ODBC Desktop Database Drivers--What's New

This Help file describes additions and changes made to the ODBC Desktop Database Drivers for version 2.0.   These drivers include the Microsoft Access driver, the Btrieve driver, the dBASE driver, the Microsoft Excel driver, the Microsoft FoxPro driver, the Paradox driver, and the Text driver.   When a change applies to one or more drivers, but not all drivers, the name of each applicable driver is shown in bold.   When text has been deleted from the Help file, but for clarity is retained in this changes file, the text has been placed in brackets preceded by DELETED:.

Added Features

The following specific features have been added to the ODBC Desktop Database Drivers for version 2.0.

Architecture Changes/
Addition of Native 32-Bit Drivers
Overview
Driver ISAMs
Hardware and Software Requirements

New Data Sources
Access 2.0
Excel 5
FoxPro 2.6
Paradox 4.X
Driver ISAMs

Scrollable Cursors
ODBC API Functions Affecting Cursor Support
SQLGetInfo Return Values Affecting Cursor Support
Positioned UPDATE and DELETE Operations

Heterogeneous Joins/Attached Table Support (Microsoft Access Driver)
Supported.

Updatable Joins
Supported.   (For more information on updatable joins, see the documentation for your database management system.)

Transaction Support (Microsoft Access Driver)
API Functions Affecting Transaction Support

Procedure Invocation (Microsoft Access Driver)
CALL Statement

General Changes

The following topics show general changes made to the ODBC Desktop Database Drivers for version 2.0.

Setup Changes
Setup Options
ODBC Setup Dialog Box
Data Source Conversion
Connecting to a Data Source
Repairing a Database (Microsoft Access)
Compacting a Database (Microsoft Access)
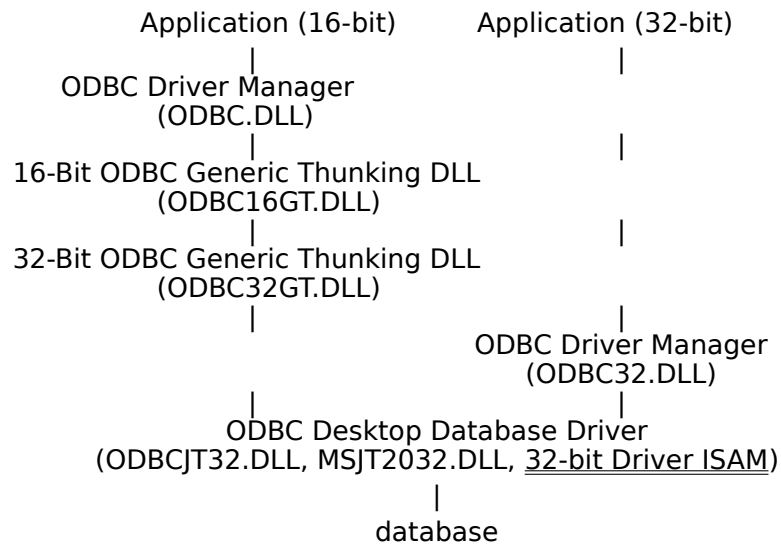
Overview

The ODBC Desktop Database Drivers allow you to open and query a database through the Open Database Connectivity (ODBC) interface.

The application/driver architecture for Win16/Win32s applications is:

```
        Application (16-bit)      Application (32-bit)
                |                          |
                        32-Bit Thunking Driver Manager
                                (ODBC32.DLL)
                |                          |
                        ODBC Universal Thunking DLL
                                (ODBC16UT.DLL)
                |                          |
                        ODBC Driver Manager
                                (ODBC.DLL)
                                   |
                        ODBC Desktop Database Driver
        (ODBCJT16.DLL, MSAJT200.DLL, 16-bit Driver ISAM))
                                   |
                                database
```
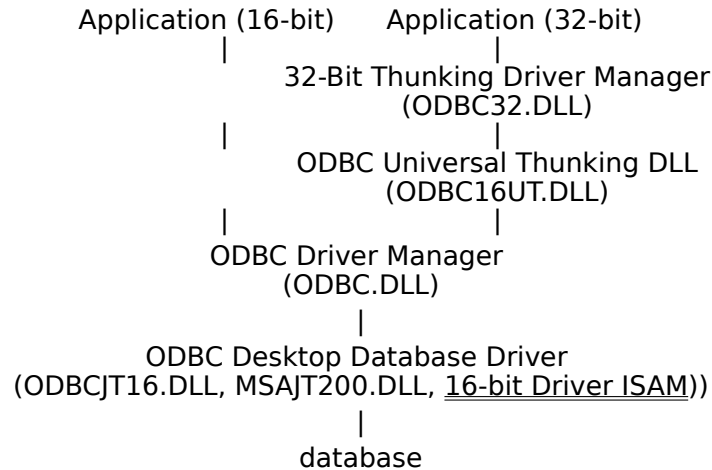
The application/driver architecture for Win32 applications is:

```
        Application (16-bit)      Application (32-bit)
                |                          |
        ODBC Driver Manager
                (ODBC.DLL)
                |                          |
        16-Bit ODBC Generic Thunking DLL
                (ODBC16GT.DLL)
                |                          |
        32-Bit ODBC Generic Thunking DLL
                (ODBC32GT.DLL)
                |                          |
                                ODBC Driver Manager
                                        (ODBC32.DLL)
                |                          |
                        ODBC Desktop Database Driver
        (ODBCJT32.DLL, MSJT2032.DLL, 32-bit Driver ISAM)
                                   |
                                database
```

Driver ISAMs

The ODBC Desktop Database Driver ISAMs are as follows:

Driver	16-bit ISAM	32-bit ISAM	Applicability
Microsoft Access	None (Included in MSAJT200.DLL)	None (Included in MSJT2032.DLL)	Microsoft Access versions 1.0, 1.1, 2.0
Btrieve	BTRV200.DLL	not available	Btrieve versions 5.1 and 6.x
dBase	XBS200.DLL	MSXB2032.DLL	dBase versions III and IV
Microsoft Excel	MSXL2016.DLL	MSXL2032.DLL	Microsoft Excel versions 3.0, 4.0, 5.0
			(For versions earlier than 3.0, see note below)
Microsoft FoxPro	XBS200.DLL	MSXB2032.DLL	Microsoft FoxPro versions 2.0, 2.5, 2.6
Paradox	PDX200.DLL	MSPX2032.DLL	Paradox versions 3.5, 4.0, 4.5
Text	MSTX2016.DLL	MSTX2032.DLL	ASCII Text

Note   The Microsoft Excel driver is not able to read Microsoft Excel files whose format is earlier than version 3.0.   These files can be opened in a later release of Microsoft Excel and saved in version 3, 4, or 5 format that the driver is able to read.

Hardware and Software Requirements
Hardware Requirements

To use the ODBC Desktop Database Drivers, you must have:

- An IBM-compatible personal computer with an 80386 or higher processor and a VGA or higher-resolution graphics card.
- A hard disk with 6MB of free disk space for 16-bit or 32-bit drivers.
- At least 4MB of random-access memory (RAM).

Software Requirements

To access data with the ODBC driver, you must have:

- The ODBC driver.
- The ODBC Driver Manager 2.0 (16-bit ODBC.DLL or 32-bit ODBC32.DLL).
- A computer running MS-DOS 3.3 or later.   (MS-DOS 5.0 or later is recommended.)
- Microsoft Windows 3.1 or later to install and run 16-bit drivers, or Microsoft Windows NT 3.5 or later to install and run 32-bit drivers.   (Note that 32-bit applications can be run on Win32s, and that 16-bit applications can be run on Windows on Windows (WOW) on Windows NT.)
- The stack size for an application using a Microsoft ODBC driver should be at least 20K.

When using Microsoft Windows NT 3.5 or later, the 32-bit driver is thread-safe, but only through the use of a global semaphore that controls access to the driver.   Concurrent use of the driver will be very limited under Windows NT.

When running multiple 16-bit applications on Windows on Windows, the applications must be run in separate memory spaces. (The same memory space cannot be used because ODBC does not support multiple environments in the same process.) To run an application in a separate memory space, highlight the application's icon in the Microsoft Windows NT Program Manager, choose the File-Properties command, then choose Run in Separate Memory Space.

Data Source Conversion

If a version 2.0 driver replaces an installed version 1.0 driver, existing data sources associated with the version 1.0 driver will need to be converted before they can be used with the new driver.   (Data sources to be converted can be either those associated with ODBC version 1.0 drivers, or those associated with the ODBC version 2.0 driver ODBCJT16.DLL, which was used with Microsoft Access 2.0 data sources.)   This process converts all data source names (DSNs) associated with the version 1.0 driver to version 2.0 DSNs.

For Windows 3.1 or later, the keywords in the ODBC.INI files are changed.   The ODBCINST.INI and ODBCISAM.INI files are also changed.   For the **Microsoft Access 2.0** driver, the name of the setup DLL in ODBCINST.INI is also changed.

If you choose to install only 32-bit components, the data sources will be converted to 32-bit data sources; otherwise, the data sources will be converted to 16-bit data sources.

Converting to Version 2.0 DSNs

Data source conversion is performed when you run driver setup.   For each new driver installed for which a version 1.0 driver was previously installed, you will be asked whether you want to convert the existing data sources associated with that driver.   If so, choose Yes, and the conversion will be performed automatically.

When the **Text** driver is used, all converted Text DSNs will be given the same default format as the last converted DSN.

Reverting to Version 1.0 DSNs

On Windows 3.1 or later, data sources can be converted back to version 1.0.   During the original conversion process, ODBC.INI, ODBCINST.INI, and ODBCISAM.INI are not deleted, but are renamed with .BAK extensions.   If there are any versions of these files with BAK extensions, they are renamed with BA1 extensions; BA1 files are renamed BA2; and so on. To convert back to version 1.0, rename ODBC.BAK, ODBCINST.BAK, and ODBCISAM.BAK with INI extensions; and rename BA1, BA2, etc., back one level.

Adding, Modifying, and Deleting a Data Source

Note   When the dBASE or Microsoft FoxPro driver is used, changes in the data source (DSN) setup for either driver will affect the DSN setup of the other driver. Changes to the dBASE DSN setup will change the Microsoft FoxPro DSN setup, and vice versa.

To add, modify, or delete a data source dynamically

You can call the SQLConfigDataSource API function to add, modify, or delete a data source dynamically.   This function uses keywords to set connect options that are also set through the Setup dialog box.   This function should be used when you want to add, modify, or delete a data source without displaying the Setup dialog box.

ODBC Setup Dialog Box

The ODBC Setup dialog box contains the following changed fields:

Control	Description
Collating Sequence	(This control is displayed only for the **Btrieve, dBASE, Microsoft FoxPro,** or **Paradox** driver.)
	The sequence in which the fields are sorted.
	When the **Btrieve, dBASE,** or **Microsoft FoxPro** driver is used, the sequence can be: ASCII or International.
	When the **Paradox** driver is used, the sequence can be: ASCII, International, Swedish-Finnish, or Norwegian-Danish.
Database	(This control is displayed only for the **Microsoft Access** driver.)
	The full path of the Microsoft Access database you want to access. Use the Select button to select an existing database. Use the Create button to create and select a new Microsoft Access database.
	A Microsoft Access data source can be set up without selecting or creating a database. If no database is provided upon setup, the user will be prompted to choose a database file when connecting to the data source.
	Use the Repair button to repair a damaged database. Use the Compact button to compact a database.
Define Format	(This control is displayed only for the **Text** driver.)
	Displays the Define Text Format dialog box and enables you to specify the schema for individual tables in the data source directory.
Directory	Displays the currently selected directory. (This does **not** apply to the **Microsoft Access** driver.)
	For **Microsoft Excel** 3.0/4.0 files, the path display is labeled "Directory", while for Microsoft Excel 5.0 files, the path display is labeled "Workbook".
Read Only	Designates the database as read-only.
Rows to Scan	(This control is displayed only for the **Microsoft Excel** or **Text** driver.)
	The number of rows to scan to determine the data type of each column. The data type is determined given the maximum number of kinds of data found. If data is encountered that does not match the data type guessed for the column, the data type will be returned as a NULL value.
	For the **Microsoft Excel** driver, you may enter a number from 1 to 16 for the rows to scan; however, the value will always default to 1. (A number outside the limit will return an error.)
	For the **Text** driver, you may enter a number from 1 to 32767 for the number of rows to scan; however, the value will always default to 1. (A number outside the limit will return an error.)
Select Directory	Displays a dialog box where you can select a directory

	containing the files you want to access.
	For Microsoft Excel 3.0/4.0 files, the path selection button is labeled "Select Directory", while for Microsoft Excel 5.0 files, the path selection button is labeled "Select Workbook".
Sort Order	(This control is displayed only for the **Microsoft Access** driver.)
	Sets a default sort order for the database. Can be:
	General (English,French,German,Portuguese,Italian), Spanish, Dutch, Swedish/Finnish, Norwegian/Danish, Icelandic, Czech, Hungarian, Polish, Russian, Turkish, Arabic, Hebrew, Greek.
System Database	(This control is displayed only for the **Microsoft Access** driver.)
	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 if no system database will be used.
Version	Enables you to select the version of the files used.
Workbook	For Microsoft Excel 5.0 files, displays the currently selected workbook. Before you add the data source, you must use the Select Workbook button to select a directory.
	Worksheets within workbooks are treated as SYSTEM TABLES. Defined or named ranges within a worksheet are treated as TABLES.

System Database

Select the System Database option button to select the system database to be used.   The ODBC Microsoft Access driver prompts the user for a name and password.   The default name is Admin and the default password in Microsoft Access for the Admin user is an empty string.

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.

Select the None option button to log into the Microsoft Access database as the Admin user; no system database will be used.

For more information about system databases and the Admin user, see the Microsoft Access documentation.

Select Directory

When defining a data source directory (for all drivers except **Microsoft Access**), specify the directory where your most commonly used files are located.   The ODBC driver uses this directory as the default directory.   Copy other files into this directory if they are used frequently.   Alternatively, you can qualify filenames in a SELECT statement with the directory name:

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

Or, you can specify a new default directory by using the **SQLSetConnectOption** function with the SQL_CURRENT_QUALIFIER option.

Paths for Additional Directories   (Btrieve)

If you want to open Btrieve data files in directories other than the default, list these directories by the XTRPATH variable; for example, XTRPATH = path [; path [; path]].   This variable can be set in either of the following ways:

- Use the SET command in the AUTOEXEC.BAT file.   For example, SET XTRPATH = C:\BTRIEVE\DATA; F:\DATA.
- Use the Btrieve environment (.ENV) file.   For other applications to recognize the Btrieve environment file, the file's path and filename must be specified by setting the XTRENV variable in the AUTOEXEC.BAT file (using the SET command).

The path for additional Btrieve directories can also be set in the application's initialization file.   With the Microsoft Access application, for instance, the XTRPATH variable can be set in the [Btrieve ISAM] section of the ODBCDDP.INI file.   If specifying the path in the Btrieve environment (.ENV) file, the path and filename of the environment file can be set in the MSACC20.INI file, rather than in AUTOEXEC.BAT.

Directory/Workbook Path Displays   (Microsoft Excel)

For Microsoft Excel 3.0/4.0 files, the path display is labeled "Directory", and the path selection button is labeled "Select Directory".   For Microsoft Excel 5.0 files, the path display is labeled "Workbook", and the path selection button is labeled "Select Workbook".   When defining a data source directory, specify the directory where your most commonly used Microsoft Excel files are located for Microsoft Excel 3/4, or the directory where the workbook file is located for Microsoft Excel 5.   The Use Current Directory control is deleted for Microsoft Excel 5.0.

Define Text Format

When the **Text** driver is used, the Define Text Format dialog box enables you to define the format for columns in a selected file.   This dialog box enables you to specify the schema for each data table. This information is written to a SCHEMA.INI file in the data source directory.   A separate SCHEMA.INI is created for each text data source directory.

Note   The same default file format applies to all new text data tables. All files created by the CREATE TABLE statement inherit those same default format values, which are set by selecting file format values in the Define Text Format dialog box with <default> chosen in the Tables list box.

Section	Information
Characters	ANSI or OEM. OEM specifies a non-ANSI character set.
Column Name Header	Indicates whether the columns of the first row of the selected table are to be used as column names. Either TRUE or FALSE.
Columns	Contains a list of the column names for each column in the selected table. The order of the columns reflects the order of the columns in the table.
Data Type	Can be Char, Date, Float, or Integer. Date data types can be in the following formats: "dd-mmm-yy", "mm-dd-yy", "mmm-dd-yy", "yyyy-mm-dd", or "yyyy-mmm-dd". "mm" denotes numbers for months; "mmm" denotes letters for months.
Delimiter	Specifies the custom delimiter character to be used to separate columns. Enabled when the "Custom Delimiter" format is selected. Double quotations marks (") cannot be used as the delimiter character. (Note that the delimiter cannot be specified in hexadecimal or decimal format.)
Format	Either delimited or fixed length. If delimited, indicates the type of delimiter used: comma (CSV), tab, or special character (custom).
	If Format is fixed-length and Column Name Header is TRUE, the first line must be comma-delimited.
Guess	Automatically generates the column's data type, name, and width values for the columns in the selected table by scanning the table's contents according to the Format list box selection. Enabled when the table format is delimited. Any previously defined columns in the Columns list are cleared and replaced with new entries. If Column Name Header is not checked, column names are generated

	automatically as "F1", "F2", and so on. Note that no default value is shown in the Data Type box.
Name	Displays the name of the selected column. May be used to specify a new column name for either an existing column or a new column.
	If Column Name Header is TRUE, the column name displayed is ignored.
Rows to Scan	The number of rows that Setup or the driver will scan when setting the columns and column data types based upon existing data.
	You may enter a number from 1 to 32767 for the number of rows to scan; however, the value will always default to 1. (A number outside the limit will return an error.)
Tables	Contains a list of all files in the directory selected in the Text Setup dialog box that match the list of extensions specified.
	Selecting <default> sets the table attributes for all files in the specified directory that are not explicitly defined. When <default> is selected, the "Columns" group is disabled.
Width	The width of the column may be changed for Char columns.
	For other data types, the width is displayed but cannot be changed.

schema

The schema includes information about each table (text file) in a data source, including the table's format, the number of rows to scan to determine column types, whether the first row of the table contains column names, whether the source file is written using an OEM or ANSI codepage, and each column's name, data type, and width.

SCHEMA.INI File

When the **Text** driver is used, the SCHEMA.INI file in the data source directory contains format information for the source text files.   This information is used to specify text file format at data source setup time.   Each text data directory has a separate SCHEMA.INI file. File format information entered in the Define Text Format dialog box is written to the SCHEMA.INI file, which is automatically created for this purpose.

Note   The same default file format applies to all new text data tables. All files created by the CREATE TABLE statement inherit those same default format values, which are set by selecting file format values in the Define Text Format dialog box with <default> chosen in the Tables list box.

File formats in SCHEMA.INI have the following sections:

Section	Information
ColNameHeader	Indicates whether the first record of data will specify the column names. Either TRUE or FALSE.
Format	Can be FIXEDLENGTH, TABDELIMITED, CSVDELIMITED (by a comma), or DELIMITED() (by the special character specified in the parentheses). The special character is one character in length and can be in character, decimal, or hexadecimal format.
	If Format is FIXEDLENGTH and ColNameHeader is TRUE, the first line must be comma-delimited.
MaxScanRows	The number of rows to be scanned when setting a column's data type based upon existing data.
	You may enter a number from 1 to 32767 for the number of rows to scan; however, the value will always default to 1. (A number outside the limit will return an error.)
Character Set	OEM or ANSI.
DateTimeFormat	Indicates date and time format. Entered as a string. Should be specified if all datetime fields in the SCHEMA.INI file will be handled with the same format. If this entry is absent, the short date picture and time options in the control panel will be used.
CurrencySymbol	Indicates the currency symbol to be used for currency values in the text file. If this entry is absent, the value in the control panel will be used.
CurrencyFormat	Indicates the format for the currency symbol and value. Can be "0" (currency symbol prefix with no separation, such as $1), "1" (currency symbol suffix with no separation, such as 1$), "2" (currency symbol prefix with one-character separation, such as $ 1), or "3" (currency symbol suffix with one-character separation, such as 1 $).

CurrencyDigits	Indicates the number of digits used for the fractional part of a currency amount.
CurrencyNegative Format	Indicates placement of the minus sign in currency format. Can be $1, -$1, $-1, $1-, 1$, -1$, 1-$, 1$-, -1 $, -$ 1, 1 $-, $ 1-, $ -1, 1- $, $ 1, or 1 $.
DecimalSymbol	Can be any single character used to separate the integer from the fractional part of a number.
NumberDigits	Indicates the number of decimal digits in the fractional portion of a number.
NumberLeading Zeros	Indicates whether a decimal value less than 1 and greater than -1 should contain leading zeros. Can be either "0" (no leading zeros) or "1".
Col1, Col2,...	Column names, types, and width.
	If ColNameHeader is TRUE, the column names listed here are ignored.

Connecting to a Data Source

Password Entry

Passwords must be entered at connection time, either by password in the connection string (UID and PWD for **Microsoft Access** or PWD for **Paradox**), or with the *szAuthStr* argument in the **SQLConnect** function.

Password-Protected Files

When the Paradox driver is used, after a password-protected file has been opened by a user, other users are not allowed to open the same file.

Heterogeneous Joins/Attached Tables

When the Microsoft Access driver is used, attached tables and heterogeneous joins are supported.

Opening Microsoft Excel Tables

The Microsoft Excel driver is not able to read Microsoft Excel files whose format is earlier than version 3.0.   These files can be opened in a later release of Microsoft Excel and saved in version 3, 4, or 5 format that the driver is able to read.

Microsoft Excel 3 (BIFF3) and Microsoft Excel 4 (BIFF4) file formats must have a database range defined.   The Microsoft Excel driver recognizes named ranges as tables.   Databases specified by a named range will increase as new data is added.   If any new data would overwrite existing data in a worksheet (outside the named range that represents the data), an error will be returned.

All Microsoft Excel tables (spreadsheets) that are created and opened for inserting are opened exclusive, and can be opened by only one user at a time.   Otherwise, Microsoft Excel tables are opened shared.   "Read Only" allows multiple users to view .XLS files.

Referential Integrity

Referential integrity is supported only for a **Microsoft Access** data source, and only if the referential integrity is set up for the data source by the **Microsoft Access** database management system.

Select Database (Advanced)

When the **Microsoft Access** driver is used, the Select button may be pressed in the Microsoft Access 2.0 Setup dialog box to select an existing database.

The Open Database dialog box contains the following fields:

Control	Description
File Name	Name of the existing database file with an .MDB extension.
Exclusive	If the Exclusive box is selected, the 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 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.
Read Only	Designates the database as read-only.

Create Database (Advanced)

When the **Microsoft Access** driver is used, the Create button may be pressed in the Microsoft Access 2.0 Setup dialog box to create a new database.

The New Database dialog box contains the following fields:

Control	Description
File Name	Name of the database file with an .MDB extension.
Exclusive	If the Exclusive box is selected, the 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 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.
Sort Order	Sets a default sort order for the database. Can be: General (English,French,German,Portuguese,Italian), Spanish, Dutch, Swedish/Finnish, Norwegian/Danish, Icelandic, Czech, Hungarian, Polish, Russian, Turkish, Arabic, Hebrew, Greek.

Database Repair (Advanced)

When the **Microsoft Access** driver is used, if a client failure occurs during the commit process (after **SQLTransact** has been issued, but before the function returns), the affected database can be corrupted and need to be repaired.   Repair can be performed either upon setup through the Microsoft Access driver setup dialog box, or programmatically by using the **SQLConfigDataSource** function.

To repair the database through the Microsoft Access driver setup dialog box, click the Repair button.   Select the file and path, and then click OK.

To repair the database programmatically, include "REPAIR_DB=<path name>" in the list of attributes (*lpszAttributes*) of the **SQLConfigDataSource** function.   In this function, <path name> is the full path to the Microsoft Access database.

When repairing a password-protected file, a password and user id must be entered in the advanced options dialog box (or in the list of attributes for **SQLConfigDataSource**, if repairing programmatically).   The password and user id should be removed after the repair.

These operations will run the standard Microsoft Access repair process on the database. Data in Microsoft Access databases could be affected by either of these actions.

Note that a database indicated as corrupted may in fact not be corrupted.   The indication may be prompted by an improper shutdown that does not clear the appropriate bits.   In this and other cases, a programmer may want to keep the corruption message box transparent to the end user.   This can be accomplished by passing SQL_DRIVER_NOPROMPT with the SQLDriverConnect function.

For more information about database repair in Microsoft Access, refer to the Microsoft Access help file.

Database Compaction (Advanced)

When the **Microsoft Access** driver is used, data compaction can be performed on a database file either upon setup through the Microsoft Access driver setup dialog box, or programmatically by using the **SQLConfigDataSource** function.

To perform data compaction on a database file through the Microsoft Access driver setup dialog box, click the Compact button.   From the Database to Compact From dialog box, select the file and path, and then click OK.   From the Database to Compact Into dialog box, select the file and path, and a sort order, and then click OK.   The name of the file to compact into can be the same as the file to be compacted.

To perform data compaction on a database file programmatically, include "COMPACT_DB=<source db> <dest db> <sort order>" in the list of attributes (*lpszAttributes*) of the **SQLConfigDataSource** function.   In this function, <source db> is the full path to the Microsoft Access database to be compacted and <dest db> is the full path for the compacted Microsoft Access database.

When compacting a password-protected file, a password and user id must be entered in the advanced options dialog box (or in the list of attributes for **SQLConfigDataSource**, if compacting programmatically).   The password and user id should be removed after the compaction.

These operations will run the standard Microsoft Access compaction process on the database.

For more information about database compaction in Microsoft Access, refer to the Microsoft Access help file.

SQL Statements (Advanced)

For information about changes to SQL grammar, see the following topics:

For Advanced Users

- Additional Supported ODBC SQL Grammar
- Driver-Specific SQL Grammar
- Limitations to SQL Grammar

Positioned UPDATE and DELETE Operations

When the **Microsoft Access, Btrieve, dBASE, Microsoft FoxPro,** or **Paradox** driver is used, positioned UPDATE and DELETE operations are supported through the **SQLSetPos** function.   The WHERE CURRENT OF syntax is not supported.

Procedure Invocation (Microsoft Access)

When the **Microsoft Access** driver is used, procedures can be invoked from the driver by using the **SQLExecDirect** or **SQLPrepare** function with the following syntax: {CALL *procedure-name* [(*parameter*[,*parameter*]...)]}. Note that expressions and constants are not supported as parameters to a called procedure.

Additional Supported ODBC SQL Grammar (Advanced)

The ODBC Desktop Database Drivers completely support the following additional SQL statements and clauses in the Core and Extended ODBC grammar, as changed for the Desktop Database Drivers version 2.0:

Core and Extended grammar	Comments
ALTER TABLE statement (Core)	When the **Microsoft Access, Btrieve, dBASE, Microsoft FoxPro,** or **Paradox** driver is used, the ALTER TABLE statement is supported with add and drop table constraint definition clauses. (ALTER TABLE statements are not supported for **Microsoft Excel** and **Text** drivers.)
Binary Literals (Extended)	Supported.
EXISTS predicate (Core)	Supported.
Procedure Invocation (Extended)	When the **Microsoft Access** driver is used, procedures can be invoked from the driver by using the **SQLExecDirect** or **SQLPrepare** function with the following syntax: {CALL *procedure-name* [(*parameter*[,*parameter*]...)]}. Note that expressions are not supported as parameters to a called procedure.
	If a procedure name includes a dash, the name must be delimited with backquotes (`).
SELECT statement (Extended) (including GROUP BY, HAVING, and UNION clauses)	Supported.
Subqueries (SUBSELECTs) (Core)	Supported.

Driver-Specific SQL Grammar Supported (Advanced)

The Microsoft ODBC drivers support the following driver-specific SQL grammar changes, as changed for the Desktop Database Drivers version 2.0:

Driver-specific SQL grammar	Comments
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.
	When the **Paradox** driver is used, the COUNT function includes NULL as well as non-NULL values in the count.
CREATE INDEX	The syntax of the CREATE INDEX statement is: CREATE [UNIQUE] INDEX *index-name* ON *table-name* (*column-identifier* [ASC][DESC][, *column-identifier* [ASC][DESC]...]) WITH <*index option list*> where <*index option list*> can be: PRIMARY | DISALLOW NULL | IGNORE NULL
	Only the **Microsoft Access** driver uses the DISALLOW NULL and IGNORE NULL index options. The **Btrieve, dBASE, Microsoft FoxPro,** and **Paradox** drivers accept the syntax, but ignore the presence of either option.
	When the **Paradox** driver is used, the <u>CREATE INDEX</u> statement creates Paradox primary key files and secondary files.
Date literals	The MM-DD-YYYY format is supported. (This has been changed from the former format of YYYY-MM-DD.)
DROP INDEX	When the **Microsoft Access, Btrieve, dBASE, Microsoft FoxPro,** or **Paradox** driver is used, the syntax of the DROP INDEX statement is "DROP INDEX a on b" where "a" is the name of the index and "b" is the name of the table (not DROP INDEX *index-name*).
	When the **Paradox** driver is used, the DROP INDEX statement deletes Paradox secondary index files and does not use the ODBC SQL grammar.
	Not supported for the **Microsoft Excel** or **Text** driver.
Fixed-Width Text File	When the **Text** driver is used, the last column of a fixed-width text file can be variable length.

Limitations to SQL Grammar (Advanced)

The ODBC Desktop Database Drivers impose the following limitations on the SQL grammar, as changed for the Desktop Database Drivers version 2.0:

Grammar	Limitation
Aggregate functions	An aggregate function and a non-aggregate column reference cannot both be used as arguments to a single SQL statement.
ALTER TABLE statement	When the **Btrieve, dBASE, Microsoft FoxPro,** or **Paradox** driver is used, once an index has been created and a new record added, the structure of the table cannot be changed by the ALTER TABLE statement unless the index is dropped and the contents of the table are deleted.
AND predicates	A maximum of 40 supported.
CALL statement	Expressions are not supported as parameters to a called procedure. (**Microsoft Access** only)
Column names	Column names can contain any valid characters (for example, spaces). If column names contain any characters except letters, numbers, and underscores, the name must be delimited by enclosing it in back quotes (`).
	When the **Microsoft Access** or **Microsoft Excel** driver is used, column names are limited to 64 characters, and longer names generate an error. When the **Paradox** driver is used, the maximum column name is 25 characters. When the **Text** driver is used, the maximum column name is 64 characters, and longer names are truncated.
	When the **Microsoft Excel** driver is used, the '$' character is used in place of the '!' character, which is not legal.
	When the **Text** driver is used, the driver provides a default name if a column name is not specified. For example, the driver calls the first column F1, the second column F2, and so on.
CONVERT function	Type conversion failures result in the affected column being set to NULL.
	Neither the DATE nor TIMESTAMP data type can be converted to another data type (or itself) by the CONVERT function.
CREATE INDEX statement	Not supported for the **Microsoft Excel** or **Text** driver.
	The syntax of the CREATE INDEX statement is: CREATE [UNIQUE] INDEX *index-name* ON *table-name* (*column-identifier* [ASC][DESC][, *column-identifier* [ASC][DESC]...]) WITH <*index*

	option list> where *<index option list>* can be: PRIMARY | DISALLOW NULL | IGNORE NULL
	Only the **Microsoft Access** driver uses the DISALLOW NULL and IGNORE NULL index options. The **Btrieve, dBASE, Microsoft FoxPro,** and **Paradox** drivers accept the syntax, but ignore the presence of either option.
	An index can be defined on a maximum of 10 columns. If more than 10 columns are included in a CREATE INDEX statement, the index will not be recognized and the table will be treated as though no index were created.
	The **dBASE** driver cannot create an index on a LOGICAL column.
	When the **dBASE** or **Microsoft FoxPro** driver is used, the index created by a CREATE UNIQUE INDEX statement is actually non-unique, and duplicate values can be inserted into the indexed column.
	When the **Paradox** driver is used, a unique index must be defined.
CREATE TABLE statement	When the **Microsoft Access, Btrieve, Microsoft Excel**, or **Paradox** driver is used, and the length of a text or binary column is not specified (or is specified as 0), the column length will be set to 255.
	When the **dBASE** or **Microsoft FoxPro** driver is used, and the length of a text or binary column is not specified (or is specified as 0), the column length will be set to 254.
	A maximum of 255 columns is supported.
	When the **Microsoft Excel5** driver is used, a worksheet cannot be created with the same name as a worksheet that was previously dropped. The DROP TABLE statement clears the worksheet, but does not delete the worksheet name.
Date Arithmetic	Date arithmetic is not supported for subtracting a DATE data type from a DATE data type.
DELETE statement	Not supported for the **Text** drivers. Note that the INSERT statement is supported for the Text driver.
DISTINCT keyword	Not supported for Long Text fields (**Microsoft Access**) or Memo fields (**dBASE** or **FoxPro**).
DROP TABLE statement	When the **Microsoft Excel5** driver is used, the DROP TABLE statement clears

	the worksheet, but does not delete the worksheet name. The worksheet name still exists in the workbook, so another worksheet cannot be created with the same name.
FROM clause	The maximum number of tables in a FROM clause is 16.
HAVING clause	The maximum number of search conditions in a HAVING clause is 40.
Identifiers	If an identifier contains a space or a special symbol, the identifier must be enclosed in back quotes. A valid name is a string of no more than 64 characters, of which the first character must not be a space. Valid names cannot include control characters or the following special characters: ` | # * ? [ ] . ! $.
	Do not use reserved words listed in the SQL grammar in Appendix C of the *Microsoft ODBC Programmer's Reference* as identifiers (that is, table or column names), unless you surround the word in back quotes (`).
Index Names	When the **Btrieve** driver is used, an index cannot have the same name as any of the column names in the table upon which the index is defined. Only 24 index "segments" are allowed per table.
	When the **Paradox** driver is used, a primary index must have the same name as the table upon which it is defined. Other unique or non-unique indexes must have the same name as the column upon which they are defined.
INSERT statement	Inserted data is truncated on the right without warning if it is too long to fit into the column.
	If a column reference is left out of an INSERT statement, a parameter size error is returned instead of a syntax error.
	Attempting to insert a value that is out of the range of a column's data type causes a NULL to be inserted into the column.
	A table is not updatable by the **Paradox** driver if a unique index is not defined on the table. (Note that this is not true for an empty table, which can be updated even if a unique index is not defined on the table.)
	When the **Text** driver is used, the length of any row must be less than or equal to 65,543 bytes.

LIKE predicate	If data in a column is longer than 255 characters, the LIKE comparison will be based only on the first 255 characters.
	A LIKE used in a procedure is supported only with constant patterns.
	Use of an escape clause in a LIKE predicate is not supported.
Parameterized Queries	A parameterized query can be called using the following syntax: CALL *query-name* [(*parameter*[,*parameter*]...)]. A parameterized query cannot be used in a SELECT statement: SELECT * FROM *query-name* WHERE *column* = ?.
Scalar Functions	Scalar functions are supported only by using the ODBC canonical format.
SELECT Statement	An aggregate-function column cannot be mixed with a non-aggregate column in a SELECT statement.
	The use of an asterisk (to select all columns) in a SELECT statement containing a GROUP BY clause is not supported. The names of the columns to be selected must be specified.
	The use of a vertical bar in a SELECT statement is not supported.
	When using a column alias in a SELECT statement, the word "as" must precede the alias. For example, "SELECT col1 as a from b". Without the "as", the statement will return an error.
	If an incorrect column name is entered into a SELECT statement, a SQLSTATE 07001 error, "Wrong Number of Parameters", is returned instead of a SQLSTATE S0022 error, "Column Not Found".
Strings	The maximum length of an SQL statement string is 65,000 characters.
	When the **Microsoft Access** driver is used, only ANSI string constants (with single quotation marks, not double quotation marks) are supported.
Table Names	Table names can contain any valid characters (for example, spaces). If table names contain any characters except letters, numbers, and underscores, the name must be delimited by enclosing it in back quotes (`).
	When the **Microsoft Excel** driver is used, if a table name is not qualified by a database reference, the default database is implied.
	The **Microsoft Excel** table name that references <filename> is supported for Microsoft Excel 3.0 and 4.0 files. The

	Microsoft Excel table name that references <workbook-name> is supported for Microsoft Excel 5.0 files.
	The "$" character is used in place of the "!" character, which is not legal.
	Special DOS keywords CON, AUX, LPT1, and LPT2 should not be used as table names.
Table references	A maximum of 16 table references can be included in any query statement. For 16-bit drivers, however, table references may be limited to a number less than 16, especially for highly complex queries, because of the compilation segment limit.
	For 32-bit drivers, the limit of 16 table references is always valid.
Views	Not supported by the **Btrieve, dBASE, Microsoft Excel, Microsoft FoxPro, Paradox,** or **Text** driver.
WHERE clause	The maximum number of clauses in a WHERE clause is 40.
	LONGVARBINARY and LONGVARCHAR columns can be searched for using literals up to 255 characters in length, but not using parameters.
WHERE CURRENT OF clause	Not supported.

CREATE INDEX (Advanced)

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

- The primary index must be created before any rows are added to the table.
- A primary index must be defined upon the first "n" columns in a table.
- Only one primary index is allowed per table.
- A table cannot be updated by the Paradox driver if a primary index is not defined on the table.   (Note that this is not true for an empty table, which can be updated even if a unique index is not defined on the table.)
- The *index-name* argument for a primary 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*.X*nn* and *table-name*.Y*nn*, 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 primary index must exist for that table.

Note that when two CREATE INDEX statements are performed, the first statement will always create a primary index with the same name as the table and the second statement will always create a non-unique index with the same name as the column. These indexes will be named this way even if different names are entered in the CREATE INDEX statements and even if the index is labeled UNIQUE in the second CREATE INDEX statement.

Data Types (Advanced)

Select the driver data types you want to display:

- **Microsoft Access**
- **Btrieve**
- **dBASE**
- **Microsoft Excel**
- **Microsoft FoxPro**
- **Paradox**
- **Text**

Microsoft Access Data Types (Advanced)

The following table shows added, changed, or deleted data types and ODBC SQL data types for the ODBC Microsoft Access Driver.   Added, changed, or deleted data types are shown in bold.

Microsoft Access data type	Data type (CREATETABLE)	ODBC SQL data type
Binary	**Binary**	**SQL_BINARY**
Bit	Bit	SQL_BIT
Char	**Char**	**SQL_VARCHAR**
Counter	**Counter**	**SQL_INTEGER**
Currency	Currency	SQL_NUMERIC
Date/Time	DateTime	SQL_TIMESTAMP
Long Binary	**Long Binary**	**LONGVARBINARY**
Long Text	**Long Text**	**SQL_LONGVARCHAR**
Memo	**Long Text** **[DELETE:** Memo]	SQL_LONGVARCHAR
Number (FieldSize=Single)	**Single** **[DELETE**: Float**]**	SQL_REAL
Number (FieldSize=Double)	**Double** **[DELETE**: Float]	SQL_DOUBLE
Number (FieldSize=Byte)	Unsigned Byte	SQL_TINYINT
Number (FieldSize=Integer)	Short	SQL_SMALLINT
Number (FieldSize=Long Integer)	Long	SQL_INTEGER
OLE	LongBinary	SQL_LONGVARBINARY
Text	**VarChar** **[DELETE:**Text]	SQL_**VAR**CHAR
VarBinary **[DELETE:** Yes/No]	**VarBinary** **[DELETE:** Bit]	**SQL_VARBINARY** **[DELETE:** SQL_BIT]

The following table shows limitations on Microsoft Access data types.

Data type	Limitation
BINARY or VARCHAR	Creating a BINARY or VARCHAR column of zero or unspecified length actually returns a 255-byte column.
BIT	**Bit columns will display as 0 or 1 when a SELECT is performed, but a query making a direct comparison of a BIT data type to the number "1" will not succeed. If a direct comparison must be made, the comparison should be made to TRUE or <>0.**
CHAR, LONGVARCHAR, VARCHAR	**A zero-length string should not be inserted into a Microsoft Access CHAR, LONGVARCHAR, or VARCHAR column.**

More limitations on data types can be found in Data Type Limitations.

Btrieve Data Types (Advanced)

The following table shows added, changed, or deleted data types and ODBC SQL data types for the ODBC Btrieve Driver.   Added, changed, or deleted data types are shown in bold.

Btrieve data type	ODBC data type
DATE	SQL_DATE
FLOAT4	SQL_REAL
FLOAT8	SQL_DOUBLE
INTEGER1	SQL_TINYINT
INTEGER2	SQL_SMALLINT
INTEGER4	SQL_INTEGER
LOGICAL1	SQL_BIT
MONEY	SQL_**NUMERIC**
NOTE	SQL_LONGVARCHAR
STRING	SQL_**VAR**CHAR

The following table shows limitations on Btrieve data types.

Data type	Limitation
LOGICAL1	**LOGICAL1 columns will display as 0 or 1 when a SELECT is performed, but a query making a direct comparison of a LOGICAL1 data type to the number "1" will not succeed. If a direct comparison must be made, the comparison should be made to TRUE or <>0.**
STRING columns	Creating a STRING column of zero or unspecified length actually returns a 255-byte column.

More limitations on data types can be found in <u>Data Type Limitations</u>.

dBASE Data Types (Advanced)

The following table shows added, changed, or deleted data types and ODBC SQL data types for the ODBC dBASE Driver. Added, changed, or deleted data types are shown in bold.

dBASE data type	ODBC data type
Char [DELETE: Character]	SQL_**VAR**CHAR
Date	SQL_DATE
Logical	SQL_BIT
Memo	SQL_LONGVARCHAR
Numeric (BCD)	SQL_DOUBLE

The following table shows limitations on dBASE data types.

Data type	Limitation
CHAR columns	Creating a CHAR column of zero or unspecified length actually returns a 254-byte column.
Encrypted data	The dBASE driver does not support encrypted dBASE tables.
LOGICAL	**LOGICAL columns will display as 0 or 1 when a SELECT is performed, but a query making a direct comparison of a LOGICAL data type to the number "1" will not succeed. If a direct comparison must be made, the comparison should be made to TRUE or <>0.** **The dBASE driver cannot create an index on a LOGICAL column.**

More limitations on data types can be found in Data Type Limitations.

Microsoft Excel Data Types (Advanced)

The following table shows added, changed, or deleted data types and ODBC SQL data types for the ODBC Microsoft Excel Driver.   Added, changed, or deleted data types are shown in bold.

Microsoft Excel data type	ODBC SQL data type
DateTime	SQL_TIMESTAMP
Integer	**SQL_INTEGER**
Logical	SQL_BIT
Number	SQL_DOUBLE
Text	SQL_**VAR**CHAR

The following table shows limitations on Microsoft Excel data types.

Data type	Limitation
Encrypted data	The Microsoft Excel driver cannot read encrypted data.
Error Strings	**The Microsoft Excel driver cannot return a character string for the Microsoft Excel error values (#N/A!, #VALUE!, #REF!, #DIV/0!, #NUM!, #NAME?, and #NULL!), but returns a NULL instead.**
LOGICAL	**LOGICAL columns will display as 0 or 1 when a SELECT is performed, but a query making a direct comparison of a LOGICAL data type to the number "1" will not succeed. If a direct comparison must be made, the comparison should be made to TRUE or <>0.**
	When a LOGICAL value is returned in a column other than a LOGICAL column, the Microsoft Excel driver cannot convert that LOGICAL value to a TRUE or FALSE character string, but converts the value to NULL instead.
TEXT columns	Creating a TEXT column of zero or unspecified length actually returns a 255-byte column.

More limitations on data types can be found in Data Type Limitations.

Microsoft FoxPro Data Types (Advanced)

The following table shows added, changed, or deleted data types and ODBC SQL data types for the ODBC Microsoft FoxPro Driver.   Added, changed, or deleted data types are shown in bold.

Microsoft FoxPro data type	ODBC data type
Char **[DELETE:** Character]	SQL_**VAR**CHAR
Date	SQL_DATE
General	SQL_LONGVARBINARY
Logical	SQL_BIT
Memo	SQL_LONGVARCHAR
Numeric (BCD)	SQL_DOUBLE

The following table shows limitations on Microsoft FoxPro data types.

Data type	Limitation
CHAR	Creating a CHAR column of zero or unspecified length actually returns a 254-byte column.
LOGICAL	**LOGICAL columns will display as 0 or 1 when a SELECT is performed, but a query making a direct comparison of a LOGICAL data type to the number "1" will not succeed. If a direct comparison must be made, the comparison should be made to TRUE or <>0.**

More limitations on data types can be found in <u>Data Type Limitations</u>.

Paradox Data Types (Advanced)

The following table shows added, changed, or deleted data types and ODBC SQL data types for the ODBC Paradox Driver.   Added, changed, or deleted data types are shown in bold.

Paradox data type	ODBC SQL data type
Alphanumeric	SQL_**VAR**CHAR
Date	SQL_DATE
Image *	**SQL_LONGVARBINARY**
Number	SQL_DOUBLE
Short	SQL_SMALLINT

* Only valid for Paradox versions 4x.

The following table shows limitations on Paradox data types.

Data type	Limitation
Alphanumeric	Creating an Alphanumeric column of zero or unspecified length actually returns a 255-byte column.

More limitations on data types can be found in <u>Data Type Limitations</u>.

Text Data Types (Advanced)

The following table shows added, changed, or deleted data types and ODBC SQL data types for the ODBC Text Driver.   Added, changed, or deleted data types are shown in bold.

Text data type	ODBC data type
CHAR	SQL_**VAR**CHAR
DATETIME	SQL_**TIMESTAMP**
FLOAT	SQL_DOUBLE
INTEGER	SQL_INTEGER
LONGCHAR	SQL_LONGVARCHAR

The following table shows limitations on Text data types.

Data type	Limitation
CHAR	Creating a CHAR column of zero or unspecified length actually returns a 255-bit column.
INTEGER	Valid values for INTEGER columns are 32767 to -32766.

More limitations on data types can be found in Data Type Limitations.

Limitations to Data Types (Advanced)

The following limitations new to version 2.0 apply to data types.

Data type	Limitation
All data types	Type conversion failures result in the affected column being set to NULL.
BIT	TRUE is represented as -1, not 1, so comparisons may not operate properly.
	A NULL is inserted as a 0.
	BIT columns will display as 0 or 1 when a SELECT is performed, but a query making a direct comparison of a BIT data type to the number "1" will not succeed. If a direct comparison must be made, the comparison should be made to TRUE or <>0.
	A value inserted into a column of type BIT (whether character or numeric) that isn't a 0 or a '0' is treated as TRUE.
	Creating a zero-length BINARY column actually returns a 255-bit BINARY column.
DATE	The DATE data type cannot be converted to another data type (or itself) by the CONVERT function.
DECIMAL (Exact Numeric)	Not supported.
Floating-Point Data Types	The number of decimal places in a floating-point number may be limited by the number format set in the International section of the Windows Control Panel.
TIMESTAMP	The TIMESTAMP data type cannot be converted to itself by the CONVERT function.
TINYINT	TINYINT values are always unsigned.

Error Messages (Advanced)

The following table shows the error messages returned by the Driver Manager and the Driver ISAM, as changed for the Desktop Database Drivers version 2.0:

Error message	Error location
[Microsoft][ODBC DLL]*message-text*	Driver Manager (16-bit ODBC.DLL or

[Microsoft][ODBC *driver-name*][ODBC File Library]*message-text*

32-bit ODBC32.DLL)
Driver ISAM
(see <u>Driver ISAMs Table</u>)

SQLGetInfo Return Values Affecting Cursor Support

The following table lists **SQLGetInfo** return values that affect cursor support and have changed for version 2.0.   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 returned by **SQLGetInfo**, see the *Microsoft ODBC SDK Programmer's Reference, Version 2.0*.

fInfoType value (#define)	Returned value
SQL_CURSOR_COMMIT_BEHAVIOR	SQL_CB_CLOSE [DELETE: 2]
SQL_CURSOR_ROLLBACK_BEHAVIOR	SQL_CB_CLOSE [DELETE: 0]
SQL_MAX_CURSOR_NAME_LEN	64 [DELETE: 18]
SQL_POS_OPERATIONS	SQL_POS_POSITION	SQL_POS_REFRESH	SQL_POS_UPDATE	SQL_POS_DELETE	SQL_POS_ADD
SQL_POSITIONED_STATEMENTS	0

SQLGetInfo Return Values (Programming)

The following table lists C-language #defines for the *fInfoType* argument and the corresponding values returned by **SQLGetInfo**. Added, changed, or deleted #defines and return values are shown in bold. 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 2.0*.

fInfoType value (#define)	Returned value
SQL_ACCESSIBLE_PROCEDURES	**"Y" (Microsoft Access)** **"N" (Btrieve, dBASE, Microsoft Excel, Microsoft FoxPro, Paradox, Text)**
SQL_ACCESSIBLE_TABLES	"Y"
SQL_ACTIVE_CONNECTIONS	**64 [DELETE: 0]**
SQL_ACTIVE_STATEMENTS	0
SQL_ALTER_TABLE	**SQL_AT_ADD_COLUMN	SQL_AT_DROP_COLUMN (Microsoft Access, Btrieve, dBASE, Microsoft FoxPro, Paradox)** **0 (Microsoft Excel, Text)**
SQL_BOOKMARK_PERSISTANCE	**SQL_BP_SCROLL	SQL_BP_UPDATE**
SQL_COLUMN_ALIAS	**"Y"**
SQL_CONCAT_NULL_BEHAVIOR	**SQL_CB_NON_NULL**
SQL_CONVERT_BIGINT	0
SQL_CONVERT_BINARY	**SQL_CVT_DOUBLE	SQL_CVT_FLOAT |**

	SQL_CVT_INTEGER | **SQL_CVT_NUMERIC |** **SQL_CVT_REAL |** **SQL_CVT_SMALLINT |** **SQL_CVT_VARCHAR** **[DELETE:** 0**]**
SQL_CONVERT_BIT	0
SQL_CONVERT_CHAR	**SQL_CVT_DOUBLE |** **SQL_CVT_FLOAT |** **SQL_CVT_INTEGER |** **SQL_CVT_NUMERIC |** **SQL_CVT_REAL |** **SQL_CVT_SMALLINT |** **SQL_CVT_VARCHAR**
SQL_CONVERT_DATE	**SQL_CVT_DOUBLE |** **SQL_CVT_FLOAT |** **SQL_CVT_INTEGER |** **SQL_CVT_NUMERIC |** **SQL_CVT_REAL |** **SQL_CVT_SMALLINT |** **SQL_CVT_VARCHAR**
SQL_CONVERT_DECIMAL	0
SQL_CONVERT_DOUBLE	**SQL_CVT_DOUBLE |** **SQL_CVT_FLOAT |** **SQL_CVT_INTEGER |** **SQL_CVT_NUMERIC |** **SQL_CVT_REAL |** **SQL_CVT_SMALLINT |** **SQL_CVT_VARCHAR**
SQL_CONVERT_FLOAT	**SQL_CVT_DOUBLE |** **SQL_CVT_FLOAT |** **SQL_CVT_INTEGER |** **SQL_CVT_NUMERIC |** **SQL_CVT_REAL |** **SQL_CVT_SMALLINT |** **SQL_CVT_VARCHAR**
SQL_CONVERT_FUNCTIONS	**SQL_FN_CVT_CONVERT**
SQL_CONVERT_INTEGER	**SQL_CVT_DOUBLE |** **SQL_CVT_FLOAT |** **SQL_CVT_INTEGER |** **SQL_CVT_NUMERIC |** **SQL_CVT_REAL |** **SQL_CVT_SMALLINT |** **SQL_CVT_VARCHAR**
SQL_CONVERT_LONGVARBINARY	**SQL_CVT_DOUBLE |** **SQL_CVT_FLOAT |** **SQL_CVT_INTEGER |** **SQL_CVT_NUMERIC |** **SQL_CVT_REAL |** **SQL_CVT_SMALLINT |** **SQL_CVT_VARCHAR** **[DELETE:** 0**]**
SQL_CONVERT_LONGVARCHAR	**SQL_CVT_DOUBLE |** **SQL_CVT_FLOAT |** **SQL_CVT_INTEGER |** **SQL_CVT_NUMERIC |** **SQL_CVT_REAL |**

	SQL_CVT_SMALLINT | SQL_CVT_VARCHAR
SQL_CONVERT_NUMERIC	**SQL_CVT_DOUBLE | SQL_CVT_FLOAT | SQL_CVT_INTEGER | SQL_CVT_NUMERIC | SQL_CVT_REAL | SQL_CVT_SMALLINT | SQL_CVT_VARCHAR**
SQL_CONVERT_REAL	**SQL_CVT_DOUBLE | SQL_CVT_FLOAT | SQL_CVT_INTEGER | SQL_CVT_NUMERIC | SQL_CVT_REAL | SQL_CVT_SMALLINT | SQL_CVT_VARCHAR**
SQL_CONVERT_SMALLINT	**SQL_CVT_DOUBLE | SQL_CVT_FLOAT | SQL_CVT_INTEGER | SQL_CVT_NUMERIC | SQL_CVT_REAL | SQL_CVT_SMALLINT | SQL_CVT_VARCHAR**
SQL_CONVERT_TIME	**SQL_CVT_DOUBLE | SQL_CVT_FLOAT | SQL_CVT_INTEGER | SQL_CVT_NUMERIC | SQL_CVT_REAL | SQL_CVT_SMALLINT | SQL_CVT_VARCHAR**
SQL_CONVERT_TIMESTAMP	**SQL_CVT_DOUBLE | SQL_CVT_FLOAT | SQL_CVT_INTEGER | SQL_CVT_NUMERIC | SQL_CVT_REAL | SQL_CVT_SMALLINT | SQL_CVT_VARCHAR**
SQL_CONVERT_TINYINT	**SQL_CVT_DOUBLE | SQL_CVT_FLOAT | SQL_CVT_INTEGER | SQL_CVT_NUMERIC | SQL_CVT_REAL | SQL_CVT_SMALLINT | SQL_CVT_VARCHAR**
SQL_CONVERT_VARBINARY	**SQL_CVT_DOUBLE | SQL_CVT_FLOAT | SQL_CVT_INTEGER | SQL_CVT_NUMERIC | SQL_CVT_REAL | SQL_CVT_SMALLINT | SQL_CVT_VARCHAR** **[DELETE:** 0**]**
SQL_CONVERT_VARCHAR	**SQL_CVT_DOUBLE | SQL_CVT_FLOAT | SQL_CVT_INTEGER | SQL_CVT_NUMERIC | SQL_CVT_REAL |**

	SQL_CVT_SMALLINT | SQL_CVT_VARCHAR [DELETE: 0]
SQL_CORRELATION_NAME	**SQL_CN_DIFFERENT**
SQL_CURSOR_COMMIT_BEHAVIOR	**SQL_CB_CLOSE** [DELETE: 2]
SQL_CURSOR_ROLLBACK_BEHAVIOR	**SQL_CB_CLOSE** [DELETE: 0]
SQL_DATA_SOURCE_NAME	**The DSN from ODBC.INI, or "" if DRIVER keyword is used in ODBC.INI**
SQL_DATA_SOURCE_READ_ONLY	**"N"** **(Microsoft Access, Btrieve, dBASE, Microsoft FoxPro, Paradox)** **"Y"** **(Microsoft Excel, Text)** **[DELETE: (The driver does not check to see whether the disk drive is read-only.)]**
SQL_DATABASE_NAME	**Filename**
SQL_DBMS_NAME	**"ACCESS" [Microsoft Access]** **"BTRIEVE" [Btrieve]** **"DBASE" [dBASE]** **"EXCEL" [Microsoft Excel]** **"FOXPRO" [Microsoft FoxPro]** **"PARADOX" [Paradox]** **"TEXT" [Text]**
SQL_DBMS_VER	**"1.0"**, "1.1**", or "2.0"** **[Microsoft Access]** **"5.1" or "6.0" [Btrieve]** **"3.0" or "4.0" [dBASE]** **"3.0", "4.0", or "5.0"** **[Microsoft Excel]** **"2.0", "2.5", or "2.6"** **[Microsoft FoxPro]** **"3.x" or "4.x" [Paradox]** **"1.0" [Text]**
SQL_DEFAULT_TXN_ISOLATION	**SQL_TXN_READ_COMMITTED [Microsoft Access]**
SQL_DRIVER_NAME	**"ODBCJT16.DLL" for 16-bit, or "ODBCJT32.DLL" for 32-bit** **[DELETE: SIMBA.DLL]**
SQL_DRIVER_ODBC_VER	**"2.01"**
SQL_DRIVER_VER	" **2.0**1.*nnnn*" (*nnnn* specifies the build date.) **[DELETE: 1.01]**
SQL_EXPRESSIONS_IN_ORDERBY	"Y"
SQL_FETCH_DIRECTION	SQL_FD_FETCH_NEXT | **SQL_FD_FETCH_FIRST |** **SQL_FD_FETCH_LAST |**

	SQL_FD_FETCH_PRIOR | SQL_FD_FETCH_ABSOLUTE | SQL_FD_FETCH_RELATIVE | SQL_FD_FETCH_BOOKMARK
SQL_FILE_USAGE	**SQL_FILE_QUALIFIER (Microsoft Access, Btrieve, Microsoft Excel5.0)**
	SQL_FILE_TABLE (dBASE, Microsoft Excel3.0/4.0, Microsoft FoxPro, Paradox, Text)
SQL_GETDATA_EXTENSIONS	**SQL_GD_ANY_ORDER | SQL_GD_ANY_COLUMN | SQL_GD_BLOCK | SQL_GD_BOUND**
SQL_GROUP_BY	**SQL_GB_GROUP_BY_ CONTAINS_SELECT**
SQL_IDENTIFIER_CASE	**SQL_IC_MIXED**
SQL_IDENTIFIER_QUOTE_CHAR	"`"" (**backquote**)
	[DELETE: double quotation mark].
SQL_KEYWORDS	**ALPHANUMERIC, AUTOINCREMENT, BINARY, BOOLEAN, BYTE, CURRENCY, DATABASE, DATABASENAME, DATETIME, DISTINCTROW, DOUBLEFLOAT, FLOAT4, FLOAT8, GENERAL, IEEEDOUBLE, IEEESINGLE, INT, INTEGER1, INTEGER2, INTEGER4, LOGICAL, LOGICAL1, LONG, LONGBINARY, LONGCHAR, LONGTEXT, MEMO, MONEY, NOTE, NUMBER, OLEOBJECT, OWNERACCESS, PARAMETERS, PERCENT, PIVOT, REAL, SHORT, SINGLE, SINGLEFLOAT, STDEV, STDEVP, STRING, TABLEID, TEXT, TOP, TRANSFORM, UNSIGNEDBYTE, VAR, VARBINARY, VARP, YESNO**
SQL_LIKE_ESCAPE_CLAUSE	**"Y"**
SQL_LOCK_TYPES	**SQL_LOCK_NO_CHANGE**
SQL_MAX_BINARY_LITERAL_LEN	**255**
SQL_MAX_CHAR_LITERAL_LEN	**255 (Microsoft Access)** **255 (Btrieve)** **254 (dBASE)** **255 (Microsoft Excel)** **254 (Microsoft FoxPro)** **255 (Paradox)** **255 (Text)**

SQL_MAX_COLUMN_NAME_LEN	**64 [Microsoft Access]** **20 [Btrieve]** **10 [dBASE]** **64 [Microsoft Excel 5.0]** **30 [Microsoft Excel 3.0/4.0]** **10 [Microsoft FoxPro]** **25 [Paradox]** **64 [Text]**
SQL_MAX_COLUMNS_IN_GROUP_B Y	**10**
SQL_MAX_COLUMNS_IN_INDEX	**32 (Microsoft Access)** **0 (Limit Unknown or Not Applicable) (Btrieve, DBase, Microsoft FoxPro, Paradox)** **0 (Microsoft Excel, Text)**
SQL_MAX_COLUMNS_IN_ORDER_B Y	**10**
SQL_MAX_COLUMNS_IN_SELECT	**255**
SQL_MAX_COLUMNS_IN_TABLE	**255** **(Microsoft Access, dBASE, Microsoft Excel, Microsoft FoxPro, Paradox, Text)** **254** **(Btrieve)**
SQL_MAX_CURSOR_NAME_LEN	**64 [DELETE: 18]**
SQL_MAX_INDEX_SIZE	**255 [Microsoft Access]** **4096 [Btrieve]** **220 [dBASE]** **0 [Microsoft Excel]** **512 [Microsoft FoxPro]** **1350 [Paradox]** **0 [Text]**
SQL_MAX_OWNER_NAME_LEN	**0**
SQL_MAX_PROCEDURE_NAME_LEN	**64 [Microsoft Access]** **0 [Btrieve, dBASE, Microsoft Excel, Microsoft FoxPro, Paradox, Text]**
SQL_MAX_QUALIFIER_NAME_LEN	66
SQL_MAX_ROW_SIZE	**2096 [Microsoft Access]** **4096 [Btrieve]** **4000 [dBASE]** **65535 [Microsoft Excel]** **65500 [Microsoft FoxPro 2.5 and above]** **4000 [Microsoft FoxPro 2.0]** **1350 [Paradox]** **65535 [Text]**
SQL_MAX_ROW_SIZE_INCLUDES_ LONG	**"N"** **[Microsoft Access, dBase, Microsoft FoxPro, Paradox]** **"Y"** **[Btrieve, Microsoft Excel,**

	Text]
SQL_MAX_STATEMENT_LEN	**65000**
SQL_MAX_TABLE_NAME_LEN	**64 (Microsoft Access)** **20 (Btrieve)** **12 (dBASE)** **31 (Microsoft Excel 5.0)** **12 (Microsoft Excel** **3.0/4.0)** **12 (Microsoft FoxPro)** **12 (Paradox)** **12 (Text)**
SQL_MAX_TABLES_IN_SELECT	**16**
SQL_MAX_USER_NAME_LEN	**0**
SQL_MULT_RESULT_SETS	**"N"**
SQL_MULTIPLE_ACTIVE_TXN	**"Y" [DELETE: N]**
SQL_NEED_LONG_DATA_LEN	**"N"**
SQL_NON_NULLABLE_COLUMNS	**SQL_NNC_NULL** **[DELETE: 1]**
SQL_NULL_COLLATION	**SQL_NC_LOW**
SQL_NUMERIC_FUNCTIONS	**SQL_FN_NUM_ABS |** **SQL_FN_NUM_ATAN |** **SQL_FN_NUM_CEILING |** **SQL_FN_NUM_COS |** **SQL_FN_NUM_EXP |** **SQL_FN_NUM_FLOOR |** **SQL_FN_NUM_LOG |** SQL_FN_NUM_MOD | **SQL_FN_NUM_POWER |** **SQL_FN_NUM_RAND |** **SQL_FN_NUM_SIGN |** **SQL_FN_NUM_SIN |** **SQL_FN_NUM_SQRT |** **SQL_FN_NUM_TAN**
SQL_ODBC_API_CONFORMANCE	**SQL_OAC_LEVEL1** **[DELETE: 1]**
SQL_ODBC_SAG_CLI_CONFORMANCE	**SQL_OSCC_NOT_COMPLIAN** **T** **[DELETE: 1]**
SQL_ODBC_SQL_CONFORMANCE	**SQL_OSC_MINIMUM** **[DELETE: 0]**
SQL_ODBC_SQL_OPT_IEF	**"N"**
SQL_ODBC_VER	**From Driver Manager**
SQL_ORDER_BY_COLUMNS_IN_ **SELECT**	**"N"**
SQL_OUTER_JOINS	**"Y"**
SQL_OWNER_TERM	**""**
SQL_OWNER_USAGE	**0**
SQL_POS_OPERATIONS	**SQL_POS_POSITION |** **SQL_POS_REFRESH |** **SQL_POS_UPDATE |** **SQL_POS_DELETE |** **SQL_POS_ADD** **(Microsoft Access, Btrieve,** **dBASE, Microsoft FoxPro,**

	Paradox)
	SQL_POS_POSITION | SQL_POS_REFRESH (Microsoft Excel, Text)
SQL_POSITIONED_STATEMENTS	**0**
SQL_PROCEDURE_TERM	**"QUERY" (Microsoft Access)**
	"" (Btrieve, dBASE, Microsoft Excel, Microsoft FoxPro, Paradox, Text)
SQL_PROCEDURES	**"Y" (Microsoft Access)**
	"N" (Btrieve, dBASE, Microsoft Excel, Microsoft FoxPro, Paradox, Text)
SQL_QUALIFIER_LOCATION	**SQL_QL_START**
SQL_QUALIFIER_NAME_SEPARATOR	**"." (Microsoft Access, Microsoft Excel 5)**
	"\" (Btrieve, dBASE, Microsoft Excel 4, Microsoft FoxPro, Paradox, Text)
SQL_QUALIFIER_TERM	"**Database**" **[Microsoft Access]** "**Workbook**" **[Microsoft Excel 5.0]** "**Directory**" **[Btrieve, DBase, Microsoft Excel 3.0/4.0, Microsoft FoxPro, Paradox, Text]**
	[DELETE: Directory]
SQL_QUALIFIER_USAGE	**SQL_QU_DML_STATEMENTS | SQL_QU_TABLE_DEFINITION | SQL_QU_INDEX_DEFINITION | SQL_QU_PROCEDURE_ INVOCATION (Microsoft Access)**
	SQL_QU_DML_STATEMENTS | SQL_QU_TABLE_DEFINITION | SQL_QU_INDEX_DEFINITION (Btrieve, dBASE, Microsoft FoxPro, Paradox)
	SQL_QU_DML_STATEMENTS | SQL_QU_TABLE_DEFINITION (Microsoft Excel, Text)
SQL_QUOTED_IDENTIFIER_CASE	**SQL_IC_MIXED**
SQL_ROW_UPDATES	**"N"** **[DELETE: Y]**

SQL_SCROLL_CONCURRENCY	**SQL_SCCO_READ_ONLY | SQL_SCCO_LOCK | SQL_SCCO_OPT_VALUES (Microsoft Access, Btrieve, dBASE, Microsoft FoxPro, Paradox)**
	SQL_SCCO_READ_ONLY **(Microsoft Excel, Text)**
SQL_SCROLL_OPTIONS	SQL_SO_FORWARD_ONLY | **SQL_SO_STATIC | SQL_SO_KEYSET_DRIVEN**
SQL_SEARCH_PATTERN_ESCAPE	"\"
SQL_SERVER_NAME	**"ACCESS" [Microsoft Access] "BTRIEVE" [Btrieve] "DBASE" [dBASE] "EXCEL" [Microsoft Excel] "FOXPRO" [Microsoft FoxPro] "PARADOX" [Paradox] "TEXT" [Text]**
SQL_SPECIAL_CHARACTERS	**"~`@#$%^&*_-+=\}{"';:?/><,.!'[]"**
SQL_STATIC_SENSITIVITY	**SQL_SS_ADDITIONS | SQL_SS_DELETIONS | SQL_SS_UPDATES**
SQL_STRING_FUNCTIONS	**SQL_FN_STR_ASCII | SQL_FN_STR_CHAR |** 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_SPACE |** SQL_FN_STR_SUBSTRING | SQL_FN_STR_UCASE
SQL_SUBQUERIES	**SQL_SQ_COMPARISON | SQL_SQ_EXISTS | SQL_SQ_IN | SQL_SQ_QUANTIFIED | SQL_SQ_CORRELATED_ SUBQUERIES**
SQL_SYSTEM_FUNCTIONS	**0**
SQL_TABLE_TERM	"TABLE"
SQL_TIMEDATE_ADD_INTERVALS	**0**
SQL_TIMEDATE_DIFF_INTERVALS	**0**
SQL_TIMEDATE_FUNCTIONS	SQL_FN_TD_CURDATE | SQL_FN_TD_CURTIME | SQL_FN_TD_DAYOFMONTH | SQL_FN_TD_DAYOFWEEK | **SQL_FN_TD_DAYOFYEAR | SQL_FN_TD_HOUR |**

	SQL_FN_TD_MINUTE | SQL_FN_TD_MONTH | **SQL_FN_TD_NOW |** **SQL_FN_TD_SECOND |** **SQL_FN_TD_WEEK |** SQL_FN_TD_YEAR
SQL_TXN_CAPABLE	**SQL_TC_ALL** **(Microsoft Access)** 0 (Btrieve, dBASE, Microsoft Excel, Microsoft FoxPro, Paradox, Text)
SQL_TXN_ISOLATION_OPTION	**SQL_TXN_READ_COMMITTED**
SQL_UNION	**SQL_U_UNION_ALL |** **SQL_U_UNION**
SQL_USER_NAME	**Not supported**

ODBC API Functions Affecting Cursor Support

The following ODBC API functions affect cursor support.

Function	Description
SQLExtendedFetch	Supported.
SQLSetPos	SQL_LOCK_NO_CHANGE is supported for *fLock*. SQL_LOCK_EXCLUSIVE and SQL_LOCK_UNLOCK are not supported.
SQLSetScrollOptions	Forward and static cursors are supported for SQL_CONCUR_READ_ONLY.
	Only keyset-driven cursors are supported for SQL_CONCUR_LOCK.
	Dynamic cursors and mixed cursors are not supported.
SQLSetStmtOption	The only valid keyset size (SQL_KEYSET_SIZE) is 0, because mixed and dynamic cursors are not supported.
	SQL_ROW_NUMBER returns 0 to indicate that a row number cannot be determined.
	SQL_SIMULATE_CURSOR is not supported.

ODBC API Functions Affecting Transaction Support

The following ODBC API functions affect transaction support.   Transactions are supported for the **Microsoft Access** driver only.

Function	Description
SQLSetConnectOption	SQL_TXN_ISOLATION is always SQL_TXN_READ_COMMITTED.
	When the **Microsoft Access** driver is used, the SQL_AUTOCOMMIT option may be set to either ON or OFF, because the Microsoft Access driver supports transactions.
	The **Btrieve, dBASE, Microsoft Excel, Microsoft FoxPro, Paradox,** and **Text** drivers only support SQL_AUTOCOMMIT being set to ON (the default state), because they do not support manual commits.
SQLTransact	When the **Microsoft Access** driver is used, COMMIT and ROLLBACK are supported.
	When the **Microsoft Access** driver is used, if a failure occurs

during the commit process, the affected database can be repaired using the Repair Database option in the Microsoft Access driver setup, or through the use of the REPAIR_DB keyword in the **SQLConfigDataSource** function. This database repair process could corrupt the data, however.

ODBC API Functions (Programming)

The ODBC Desktop Database Drivers support all Core and Level 1 functions and the following Level 2 functions:

- **SQLData Sources**
- **SQLDrivers**
- **SQLExtendedFetch**
- **SQLMoreResults**
- **SQLNativeSQL**
- **SQLNumParams**
- **SQLProcedures**
- **SQLProcedureColumns**
- **SQLSetPos**
- **SQLSetScrollOptions**

The following table describes how the ODBC Desktop Database Drivers implement specific functions.

Function	Description
SQLColAttributes	LONGVARBINARY and LONGVARCHAR columns are reported as SQL_UNSEARCHABLE.
	For LONGVARBINARY data, SQL_COLUMN_DISPLAY_SIZE is the maximum length of the column, not the maximum length of the column times 2.
	The SQL_COLUMN_MONEY statement option of **SQLColAttributes** returns FALSE for a column of **Paradox** data type CURRENCY, which maps to the ODBC SQL data type SQL_DOUBLE.
SQLConnect	When the **Microsoft Access** or **Paradox** driver is used, the PWD keyword should not include any of the special characters (see SQL_SPECIAL_CHARACTERS in SQLGetInfo Return Values).
SQLDriverConnect	**SQLDriverConnect** enables you to connect to a driver without creating a data source (DSN).
	When the **Microsoft Access** or **Paradox** driver is used, the PWD keyword should not include any of

	the special characters (see SQL_SPECIAL_CHARACTERS in SQLGetInfo Return Values).
	When the **Paradox** driver is used, after a password-protected file has been opened by a user, other users are not allowed to open the same file.
SQLGetTypeInfo	The name of the type (TYPE_NAME) returned in the table produced by **SQLGetTypeInfo** will be the name most commonly used by the data source.
	When the **Text** driver is used, **SQLGetTypeInfo** returns a CASE_SENSITIVE value of FALSE for the text data types (CHAR and LONGCHAR), when the data types actually are case-sensitive.
SQLPrepare	A statement prepared with an incorrect number of columns returns an error at execution time, not upon statement preparation.
SQLProcedureColumns	Supported for **Microsoft Access** DSNs only.
	Pattern matching is not supported for *szProcName*.
	SQL_PARAM_INPUT or SQL_RESULT_COL is returned for COLUMN_TYPE.
SQLProcedures	SQL_PT_PROCEDURE is returned in PROCEDURE_TYPE for all procedures.
SQLSetConnectOption	When the **Microsoft Access** driver is used, SQL_TXN_ISOLATION is always SQL_TXN_READ_COMMITTED.
	SQL_TRANSLATE_DLL and SQL_TRANSLATION_OPTION are not supported.
	SQL_LOGIN_TIMEOUT is not supported.
	When the **Microsoft Access** driver is used, the SQL_AUTOCOMMIT option may be set to either ON or OFF, because the Microsoft Access driver supports transactions.
	The **Btrieve, dBASE, Microsoft Excel, Microsoft FoxPro, Paradox,** and **Text** drivers only support SQL_AUTOCOMMIT being set to ON (the default state), because they do not support

	transactions.
SQLSetPos	SQL_LOCK_NO_CHANGE is supported for *fLock*. SQL_LOCK_EXCLUSIVE and SQL_LOCK_UNLOCK are not supported.
	SQLSetPos supports updatable joins. (For more information on updatable joins, see the documentation for your database management system.)
SQLSetScrollOptions	Forward and static cursors are supported for SQL_CONCUR_READ_ONLY.
	Only keyset-driven cursors are supported for SQL_CONCUR_LOCK.
	Dynamic cursors and mixed cursors are not supported.
SQLSetStmtOption	The only valid keyset size (SQL_KEYSET_SIZE) is 0, because mixed and dynamic cursors are not supported.
	SQL_ROW_NUMBER returns 0 to indicate that a row number cannot be determined.
	SQL_SIMULATE_CURSOR is not supported.
SQLSpecialColumns	A unique index will be returned (if one exists) for the SQL_BEST_ROWID flag in *fColType*. No result set will be returned for the SQL_ROWVER flag. All row IDs have a scope of SQL_SCOPE_SESSION.
SQLStatistics	Filtering is based on uniqueness (*fUnique*). The *fAccuracy* parameter is ignored.
	NON_UNIQUE, SEQ_IN_INDEX, COLUMN_NAME, and COLLATION are index-dependent.
	NULL is always returned in the INDEX_QUALIFIER and PAGES columns.
	Only SQL_TABLE_STAT or SQL_INDEX_OTHER will be returned for TYPE.
	CARDINALITY will be returned for the **Microsoft Access** driver. CARDINALITY will not be returned for the **Btrieve, dBASE, Microsoft Excel, Microsoft FoxPro, Paradox,** or **Text** driver.
SQLTables	When the **Microsoft Access** driver is used, "SYSTEM TABLE" is supported for *szTableType* for

system tables, "SYNONYM" is supported for attached tables, and "VIEW" is supported for row returning queries.

When the **Btrieve, dBASE, Microsoft Excel 3.0/4.0, Microsoft FoxPro, Paradox,** or **Text** driver is used, "TABLE" is the only table type supported.

When the **Microsoft Excel 5.0** driver is used, sheet names (tables with a "$" on the end) are returned as "SYSTEM TABLE", and "TABLE" is returned for all other sheets.

When the **Text** driver is used, the list of files returned by SQLTables is determined by the file extensions in the Extensions List box in the ODBC Text Setup dialog box.

SQLTransact	When the **Microsoft Access** driver is used, COMMIT and ROLLBACK via the **SQLTransact** function are supported.

When the **Btrieve, dBASE, Microsoft Excel, Microsoft FoxPro, Paradox,** or **Text** driver is used, this function supports COMMIT, but not ROLLBACK.

When the **Microsoft Access** driver is used, if a failure occurs during the commit process, the affected database can be repaired using the Repair Database option in the Microsoft Access driver setup, or through the use of the REPAIR_DB keyword in the **SQLConfigDataSource** function.

SQLConfigDataSource Implementation (Programming)

The **SQLConfigDataSource** function that is used to add, modify, or delete a data source dynamically uses the following arguments.

Keyword	Description
CHARACTERSET	For the **Text** driver, OEM or ANSI.
COLNAMEHEADER	For the **Text** driver, indicates whether the first record of data will specify the column names. Either TRUE or FALSE.
COMPACT_DB	For the **Microsoft Access** driver, performs <u>data compaction</u> on a database file.
CREATE_DB	For the **Microsoft Access** driver, creates a database file.
DBQ	For the **Microsoft Access, Btrieve,** or **Paradox** driver, the name of the database file.
	For the **Microsoft Excel 5** driver, the name of the workbook file.
DEFAULTDIR	The path specification to the database file (for the **Microsoft Access** driver) or directory (for the **Btrieve, dBASE, Microsoft Excel, Microsoft FoxPro, Paradox,** or **Text** driver).
DESCRIPTION	A description of the data in the data source.
DRIVER	The path specification to the driver DLL.
DRIVERID	An integer ID for the driver. 25 (Microsoft Access) 20 (Btrieve) 21 (dBASEIII) 277 (dBASEIV) 534 (Microsoft Excel3) 278 (Microsoft Excel4) 22 (Microsoft Excel5) 24 (Microsoft FoxPro2.0) 280 (Microsoft FoxPro2.5) 536 (Microsoft FoxPro2.6) 26 (Paradox3) 282 (Paradox4) 27 (Text)
EXCLUSIVE	Determines whether the database will be opened in exclusive mode (accessed by only one user at a time) or shared mode (accessed by more than one user at a time). Can be true (exclusive mode) or false (shared mode). (This

	does not apply to the **Microsoft Excel** or **Text** driver.)
EXTENSIONS	Lists the filename extensions of the **Text** files on the data source.
FIL	File type (MS Access for Microsoft Access; BTRIEVE; DBASE3 or DBASE4; EXCEL 3.0, 4.0, or 5.0 for Microsoft Excel; FOXPRO 2.0, 2.5, or 2.6 for Microsoft FoxPro; PARADOX 3.x or 4.x). (This does not apply to the **Text** driver.)
FILETYPE	File type for the **Text** driver (Text).
FORMAT	For the **Text** driver, can be FIXEDLENGTH, TABDELIMITED, CSVDELIMITED (by a comma), or DELIMITED() (by the special character specified in the parentheses). The special character is one character in length and can be in character, decimal, or hexadecimal format.
JETINIPATH	Path to the initialization file.
MAXSCANRO WS	For the **Microsoft Excel** or **Text** driver, the number of rows to be scanned when setting a column's data type based upon existing data.
	For the **Microsoft Excel** driver, you may enter a number from 1 to 16 for the rows to scan; however, the value will always default to 1. (A number outside the limit will return an error.)
	For the **Text** driver, you may enter a number from 1 to 32767 for the number of rows to scan; however, the value will always default to 1. (A number outside the limit will return an error.)
PWD	For the **Microsoft Access** or **Paradox** driver, the password.
READONLY	TRUE to make file read-only; FALSE to make file not read-only.
REPAIR_DB	For the **Microsoft Access** driver, repairs a database damaged by a failure that occurs during the commit process.
SYSTEMDB	For the **Microsoft Access**

| | | driver, the path specification to system database file. |
| UID | | For the **Microsoft Access** driver, the user ID name used for login. |

SQLDriverConnect Implementation (Programming)

The following table shows the minimum keywords required to connect to each driver, and provides an example of keyword/value pairs used with **SQLDriverConnect**. For a full list of DRIVERID values, see SQLConfigDataSource Implementation.

Note   If DBQ or DefaultDir is not specified for the **Btrieve, dBASE, Excel 3/4, FoxPro, Paradox,** or **Text** driver, the driver will connect to the current directory.

Driver	Keywords Required	Examples
Microsoft Access	Driver, DBQ	Driver={Microsoft Access Driver (*.mdb)}; DBQ=c:\temp\sample.mdb
Btrieve	Driver	Driver={Microsoft Btrieve Driver (file.dd)}; DBQ=c:\temp
dBASE	Driver, DriverID	Driver={Microsoft dBASE Driver (*.dbf)}; DBQ=c:\temp; DriverID=277
Microsoft Excel3/4	Driver, DriverID	Driver={Microsoft Excel Driver (*.xls)}; DBQ=c:\temp; DriverID=278
Microsoft Excel5	Driver, DriverID, DBQ	Driver={Microsoft Excel Driver (*.xls)}; DBQ=c:\temp\sample.xls;DriverID=22
Microsoft FoxPro	Driver, DriverID	Driver={Microsoft FoxPro Driver (*.dbf)}; DBQ=c:\temp;DriverID=536
Paradox	Driver, DriverID	Driver={Microsoft Paradox Driver (*.db); DBQ=c:\temp;DriverID=26
Text	Driver	Driver={Microsoft Text Driver (*.txt;*.csv)}; DefaultDir=c:\temp

Implementation Issues (Programming)

The following information might affect the use of the Microsoft ODBC drivers.

Issue	Implementation
Read-Only Status	When the **Microsoft Excel** or **Text** driver is used, data source tables are read-only. Even though tables have read-only status, however, applications can perform insertions and updates for Microsoft Excel tables, or insertions for Text tables.
Translation DLLs	Not Supported.

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.