ODBC Microsoft Access Driver

For All Users

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

Overview

Hardware and Software Requirements

Setting Up the ODBC Microsoft Access Driver

Adding, Modifying, and Deleting a Microsoft Access Data Source

Connecting to a Microsoft Access Data Source

Using the ODBC Microsoft Access Driver

For Advanced Users

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

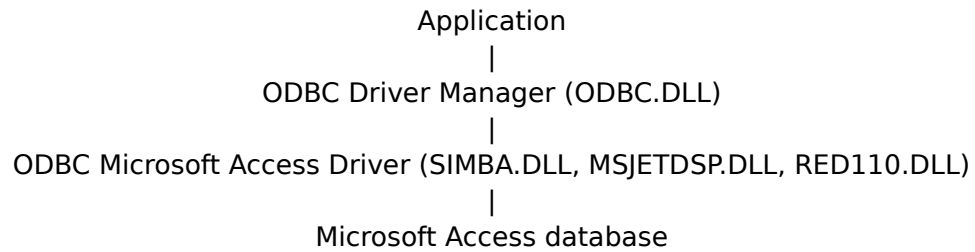SQLGetInfo Return Values (Programming)

ODBC API Functions (Programming)

Implementation Issues (Programming)

Overview

The ODBC Microsoft Access driver allows you to open and query a Microsoft Access database through the Open Database Connectivity (ODBC) interface.

The application/driver architecture is:

```
                        Application
                            |
              ODBC Driver Manager (ODBC.DLL)
                            |
ODBC Microsoft Access Driver (SIMBA.DLL, MSJETDSP.DLL, RED110.DLL)
                            |
                  Microsoft Access database
```

See Also

For All Users

Hardware and Software Requirements

To access Microsoft Access data, you must have:

- The ODBC Microsoft Access driver.
- The 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 Microsoft Windows 3.0a) installed on your computer.

If you do not have Microsoft Access installed on your computer, make the following settings in your CONFIG.SYS and AUTOEXEC.BAT files:

- Add the line FILES= 50 to CONFIG.SYS.
- Add the line SHARE.EXE /L:200 to AUTOEXEC.BAT. If you are working with very large databases, add SHARE.EXE /L:500.

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

See Also

For All Users

Setting Up the ODBC Microsoft Access Driver

To set up the ODBC Microsoft Access 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 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 Access 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 Access Data from the Available ODBC Drivers list.
7 Choose the OK button to install the driver.

> **Note**   The ODBC Microsoft Access 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 Access 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 Access 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 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 Access driver from the Installed ODBC Drivers list.
4 Choose the Delete button.
   The ODBC Microsoft Access setup program asks if you want to remove the driver and all the data sources that use the driver.
5 Choose the Yes button.

See Also

For All Users

Adding, Modifying, and Deleting a Microsoft Access Data Source

Before you can access data with the ODBC Microsoft Access driver, you must add a <u>data source</u> for it.   You can change or delete a data source at any time.

To add a Microsoft Access 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 3.0a, start the ODBC Administrator by double-clicking the Microsoft ODBC Administrator icon in the Microsoft ODBC program group.

2 In the Data Sources dialog box, choose the Add button.
3 In the Add Data Source dialog box, select Access from the Installed ODBC Drivers list and choose OK.
4 In the <u>ODBC Microsoft Access Setup dialog box</u>, enter information to set up the data source.

To modify a Microsoft Access 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 3.0a, start the ODBC Administrator by double-clicking the Microsoft ODBC Administrator icon in the Microsoft ODBC program 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 <u>ODBC Microsoft Access Setup dialog box</u>, enter information to set up the data source.

To delete a Microsoft Access 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 3.0a, start the ODBC Administrator by double-clicking the Microsoft ODBC Administrator icon in the Microsoft ODBC program 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

Microsoft Access data source

A Microsoft Access data source specifies the Access database (.MDB) you want to use and the information needed to get to that database.

ODBC Microsoft Access Setup Dialog Box

The ODBC Microsoft Access 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

The full path of the Microsoft Access database you want to access.   Use the Select Database button to select an existing database.   Use the Create Database button to create and select a new Microsoft Access database.

Options

Displays the following options:

System Database

The full path of the Microsoft Access system database to be used with the Microsoft Access database you want to access.   Select the None option button to log in to the Microsoft Access database as the Admin user; no system database will be used.   Select the Database option button to use the specified system database.   If the Database option button is selected, use the Select Database button to select a different system database. For more information about system databases and the Admin user, see the Microsoft Access documentation.

Caution   The Page Timeout and Buffer Size options apply to all data sources that use the ODBC Microsoft Access driver.

Page Timeout

Specifies 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).

Buffer Size

The size of the internal buffer, in kilobytes, that is used by Microsoft Access to transfer data to and from the disk.   The default buffer size is 256K (displayed as 256).

Exclusive

If the Exclusive box is selected, the Microsoft Access database will be opened in Exclusive mode and can be accessed by only one user at a time.   If the Exclusive box is cleared, the Microsoft Access database 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.

See Also
For All Users

Connecting to a Microsoft Access Data Source

When you connect to a Microsoft Access data source, an application may prompt you to enter the name of a Microsoft Access database.   If you are prompted, enter or select the full path of the Microsoft Access database you want to access.

If the data source uses a system database, an application may prompt you for your name and password.   The default name is Admin and the default password in Access for the Admin user is an empty string.

See Also

For All Users

Adding, Modifying, and Deleting a Microsoft Access Data Source

Using the ODBC Microsoft Access Driver

For Advanced Users

Connection Strings (Advanced)

Using the ODBC Microsoft Access Driver

The following information may be helpful when using the ODBC Microsoft Access driver:

Columns

- Column names are limited to 64 characters or less.   Column names longer than 64 characters generate an error.
- The driver allows column names to contain any valid Microsoft Access 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 expects the Microsoft Access database to have column names in the first row of the database.
- The driver supports a maximum of 255 columns when a table is created.

Tables

- The ODBC Microsoft Access driver does not lock access to tables, although you (or a user) can do this from within Microsoft Access.   Multiple users can read, but not update tables.
- The driver supports table name lengths of up to 64 characters.
- The driver allows table names to contain any valid Microsoft Access characters (for example, spaces).   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(").

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 (').

Character Sets

The ODBC Microsoft Access driver supports the following character sets:

- ENGLISH/FRENCH/GERMAN/PORTUGUESE/ITALIAN
- SPANISH
- NORDIC
- DUTCH

System Security

If you specify a system database with the System Database option in the ODBC Microsoft Access Setup dialog box, the ODBC Microsoft Access driver prompts the user for a name and password.   The default name is Admin and the default password in Access for the Admin user is an empty string.   If you do not specify a system database, the ODBC Microsoft Access driver automatically logs in as the Admin user and no password is required.

To increase the security of your Microsoft Access database, create a new user to replace the Admin user and delete the Admin user, or change the objects to which the Admin user has access.   For more information, see the Microsoft Access documentation.

Connection Strings (Advanced)

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

Keyword	Description
DSN	Name of the Microsoft Access data source.
UID	The user login ID.
PWD	The user-specified password.
DBQ	The Microsoft Access database.
FIL	File type (RedISAM).

For example, to open a Microsoft Access data source named Accounting in the directory C:\ACCESS, use the following connection string:

```
DSN=Accounting;UID=Smith;PWD=Sesame;DBQ=C:\ACCESS\ACCT.MDB;FIL=RedISAM
```

See Also

For All Users

SQL Statements (Advanced)

The ODBC Microsoft Access 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)

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

Core and Extended Grammar	Comments
Approximate numeric literal	Supported.
AVG(*expression*), COUNT(*), MAX(*expression*), MIN(*expression*), and SUM(*expression*)	See also the description of COUNT(*expression*) 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:
	
	`SELECT * FROM Emp E1`
	`WHERE E1.LastName = 'Smith'`

Exact numeric literal	Supported.
[GROUP BY *column-name* [,*column-name*]...]	Supported.
[HAVING *search-condition*]	Supported.

IN (*valuelist*)	Implemented as specified in the ODBC core grammar. For example:
	
	`SELECT * FROM EMP`
	`WHERE Dept`
	`IN ('Sales','Marketing')`

INSERT supports pathnames with table names.	Supported.

See Also

For Advanced Users

Driver-specific ODBC SQL Grammar Suported (Advanced)

Limitations to ODBC SQL Grammar (Advanced)

Driver-specific ODBC SQL Grammar Supported (Advanced)

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

Driver-specific ODBC SQL Grammar	Comments
BETWEEN predicate	The syntax:
	expression1 BETWEEN *expression2* AND *expression3*
	returns True only if *expression1* is greater than or equal to *expression2* and *expression1* is less than or equal to *expression3*.
COUNT(*expression*)	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:
	`  SELECT COUNT(A+B) FROM Q`
	counts all the rows in Q where A+B does not equal NULL.
Date literals	The YYYY-MM-DD format is supported.
GROUP BY *expression-list*	GROUP BY supports an expression list as well as a column name.
ORDER BY *expression-list*	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.
	`  SELECT * FROM emp` `  ORDER BY a+b,c+d,e`
ORDER BY with GROUP BY	ORDER BY can be performed on any expression in the GROUP BY *expression-list* or any column in the result set.
Outer Joins	A SELECT statement can contain a list of OUTER JOIN clauses.
Scalar Functions	Supported.
USE [*drive*:]*dir**filename*	Sets the current database file. *drive* is a valid drive name, *dir* is any valid MS-DOS directory name, and *filename* is the name of the Microsoft Access database file (including the .MDB extension).
	For example, the following changes the current database to C:\DBDIR\MYDATA.MDB:
	`  USE C:\DBDIR\MYDATA.MDB`
	USE does the same thing as setting DataDirectory to a Microsoft Access database file in your ODBC.INI file.

See Also
 For Advanced Users

Outer Joins (Advanced)

The ODBC Microsoft Access 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

Limitations to ODBC SQL Grammar (Advanced)

The ODBC Microsoft Access driver impose the following limitations on the ODBC SQL grammar:

Grammar	Limitation
AND predicates	A maximum of 300 supported.
Comparison predicate	For SQL_BIT data, comparisons can be made for equality and inequality only (= and <> operators).
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	The NOT NULL constraint in the CREATE TABLE statement is not supported except for bit columns.
SELECT DISTINCT	The DISTINCT keyword does not apply to binary data.
Set Functions	The 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.

See Also
For Advanced Users
Additional Supported ODBC SQL Grammar (Advanced)
Driver-specific ODBC SQL Grammar Supported (Advanced)

Data Types (Advanced)

The following table shows the Microsoft Access data types, data types used to create tables, and ODBC SQL data types.

Microsoft Access data type	Data type (create table)	ODBC SQL data type
Currency	Currency	SQL_NUMERIC
Date/Time	DateTime	SQL_TIMESTAMP
Memo	Memo	SQL_LONGVARCHAR
Number (FieldSize=Single)	SingleFloat	SQL_REAL
Number (FieldSize=Double)	DoubleFloat	SQL_DOUBLE
Number (FieldSize=Byte)	UnsignedByte	SQL_TINYINT
Number (FieldSize=Integer)	Short	SQL_SMALLINT
Number (FieldSize=Long Integer)	Long	SQL_INTEGER
OLE Object	LongBinary	SQL_LONGVARBINARY
Text	Text	SQL_CHAR
Yes/No	Bit	SQL_BIT

Note   **SQLGetTypeInfo** returns ODBC SQL data types.   All conversions in Appendix D of the *Microsoft ODBC Programmer's Reference* are supported for the SQL data types listed above.

When converting Microsoft Access 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 Access 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 Access driver uses unsigned single-byte integers.

See Also

For Advanced Users

Error Messages (Advanced)

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

Native Error

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

SQLSTATE

For errors that occur in the data source, the ODBC Microsoft Access 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 Access driver or Driver Manager generates the appropriate SQLSTATE.

Error Message

For errors that occur in the data source, the ODBC Microsoft Access driver returns an error message returned to it by the ODBC File Library. For errors that occur in the ODBC Microsoft Access 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 Access 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 Access ISAM (RED110.DLL)

SQLGetInfo Return Values (Programming)

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*.

fInfoType 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_CHAR | SQL_CVT_DOUBLE | SQL_CVT_INTEGER | SQL_CVT_LONGVARCHAR | SQL_CVT_NUMERIC | SQL_CVT_REAL | SQL_CVT_SMALLINT | SQL_CVT_TINYINT
SQL_CONVERT_CHAR	SQL_CVT_BIT | SQL_CVT_CHAR | SQL_CVT_DOUBLE | SQL_CVT_INTEGER | SQL_CVT_LONGVARCHAR | SQL_CVT_REAL | SQL_CVT_SMALLINT | SQL_CVT_TIMESTAMP | SQL_CVT_TINYINT
SQL_CONVERT_DATE	SQL_CVT_CHAR | SQL_CVT_TIMESTAMP
SQL_CONVERT_DECIMAL	0
SQL_CONVERT_DOUBLE	SQL_CVT_BIT | SQL_CVT_CHAR | SQL_CVT_DOUBLE | SQL_CVT_INTEGER | SQL_CVT_LONGVARCHAR | SQL_CVT_NUMERIC | SQL_CVT_REAL | SQL_CVT_SMALLINT | SQL_CVT_TINYINT
SQL_CONVERT_FLOAT	SQL_CVT_BIT | SQL_CVT_CHAR | SQL_CVT_DOUBLE | SQL_CVT_INTEGER | SQL_CVT_LONGVARCHAR | SQL_CVT_NUMERIC | SQL_CVT_REAL | SQL_CVT_SMALLINT | SQL_CVT_TINYINT

SQL_CONVERT_FUNCTIONS	SQL_FN_CVT_CONVERT
SQL_CONVERT_INTEGER	SQL_CVT_BIT | SQL_CVT_CHAR | SQL_CVT_DOUBLE | SQL_CVT_INTEGER | SQL_CVT_LONGVARCHAR | SQL_CVT_NUMERIC | SQL_CVT_REAL | SQL_CVT_SMALLINT | SQL_CVT_TINYINT
SQL_CONVERT_LONGVARBINARY	0
SQL_CONVERT_LONGVARCHAR	SQL_CVT_CHAR | SQL_CVT_LONGVARCHAR
SQL_CONVERT_NUMERIC	SQL_CVT_BIT | SQL_CVT_CHAR | SQL_CVT_DOUBLE | SQL_CVT_INTEGER | SQL_CVT_LONGVARCHAR | SQL_CVT_NUMERIC | SQL_CVT_REAL | SQL_CVT_SMALLINT | SQL_CVT_TINYINT
SQL_CONVERT_REAL	SQL_CVT_BIT | SQL_CVT_CHAR | SQL_CVT_DOUBLE | SQL_CVT_INTEGER | SQL_CVT_LONGVARCHAR | SQL_CVT_NUMERIC | SQL_CVT_REAL | SQL_CVT_SMALLINT | SQL_CVT_TINYINT
SQL_CONVERT_SMALLINT	SQL_CVT_BIT | SQL_CVT_CHAR | SQL_CVT_DOUBLE | SQL_CVT_INTEGER | SQL_CVT_LONGVARCHAR | SQL_CVT_NUMERIC | SQL_CVT_REAL | SQL_CVT_SMALLINT | SQL_CVT_TINYINT
SQL_CONVERT_TIME	SQL_CVT_CHAR | SQL_CVT_TIMESTAMP
SQL_CONVERT_TIMESTAMP	SQL_CVT_CHAR | SQL_CVT_TIMESTAMP
SQL_CONVERT_TINYINT	SQL_CVT_BIT | SQL_CVT_CHAR | SQL_CVT_DOUBLE | SQL_CVT_INTEGER | SQL_CVT_LONGVARCHAR | SQL_CVT_NUMERIC | SQL_CVT_REAL | SQL_CVT_SMALLINT | SQL_CVT_TINYINT
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	"ACCESS"
SQL_DBMS_VER	"1.1"
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	64
SQL_MULT_RESULT_SETS	"N"
SQL_MULTIPLE_ACTIVE_TXN	"N"
SQL_NON_NULLABLE_COLUMNS	1
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	"ACCESS"
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)

  For Programmers
    Scalar Functions (Programming)

Scalar Functions (Programming)

The ODBC Microsoft Access 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)

The ODBC Microsoft Access 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 Access driver.

Function	Description
SQLDriverConnect	The following keywords are supported in the <u>connection string</u>: **DSN**, **UID**, **PWD**, **DBQ**, and **FIL**.
SQLGetConnectOption **SQLSetConnectOption**	These functions support the SQL_ACCESS_MODE, SQL_CURRENT_QUALIFIER, SQL_OPT_TRACE, and SQL_OPT_TRACEFILE connection options. **SQLGetConnectOption** also supports the SQL_AUTOCOMMIT option.
SQLGetCursorName **SQLSetCursorName**	Since the driver does not support a positioned update or delete (for example, WHERE CURRENT OF *cursorname*), these functions are supported, but cannot be used for positioned updates.
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 Microsoft Access driver returns 3, since each Microsoft Access file is a complete database.
SQLGetStmtOption **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.
SQLGetTypeInfo	Only data type names returned by **SQLGetTypeInfo** can be used with CREATE statements.
SQLMoreResults	This function always returns SQL_NO_DATA_FOUND.
SQLTables	The Microsoft Access driver does not return information on tables (for example, SQL Server tables) attached to a Microsoft Access database.
SQLTransact	This function supports COMMIT, but not ROLLBACK.

See Also
For Advanced Users
Error Messages (Advanced)
For Programmers
Implementation Issues (Programming)

SQL_LOCK_TABLES Statement Option (Programming)

The ODBC Microsoft Access 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 Microsoft Access 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 information might affect the use of the ODBC Microsoft Access driver.

Accessing Attached Tables

The Microsoft Access driver does not recognize a table attached to an existing Microsoft Access database. **SQLTables** does not return the table name, and if you use SELECT on an attached table (for example, a SQL Server table), the driver returns an error message.

Arithmetic Errors

The Microsoft Access 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 Microsoft Access driver stops inserting or updating data when the first arithmetic error is encountered.

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.