

## **ODBC Microsoft Excel Driver**

### **For All Users**

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

[Overview](#)

[Hardware and Software Requirements](#)

[Setting Up the ODBC Microsoft Excel Driver](#)

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

[Connecting to a Microsoft Excel Data Source](#)

[Using the ODBC Microsoft Excel Driver](#)

### **For Advanced Users**

The following topics discuss how to use the ODBC Microsoft Excel 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 Microsoft Excel 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 \(Programming\)](#)

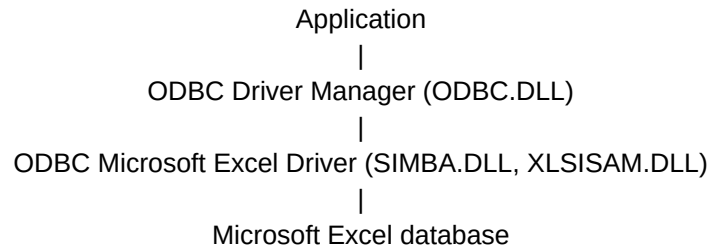
[Implementation Issues \(Programming\)](#)

## Overview

See Also

In Microsoft Excel, a range of worksheet data can be defined as a database. The ODBC Microsoft Excel driver allows you to open and query a Microsoft Excel database through the Open Database Connectivity ([ODBC](#)) interface.

The application/driver architecture is:



**See Also**

For All Users

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

[Connecting to a Microsoft Excel Data Source](#)

[Hardware and Software Requirements](#)

[Setting Up the ODBC Microsoft Excel Driver](#)

[Using the ODBC Microsoft Excel Driver](#)

## Hardware and Software Requirements

See Also

To access a Microsoft Excel database, you must have:

- The ODBC Microsoft Excel driver.  
The ODBC Microsoft Excel driver can access Microsoft Excel version 3.0 (BIFF3), or 4.0 (BIFF4) worksheet files.
- ODBC Driver Manager 1.0 (ODBC.DLL).
- A computer running MS-DOS 3.3 or later.
- Microsoft Windows 3.0a or later.

To add, modify, or delete drivers or data sources, you should have the ODBC Control Panel option (or the ODBC Administrator program if you're running Windows 3.0a) installed on your computer.

For more information about Microsoft Excel databases, see the Microsoft Excel documentation.

**See Also**

For All Users

[Setting Up the ODBC Microsoft Excel Driver](#)

## Setting Up the ODBC Microsoft Excel Driver

See Also

### To set up the ODBC Microsoft Excel 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 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 Add Driver dialog box, enter the name of the drive and directory containing the ODBC Microsoft Excel driver in the text box. Or choose the Browse button to select a drive and directory name.
- 5 Choose the OK button.
- 6 In the Install Drivers dialog box, choose Excel from the Available ODBC Drivers list.
- 7 Choose the OK button to install the driver.

---

**Note** The ODBC Microsoft Excel 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 Microsoft Excel driver, regardless of whether it has been installed. Choose the Yes button to install the driver.

---

Before using the driver, you must add a data source for it.

### To delete the ODBC Microsoft Excel 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 Microsoft Excel driver from the Installed ODBC Drivers list.
- 4 Choose the Delete button.
- 5 The ODBC Microsoft Excel 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](#)

## Adding, Modifying, and Deleting a Microsoft Excel Data Source

See Also

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

### To add a Microsoft Excel 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 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 Excel from the Installed ODBC Drivers list and choose OK.
- 4 In the ODBC Microsoft Excel Setup dialog box, enter information to set up the data source.

### To modify a Microsoft Excel 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 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 Microsoft Excel Setup dialog box, enter information to set up the data source.

### To delete a Microsoft Excel 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 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 Microsoft Excel Data Source](#)

[Setting Up the ODBC Microsoft Excel Driver](#)

**Excel data source**

An Excel data source specifies the default directory in which the ODBC Excel driver searches for Excel files you want to access.

## ODBC Microsoft Excel Setup Dialog Box

See Also

The ODBC Microsoft Excel Setup dialog box contains 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."

### Database Directory

Displays the currently selected directory. 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.

When defining a Microsoft Excel data source directory, specify the directory where your most commonly used Microsoft Excel files are located. The ODBC Microsoft Excel driver uses this directory as the default directory. Copy other Microsoft Excel files into this directory if they are used frequently. Alternatively, you can qualify file names in a SELECT statement with the directory name:

```
SELECT * FROM C:\MYDIR\EMP
```

Or, you can use the USE statement to specify a new default directory:

```
USE C:\MYDIR
```

### Options

Displays the following option:

### Rows to Scan

The number of rows to scan to determine the data type of each column. To scan the entire file, enter 0.

**See Also**

For All Users

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

## Connecting to a Microsoft Excel Data Source

### See Also

When you connect to a Microsoft Excel data source, an application may prompt you to enter the name of a directory. If you are prompted, enter or select the directory containing the Microsoft Excel database files you want to access.

**See Also**

For All Users

[Adding, Modifying, and Deleting a Microsoft Excel Data Source  
Using the ODBC Microsoft Excel Driver](#)

For Advanced Users

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

## Using the ODBC Microsoft Excel Driver

The following information may be useful when using the ODBC Microsoft Excel driver:

### Columns

- Column names are limited to 64 characters or less. Column names longer than 64 characters generate an error condition.
- The driver allows column names to contain any valid Microsoft Excel characters (for example, spaces). If column names contain any characters except letters, numbers, and underscores, they must be delimited. To delimit a column name, enclose the name in double quotes("").
- The driver supports a maximum of 255 columns when a table is created.

### Literals

- The maximum length of any literal (for example, a string) is 1000 characters.
- A character string literal can be any ANSI character (1 - 255 decimal). Use two consecutive single quotation marks (') to represent one single quotation mark (').

### Tables

- All Microsoft Excel tables (Microsoft Excel spreadsheets) are opened shared (that is, more than one user can open them at a time), except for those created and opened for inserting, which are opened exclusive and can be opened by only one user at a time.
- The ODBC Microsoft Excel driver cannot access data in password-protected tables.
- The ODBC Microsoft Excel driver creates new tables in Microsoft Excel version 3.0 (BIFF3) format.
- The Microsoft Excel driver cannot insert data into an existing table. Instead, an application must create a new table and insert the data into that table. The application cannot perform other operations on the table until it is closed and reopened. No more data can be inserted into the table after it has been closed.
- The Microsoft Excel driver cannot update or delete data in an existing table.
- The Microsoft Excel driver allows table names to contain any valid MS-DOS characters. If table names contain any characters except letters, numbers, and underscores, they must be delimited. To delimit a table name, enclose the name in double quotes("").

## Connection Strings (Advanced)

See Also

The connection string for the ODBC Microsoft Excel driver uses the following keywords:

<b>Keyword</b>	<b>Description</b>
<b>DSN</b>	Name of the Microsoft Excel data source.
<b>DBQ</b>	The Microsoft Excel directory.
<b>FIL</b>	File type (EXCEL).

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

```
DSN=Accounting;DBQ=C:\ACCT;FIL=EXCEL
```



**See Also**

For All Users

[Connecting to a Microsoft Excel Data Source](#)

For Programmers

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

## SQL Statements (Advanced)

[See Also](#)

The ODBC Microsoft Excel Driver supports most SQL statements and clauses in the ODBC minimum grammar. While the driver supports the grammar for reading data, it has limitations for writing data. 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 \(Advanced\)](#)

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

**See Also**

For Advanced Users

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

## Additional Supported ODBC SQL Grammar (Advanced)

See Also

The following table lists core and extended ODBC SQL grammar supported by the ODBC Microsoft Excel Driver.

Core and Extended Grammar Supported	Comments
Approximate numeric literal	Supported.
AVG( <i>expression</i> ), COUNT(*), MAX( <i>expression</i> ), MIN( <i>expression</i> ), and SUM( <i>expression</i> )	See also the description of COUNT( <i>expression</i> ) in <u>Driver-specific ODBC SQL Grammar</u> .
BETWEEN predicate	Supported.
Correlation names are fully supported, including within the table list.	For example, in the following string, E1 is the correlation name for the table named Emp: <pre>SELECT * FROM Emp E1 WHERE E1.LastName = 'Smith'</pre>
Exact numeric literal	Supported.
[GROUP BY <i>column-name</i> [, <i>column-name</i> ]...]	Supported.
[HAVING <i>search-condition</i> ]	Supported.
IN ( <i>valuelist</i> )	Implemented as specified in the ODBC core grammar. For example: <pre>SELECT * FROM EMP WHERE Dept IN ('Sales', 'Marketing')</pre>
INSERT supports pathnames with table names.	Supported.

**See Also**

For Advanced Users

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

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

## Driver-specific ODBC SQL Grammar Supported (Advanced)

[See Also](#)

The ODBC Microsoft Excel driver supports the following driver-specific ODBC SQL grammar:

Driver-specific ODBC SQL Grammar	Comments
BETWEEN predicate	The syntax: <i>expression1</i> BETWEEN <i>expression2</i> AND <i>expression3</i> returns True only if <i>expression1</i> is greater than or equal to <i>expression2</i> and <i>expression1</i> is less than or equal to <i>expression3</i> .
COUNT( <i>expression</i> )	Counts all non-NULL values for an expression across a predicate. This function behaves like other set functions, such as SUM, AVG, MIN, and MAX. For example: <pre>SELECT COUNT (A+B) FROM Q</pre> counts all the rows in Q where A+B does not equal NULL.
Date arithmetic	The driver supports adding and subtracting an integer from a date. The integer specifies the number of days to add or subtract. The driver also supports subtracting one date from another to return a number of days.
Date literals	The YYYY-MM-DD format is supported.
GROUP BY <i>expression-list</i>	GROUP BY supports an expression list as well as a column name.
ORDER BY <i>expression-list</i>	If the expression is a single integer literal, it is interpreted as the number of the column in the result set. Ordering is done on one of the result table columns. No ordering is allowed on Set functions or an expression that contains a Set function. For example, in the following clauses the table is ordered by three key expressions: a+b, c+d, and e. <pre>SELECT * FROM emp ORDER BY a+b, c+d, e</pre>
ORDER BY with GROUP BY	ORDER BY can be performed on any expression in the GROUP BY <i>expression-list</i> or any column in the result set.
<a href="#">Outer Joins</a>	A SELECT statement can contain a list of OUTER JOIN clauses.
<a href="#">Scalar Functions</a>	Supported.
Table names that occur in the FROM clause of SELECT, after the INTO clause in INSERT, and after CREATE and DROP TABLE can contain a valid pathname, table name, and filename extension.	Use of a table name elsewhere in an SQL statement does not support the use of pathnames or extensions but will accept only the primary name (for example, EMP FROM C:\ABC\EMP.XLS). If no extension is specified, .XLS is assumed. Correlation names can also be used. For example: <pre>SELECT * FROM C:\ABC\EMP WHERE EMP.COL1 = 'aaa'</pre>
USE [ <i>drive:</i> ] <i>dir</i>	Sets the current database directory. <i>drive</i> is a valid drive name and <i>dir</i> is any valid MS-

DOS directory name.

For example, the following changes the current directory to C:\DBDIR:

```
USE C:\DBDIR
```

USE does the same thing as setting DataDirectory to an MS-DOS directory in your ODBC.INI file.

**See Also**

For Advanced Users

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

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



## Outer Joins (Advanced)

See Also

The ODBC Microsoft Excel 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 \(Advanced\)](#)

## Limitations to ODBC SQL Grammar (Advanced)

See Also

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

<b>Grammar</b>	<b>Limitation</b>
AND predicates	A maximum of 300 supported.
Comparison predicate	For SQL_BIT data, comparisons can be made for equality and inequality only (= and <> operators).
CREATE INDEX, DELETE, DROP INDEX, and UPDATE	Not supported.
LIKE predicate	If data in a column is longer than 255 characters, the LIKE comparison will be based only on the first 255 characters.
Sort Keys	<p>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.</p> <p>If the length of the data in a column is greater than 255 characters, sorting will be based on the first 255 characters.</p>

**See Also**

For Advanced Users

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

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

## Data Types (Advanced)

[See Also](#)

The following table shows how Microsoft Excel driver data types are mapped to ODBC SQL data types. The Microsoft Excel driver assigns these data types to column in Microsoft Excel tables based on the data in the column.

Microsoft Excel Driver Data Type	ODBC SQL Data Type
LOGICAL	SQL_BIT
DATE	SQL_DATE
NUMBER	SQL_DOUBLE
TIME	SQL_TIME
DATETIME	SQL_TIMESTAMP
TEXT	SQL_VARCHAR

---

**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.

---

### Limitations

The ODBC Microsoft Excel driver and Microsoft Excel impose the following limitations on the data types:

- SQL\_VARCHAR columns are limited to 255 bytes or less.
- When the rows of a column contain more than one Microsoft Excel data type, the ODBC Microsoft Excel driver assigns the SQL\_VARCHAR data type to the column. There is one exception to this: if the column only contains two or three of the date/time data types (DATE, TIME, and DATETIME), the ODBC Microsoft Excel driver assigns the SQL\_TIMESTAMP data type to the column.
- For the FALSE and TRUE values of the Microsoft Excel LOGICAL data type, the driver returns the values 0 and 1 if the column is an SQL\_BIT column, "FALSE" and "TRUE" if the column is an SQL\_VARCHAR column.
- For the Microsoft Excel error values (#N/A!, #VALUE!, #REF!, #DIV/0!, #NUM!, #NAME?, and #NULL!), the ODBC Microsoft Excel driver returns a corresponding character string. For example, for the error value #N/A!, the ODBC Microsoft Excel driver returns "#N/A!".
- The Microsoft Excel LOGICAL values and error values cannot be inserted into a Microsoft Excel table. For example, inserting the character string "#N/A!" into an SQL\_VARCHAR column results in a Microsoft Excel cell containing the string "#N/A!," not the error value #N/A!.
- When converting Microsoft Excel 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 Microsoft Excel 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 Microsoft Excel driver uses unsigned single-byte integers.

**See Also**

For Advanced Users

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

## Error Messages (Advanced)

When an error occurs, the ODBC Microsoft Excel Driver returns the native error number, the SQLSTATE (an ODBC error code), and an error message.

### Native Error

For errors that occur in the data source, the Microsoft driver returns the native error returned to it by the ODBC File Library. For errors that are detected by the driver or the Driver Manager, the ODBC Microsoft Excel Driver returns a native error of zero.

### SQLSTATE

For errors that occur in the data source, the ODBC Microsoft Excel Driver maps the returned native error to the appropriate SQLSTATE. For errors that are detected by the driver or the Driver Manager, the ODBC Microsoft Excel Driver or Driver Manager generates the appropriate SQLSTATE.

### Error Message

For errors that occur in the data source, the ODBC Microsoft Excel Driver returns an error message returned to it by the ODBC File Library. For errors that occur in the ODBC Microsoft Excel Driver or the Driver Manager, the driver returns an error message based on the text associated with the SQLSTATE.

Error messages have the following format:

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

where the prefixes in brackets ([ ]) identify the location of the error. When the error occurs in the Driver Manager or Simba driver, *data-source* is not given. 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.

The following table shows the error messages returned by the Driver Manager, Simba driver, and Microsoft Excel ISAM:

Error Message	Error location
[Microsoft][ODBC DLL]message-text	Driver Manager (ODBC.DLL)
[Microsoft][ODBC Single-Tier Driver]message-text	Simba Driver (SIMBA.DLL)
[Microsoft][ODBC Single-Tier Driver][ODBC File Library]message-text	Microsoft Excel ISAM (XLSISAM.DLL)

## SQLGetInfo Return Values (Programming)

[See Also](#)

The following table lists the C language #defines for the *fInfoType* argument and the corresponding values returned by **SQLGetInfo**. This information can be retrieved by passing the listed C language #defines to **SQLGetInfo** in the *fInfoType* 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>fInfoType</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_BIT   SQL_CVT_DOUBLE   SQL_CVT_VARCHAR
SQL_CONVERT_CHAR	SQL_CVT_BIT   SQL_CVT_DATE   SQL_CVT_DOUBLE   SQL_CVT_TIME   SQL_CVT_TIMESTAMP   SQL_CVT_VARCHAR
SQL_CONVERT_DATE	SQL_CVT_DATE   SQL_CVT_TIMESTAMP   SQL_CVT_VARCHAR
SQL_CONVERT_DECIMAL	0
SQL_CONVERT_DOUBLE	SQL_CVT_BIT   SQL_CVT_DOUBLE   SQL_CVT_VARCHAR
SQL_CONVERT_FLOAT	SQL_CVT_BIT   SQL_CVT_DOUBLE   SQL_CVT_VARCHAR
SQL_CONVERT_FUNCTIONS	SQL_FN_CVT_CONVERT
SQL_CONVERT_INTEGER	SQL_CVT_BIT   SQL_CVT_DOUBLE   SQL_CVT_VARCHAR
SQL_CONVERT_LONGVARBINARY	0
SQL_CONVERT_LONGVARCHAR	SQL_CVT_VARCHAR
SQL_CONVERT_NUMERIC	SQL_CVT_BIT   SQL_CVT_DOUBLE   SQL_CVT_VARCHAR
SQL_CONVERT_REAL	SQL_CVT_BIT   SQL_CVT_DOUBLE   SQL_CVT_VARCHAR
SQL_CONVERT_SMALLINT	SQL_CVT_BIT   SQL_CVT_DOUBLE   SQL_CVT_VARCHAR
SQL_CONVERT_TIME	SQL_CVT_TIME   SQL_CVT_TIMESTAMP   SQL_CVT_VARCHAR
SQL_CONVERT_TIMESTAMP	SQL_CVT_DATE



	SQL_CVT_TIME
	SQL_CVT_TIMESTAMP
	SQL_CVT_VARCHAR
SQL_CONVERT_TINYINT	SQL_CVT_BIT
	SQL_CVT_DOUBLE
	SQL_CVT_VARCHAR
SQL_CONVERT_VARBINARY	0
SQL_CONVERT_VARCHAR	SQL_CVT_BIT
	SQL_CVT_DATE
	SQL_CVT_DOUBLE
	SQL_CVT_TIME
	SQL_CVT_TIMESTAMP
	SQL_CVT_VARCHAR
SQL_CORRELATION_NAME	2
SQL_CURSOR_COMMIT_BEHAVIOR	2
SQL_CURSOR_ROLLBACK_BEHAVIOR	0
SQL_DATA_SOURCE_READ_ONLY	"Y"
SQL_DBMS_NAME	"EXCEL"
SQL_DBMS_VER	"4.0"
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	64
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	"N"
SQL_SCROLL_CONCURRENCY	SQL_SCCO_READ_ONLY
SQL_SCROLL_OPTIONS	SQL_SO_FORWARD_ONLY

SQL_SEARCH_PATTERN_ESCAPE	"\" (backslash)
SQL_SERVER_NAME	"EXCEL"
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

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

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

For Programmers

[Scalar Functions \(Programming\)](#)

## Scalar Functions (Programming)

The ODBC Microsoft Excel driver support the following scalar functions:

CONCAT	LCASE	RIGHT
CONVERT	LEFT	RTRIM
CURDATE	LENGTH	SUBSTRING
CURTIME	LOCATE	UCASE
DATABASE	LTRIM	USER
DAYOFMONTH	MOD	YEAR
DAYOFWEEK	MONTH	

For information about the arguments and return values of scalar functions, see Appendix G of the *Microsoft ODBC SDK Programmer's Reference*.

## ODBC API Functions (Programming)

[See Also](#)

The ODBC Microsoft Excel 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 Microsoft Excel driver:

Function	Description
<b>SQLColumns</b>	This function returns results only for Microsoft Excel files with a .XLS filename extension.
<b>SQLDriverConnect</b>	The following keywords are supported in the <u>connection string</u> : <b>DSN</b> , <b>DBQ</b> , and <b>FIL</b> .
<b>SQLGetConnectOption</b> and <b>SQLSetConnectOption</b>	These functions support the SQL_ACCESS_MODE, SQL_CURRENT_QUALIFIER, SQL_OPT_TRACE, and SQL_OPT_TRACEFILE connection options. <b>SQLGetConnectOption</b> also supports the SQL_AUTOCOMMIT option.
<b>SQLGetCursorName</b> and <b>SQLSetCursorName</b>	These functions are supported, but cannot be used for positioned updates or deletes (for example, WHERE CURRENT OF <i>cursorname</i> ).
<b>SQLGetData</b>	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.
<b>SQLGetInfo</b>	<b>SQLGetInfo</b> 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 Microsoft Excel driver returns 1, since each Microsoft Excel file is a table.
<b>SQLGetStmtOption</b> and <b>SQLSetStmtOption</b>	These functions support the SQL_MAX_LENGTH, SQL_MAX_ROWS, and SQL_NOSCAN statement options.
<b>SQLGetTypeInfo</b>	Only data type names returned by

<b>SQLMoreResults</b>	<b>SQLGetTypeInfo</b> can be used with CREATE statements.
<b>SQLSpecialColumns</b>	<b>SQLMoreResults</b> always returns SQL_NO_DATA_FOUND. It cannot return additional results.
<b>SQLTables</b>	This function always returns SQL_SUCCESS and an empty result set.
<b>SQLTransact</b>	The table names returned by <b>SQLTables</b> have no filename extensions. The function only returns Excel files containing defined database ranges.
	This function supports COMMIT but not ROLLBACK.

### See Also

For Advanced Users

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

For Programmers

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

## Implementation Issues (Programming)

The following information might affect the use of the ODBC Microsoft Excel Driver.

### Arithmetic Errors

The Microsoft Excel 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, however, the Microsoft Excel driver stops inserting data when the first arithmetic error is encountered.

### Closing Open Tables (Files)

Calling **SQLFreeStmt** with the SQL\_CLOSE option changes the statement state but does not close the tables used by the *hstmt*. To close the tables currently used by *hstmt*, you must call **SQLFreeStmt** with the SQL\_DROP option.

In the following example, when **SQLFreeStmt** is called, the emp and dept tables remain open:

```
SQLPrepare(hStmt, "SELECT * FROM emp, dept  
WHERE emp.dept = dept.dept_id", SQL_NTS);  
SQLExecute(hStmt);  
/* .SQLFetch until NO_DATA_FOUND  
SQLFreeStmt(hStmt, SQL_CLOSE);  
SQLPrepare(hStmt, "SELECT * FROM emp", SQL_NTS);
```

---

**Note** Each file used by the ODBC Microsoft Excel 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.

---

### Sorting with DISTINCT, GROUP BY, or ORDER BY

DISTINCT, GROUP BY, and ORDER BY always result in a sort. A temporary table is created from the data and the sort occurs on the temporary table. This type of sort is not based on dynamic data since the temporary table is built from data found in the original data file at **SQLExecute** time.

**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.

