

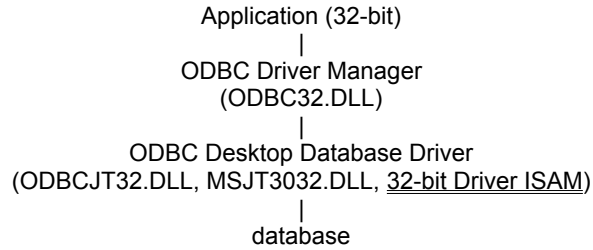
## Overview

### See Also

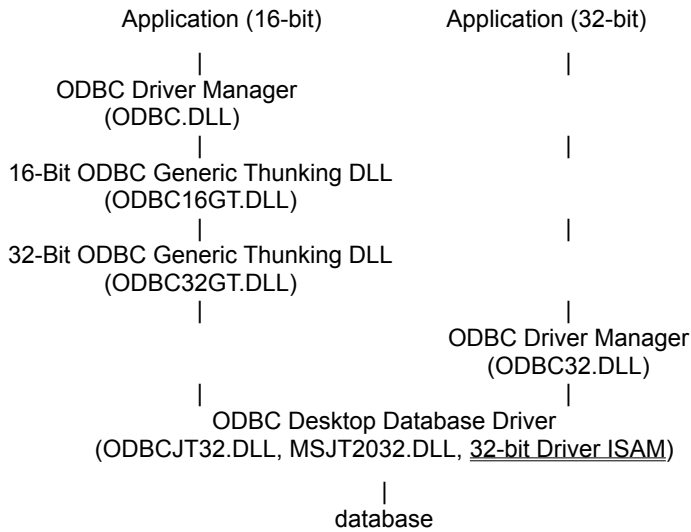
The Microsoft ODBC Desktop Database Drivers allow you to open and query a database through the Open Database Connectivity (ODBC) interface. These drivers are designed for use on Microsoft Windows 95 or later, or Windows NT 3.51 or later. Only 32-bit applications are supported on Windows 95 or later; 16-bit and 32-bit applications are supported on Windows NT 3.51 or later.

The ODBC Desktop Database Drivers include 32-bit drivers for Microsoft Access, dBASE, Microsoft Excel, Microsoft FoxPro, Paradox, and Text. No 16-bit drivers are included.

The application/driver architecture on Windows 95 or later is:



The application/driver architecture on Windows NT 3.51 or later is:



## Driver ISAM Files

### See Also

The ODBC Desktop Database Drivers include ISAM files. These files are as follows:

<b>Driver</b>	<b>32-bit ISAM</b>	<b>Versions</b>
<b>Microsoft Access</b>	None (Included in MSJT3032.DLL)	Microsoft Access version 7.0
<b>Microsoft Access</b>	MSRD2X32.DLL	Microsoft Access versions 1.0, 1.1, 2.0
<b>dBase</b>	MSXB3032.DLL	dBase versions III, IV, and V (dBASE for Windows)
<b>Microsoft Excel</b>	MSXL3032.DLL	Microsoft Excel versions 3.0, 4.0, 5.0 (For versions earlier than 3.0, see note below)
<b>Microsoft FoxPro</b>	MSXB3032.DLL	Microsoft FoxPro versions 2.0, 2.5, 2.6
<b>Paradox</b>	MSPX3032.DLL	Paradox versions 3.5, 4.0, 4.5, 5.x (Paradox for Windows)
<b>Text</b>	MSTX3032.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.

## ODBC Component Files

### See Also

The following table lists the files required by each component of the ODBC Desktop Database Drivers. These files are installed in the WINDOWS\SYSTEM directory for Windows 95 or later, or the WINDOWS\SYSTEM32 directory for Windows NT 3.51 or later. If ODBC files were previously installed in a different directory, make sure that you use the newer files in the WINDOWS\SYSTEM (or SYSTEM32) directory.

Some files are required by more than one component. These files must be redistributed if you intend to redistribute any of the ODBC Desktop Database Drivers with your commercial application.

The following files are common for each of the ODBC Desktop Database Drivers 3.0:

CTL3D32.DLL (Windows NT only)	ODBCINST.CNT
DS16GT.DLL	ODBCINST.HLP
DS32GT.DLL	ODBCINT.DLL
MSJINT32.DLL	ODBCJET.HLP
MSJT3032.DLL	ODBCJET.CNT
MSJTER32.DLL	ODBCJI32.DLL
MSVCRT20.DLL (Windows NT only)	ODBCJT32.DLL
ODBC16GT.DLL	ODBCJTNW.HLP
ODBC32.DLL	ODBCJTNW.CNT
ODBC32GT.DLL	ODBCTL32.DLL
ODBCCP32.CPL	VEN2232.OLB
ODBCCP32.DLL	VBAJET32.DLL
ODBCCR32.DLL	VBAR2232.DLL

The following files are unique to each driver:

<b>Microsoft Access</b>	MSRD2X32.DLL
<b>dBASE</b>	MSXB3032.DLL ODDBSE32.DLL
<b>Microsoft Excel</b>	MSXL3032.DLL ODEXL32.DLL
<b>Microsoft FoxPro</b>	MSXB3032.DLL ODFOX32.DLL
<b>Paradox</b>	MSPX3032.DLL ODPDX32.DLL
<b>Text</b>	MSTX3032.DLL ODTEXT32.DLL

## 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.
- At least 4MB of random-access memory (RAM).

### Software Requirements

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

- The ODBC driver.
- The 32-bit ODBC Driver Manager, version 2.5 or later (ODBC32.DLL). Note that this file is included with the ODBC Desktop Database Drivers.
- Microsoft Windows 95 or later, or Windows NT 3.51 or later.
- The stack size for an application using a Microsoft ODBC driver should be at least 20K.

When using Microsoft Windows NT 3.51 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.

#### **Driver-Specific Hardware and Software Requirements**

- The **Microsoft Access**, **dBASE**, and **Microsoft FoxPro** driver may require changes in the AUTOEXEC.BAT or CONFIG.SYS files.

## **CONFIG.SYS and AUTOEXEC.BAT Changes**

If you are working with the **Microsoft Access** ODBC driver and do not have Microsoft Access installed on your computer, add the line "FILES= 50" to your CONFIG.SYS file and add the line "SHARE.EXE /L:200" to AUTOEXEC.BAT. If you are working with very large databases, add "SHARE.EXE /L:500" to AUTOEXEC.BAT.

Before you use the **dBASE** or **Microsoft FoxPro** driver, you must add the line "SHARE.EXE /L:200" to your AUTOEXEC.BAT file. The SHARE command allows file sharing and locking in a multitasking environment. The /L option specifies the number of files that can be locked at one time.

## **Setting Up Drivers**

Drivers can no longer be added or deleted from the Control Panel or Administrator. The Add and Delete buttons have been removed from the Drivers dialog box that is displayed when the ODBC icon in the Control Panel (or Administrator) is chosen, then the Drivers button in the Data Sources dialog box is chosen. Adding or deleting drivers is only supported during installation of the ODBC Desktop Database Drivers.

## **Adding a Data Source**

### See Also

- 1** Double-click the Control Panel icon. In the Control Panel window, double-click the ODBC icon.
- 2** In the Data Sources dialog box, choose the Add button.
- 3** In the Add Data Source dialog box, select the driver's name from the Installed ODBC Drivers list and choose OK.
- 4** In the ODBC Setup dialog box, enter information to set up the data source. (Note that the Description entry cannot exceed 79 characters.)

## Adding a System Data Source

A data source can be set up with a system data-source name (DSN) that can be used by more than one user on the same machine. The system DSN can also be used by a system-wide service, which can then gain access to the data source even if no user is logged onto the machine.

- 1** Double click the Control Panel icon. In the Control Panel window, double-click the ODBC icon.
- 2** In the Data Sources dialog box, choose the System DSN button.
- 3** In the System Data Sources dialog box, choose the Add button.
- 4** In the Add Data Source dialog box, select the driver's name from the Installed ODBC Drivers list and choose OK.
- 5** In the ODBC Setup dialog box, enter information to set up the data source. (Note that the description entry cannot exceed 79 characters). Choose OK.



## **Modifying a Data Source**

- 1** Double-click the Control Panel icon. In the Control Panel window, double-click the ODBC icon.
- 2** In the Data Sources dialog box, select the data source from the Data Sources list.
- 3** Choose the Setup button.
- 4** In the ODBC Setup dialog box, enter information to set up the data source. (Note that the Description entry cannot exceed 79 characters.)

## **Deleting a Data Source**

- 1** Double-click the Control Panel icon. In the Control Panel window, double-click the ODBC icon.
- 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.

## **Data Source Conversion**

The 32-bit data sources used with the ODBC Desktop Database Drivers version 2.0 are automatically converted to new 32-bit data sources for version 3.0 drivers. No conversion is provided for 16-bit data sources. To change a 16-bit data source to a 32-bit data source, create a new 32-bit data source, then (optionally) delete the old 16-bit data source. A 32-bit data source and a 16-bit data source cannot share the same name.

## **Adding, Modifying, or Deleting a Data Source Dynamically**

### See Also

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 when connecting through the Control Panel are 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.

## Product Support

### CompuServe

The Microsoft Connection on CompuServe provides online technical information for Microsoft products, including the ODBC Desktop Database Drivers. With the Microsoft Connection, you can exchange messages with Microsoft professionals and experienced Microsoft users, and you can download free software—such as patches, tools, and add-ons—provided by Microsoft and CompuServe members.

By using the Microsoft Connection, you can access the Microsoft Developer Services area. You are encouraged to use this area to speak directly to Microsoft about developer-related issues. The Microsoft Developer Services area offers the following advantages:

- **Developer Forums** The ODBC section of the Windows Extension (WinEXT) forum provides information about the ODBC API, application development, and driver development (GO WINEXT). The ODBC Desktop Drivers section of the WinEXT forum provides information about the use of the ODBC Desktop Database Drivers. The section leads for these sections are from Microsoft Product Support and can help answer your questions about the Desktop Database Drivers.
- **Confidential Technical Service Requests** Microsoft offers private (fee-based per incident) technical support to help solve your more complex development problems. For more details, see the Microsoft Developer Services area.
- **Microsoft Knowledge Base** This up-to-date reference tool, compiled by Microsoft Product Support, contains developer-specific technical information about Microsoft products (GO MSKB).

To connect to the Microsoft Connection, type **GO MICROSOFT** at the CompuServe “!” prompt. For information about establishing a CompuServe account, call (800) 848-8199, 8:00 a.m. to 10:00 p.m. EST. Ask for operator 230 and receive a \$15 connect-time usage credit.

### Support Plans

Microsoft also offers pay-as-you-go telephone support from a Microsoft engineer, and a variety of annual paid support plans. In the United States, call (800) 936-3500 for further information on Microsoft support options. Outside of the United States, contact your local Microsoft Subsidiary for information regarding the availability of these products and services.

## ODBC Setup Dialog Box

The ODBC Setup dialog box contains the following fields:

- Approximate Row Count
- Buffer Size
- Collating Sequence
- Data Source Name
- Database
- τις Φορματ\_
- Description
- Directory
- Exclusive
- Extensions List
- Network Directory
- Page Timeout
- Read Only
- Rows to Scan
- Select Directory
- Select Indexes
- Show Deleted Rows
- Sort Order
- System Database
- Use Current Directory
- User Name
- Version
- Workbook

## Approximate Row Count

(This control is displayed only for the **dBASE** and **Microsoft FoxPro** drivers.)

Determines whether table size statistics are approximated. Note that this option applies to all data sources that use the ODBC driver.

## Buffer Size

(This control is displayed only for the **Microsoft Access** driver.)

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). Note that this option applies to all data sources that use the ODBC driver.



## Collating Sequence

(This control is displayed only for the **dBASE**, **Microsoft FoxPro**, and **Paradox** drivers.)

The sequence in which the fields are sorted.

When the **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.

**Data Source Name**

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

## Database

### See Also

(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**

### See Also

(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.

## **Description**

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

## Directory

Displays the currently selected directory. Before you add the data source, you must either use the Select Directory button to select a directory, or select the Use Current Directory check box to use the application's current working directory. (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".

## **Exclusive**

(This control is displayed only for the **dBASE**, **Microsoft Access**, **Microsoft FoxPro**, and **Paradox** drivers.)

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.

## Extensions List

(This control is displayed only for the **Text** driver.)

Lists the filename extensions of the text files on the data source. To use all files in the directory, select the Default (\*.\*) check box. To use only those files with certain extensions, clear the Default (\*.\*) check box and add each extension you want to use.

To add an extension, type the extension in the Extension box and click the Add button. The extension must use the format \*.xxx. For example, to use .DAT files, type the extension \*.DAT. To remove an extension, highlight the extension in the Extensions list and click the Remove button. Removing all extensions is like checking the Default (\*.\*) check box: all files will be displayed.



## Network Directory

(This control is displayed only for the **Paradox** driver.)

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

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

## Page Timeout

(This control is displayed only for the **dBASE**, **Microsoft Access**, **Microsoft FoxPro**, and **Paradox** drivers.)

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). Note that this option applies to all data sources that use the ODBC driver.

## **Read-Only**

Designates the database as read-only.

## Rows to Scan

(This control is displayed only for the **Microsoft Excel** and **Text** drivers.)

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

(This control is displayed only for the **dBASE**, **Microsoft Excel**, **Microsoft FoxPro**, **Paradox**, and **Text** drivers.)

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".

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.

### 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.

## Select Indexes

### See Also

(This control is displayed only for the **dBASE** and **Microsoft FoxPro** drivers.)

Displays the Select Indexes dialog box where you can associate dBASE or Microsoft FoxPro files with index files. The Select Indexes dialog box contains the following fields.

### **Tables**

Displays a list of the files in the currently selected data source.

### **Indexes**

Displays the indexes assigned to the currently selected file in the Tables list.

### **List Files of Type**

Displays and allows you to choose the types of files to display in the Indexes list.

### **To associate an index with a table**

- 1 From the Tables list, select a file.
- 2 From the Indexes list, select an index.
- 3 Choose the OK button to save the table/index associations.

---

**Note** dBASE III indexes or Microsoft FoxPro 2.0 indexes must be assigned using this dialog box for the driver to recognize them.

---

---

**Note** The ODBC dBASE and Microsoft FoxPro drivers do not support Clipper .NTX index files.

---

## Show Deleted Rows

(This control is displayed only for the **dBASE** and **Microsoft FoxPro** drivers.)

Specifies whether or not rows that have been marked as deleted should be displayed.

## Sort Order

(This control is displayed only for the **Microsoft Access** drivers.)

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** drivers.)

The full path of the Microsoft Access system database to be used with the Microsoft Access database you want to access.

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 if no system database will be used. 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.

## **Use Current Directory**

(This control is displayed only for the **dBASE, Microsoft Excel, Microsoft FoxPro, Paradox, Text** drivers.)

When selected, makes the application's current working directory the data source directory and disables the Select Directory option. When cleared, enables you to select the data source directory using the Select Directory option.

The Use Current Directory control is deleted for **Microsoft Excel 5.0**.

## User Name

(This control is displayed only for the **Paradox** driver.)

The Paradox user name.

## **Version**

Enables you to select the version of the files used.

## **Workbook**

(This control is displayed only for the **Microsoft Excel 5.0** driver.)

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.

## Select Database

Enter the full path of an existing **Microsoft Access** database you want to access.

## Create Database

Enter the full path of the new **Microsoft Access** database you want to create.

## Select Network Directory

Select the full path of the directory containing the PDOXUSRS.NET file. If the directory does not contain a PDOXUSRS.NET file, the Paradox driver creates one. For information about the PDOXUSRS.NET file, see the Paradox documentation.



## Defining Text Format

### See Also

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.

---

<b>Section</b>	<b>Information</b>
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

### See Also

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:

<b>Section</b>	<b>Information</b>
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. The special character can be any single character except a double quotation mark (“”).  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.)
CharacterSet	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.
CurrencyPosFormat	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 \$). If this entry is absent, the default value in the control panel is used.
CurrencyDigits	Indicates the number of digits used for the fractional part of a currency amount. If this entry is absent, the default value in the control panel is used.
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 \$. If this entry is absent, the default value in the control panel is used.
CurrencyThousandSymbol	Indicates the single-character symbol to be used for separating currency values in the text file by thousands. If this entry is absent, the default value in the control panel is used.
CurrencyDecimalSymbol	Can be set to any single character that is used to separate the whole from the fractional part of a currency amount. If this entry is absent, the default value in the control panel is used.
DecimalSymbol	Can be any single character used to separate the integer from the fractional part of a number. If this entry is absent, the value in the control panel will be used.
NumberDigits	Indicates the number of decimal digits in the fractional portion of a number. If this entry is absent, the value in the control panel will be used.
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,...	<p>Column names, types, and width.</p> <p>If ColNameHeader is TRUE, the column names listed here are ignored.</p> <p>The format of this entry should be:</p> <p><i>Coln=columnName type [Width #]</i></p> <p><i>columnName</i>: Column names with embedded spaces should be enclosed in quotation marks.</p> <p><i>type</i>: can be BIT, BYTE, SHORT, LONG, CURRENCY, SINGLE, DOUBLE, DATETIME, TEXT, or MEMO.</p> <p>In addition, the following ODBC Text Driver types are supported:</p> <p>CHAR (same as TEXT)  FLOAT (same as DOUBLE)  INTEGER (same as SHORT)  LONGCHAR (same as MEMO)  DATE <i>date format</i></p>

## Text File Format (Advanced)

### See Also

The ODBC **Text** driver supports both delimited and fixed-width text files. A text file consists of an optional header line and zero or more text lines.

Although the header line uses the same format as the other lines in the text file, the ODBC text driver interprets the header line entries as column names, not data.

A delimited text line contains one or more data values separated by delimiters: commas, tabs, or a custom delimiter. The same delimiter must be used throughout the file. Null data values are denoted by two delimiters in a row with no data between them. Character strings in a delimited text line can be enclosed in double quotation marks (" ").

The width of each data entry in a fixed-width text line is specified in a schema. Null data values are denoted by blanks.

Tables are limited to a maximum of 255 fields. Field names are limited to 64 characters, and field widths are limited to 32,766 characters. Records are limited to 65,000 bytes.

A text file can be opened only for a single user. Multiple users are not supported.

The following grammar, written for programmers, defines the format of a text file that can be read by the ODBC text driver. Non-italics represent characters that must be entered as shown, italics represent arguments that are defined elsewhere in the grammar, brackets ([]) represent optional items, braces ({} ) delimit a list of mutually exclusive choices, vertical bars (|) separate these choices, and ellipses (...) represent items that can be repeated one or more times.

The format of a text file is:

```
text-file ::=  
    [delimited-header-line] [delimited-text-line]... end-of-file |  
    [fixed-width-header-line] [fixed-width-text-line]... end-of-file
```

```
delimited-header-line ::= delimited-text-line
```

```
delimited-text-line ::=  
    blank-line |  
    delimited-data [delimiter delimited-data]... end-of-line
```

```
fixed-width-header-line ::= fixed-width-text-line
```

```
fixed-width-text-line ::=  
    blank-line |  
    fixed-width-data [fixed-width-data]... end-of-line
```

```
end-of-file ::= <EOF>
```

```
blank-line ::= end-of-line
```

```
delimited-data ::= delimited-string | number | date | delimited-null
```

```
fixed-width-data ::= fixed-width-string | number | date | fixed-width-null
```

The width of each column in a fixed width text file is specified in the SCHEMA.INI file.

```
end-of-line ::= <CR> | <LF> | <CR><LF>
```

```
delimited-string ::= unquoted-string | quoted-string
```

```
unquoted-string ::= [character | digit] [character | digit | quote-character]...
```

```
quoted-string ::=  
    quote-character  
    [character | digit | delimiter | end-of-line | embedded-quoted-string]...  
    quote-character
```

```
embedded-quoted-string ::=  
    quote-character quote-character  
    [character | digit | delimiter | end-of-line]  
    quote-character quote-character
```

```
fixed-width-string ::= [character | digit | delimiter | quote-character] ...
```

*character* ::= any character except:

*delimiter*  
*digit*  
*end-of-file*  
*end-of-line*  
*quote-character*

*digit* ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

*delimiter* ::= , | <TAB> | *custom-delimiter*

*custom-delimiter* ::= any character except:

*end-of-file*  
*end-of-line*  
*quote-character*

The delimiter in a custom-delimited text file is specified in the SCHEMA.INI file.

*quote-character* ::= "

*number* ::= *exact-number* | *approximate-number*

*exact-number* ::= [+ | -] {*unsigned-integer*[.*unsigned-integer*] |  
*unsigned-integer*. |  
*unsigned-integer*

*approximate-number* ::= *exact-number*{e | E}[+ | -]*unsigned-integer*

*unsigned-integer* ::= {*digit*}...

*date* ::=

*mm date-separator dd date-separator yy* |  
*mmm date-separator dd date-separator yy* |  
*dd date-separator mmm date-separator yy* |  
*yyyy date-separator mm date-separator dd* |  
*yyyy date-separator mmm date-separator dd*

*mm* ::= *digit* [*digit*]

*dd* ::= *digit* [*digit*]

*yy* ::= *digit digit*

*yyyy* ::= *digit digit digit digit*

*mmm* ::= Jan | Feb | Mar | Apr | May | Jun | Jul | Aug | Sep | Oct | Nov | Dec

*date-separator* ::= - | / | .

*delimited-null* ::=

For delimited files, a NULL is represented by no data between two delimiters.

*fixed-width-null* ::= <SPACE>...

For fixed width files, a NULL is represented by spaces.

## Select Unique Record Identifier

Select the column(s) that uniquely identify a record for a table.

**data source**

When the **Microsoft Access** driver is used, the data source specifies the Microsoft Access database (.MDB) you want to use and the information needed to get to that database.

When the **dBASE, Microsoft Excel 3.0/4.0, Microsoft FoxPro, Paradox,** or **Text** driver is used, the data source specifies the default data directory in which the ODBC driver searches for data files you want to access, as well as other information.

When the **Microsoft Excel 5.0** driver is used, the data source specifies the default data workbook in which the ODBC driver searches for data files you want to access, as well as other information.



## Password Entry

### See Also

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.

When the **Microsoft Access** driver is used, 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 Microsoft Access for the Admin user is an empty string.

## **Heterogeneous Joins/Attached Tables**

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

The **Microsoft Access** driver should not be used to attach a table to another table from within the Microsoft Access database management system. The table should be attached directly.

Note that the use of the ODBC Desktop Database Drivers from within Data Access Object (DAO) code is not 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.

## **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.

## **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.

## **Lotus 1-2-3 Data/EMS Data**

The ODBC Desktop Database Drivers 3.0 do not support Lotus 1-2-3 data or EMS data.

## **Visual Basic Support**

The Connect and Attributes properties of a TableDef object in Visual Basic can be used to connect to an ODBC data source using the ODBC Desktop Database Drivers 3.0. For more information, see the Visual Basic documentation.

## Connection Strings (Advanced)

### See Also

The connection string for the ODBC drivers uses the following keywords:

<b>Keyword</b>	<b>Description</b>
DSN	Name of the data source
DBQ	Name of the directory
FIL	File type (MS Access for Microsoft Access; DBASE3, DBASE4, or DBASE5; EXCEL for Microsoft Excel; FOXPRO 2.0, 2.5, or 2.6 for Microsoft FoxPro; PARADOX; or TEXT)

### Driver-Specific Keywords

In addition to the keywords shown above, the **Microsoft Access** driver uses UID and PWD keywords and the **Paradox** driver uses the PWD keyword.

<b>Keyword</b>	<b>Description</b>
UID	The user login ID (Microsoft Access only)
PWD	The user-specified password.

---

**Note** 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 Returned Values](#)).

---



## Select Database (Advanced)

When the **Microsoft Access** driver is used, the Select button may be pressed in the Microsoft Access 3.0 Setup dialog box to select an existing database.

The Open Database dialog box contains the following fields:

<b>Control</b>	<b>Description</b>
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 to create a new database.

---

**Note** A database created using the Create button will only work with Microsoft Access 7.0.

---

The New Database dialog box contains the following fields:

<b>Control</b>	<b>Description</b>
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 (*pszAttributes*) 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, select the version of the resulting compacted file 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 (*IpszAttributes*) 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.

## Additional Supported ODBC SQL Grammar (Advanced)

The ODBC drivers completely support the Minimum ODBC grammar and the following clauses in the Core and Extended ODBC grammar. Additional information may be displayed by jumping to the ALTER TABLE statement, correlation names, and procedure invocation topics.

- Approximate numeric literal (Core)
- ALTER TABLE statement (Core)
- Binary Literals (Extended)
- BETWEEN predicate (Core)
- Correlation names (Core)
- EXISTS predicate (Core)
- IN (*valuelist*) (Core)
- Procedure Invocation (Extended)
- SELECT statement (Extended) (including GROUP BY, HAVING, and UNION clauses)
- Subqueries (SUBSELECTs) (Core)

## **ALTER TABLE Statement (Core)**

### See Also

When the **Microsoft Access**, **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 the **Microsoft Excel** or **Text** driver.)

## Correlation Names (Core)

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'
```

## 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 (`).

A parameterized query can be called using the above statement.

## **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.

When the **Paradox** driver is used, the COUNT function includes NULL as well as non-NULL values in the count.

## CREATE INDEX Statement

### See Also

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 **dBASE**, **Microsoft FoxPro**, and **Paradox** drivers accept the syntax, but ignore the presence of either option.

When the **Paradox** driver is used, the CREATE INDEX statement creates Paradox primary key files and secondary files.

## **Date Arithmetic**

The driver supports adding and subtracting an integer from a DATE column. The integer specifies the number of days to add or subtract.

## Date Literals

For maximum interoperability, applications should pass date literals in the ODBC canonical format using escape-clause syntax: {d 'value'}, where *value* is in the form "yyyy-mm-dd".

## **DROP INDEX Statement**

When the **Microsoft Access**, **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.

The DROP INDEX statement is 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.

## **GROUP BY *expression-list***

GROUP BY supports an expression list as well as a column name. The select list of a SELECT statement that has a GROUP BY clause can only include expressions from the GROUP BY clause or set functions.

## **ORDER BY *expression-list***

Expressions can be used in the ORDER BY clause. 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
```

No ordering is allowed on set functions or an expression that contains a set function.



## **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. Nested OUTER JOINS are supported. For more information on OUTER JOINS, refer to the documentation for the applicable Database Management System.

## Scalar Functions

The Microsoft ODBC Desktop Database drivers 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*.

## Table Names

When the **dBASE**, **Microsoft Excel**, **Microsoft FoxPro**, **Paradox**, or **Text** driver is used, table names that occur in the FROM clause of SELECT or DELETE, after the INTO clause in INSERT, and after UPDATE, CREATE TABLE, and DROP TABLE can contain a valid path, primary name, and filename extension.

Use of a table name elsewhere in an SQL statement does not support the use of paths or extensions but will accept only the primary name (for example, EMP FROM C:\ABC\EMP).

Correlation names (aliases) can be used. For example:

```
SELECT *  
FROM C:\ABC\EMP T1  
WHERE T1.COL1 = 'aaa'
```

## **Aggregate Function Limitations**

An aggregate function and a non-aggregate column reference cannot both be used as arguments to a single SQL statement.

## **ALTER TABLE Statement Limitations**

### See Also

When the **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 Predicate Limitations**

A maximum of 40 is supported.

## **CALL Statement Limitations**

(This limitation applies to the **Microsoft Access** driver only.)

Expressions are not supported as parameters to a called procedure.



## Column Name Limitations

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 **dBASE** or **Microsoft FoxPro** driver is used, characters with an ASCII value greater than 127 are converted to underscores.

When the **Microsoft Excel** driver is used, a name that in Microsoft Excel would use the “!” character must use the “\$” character instead, because the “!” character is not legal in an ODBC name, even when the name is enclosed in back quotes “`”.

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 Limitations**

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 Limitations**

### See Also

The CREATE INDEX statement is not supported for the **Microsoft Excel** or **Text** driver.

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, response time on large files can be improved by building an .MDX (or .NDX) index on the column (field) specified in the WHERE clauses of a SELECT statement.

Existing .MDX indexes will automatically be applied for =, >, <, >=, =<, and BETWEEN operators in a WHERE clause, and LIKE predicates, as well as in join predicates.

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 upon a contiguous subset of the columns in a table, including the first column. A table cannot be updated by the Paradox driver if a unique index is not defined on the table.

## CREATE TABLE Statement Limitations

When the **Microsoft Access**, **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. When the **Microsoft Excel5** driver is used to access an Excel version 5.0 worksheet, a DROP TABLE statement clears the worksheet, but does not delete the worksheet name.

When the **Paradox** driver is used, columns cannot be added once an index has been defined on a table. If the first column of the argument list of a CREATE TABLE statement creates an index, a second column cannot be included in the argument list.

## **Date Arithmetic Limitations**

Date arithmetic is not supported for subtracting a DATE data type from a DATE data type.

## **DELETE Statement Limitations**

The DELETE statement is not supported for the **Microsoft Excel** or **Text** driver. Note that the INSERT statement is supported for the Text driver.

For the **Paradox** driver to delete a row from a table, the table must have a unique index (Paradox primary key).

## **DISTINCT Keyword Limitations**

Not supported for Long Text fields (**Microsoft Access**) or Memo fields (**dBASE** or **FoxPro**).

## **DROP TABLE Statement Limitations**

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 Limitations**

The maximum number of tables in a FROM clause is 16.

## **HAVING Clause Limitations**

The maximum number of search conditions in a HAVING clause is 40.

## Identifiers Limitations

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 the reserved words listed in the SQL grammar in Appendix C of the *Microsoft ODBC Programmer's Reference* (or the shorthand form of these reserved words) as identifiers (that is, table or column names), unless you surround the word in back quotes (`).

## Index Name Limitations

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 table upon which they are defined.

## INSERT Statement Limitations

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.

When a **dBASE**, **Microsoft Excel**, **Microsoft FoxPro**, **Paradox**, or **Text** driver is used, inserting a zero-length string into a column actually inserts a NULL instead.

When the **Microsoft Excel** driver is used, inserting a NULL into a column of the first row of the table, when the column has a data type other than SQL\_VARCHAR, will cause the data type of the column to change to SQL\_VARCHAR.

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, NULL values are represented by a blank padded string in fixed-length files, but are represented by no spaces in delimited files. For example, in the following row containing three fields, the second field is a NULL value:

```
"Smith:., 123
```

When the **Text** driver is used, all column values can be padded with leading spaces. The length of any row must be less than or equal to 65,543 bytes.

## **LIKE Predicate Limitations**

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.

## **NOT NULL Limitations**

The NOT NULL constraint in the CREATE TABLE statement is not supported.

## **ORDER BY Clause Limitations**

If a SELECT statement contains a GROUP BY clause and an ORDER BY clause, the ORDER BY clause can contain only a column in the result set or an expression in the GROUP BY clause.



## Parameterized Query Limitations

When the **Microsoft Access** driver is used, a parameterized query can be called using the following syntax:  
CALL *query-name* [(*parameter*[,*parameter*]...)].

## Reserved Word Limitations

Do not use the reserved words listed in the SQL grammar in Appendix C of the *Microsoft ODBC Programmer's Reference* (or the shorthand form of these reserved words) as identifiers (that is, table or column names), unless you surround the word in back quotes (`).

## **Scalar Function Limitations**

Scalar functions are supported only by using the ODBC canonical format.

## **SELECT DISTINCT Limitations**

The DISTINCT keyword does not apply to binary data.

## **SELECT Statement Limitations**

An aggregate-function column cannot be mixed with a non-aggregate column in a SELECT statement.

The select list of a SELECT statement that has a GROUP BY clause can only have expressions from the GROUP BY clause or set functions.

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. Use a parameter in the SELECT statement if you need to refer to a data value that contains a vertical bar.

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".

## **Set Functions Limitations**

The set functions (AVG, MAX, MIN, and SUM) do not support the DISTINCT keyword.

## **Sorting Limitations**

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.

## String Limitations

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 Name Limitations

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, and a table name is not qualified by a database reference, the default database is implied. A name that in Microsoft Excel would use the “!” character must use the '\$' character instead, because the “!” character is not legal in an ODBC name, even when the name is enclosed in back quotes “`”.

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.

When the **dBASE** or **Microsoft FoxPro** driver is used, characters with an ASCII value greater than 127 are converted to underscores.

When the **Microsoft Access** driver is used, the table name is limited to 64 characters.

When the **dBASE**, **Microsoft Excel 3/4**, **Microsoft FoxPro**, **Paradox**, or **Text** driver is used, special DOS keywords CON, AUX, LPT1, and LPT2 should not be used as table names.

## **Table References Limitations**

A maximum of 16 table references can be included in any query statement.

## **UPDATE Statement Limitations**

For the **Paradox** driver to update a table, the table must have a unique index (Paradox primary key).

Not supported by the **Text** driver.

## **Views Limitations**

Not supported by the **dBASE, Microsoft Excel, Microsoft FoxPro, Paradox, or Text** driver.

## **WHERE Clause Limitations**

The maximum number of clauses in a WHERE clause is 40.

LONGVARBINARY and LONGVARCHAR columns can be compared to literals of up to 255 characters in length, but cannot be compared using parameters.

## **WHERE CURRENT OF Clause Limitations**

Not supported.

## **dBASE Indexes (Advanced)**

### See Also

The ODBC **dBASE** driver automatically opens and updates dBASE IV index files. You must use the ODBC Control Panel option to associate dBASE III .NDX files with dBASE files.

The following limitations apply to the creation of dBASE indexes:

- All column names must be valid.
- All columns must be in the same ascending or descending order.
- The length of any single text column must be less than 100 bytes.
- If more than one column exists, all of the columns must be text columns and the sum of the column sizes must be less than 100 bytes.
- Memo fields cannot be indexed.
- An index must not be specified for the current set of fields (that is, duplicate indexes are not allowed).
- The index name must match the dBASE index naming convention. dBASE III requires that each index be in a separate file, each having an .NDX extension. In dBASE IV, indexes are created as tag names that are stored in a single .MDX file. The .MDX file has the same base name as the database file (for example, EMP.MDX is the index file for the EMP.DBF database).

## Microsoft FoxPro Indexes (Advanced)

### See Also

The ODBC Microsoft **FoxPro** driver automatically opens and updates Microsoft FoxPro 2.5/2.6 index files. You must use the ODBC Control Panel option to associate Microsoft FoxPro 2.0 .IDX files with Microsoft FoxPro files.

The following limitations apply to the creation of Microsoft FoxPro indexes:

- All column names must be valid.
- All columns must be in the same ascending or descending order.
- The length of any single text column must be less than 100 bytes.
- If more than one column exists, all columns must be text columns and the sum of the column sizes must be less than 100 bytes.
- Memo fields cannot be indexed.
- An index must not be specified for the current set of fields (that is, duplicate indexes are not allowed).
- The index name must match the Microsoft FoxPro index naming convention. Microsoft FoxPro 2.0 requires that each index be in a separate file, each having an .IDX extension. In Microsoft FoxPro 2.5/2.6, indexes are created as tag names that are stored in a single .CDX file. The .CDX file has the same base name as the database file (for example, EMP.CDX is the index file for the EMP.DBF database).



## CREATE INDEX for Paradox (Advanced)

### See Also

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

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

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

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

- Before a non-unique index can be created for a table, a primary index must exist for that table.
- The *index-name* argument for any index other than a primary index (unique or non-unique) must be the same as the column name.
- Only one column can be specified for a non-unique index.

Columns cannot be added once an index has been defined on a table. If the first column of the argument list of a CREATE TABLE statement creates an index, a second column cannot be included in the argument list.

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

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

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

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

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.

## Microsoft Access 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 (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
GUID	BINARY	SQL_LONGBINARY
LONG BINARY	LONGBINARY	SQL_LONGVARBINARY
LONG TEXT	LONGTEXT	SQL_LONGVARCHAR
MEMO	LONGTEXT	SQL_LONGVARCHAR
NUMBER (FieldSize= SINGLE)	SINGLE	SQL_REAL
NUMBER (FieldSize= DOUBLE)	DOUBLE	SQL_DOUBLE
NUMBER (FieldSize= BYTE)	UNSIGNEDBYTE	SQL_TINYINT
NUMBER (FieldSize= INTEGER)	SHORT	SQL_SMALLINT
NUMBER (FieldSize= LONG INTEGER)	LONG	SQL_INTEGER
OLE	LONGBINARY	SQL_LONGVARBINARY
TEXT	VARCHAR	SQL_VARCHAR
VARBINARY	VARBINARY	SQL_VARBINARY

---

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

---

The following table shows limitations on Microsoft Access data types.

Data type	Description
BINARY, VARBINARY, and VARCHAR	Creating a BINARY, VARBINARY, or VARCHAR column of zero or unspecified length actually returns a 255-byte column.
CHAR, LONGVARCHAR, and VARCHAR	A character string literal can contain any ANSI character (1-255 decimal). Use two consecutive single quotation marks (") to represent one single quotation mark (').  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](#).

## dBASE Data Types (Advanced)

The following table shows how dBASE data types are mapped to ODBC SQL data types. Note that not all ODBC SQL data types are supported.

<b>dBASE data type</b>	<b>ODBC data type</b>
CHAR	SQL_VARCHAR
DATE	SQL_DATE
FLOAT *	SQL_DOUBLE
LOGICAL	SQL_BIT
MEMO	SQL_LONGVARCHAR
NUMERIC (BCD)	SQL_DOUBLE
OLEOBJECT *	SQL_LONGBINARY

\* Only valid for dBASE version 5.x

Precision in dBASE III allows numbers with up to two-digit exponents and in dBASE IV numbers with up to three-digit exponents. Because numbers are stored as text, they are converted to numbers. If the number to convert does not fit in a field, unexplained results may occur.

While dBASE allows a precision and a scale to be specified with a Numeric data type, it is not supported by the ODBC dBASE driver. The ODBC dBASE driver always returns a precision of 15 and a scale of 0 for a Numeric data type.

A column created with the Numeric data type using the ODBC dBASE driver maps to the SQL\_DOUBLE ODBC data type. Thus the data in this column is subject to rounding. This behavior is not the same as that of the Numeric data type in dBASE (type N), which is Binary Coded Decimal (BCD).

---

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

---

The following table shows limitations on dBASE data types.

<b>Data type</b>	<b>Description</b>
CHAR	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	The dBASE driver cannot create an index on a LOGICAL column.
MEMO	The maximum length of a MEMO column is 65,500 bytes.

More limitations on data types can be found in [Data Type Limitations](#).

## Microsoft Excel Data Types (Advanced)

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

Microsoft Excel data type	ODBC data type
CURRENCY	SQL_NUMERIC
DATETIME	SQL_TIMESTAMP
LOGICAL	SQL_BIT
NUMBER	SQL_DOUBLE
TEXT	SQL_VARCHAR

---

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

---

The following table shows limitations on Microsoft Excel data types.

Data type	Description
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	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	<p>When the rows of a column contain more than one Microsoft Excel data type, the ODBC Microsoft Excel driver assigns the SQL_VARCHAR data type to the column. There is one exception to this: if the column contains only two or three of the date/time data types (DATE, TIME, and DATETIME), the ODBC Microsoft Excel driver assigns the SQL_TIMESTAMP data type to the column.</p> <p>Creating a TEXT column of zero or unspecified length actually returns a 255-byte column.</p> <p>A character string literal can contain any ANSI character (1-255 decimal). Use two consecutive single quotation marks (") to represent one single quotation mark (').</p> <p>Inserting a NULL into a column with a data type other than SQL_VARCHAR will cause the data type of the column to change to</p>

SQL\_VARCHAR.

More limitations on data types can be found in [Data Type Limitations](#).

## Microsoft FoxPro Data Types (Advanced)

The following table shows how Microsoft FoxPro data types are mapped to ODBC SQL data types. Note that not all ODBC SQL data types are supported.

Microsoft FoxPro data type	ODBC data type
CHAR	SQL_VARCHAR
DATE	SQL_DATE
GENERAL	SQL_LONGVARBINARY
LOGICAL	SQL_BIT
MEMO	SQL_LONGVARCHAR
NUMERIC (BCD)	SQL_DOUBLE

Precision in Microsoft FoxPro 2.0 allows numbers with up to two-digit exponents and in Microsoft FoxPro 2.5 numbers with up to three-digit exponents. Because numbers are stored as text, they are converted to numbers. If the number to convert does not fit in a field, unexplained results may occur.

While Microsoft FoxPro allows a precision and a scale to be specified with a Numeric data type, it is not supported by the ODBC Microsoft FoxPro driver. The ODBC Microsoft FoxPro driver always returns a precision of 15 and a scale of 0 for a Numeric data type.

A column created with the Numeric data type using the ODBC Microsoft FoxPro driver maps to the SQL\_DOUBLE ODBC data type. Thus the data in this column is subject to rounding. This behavior is not the same as that of the Numeric data type in Microsoft FoxPro (type N), which is Binary Coded Decimal (BCD).

---

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

---

The following table shows limitations on Microsoft FoxPro data types.

Data type	Description
CHAR	Creating a CHAR column of zero or unspecified length actually returns a 254-byte column.
GENERAL	The General data type is supported only for Microsoft FoxPro version 2.5 files.
MEMO	The maximum length of a MEMO column is 65,500 bytes.

More limitations on data types can be found in [Data Type Limitations](#).

## Paradox Data Types (Advanced)

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

Paradox data type	ODBC data type
ALPHANUMERIC	SQL_VARCHAR
AUTOINCREMENT #	SQL_INTEGER
BCD #	SQL_DOUBLE
BYTES #	SQL_BINARY
DATE	SQL_DATE
IMAGE *	SQL_LONGVARIABLE
LOGICAL #	SQL_BIT
LONG #	SQL_INTEGER
MEMO	SQL_LONGVARCHAR
MONEY #	SQL_DOUBLE
NUMBER	SQL_DOUBLE
SHORT	SQL_SMALLINT
TIME #	SQL_TIMESTAMP
TIMESTAMP #	SQL_TIMESTAMP

\* Only valid for Paradox versions 4.x and 5.x.

# Only valid for Paradox versions 5.x.

---

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

---

The following table shows limitations on Paradox data types.

Data type	Description
ALPHANUMERIC	Creating an Alphanumeric column of zero or unspecified length actually returns a 255-byte column.
LONG	The maximum negative value supported by the Paradox driver for the Long data type in Paradox 5.x is not $-2^{31}$ (-2147483648), as it should be since Long maps to the ODBC data type SQL_INTEGER. The maximum negative value supported for Long is actually $-2^{31} + 1$ (-2147483647).

More limitations on data types can be found in [Data Type Limitations](#).



## Text Data Types (Advanced)

The following table shows how text data types are mapped to ODBC SQL data types. Note that not all ODBC SQL data types are supported by the ODBC Text driver.

Text data type	ODBC data type
CHAR	SQL_VARCHAR
DATETIME	SQL_TIMESTAMP
FLOAT	SQL_DOUBLE
INTEGER	SQL_INTEGER
LONGCHAR	SQL_LONGVARCHAR

---

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

---

The following table shows limitations on Text data types.

Data type	Description
CHAR	Creating a CHAR column of zero or unspecified length actually returns a 255-bit column.
DATETIME	MM-DD-YY (for example, 01-17-92) MMM-DD-YY (for example, Jan-17-92) DD-MMM-YY (for example, 17-Jan-92) YYYY-MM-DD (for example, 1992-01-17) YYYY-MMM-DD (for example, 1992-Jan-17)
FLOAT	The maximum width includes the sign and decimal point. In SCHEMA.INI, the width is denoted as follows: 14.083 is Float Width 6 -14.083 is Float Width 7 +14.083 is Float Width 7 14083. is Float Width 6 ODBC always returns 8 for float columns. Float columns can also be in scientific notation, for example: -3.04E+2 is Float Width 8 25E4 is Float Width 4 <b>Note</b> Decimal and scientific notation cannot be mixed in a column.
INTEGER	Valid values for INTEGER columns are 32767 to -32766. In SCHEMA.INI, the width is denoted as follows: 14083 is Integer Width 5 0 is Integer Width 1 ODBC always returns 4 for integer columns.

More limitations on data types can be found in [Data Type Limitations](#).

## Data Type Limitations (Advanced)

The Microsoft ODBC Desktop Database Drivers impose the following limitations on the data types:

<b>Data type</b>	<b>Description</b>
All data types	Type conversion failures result in the affected column being set to NULL.
BINARY	Creating a zero-length BINARY column actually returns a 255-byte 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.
Zero-Length Strings	When a <b>dBASE</b> , <b>Microsoft Excel</b> , <b>Microsoft FoxPro</b> , <b>Paradox</b> , or <b>Text</b> driver is used, inserting a zero-length string into a column actually inserts a NULL instead.

## Error Messages (Advanced)

When an error occurs, the Microsoft ODBC Desktop Database drivers return the native error number, the SQLSTATE (an ODBC error code), and an error message.

### Native Error

For errors that occur in the driver ISAM layer, the driver returns the native error returned to it by the ODBC File Library (that is, the ODBC driver ISAM).

### SQLSTATE

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

### Error Message

For errors that occur in the data source, the ODBC driver returns an error message returned to it by the ODBC File Library. For errors that occur in the ODBC 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, *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 and driver ISAM:

<b>Error message</b>	<b>Error location</b>
<i>[Microsoft][ODBC Driver Manager] message-text</i>	Driver Manager (ODBC32.DLL)
<i>[Microsoft][ODBC driver-name] [ODBC File Library] message-text</i>	Driver ISAM (see <a href="#">Driver ISAMs Table</a> )

## SQLGetInfo Returned Values (Programming)

The following table lists the C-language #defines for the *fnInfoType* argument and the corresponding values returned by **SQLGetInfo**. This information can be retrieved by passing the listed C-language #defines to **SQLGetInfo** in the *fnInfoType* 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*.

»»	SQL_ACCESSIBLE_PROCEDURES
»»	SQL_ACCESSIBLE_TABLES
»»	SQL_ACTIVE_CONNECTIONS
»»	SQL_ACTIVE_STATEMENTS
»»	SQL_ALTER_TABLE
»»	SQL_BOOKMARK_PERSISTENCE
»»	SQL_COLUMN_ALIAS
»»	SQL_CONCAT_NULL_BEHAVIOR
»»	SQL_CONVERT_BIGINT
»»	SQL_CONVERT_BINARY
»»	SQL_CONVERT_BIT
»»	SQL_CONVERT_CHAR
»»	SQL_CONVERT_DATE
»»	SQL_CONVERT_DECIMAL
»»	SQL_CONVERT_DOUBLE
»»	SQL_CONVERT_FLOAT
»»	SQL_CONVERT_FUNCTIONS
»»	SQL_CONVERT_INTEGER
»»	SQL_CONVERT_LONGVARBINARY
»»	SQL_CONVERT_LONGVARCHAR
»»	SQL_CONVERT_NUMERIC
»»	SQL_CONVERT_REAL
»»	SQL_CONVERT_SMALLINT
»»	SQL_CONVERT_TIME
»»	SQL_CONVERT_TIMESTAMP
»»	SQL_CONVERT_TINYINT
»»	SQL_CONVERT_VARBINARY
»»	SQL_CONVERT_VARCHAR
»»	SQL_CORRELATION_NAME
»»	SQL_CURSOR_COMMIT_BEHAVIOR
»»	SQL_CURSOR_ROLLBACK_BEHAVIOR
»»	SQL_DATA_SOURCE_NAME
»»	SQL_DATA_SOURCE_READ_ONLY
»»	SQL_DATABASE_NAME
»»	SQL_DBMS_NAME
»»	SQL_DBMS_VER
»»	SQL_DEFAULT_TXN_ISOLATION
»»	SQL_DRIVER_NAME
»»	SQL_DRIVER_ODBC_VER
»»	SQL_DRIVER_VER
»»	SQL_EXPRESSIONS_IN_ORDERBY
»»	SQL_FETCH_DIRECTION
»»	SQL_FILE_USAGE
»»	SQL_GETDATA_EXTENSIONS
»»	SQL_GROUP_BY
»»	SQL_IDENTIFIER_CASE
»»	SQL_IDENTIFIER_QUOTE_CHAR
»»	SQL_KEYWORDS
»»	SQL_LIKE_ESCAPE_CLAUSE
»»	SQL_LOCK_TYPES
»»	SQL_MAX_BINARY_LITERAL_LEN
»»	SQL_MAX_CHAR_LITERAL_LEN
»»	SQL_MAX_COLUMN_NAME_LEN

» SQL\_MAX\_COLUMNS\_IN\_GROUP\_BY  
» SQL\_MAX\_COLUMNS\_IN\_INDEX  
» SQL\_MAX\_COLUMNS\_IN\_ORDER\_BY  
» SQL\_MAX\_COLUMNS\_IN\_SELECT  
» SQL\_MAX\_COLUMNS\_IN\_TABLE  
» SQL\_MAX\_CURSOR\_NAME\_LEN  
» SQL\_MAX\_INDEX\_SIZE  
» SQL\_MAX\_OWNER\_NAME\_LEN  
» SQL\_MAX\_PROCEDURE\_NAME\_LEN  
» SQL\_MAX\_QUALIFIER\_NAME\_LEN  
» SQL\_MAX\_ROW\_SIZE  
» SQL\_MAX\_ROW\_SIZE\_INCLUDES\_LONG  
» SQL\_MAX\_STATEMENT\_LEN  
» SQL\_MAX\_TABLE\_NAME\_LEN  
» SQL\_MAX\_TABLES\_IN\_SELECT  
» SQL\_MAX\_USER\_NAME\_LEN  
» SQL\_MULT\_RESULT\_SETS  
» SQL\_MULTIPLE\_ACTIVE\_TXN  
» SQL\_NEED\_LONG\_DATA\_LEN  
» SQL\_NON\_NULLABLE\_COLUMNS  
» SQL\_NULL\_COLLATION  
» SQL\_NUMERIC\_FUNCTIONS  
» SQL\_ODBC\_API\_CONFORMANCE  
» SQL\_ODBC\_SAG\_CLI\_CONFORMANCE  
» SQL\_ODBC\_SQL\_CONFORMANCE  
» SQL\_ODBC\_SQL\_OPT\_IEF  
» SQL\_ODBC\_VER  
» SQL\_OJ\_CAPABILITIES  
» SQL\_ORDER\_BY\_COLUMNS\_IN\_SELECT  
» SQL\_OUTER\_JOINS  
» SQL\_OWNER\_TERM  
» SQL\_OWNER\_USAGE  
» SQL\_POS\_OPERATIONS  
» SQL\_POSITIONED\_STATEMENTS  
» SQL\_PROCEDURE\_TERM  
» SQL\_PROCEDURES  
» SQL\_QUALIFIER\_LOCATION  
» SQL\_QUALIFIER\_NAME\_SEPARATOR  
» SQL\_QUALIFIER\_TERM  
» SQL\_QUALIFIER\_USAGE  
» SQL\_QUOTED\_IDENTIFIER\_CASE  
» SQL\_ROW\_UPDATES  
» SQL\_SCROLL\_CONCURRENCY  
» SQL\_SCROLL\_OPTIONS  
» SQL\_SEARCH\_PATTERN\_ESCAPE  
» SQL\_SERVER\_NAME  
» SQL\_SPECIAL\_CHARACTERS  
» SQL\_STATIC\_SENSITIVITY  
» SQL\_STRING\_FUNCTIONS  
» SQL\_SUBQUERIES  
» SQL\_SYSTEM\_FUNCTIONS  
» SQL\_TABLE\_TERM  
» SQL\_TIMEDATE\_ADD\_INTERVALS  
» SQL\_TIMEDATE\_DIFF\_INTERVALS  
» SQL\_TIMEDATE\_FUNCTIONS  
» SQL\_TXN\_CAPABLE  
» SQL\_TXN\_ISOLATION\_OPTION  
» SQL\_UNION  
» SQL\_USER\_NAME

## **SQL\_ACCESSIBLE\_PROCEDURES**

"Y"

(Microsoft Access)

"N"

(dBASE, Microsoft Excel, Microsoft FoxPro, Paradox, Text)

**SQL\_ACCESSIBLE\_TABLES**

"Y"

## **SQL\_ACTIVE\_CONNECTIONS**

64



## SQL\_ACTIVE\_STATEMENTS

0

## **SQL\_ALTER\_TABLE**

SQL\_AT\_ADD\_COLUMN |

SQL\_AT\_DROP\_COLUMN

(Microsoft Access, 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

**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



## **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



## **SQL\_CONVERT\_VARCHAR**

SQL\_CVT\_DOUBLE |  
SQL\_CVT\_FLOAT |  
SQL\_CVT\_INTEGER |  
SQL\_CVT\_NUMERIC |  
SQL\_CVT\_REAL |  
SQL\_CVT\_SMALLINT |  
SQL\_CVT\_VARCHAR

**SQL\_CORRELATION\_NAME**

SQL\_CN\_DIFFERENT

**SQL\_CURSOR\_COMMIT\_BEHAVIOR**

SQL\_CB\_CLOSE

**SQL\_CURSOR\_ROLLBACK\_BEHAVIOR**

SQL\_CB\_CLOSE

**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, dBASE, Microsoft FoxPro, Paradox)

"Y"

(Microsoft Excel, Text)

**SQL\_DATABASE\_NAME**

Filename

## **SQL\_DBMS\_NAME**

"ACCESS" (Microsoft Access)

"DBASE" (dBASE)

"EXCEL" (Microsoft Excel)

"FOXPRO" (Microsoft FoxPro)

"PARADOX" (Paradox)

"TEXT" (Text)



## **SQL\_DBMS\_VER**

"1.0", "1.1", "2.0", or "7.0" (Microsoft Access)

"3.0", "4.0", or "5.0" (dBASE)

"3.0", "4.0", or "5.0" (Microsoft Excel)

"2.0", "2.5", or "2.6" (Microsoft FoxPro)

"3.x", "4.x", or "5.x" (Paradox)

"1.0" (Text)

## **SQL\_DEFAULT\_TXN\_ISOLATION**

SQL\_TXN\_READ\_COMMITTED [Microsoft Access]

**SQL\_DRIVER\_NAME**

"ODBCJT32.DLL"

**SQL\_DRIVER\_ODBC\_VER**

"2.50"

**SQL\_DRIVER\_VER**

" 2.99.*nnnn*" (*nnnn* specifies the build date.)

**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, 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)

## **SQL\_KEYWORDS**

ALPHANUMERIC, AUTOINCREMENT, BINARY, BOOLEAN, BYTE, CURRENCY, DATABASE, DATABASENAME, DATETIME, DISTINCTROW, DOUBLEFLOAT, FLOAT4, FLOAT8, GENERAL, IEEEDOUBLE, IEEE SINGLE, 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

"N"

## **SQL\_LOCK\_TYPES**

SQL\_LOCK\_NO\_CHANGE

**SQL\_MAX\_BINARY\_LITERAL\_LEN**

255



## **SQL\_MAX\_CHAR\_LITERAL\_LEN**

255 (Microsoft Access)  
254 (dBASE)  
255 (Microsoft Excel)  
254 (Microsoft FoxPro)  
255 (Paradox)  
255 (Text)

## **SQL\_MAX\_COLUMN\_NAME\_LEN**

64 (Microsoft Access)  
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\_BY

10

## **SQL\_MAX\_COLUMNS\_IN\_INDEX**

32 (Microsoft Access)

0 (Limit Unknown or Not Applicable) (dBase)

0 (Microsoft Excel)

0 (Limit Unknown or Not Applicable) (Microsoft FoxPro)

0 (Limit Unknown or Not Applicable) (Paradox)

0 (Text)

## SQL\_MAX\_COLUMNS\_IN\_ORDER\_BY

10

## SQL\_MAX\_COLUMNS\_IN\_SELECT

255

## SQL\_MAX\_COLUMNS\_IN\_TABLE

255

**SQL\_MAX\_CURSOR\_NAME\_LEN**

64



## **SQL\_MAX\_INDEX\_SIZE**

255 (Microsoft Access)

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 (dBASE)  
0 (Microsoft Excel)  
0 (Microsoft FoxPro)  
0 (Paradox)  
0 (Text)

**SQL\_MAX\_QUALIFIER\_NAME\_LEN**

66

## **SQL\_MAX\_ROW\_SIZE**

2096 (Microsoft Access)  
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)

"N" (dBASE)

"Y" (Microsoft Excel)

"N" (Microsoft FoxPro)

"N" (Paradox)

"Y" (Text)

**SQL\_MAX\_STATEMENT\_LEN**

65000

## **SQL\_MAX\_TABLE\_NAME\_LEN**

- 64 (Microsoft Access)
- 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"

**SQL\_NEED\_LONG\_DATA\_LEN**

"N"

**SQL\_NON\_NULLABLE\_COLUMNS**

SQL\_NNC\_NULL

**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

**SQL\_ODBC\_SAG\_CLI\_CONFORMANCE**

SQL\_OSCC\_NOT\_COMPLIANT

**SQL\_ODBC\_SQL\_CONFORMANCE**

SQL\_OSC\_MINIMUM

SQL\_ODBC\_SQL\_OPT\_IEF

"N"

**SQL\_ODBC\_VER**

From Driver Manager

## **SQL\_OJ\_CAPABILITIES**

SQL\_OJ\_LEFT

SQL\_OJ\_RIGHT

SQL\_OJ\_NOT\_ORDERED

SQL\_OJ\_INNER

SQL\_OJ\_ALL\_COMPARISON\_OPS

**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, dBASE, Microsoft FoxPro, Paradox)

SQL\_POS\_POSITION |  
SQL\_POS\_REFRESH

(Microsoft Excel, Text)

## SQL\_POSITIONED\_STATEMENTS

0

## **SQL\_PROCEDURE\_TERM**

"QUERY"

(Microsoft Access)

""

(dBASE, Microsoft Excel, Microsoft FoxPro, Paradox, Text)

## **SQL\_PROCEDURES**

"Y"

(Microsoft Access)

"N"

(dBASE, Microsoft Excel, Microsoft FoxPro, Paradox, Text)

**SQL\_QUALIFIER\_LOCATION**

SQL\_QL\_START

## **SQL\_QUALIFIER\_NAME\_SEPARATOR**

`.`

(Microsoft Access, Microsoft Excel 5)

`\`

(dBASE, Microsoft Excel 4, Microsoft FoxPro, Paradox, Text)



## **SQL\_QUALIFIER\_TERM**

"Database" (Microsoft Access)  
"Directory" (dBASE)  
"Workbook" (Microsoft Excel 5)/  
"Directory" (Microsoft Excel3/4)  
"Directory" (Microsoft FoxPro)  
"Directory" (Paradox)  
"Directory" (Text)

## **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  
(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"

## **SQL\_SCROLL\_CONCURRENCY**

SQL\_SCCO\_READ\_ONLY |

SQL\_SCCO\_LOCK |

SQL\_SCCO\_OPT\_VALUES

(Microsoft Access, 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)

"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  
(Microsoft Access, Microsoft Excel, Text)

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\_SECOND |  
SQL\_FN\_TD\_WEEK |  
SQL\_FN\_TD\_YEAR  
(dBASE, Microsoft FoxPro, Paradox)

## **SQL\_TXN\_CAPABLE**

SQL\_TC\_ALL  
(Microsoft Access)

0  
(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

## **Supported ODBC API Functions**

The Microsoft ODBC drivers support all Core and Level 1 functions and the following Level 2 functions:

- SQLDataSources
- SQLDrivers
- SQLExtendedFetch
- SQLMoreResults
- SQLNativeSQL
- SQLNumParams
- SQLProcedures
- SQLProcedureColumns
- SQLSetPos
- SQLSetScrollOptions

## **SQLBindParameter**

When the Microsoft Excel 3.0/4.0 driver or the Microsoft Excel 5.0 driver is used, executing an INSERT statement that uses a parameter to insert a NULL into a SQL\_CHAR column will return SQL\_SUCCESS\_WITH\_INFO with SQLSTATE 01004, "Data Truncated".

## 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 *fDescType* option of **SQLColAttributes** returns FALSE for a column of **Paradox** data type CURRENCY, which maps to the ODBC SQL data type SQL\_DOUBLE.



## **SQLColumns**

A column may be shown as nullable by **SQLColumns** even if the column participates in a primary key or unique index, so cannot accept a NULL value.

## SQLConfigDataSource

### See Also

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

<b>Keyword</b>	<b>Description</b>
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 or Paradox driver, the name of the database file. For the Microsoft Excel 5 driver when accessing Excel 5.0 files, the name of the workbook file.
DEFAULTDIR	The path specification to the database file (for the Microsoft Access driver) or directory (for the 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) 21 (dBASEIII) 277 (dBASEIV) 533 (dBASEV) 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) 538 (Paradox5) 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; DBASE3, DBASE4, or DBASE5;

	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, 4.x, or 5.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.
MAXSCANROWS	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, <u>repairs a database</u> 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.

## 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 Returned Values](#)).

## SQLDriverConnect

### See Also

**SQLDriverConnect** enables you to connect to a driver without creating a data source (DSN).

The following keywords are supported in the connection string for all drivers: **DSN**, **DBQ**, and **FIL**.

For the **Microsoft Access** driver, the **UID** and **PWD** keywords are also supported. For the **Paradox** driver, the **PWD** keyword is also supported.

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 Returned 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.

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 **dBASE**, **Excel 3/4**, **Microsoft 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
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

## SQLExtendedFetch

The ODBC Desktop Database Drivers do not perform bookmark checking. If an application passes an invalid bookmark into **SQLExtendedFetch**, the results are undefined. **SQLExtendedFetch** does not always return SQLSTATE S1111 (Invalid Bookmark). It can also return SQLSTATE S1000 (General Error) with a message indicating a disk or network error.

When division-by-zero is performed in a SELECT statement, then **SQLExtendedFetch** is called, **SQLExtendedFetch** returns SQLSTATE 22012, "Division by zero", with SQL\_SUCCESS\_WITH\_INFO. For each row affected by the division-by-zero, SQL\_ROW\_ERROR is returned in the *rgfRowStatus* buffer to indicate the

error in retrieving data from the row. Rather than returning just SQLSTATE 22012, "Division by zero", **SQLExtendedFetch** should return SQLSTATE 01S01, "Error in Row", then SQLSTATE 22012, "Division by zero". Both should be returned with SQL\_SUCCESS\_WITH\_INFO.

## **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`. The returned value is a 16-bit integer that indicates how the driver directly treats files in a data source:

`SQL_FILE_NOT_SUPPORTED` = The driver is not a single-tier driver.

`SQL_FILE_TABLE` = A single-tier driver treats files in a data source as tables.

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

The ODBC driver returns `SQL_FILE_QUALIFIER` for the **Microsoft Access** driver, because each file is a complete database. The ODBC driver returns `SQL_FILE_TABLE` for the **dBASE, Microsoft Excel, Microsoft FoxPro, Paradox, or Text** driver, because each file is a table.



## **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.

## **SQLMoreResults**

This function always returns SQL\_NO\_DATA\_FOUND.

## **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

This function supports the SQL\_ACCESS\_MODE, SQL\_CURRENT\_QUALIFIER, SQL\_OPT\_TRACE, and SQL\_OPT\_TRACEFILE connection options.

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 SQL\_AUTOCOMMIT\_ON or SQL\_AUTOCOMMIT\_OFF, because the Microsoft Access driver supports transactions.

The **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.

## **SQLSetCursorName**

Because the driver does not support a positioned update or delete by the WHERE CURRENT OF *cursorname* syntax, **SQLSetCursorName** is supported, but cannot be used for positioned updates.

## **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.)

When the Paradox driver is used, performing a positioned update with **SQLSetPos** (with an *fOption* of SQL\_REFRESH, SQL\_UPDATE, SQL\_DELETE, or SQL\_ADD) against a table with no index will return SQLSTATE 42000, "Syntax error or access violation". An error message will indicate that the database cannot be updated because it is read-only. This error message will be returned even if the database is not read-only.



## **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 **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.

For **dBASE, Microsoft Excel 3.0/4.0, Microsoft FoxPro, Paradox,** or **Text** files, "TABLE" is the only table type supported.

When the **Microsoft Excel 3.0/4.0** driver is used, if you call **SQLTables** with a value for *szTableQualifier* that is not the name of an existing table, the driver will create a table with that name.

For **Microsoft Excel 5.0** files, "SYSTEM TABLE" is returned for sheet names (tables with a "\$" on the end), and "TABLE" is returned for tables within worksheets.

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 **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.

## **Arithmetic Errors**

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

## **Base Address of Drivers**

Each of the ODBC Desktop Database Drivers has been linked with a base address of 0x04c00000.



## Creating and Opening Tables

When the **Text** driver is used, a new table is created using the format specified in ODBCINST.INI. If not specified, tables are created in CSVDELIMITED format. By default, INTEGER columns default to 11 characters and FLOAT columns default to 22 characters. DATE columns use the YYYY-MM-DD format. CHAR and LONGCHAR columns are the width specified in the CREATE statement.

## Multiple *hstmts*

When the ODBC **Paradox** driver is used, if you want to use more than one *hstmt* to execute queries on a table, the table must have a unique index (Paradox primary key).

## 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.

## Setup DLL

The ODBC Administrator calls the function **ConfigDSN** when users configure data sources.

## Single-User Mode versus Multi-User Mode

The ODBC drivers can use a data source in single- or multi-user mode. The user specifies which mode to use in the ODBC Setup dialog box.

The Microsoft ODBC drivers allow only one user at a time to be connected to a single-user data source. When it uses a file in the data source, such as when it processes a SELECT or UPDATE statement, it exclusively locks the file.

The ODBC driver allows any number of users to be connected to a multi-user data source at the same time. It exclusively locks records before updating or deleting them. Other users cannot update or delete a locked record. Whether they can read the record depends on the version of the file.

For the **dBASE** driver, other users cannot read locked dBASE III records; they can read locked dBASE IV records ("dirty reads" are permitted).

For the **Microsoft FoxPro** driver, other users cannot read locked Microsoft FoxPro 2.0 records; they can read locked Microsoft FoxPro 2.5 or 2.6 records ("dirty reads" are permitted).

---

**Note** Although the ODBC driver locks records in multi-user mode, this does not imply that serializable transactions are available. Only the **Microsoft Access** driver supports transactions.

---

## **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.

