configure data sources. For the ODBC Paradox driver, this function is in a setup DLL (SIMADMIN.DLL).

Stack Size

The ODBC Paradox driver requires that any application have a stack size of at least 4K remaining after the driver is called. A stack size of 6K is recommended.

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.

