SQL92 Compliance (Advanced)

The ODBC Desktop Database Drivers and the underlying Microsoft Jet engine are not SQL92 compliant. They support many features that have been defined in SQL92. Some features supported in the driver are not supported in SQL92. For more information, see the *Microsoft Jet Database Engine Programmer's Guide*. The following are the major differences between the two:

- The SQL used by the Desktop Database Drivers supports more powerful expressions than those specified by SQL92.
- Different rules apply to the BETWEEN predicate.
- The SQL used by the Desktop Database Drivers and ANSI SQL supports different keywords.

The following SQL92 features are not supported by Microsoft Jet SQL:

- Security statements, such as GRANT and LOCK.
- DISTINCT with aggregate function references.

The following features are enhancements in the SQL used by the Desktop Database Drivers that are not specified by SQL92:

- The TRANSFORM statement providing support for crosstab queries.
- Additional aggregate functions (**StDev** and **VarP**).

Additional Supported ODBC SQL Grammar (Advanced)

The ODBC drivers completely support the Minimum ODBC 2.5 grammar and the following clauses in the Core and Extended ODBC 2.5 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)
- Outer joins (Extended)
- Procedure invocation (Extended)
- SELECT statement (Extended) (including GROUP BY, HAVING, and UNION clauses)
- Subqueries (SUBSELECTs) (Core)
- Topic

ALTER TABLE Statement (Core)

When the **Microsoft Access**, **dBASE**, 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.)

Note   When you use the **Paradox** driver without implementing the Borland Database Engine, ALTER TABLE statements are not supported; only read and append statements are allowed.

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 back quotes (`).

A parameterized query can be called using the previous 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*.

The semantics of this syntax are different for the Desktop Database Drivers and the Microsoft Jet engine. In Microsoft Jet SQL, *expression2* can be greater than *expression3* so that the statement will return TRUE only if *expression1* is greater than or equal to *expression3*, and *expression1* is less than or equal to *expression2*.

CREATE INDEX Statement

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** 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.

This statement is not supported by the Microsoft Excel or Text drivers.

Date Arithmetic

The driver supports adding and subtracting an integer from a DATE, TIME, or TIMESTAMP column. For a DATE column, the integer specifies the number of days to add or subtract. For a TIME or TIMESTAMP column, the integer specified the number of seconds to add or subtract.

Date, Time, and Timestamp Literals

For maximum interoperability, applications should pass date literals in the ODBC canonical format using escape-clause syntax:

- For date literals, {d '*value*'}, where *value* is in the form "yyyy-mm-dd"
- For time literals, {t '*value*'}, where *value* is in the form "hh:mm:ss"
- For timestamp literals {ts '*value*'}, where *value* is in the form "yyyy-mm-dd hh:mm:ss[.f…]".

DROP INDEX Statement

When the **Microsoft Access**, **dBASE**, 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 about OUTER JOINS, see the *Microsoft Jet Database Engine Programmer's Guide*.

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 F of the *Microsoft ODBC Programmer's Reference.*

Table Names

When the **dBASE**, **Microsoft Excel**, **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 file name 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

When the **dBASE** 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.

ALTER TABLE statements are not supported for the **Microsoft Excel** or **Text** driver.

Note　When you use the **Paradox** driver without implementing the Borland Database Engine, ALTER TABLE statements are not supported; only read and append statements are allowed.

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

When the **Microsoft Excel** driver is used, if column names are present, they must be in the first row. A name that in Microsoft Excel would use the "!" character must be enclosed in back quotes "`". The "!" character is converted to the "$" character, because the "!" character is not legal in an ODBC name, even when the name is enclosed in back quotes. All other valid Microsoft Excel characters (except the pipe character (|)) can be used in a column name, including spaces. A delimited identifier must be used for a Microsoft Excel column name to include a space. Unspecified column names will be replaced with driver-generated names, for example, "Col1" for the first column.

The pipe character (|) cannot be used in a column name, whether the name is enclosed in back quotes or not.

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

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** 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** 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. Only one record from a set with identical key values can be added to the index.

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 or when the Paradox driver is used without the implementation of the Borland Database Engine.

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** 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 Excel** driver is used on a **Microsoft Excel** 5.0, 7.0, or 97 data source, a worksheet cannot be created with the same name as a worksheet that was previously dropped. When the **Microsoft Excel** driver is used to access a version 5.0, 7.0, or 97 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.

The **dBASE** driver does not support packing a table to remove "deleted" values.

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**).

DROP INDEX Statement Limitations

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

DROP TABLE Statement Limitations

When the **Microsoft Excel** 5.0, 7.0, or 97 driver is used, the DROP TABLE statement clears the worksheet, but does not delete the worksheet name. Because the worksheet name still exists in the workbook, 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 <u>index</u> 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.

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**, **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, if an empty string is inserted into a column, the empty string is converted to a NULL; a searched SELECT statement that is executed with an empty string in the WHERE clause will not succeed on that column.

A table is not updatable by the **Paradox** driver under two conditions:

- When a unique index is not defined on the table. This is not true for an empty table, which can be updated with a single row even if a unique index is not defined on the table. If a single row is inserted in an empty table that does not have a unique index, an application cannot create a unique index or insert additional data after the single row has been inserted.
- If the Borland Database Engine is not implemented, only read and append statements are allowed on the Paradox 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. The Desktop Database Drivers support SQL92 LIKE pattern matching.

Use of an escape clause in a LIKE predicate is not supported.

A LIKE comparison should not be performed on a column containing data of a numeric or float data type. The results may be unpredictable. For more information, see the *Microsoft Jet Database Engine Programmer's Guide*.

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."

When the **Microsoft Excel** driver is used, if an empty string is inserted into a column, the empty string is converted to a NULL; a searched SELECT statement that is executed with an empty string in the WHERE clause will not succeed on that column.

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 SQL92 string constants (with single quotation marks, not double quotation marks) are supported.

The pipe character (|) cannot be used in a string, whether the character is enclosed in back quotes or not.

For maximum interoperability, applications should pass strings in parameters, rather than passing quoted strings.

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. If a name in Microsoft Excel includes the "!" character, it will automatically be translated to the "$" character instead.

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, 7.0, or 97 files.

When the **dBASE** 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 3.0** or **4.0**, **Paradox**, or **Text** driver is used, special MS-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). When you use the Paradox driver without implementing the Borland Database Engine, it is not possible to update a Paradox table.

Not supported by the **Text** driver.

When the **Microsoft Excel** driver is used, it is possible to update values, but a row cannot be deleted from a table based on a **Microsoft Excel** spreadsheet. As a result, the UPDATE statement is not considered officially supported by the **Microsoft Excel** driver. Only the INSERT statement is considered supported.

Views Limitations

Not supported by the **dBASE**, **Microsoft Excel**, **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)

The ODBC **dBASE** driver automatically opens and updates dBASE IV index files. You must use the Select Indexes dialog box displayed through 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).

CREATE INDEX for Paradox (Advanced)

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

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

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

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

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

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

- Before a non-unique index can be created for a table, a primary index must exist for that table.
- For Paradox 3.*x*, the *index-name* argument for any index other than a primary index (unique or non-unique) must be the same as the column name. For Paradox 4.*x* and 5.*x*, the name of such an index can be, but doesn't have to 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.

Note   When you use the Paradox driver without implementing the Borland Database Engine, only read and append statements are allowed.

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
BIGBINARY(1)	LONGBINARY	SQL_LONGVARBINARY
BINARY	BINARY	SQL_BINARY
BIT	BIT	SQL_BIT
COUNTER	COUNTER	SQL_INTEGER
CURRENCY	CURRENCY	SQL_NUMERIC
DATE/TIME	DATETIME	SQL_TIMESTAMP
GUID	GUID	SQL_GUID
LONG BINARY	LONGBINARY	SQL_LONGVARBINARY
LONG TEXT	LONGTEXT	SQL_LONGVARCHAR (2) SQL_WLONGVARCHAR(3)
MEMO	LONGTEXT	SQL_LONGVARCHAR (2) SQL_WLONGVARCHAR(3)
NUMBER (FieldSize= SINGLE)	SINGLE	SQL_REAL
NUMBER (FieldSize= DOUBLE)	DOUBLE	SQL_DOUBLE
NUMBER (FieldSize= BYTE)	UNSIGNED BYTE	SQL_TINYINT
NUMBER (FieldSize= INTEGER)	SHORT	SQL_SMALLINT
NUMBER (FieldSize= LONG INTEGER)	LONG	SQL_INTEGER
NUMERIC	NUMERIC	SQL_NUMERIC
OLE	LONGBINARY	SQL_LONGVARBINARY
TEXT	VARCHAR	SQL_VARCHAR (1) SQL_WVARCHAR(2)
VARBINARY	VARBINARY	SQL_VARBINARY

1 Access 4.0 applications only. Maximum length of 4000 bytes. Behavior similar to LONGBINARY.

2 ANSI applications only.

3 Unicode and Access 4.0 applications only.

Note   **SQLGetTypeInfo** returns ODBC data types. It will not return all Microsoft Access data types if more than one **Microsoft Access** type is mapped to the same ODBC SQL data type. All conversions in Appendix D of the *Microsoft ODBC Programmer's Reference* are supported for the SQL data types listed in the previous table.

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 510-byte column.
BYTE	Even though a **Microsoft Access** NUMBER field with a FieldSize equal to BYTE is unsigned, a negative number can be inserted into the field when using the Microsoft Access driver.
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 (').
	Procedures should be used to pass character data when using any special character in a character data type column.
DATE	Date values must be either delimited according to the ODBC canonical date format or delimited by the datetime delimiter ("#"). Otherwise, **Microsoft Access** will treat the value as an arithmetic expression and will not raise a warning or error.
	For example, the date "March 5, 1996" must be represented as {d '1996-03-05'} or #03/05/1996#; otherwise, if only 03/05/1993 is submitted, **Microsoft Access** will evaluate this as 3 divided by 5 divided by 1996. This value rounds up to the integer 0, and since the zero day maps to 1899-12-31, this is the date used.
	A pipe character (|) cannot be used in a date value, even if enclosed in back quotes.
GUID	Data type limited to Microsoft Access 4.0.
NUMERIC	Data type limited to Microsoft Access 4.0.

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.

dBASE data type	ODBC data type
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 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.

Data type	Description
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 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	The value in a LOGICAL column is returned in a SQL_C_CHAR buffer as either 0 or 1.
NUMBER	If an integer column is created, numbers that are too big for the integer data type can be entered, and data containing non-integer values can be inserted, with the result that the column may be converted to SQL_DOUBLE.
TEXT	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 datetime data types (DATE, TIME, and DATETIME), the ODBC **Microsoft Excel** driver assigns the SQL_TIMESTAMP data type to the column.
	Creating a TEXT column of zero or unspecified length actually returns a 255-byte column.
	A character string literal can contain any ANSI character (1-255 decimal). Use two consecutive single quotation marks (") to represent one single quotation mark (').
	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 SQL_VARCHAR.

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_LONGVARBINARY
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 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.
BYTES	If you insert NULL into a binary column with the Paradox5 driver, it is changed to 0.
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).
TIMESTAMP	When a value is inserted into a TIMESTAMP column by the **Paradox** driver, then subsequently retrieved from the column, the retrieved value may differ from the inserted value by as much as 1 second because of rounding.

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 in the previous table.

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.
	In delimited files, a CHAR column may or may not have double quotation mark delimiters at the beginning and the end; in fixed-length files, double quotation marks are not used as delimiters.
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)
	Mixed date separators are not allowed within a table.
	The Text ISAM formats a DATETIME field in the American or European format, depending upon the International setting in the Windows Control Panel.
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
	Note Decimal and scientific notation cannot be mixed in a column.
	NULL values are represented by a blank padded string in fixed-length files, and are omitted in delimited files.
	Float data may be padded with leading blanks.
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.
	The maximum width includes a sign. The maximum width of an INTEGER column is 11, although the width can be greater due to blanks that are allowed in fixed-format tables.
LONGCHAR	The theoretical limit on the width of a LONGCHAR column in either a fixed-length or delimited table is 65500K. The **Text** ISAM is more likely to provide reliable support up to about 32K.

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

Data Type Limitations (Advanced)

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

Data type	Description
All data types	Type conversion failures may 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.
NUMERIC	Supports maximum precision and a scale of 28
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 **dBASE**, **Microsoft Excel**, **Paradox**, or **Text** 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

Positive native error codes are generated by the Desktop Database Drivers. Negative error codes are generated by Microsoft Jet. For more information, see the *Microsoft Jet Database Engine Programmer's Guide*.

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:

Error message	Error location
[Microsoft][ODBC Driver Manager] *message-text*	Driver Manager (Odbc32.dll)
[Microsoft][ODBC *driver-name*] *message-text*	Driver ISAM (see Driver ISAMs)

SQLGetInfo Returned Values

The following table lists the C-language #defines for the *fInfoType* argument and the corresponding values returned by **SQLGetInfo**. This information can be retrieved by passing the listed C-language #defines to **SQLGetInfo** in the *fInfoType* argument. Where **SQLGetInfo** returns a 32-bit bitmask, a vertical bar (|) represents a bitwise OR. For more information about the values returned by **SQLGetInfo**, see the *Microsoft ODBC Programmer's Reference, Version 2.0* or later.

- » SQL_ACCESSIBLE_PROCEDURES
- » SQL_ACCESSIBLE_TABLES
- » SQL_ACTIVE_CONNECTIONS
- » SQL_ACTIVE_ENVIRONMENTS
- » SQL_ACTIVE_STATEMENTS
- » SQL_AGGREGATE_FUNCTIONS
- » SQL_ALTER_DOMAIN
- » SQL_ALTER_TABLE
- » SQL_ASYNC_MODE
- » SQL_BATCH_ROW_COUNT
- » SQL_BATCH_SUPPORT
- » SQL_BOOKMARK_PERSISTENCE
- » SQL_CATALOG_NAME
- » SQL_COLLATION_SEQ
- » 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_CREATE_ASSERTION
- » SQL_CREATE_CHARACTER_SET
- » SQL_CREATE_COLLATION
- » SQL_CREATE_DOMAIN
- » SQL_CREATE_SCHEMA
- » SQL_CREATE_TABLE
- » SQL_CREATE_TRANSLATION
- » SQL_CREATE_VIEW
- » SQL_CURSOR_COMMIT_BEHAVIOR

- SQL_CURSOR_ROLLBACK_BEHAVIOR
- SQL_CURSOR_SENSITIVITY
- SQL_DATA_SOURCE_NAME
- SQL_DATA_SOURCE_READ_ONLY
- SQL_DATABASE_NAME
- SQL_DATETIME_LITERALS
- SQL_DBMS_NAME
- SQL_DBMS_VER
- SQL_DDL_INDEX
- SQL_DEFAULT_TXN_ISOLATION
- SQL_DESCRIBE_PARAMETER
- SQL_DRIVER_HDBC
- SQL_DRIVER_HENV
- SQL_DRIVER_HLIB
- SQL_DRIVER_HSTMT
- SQL_DRIVER_NAME
- SQL_DRIVER_ODBC_VER
- SQL_DRIVER_VER
- SQL_DROP_ASSERTION
- SQL_DROP_CHARACTER_SET
- SQL_DROP_COLLATION
- SQL_DROP_DOMAIN
- SQL_DROP_SCHEMA
- SQL_DROP_TABLE
- SQL_DROP_TRANSLATION
- SQL_DROP_VIEW
- SQL_EXPRESSIONS_IN_ORDERBY
- SQL_FETCH_DIRECTION
- SQL_FILE_USAGE
- SQL_FORWARD_ONLY_CURSOR_ATTRIBUTES1
- 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, Paradox, Text)

SQL_ACCESSIBLE_TABLES

"Y"

SQL_ACTIVE_CONNECTIONS

64

SQL_ACTIVE_ENVIRONMENTS

0

SQL_ACTIVE_STATEMENTS

0

SQL_AGGREGATE_FUNCTIONS

All set

SQL_ALTER_DOMAIN

0

(Microsoft Access)

SQL_ALTER_TABLE

SQL_AT_ADD_COLUMN |
SQL_AT_DROP_COLUMN
(dBASE, Paradox)

0
(Microsoft Access, Microsoft Excel, Text)

SQL_ASYNC_MODE

0
(Microsoft Access)

SQL_BATCH_ROW_COUNT

0
(Microsoft Access)

SQL_BATCH_SUPPORT

0
(Microsoft Access)

SQL_BOOKMARK_PERSISTENCE

SQL_BP_SCROLL |
SQL_BP_UPDATE (1)

1 Bookmarks persist after a commit, but do not persist after a rollback.

SQL_CATALOG_NAME

"Y"

SQL_COLLATION_SEQ

Empty string

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_CVT_WVARCHAR

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_CVT_WVARCHAR

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_CVT_WVARCHAR

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_CVT_WVARCHAR

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_CVT_WVARCHAR

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_CVT_WVARCHAR

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_CVT_WVARCHAR

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_CVT_WVARCHAR

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_CVT_WVARCHAR

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_CVT_WVARCHAR

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_CVT_WVARCHAR

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_CVT_WVARCHAR

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_CVT_WVARCHAR

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_CVT_WVARCHAR

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_CVT_WVARCHAR

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_CVT_WVARCHAR

SQL_CORRELATION_NAME

SQL_CN_DIFFERENT

SQL_CREATE_ASSERTION

0
(Microsoft Access)

SQL_CREATE_CHARACTER_SET

0
(Microsoft Access)

SQL_CREATE_COLLATION

0
(Microsoft Access)

SQL_CREATE_DOMAIN

0
(Microsoft Access)

SQL_CREATE_SCHEMA

0
(Microsoft Access)

SQL_CREATE_TABLE

TABLE
(Microsoft Access)

SQL_CREATE_TRANSLATION

0
(Microsoft Access)

SQL_CREATE_VIEW

SQL_CV_CREATE_VIEW
(Microsoft Access)

SQL_CURSOR_COMMIT_BEHAVIOR

SQL_CB_CLOSE

SQL_CURSOR_ROLLBACK_BEHAVIOR

SQL_CB_CLOSE

SQL_CURSOR_SENSITIVITY

SQL_UNSPECIFIED

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, Paradox)

"Y"
(Microsoft Excel, Text)

SQL_DATABASE_NAME

File name (Microsoft Access, Microsoft Excel 5.0/7.0//97)

Current database directory (dBASE, Microsoft Excel 3.0/4.0, Paradox, Text)

SQL_DATETIME_LITERALS

0

(Microsoft Access)

SQL_DBMS_NAME

"ACCESS"   (Microsoft Access)
"DBASE"   (dBASE)
"EXCEL"   (Microsoft Excel)
"PARADOX"   (Paradox)
"TEXT"   (Text)

SQL_DBMS_VER

ISAM	Version	Format of version numbers	
Microsoft Access	2.0	02.00.0000	
	3.0	03.00.0000	
	3.5	03.50.0000	
	4.0	04.00.0000	
		(Versions 1.0 and 1.1 are not supported.)	
		(There is no difference in the data format in Microsoft Access versions 3.0, 7.0, and 97.)	
dBASE	3.0	03.0n.nnnn	
	4.0	04.0n.nnnn	
	5.0	05.0n.nnnn	
Microsoft Excel	3.0	03.0n.nnnn	
	4.0	04.0n.nnnn	
	5.0	05.0n.nnnn	
	7.0, 97	07.0n.nnnn	
Paradox	3.x	03.nn.nnnn	
	4.x	04.nn.nnnn	
	5.x	05.nn.nnnn	
Text	1.0	01.0n.nnnn	

SQL_DDL_INDEX

CREATE, DROP
(Microsoft Access)

SQL_DEFAULT_TXN_ISOLATION

SQL_TXN_READ_COMMITTED [Microsoft Access]

0 [dBASE, Microsoft Excel, Paradox, Text]

SQL_DESCRIBE_PARAMETER

0
(Microsoft Access)

SQL_DRIVER_HDBC

Handled by the Driver Manager.

SQL_DRIVER_HENV

Handled by the Driver Manager.

SQL_DRIVER_HLIB

Handled by the Driver Manager.

SQL_DRIVER_HSTMT

Handled by the Driver Manager.

SQL_DRIVER_NAME

"OdbcJt32.dll"

SQL_DRIVER_ODBC_VER

"3.51.0000"

SQL_DRIVER_VER

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

SQL_DROP_ASSERTION

0
(Microsoft Access)

SQL_DROP_CHARACTERSET

0
(Microsoft Access)

SQL_DROP_COLLATION

0
(Microsoft Access)

SQL_DROP_DOMAIN

0
(Microsoft Access)

SQL_DROP_SCHEMA

0
(Microsoft Access)

SQL_DROP_TABLE

SQL_DT_DROP_TABLE
(Microsoft Access)

SQL_DROP_TRANSLATION

0
(Microsoft Access)

SQL_DROP_VIEW

SQL_DV_DROP_VIEW
(Microsoft Access)

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 Excel 5.0, 7.0, or 97)

SQL_FILE_TABLE
(dBASE, Microsoft Excel 3.0 or 4.0, Paradox, Text)

SQL_FORWARD_ONLY_CURSOR_ATTRIBUTES1

NEXT, LOCK_NO_CHANGE, POS_POSITION, POS_UPDATE
(Microsoft Access)

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
(Microsoft Access, Microsoft Excel, Paradox, Text)

SQL_IC_UPPER (1)
(dBASE)

1 The qualifier is returned in mixed case so that Windows NT can locate the directory.

SQL_IDENTIFIER_QUOTE_CHAR

"`" (back quote)

SQL_KEYWORDS

ALPHANUMERIC, AUTOINCREMENT, BINARY, BOOLEAN, BYTE, COUNTER, CURRENCY, DATABASE, DATABASENAME, DATETIME, DISALLOW, DISTINCTROW, DOUBLEFLOAT, FLOAT4, FLOAT8, GENERAL, IEEEDOUBLE, IEEESINGLE, IGNORE, IMAGE, INTEGER1, INTEGER2, INTEGER4, LOGICAL, LOGICAL1, LONG, LONGBINARY, LONGCHAR, LONGTEXT, MEMO, MONEY, NOTE, NUMBER, OLEOBJECT, OWNERACCESS, PARAMETERS, PERCENT, PIVOT, SHORT, SINGLE, SINGLEFLOAT, STDEV, STDEVP, STRING, TABLEID, TEXT, TOP, TRANSFORM, UNSIGNEDBYTE, VAR, VARBINARY, VARP, YESNO

SQL_LIKE_ESCAPE_CLAUSE

"N"

SQL_LOCK_TYPES

SQL_LCK_NO_CHANGE

SQL_MAX_BINARY_LITERAL_LEN

255

SQL_MAX_CHAR_LITERAL_LEN

255   (Microsoft Access)
254   (dBASE)
255   (Microsoft Excel 3.0, 4.0, 5.0, or 7.0)
65535 (Microsoft Excel 97)
255   (Paradox)
255   (Text)

SQL_MAX_COLUMN_NAME_LEN

64   (Microsoft Access)
10   (dBASE)
64   (Microsoft Excel 5.0, 7.0, or 97)
30   (Microsoft Excel 3.0 or 4.0)
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)   (Paradox)
0   (Text)

SQL_MAX_COLUMNS_IN_ORDER_BY

10

SQL_MAX_COLUMNS_IN_SELECT

255

SQL_MAX_COLUMNS_IN_TABLE

255

When using the **Microsoft Excel** driver, a CREATE TABLE statement may allow 256 columns, but the 255 column limit is still valid, and an insert into column number 256 will fail.

SQL_MAX_CURSOR_NAME_LEN

64

SQL_MAX_INDEX_SIZE

255   (Microsoft Access)
220   (dBASE)
0   (Microsoft Excel)
1350   (Paradox)
0   (Text)

SQL_MAX_OWNER_NAME_LEN

0

SQL_MAX_PROCEDURE_NAME_LEN

64   (Microsoft Access)
0   (dBASE)
0   (Microsoft Excel)
0   (Paradox)
0   (Text)

SQL_MAX_QUALIFIER_NAME_LEN

66

SQL_MAX_ROW_SIZE

2096   (Microsoft Access)
4000   (dBASE)
65535   (Microsoft Excel)
1350   (Paradox)
65535   (Text)

SQL_MAX_ROW_SIZE_INCLUDES_LONG

"N"   (Microsoft Access)
"N"   (dBASE)
"Y"   (Microsoft Excel)
"N"   (Paradox)
"Y"   (Text)

SQL_MAX_STATEMENT_LEN

65000

SQL_MAX_TABLE_NAME_LEN

64   (Microsoft Access)
12   (dBASE)
31   (Microsoft Excel 5.0, 7.0, or 97)
12   (Microsoft Excel 3.0 or 4.0)
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_NON_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_ 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, Paradox)

SQL_POS_POSITION |
SQL_POS_REFRESH
(Microsoft Excel, Text)

SQL_POSITIONED_STATEMENTS

0

SQL_PROCEDURE_TERM

"QUERY"
(Microsoft Access)

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

SQL_PROCEDURES

"Y"
(Microsoft Access)

"N"
(dBASE, Microsoft Excel, Paradox, Text)

SQL_QUALIFIER_LOCATION

SQL_QL_START

SQL_QUALIFIER_NAME_SEPARATOR

"."
(Microsoft Access, Microsoft Excel 5.0, 7.0, or 97)

"\"
(dBASE, Microsoft Excel 3.0 or 4.0, Paradox, Text)

SQL_QUALIFIER_TERM

"Database"   (Microsoft Access)
"Directory"   (dBASE)
"Workbook"   (Microsoft Excel 5.0, 7.0, or 97)/
"Directory" (Microsoft Excel 3.0 or 4.0)
"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, 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, 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)
"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_LOCATE_2
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, Paradox)

SQL_TXN_CAPABLE

SQL_TC_ALL
(Microsoft Access)

SQL_TC_NONE
(dBASE, Microsoft Excel, Paradox, Text)

SQL_TXN_ISOLATION_OPTION

SQL_TXN_READ_COMMITTED
(Microsoft Access)

0
(dBASE, Microsoft Excel, Paradox, Text)

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** 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

Attribute	Comments
SQL_COLUMN_DISPLAY_SIZE	For LONGVARBINARY data, SQL_COLUMN_DISPLAY_SIZE is the maximum length of the column, not the maximum length of the column times 2.
SQL_COLUMN_MONEY	FALSE is returned for a column of **Paradox** data type CURRENCY, which maps to the ODBC SQL data type SQL_DOUBLE.
SQL_OWNER_NAME	An empty string ("") is returned in this column, since owner name is not supported.
SQL_QUALIFIER_NAME	The path to a database file is returned for **Microsoft Access**; the path to a directory is returned for **dBASE**, **Microsoft Excel**, **Paradox**, or **Text**.
SQL_COLUMN_SEARCHABLE	LONGVARBINARY and LONGVARCHAR columns are reported as SQL_UNSEARCHABLE. Fixed-length and variable-length binary and character data types are seachable, even though LONGVARBINARY and LONGVARCHAR are not.

Note   The above is not a complete list of the attributes returned by **SQLColAttributes**.

SQLColumns

Column	Comments
TABLE_QUALIFIER	The path to a database file is returned for **Microsoft Access**; the path to a directory is returned for **dBASE**, **Microsoft Excel**, **Paradox**, or **Text**.
TABLE_OWNER	NULL is returned in this column, since owner name is not supported.
NULLABLE	SQL_NO_NULLS is returned for columns that participate in a primary key or unique index.

SQLConfigDataSource

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

Keyword	Description
CHARACTERSET	For the **Text** driver, OEM or ANSI.
COLLATINGSEQUENCE	The sequence in which the fields are sorted.
	When the **dBASE** driver is used, the sequence can be: ASCII (the default) or International.
	When the **Paradox** driver is used, the sequence can be: ASCII (default), International, Swedish-Finnish, or Norwegian-Danish.
	This sets the same option as the **Collating Sequence** control in the setup dialog box.
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. Has the following format: COMPACT_DB=<path_name><optionaL_sort_order>< optional ENCRYPT keyword>.
CREATE_DB	For the **Microsoft Access** driver, creates a database file. Has the following format: CREATE_DB=<path_name><optional_sort-order><optional_ENCRYPT keyword>, where the path name is the full path to a Microsoft Access database. An error will be returned if the path name specifies an existing database. The sort order will be as set up in the New Database dialog box displayed when the Create button is pressed in the Microsoft Access Setup dialog box. If no sort order is specified, General is used.
	When using the CREATE_DB keyword, if the pathname of the Microsoft Access database to be created contains one or more spaces, then the entire pathname must be enclosed by double quotation marks, as shown in the following examples:
	"C:\PROGRAM FILES\COMMON FILES\ MyAccess.mdb"
	"C:\PROGRAM FILES\Access2.mdb"
	CREATE_DB=C:\TEMP\test.mdb (no quotation marks needed)
CREATE_SYSDB	For the **Microsoft Access** driver, creates a system database file. Has the following format: CREATE_SYSDB=<path-name><optional-sort-order>, where the path name is the full path to a Microsoft Access database. An error will be returned if the path name specifies an existing database. The sort order will be as set up in the New Database dialog box

	displayed when the Create button is pressed in the Microsoft Access Setup dialog box. If no sort order is specified, General is used.
CREATE_V2DB	For the **Microsoft Access** driver, creates a database file that is compatible with Microsoft Access 2.0. Has the following format: CREATE_V2DB=<path-name><optional-sort-order>, where the path name is the full path to a Microsoft Access database. An error will be returned if the path name specifies an existing database. The sort order will be as set up in the New Database dialog box displayed when the Create button is pressed in the Microsoft Access Setup dialog box. If no sort order is specified, General is used.
	When using the CREATE_V2DB keyword, if the pathname of the Microsoft Access database to be created contains one or more spaces, then the entire pathname must be enclosed by double quotation marks, as shown in the following examples:
	"C:\PROGRAM FILES\COMMON FILES\MyAccess.mdb"
	"C:\PROGRAM FILES\Access2.mdb"
	CREATE_V2DB=C:\TEMP\test.mdb (no quotation marks needed)
DBQ	For the **Microsoft Access** or **Paradox** driver, the name of the database file.
	For the **Microsoft Excel** driver when accessing **Microsoft Excel** 5.0, 7.0, or 97 files, the name of the workbook file.
	This sets the same option as the **Database** control in the setup dialog box.
DEFAULTDIR	The path specification to the database file (for the **Microsoft Access** driver) or directory (for the **dBASE**, **Microsoft Excel**, **Paradox**, or **Text** driver).
	This sets the same option as the **Select Directory control** in the setup dialog box, or for **Microsoft Excel**, the **Select Workbook** control in the setup dialog box.
DELETED	For the **dBASE** driver, specifies whether or not rows that have been marked as deleted can be retrieved or positioned on. If set to 1, deleted rows are not displayed; if set to 0, deleted rows are treated the same as non-deleted rows. The default is unchecked.
	This sets the same option as the **Show Deleted Rows** control in the setup dialog box.
DESCRIPTION	A description of the data in the data source.
	This sets the same option as the **Description** control in the setup dialog box.
DRIVER	The path specification to the driver DLL.
DRIVERID	An integer ID for the driver.
	25 (Microsoft Access)
	21 (dBASE III)
	277 (dBASE IV)

533 (dBASE 5.0)
534 (Microsoft Excel 3.0)
278 (Microsoft Excel 4.0)
22 (Microsoft Excel 5.0/7.0)
790 (Microsoft Excel 97)
26 (Paradox 3.x)
282 (Paradox 4.x)
538 (Paradox 5.x)
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.)
	This sets the same option as the **Exclusive** control in the setup dialog box.
EXTENSIONS	Lists the file name extensions of the **Text** files on the data source.
	This sets the same option as the **Extensions List** control in the setup dialog box.

FIL	File type
	    MS Access for Microsoft Access
	    dBase III, dBase IV, or dBase 5
	    Excel 3.0, 4.0, 5.0, 7.0, or 97 for Microsoft Excel
	    Paradox 3.x, 4.x, or 5.x)
	    Text.

FILETYPE	File type for the Text driver (Text).
FIRSTROWHASNAMES	For the **Microsoft Excel** driver, indicates whether the cells of the first row of the range contain the column names for the table (1) or not (0).
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.
IMPLICITCOMMITSYNC	Determines whether the **Microsoft Access** driver will perform internal or implicit commits asynchronously. This value is initially set to "Yes", which means that the **Microsoft Access** driver will wait for commits in an internal/implicit transaction to be completed.
	The value of this option should not be changed without careful consideration of the consequences. For more information about the option, see the *Microsoft Jet Database Engine Programmer's Guide*.
	This sets the same option as the **ImplicitCommitSync** control in the setup dialog box.
MAXBUFFERSIZE	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 512K (displayed as 512). Any integer value divisible by 256 can be used. This sets the same option as the **Buffer Size** control in

	the setup dialog box.
MAXSCANROWS	For the **Microsoft Access**, **Microsoft Excel**, or **Text** driver, the number of rows to be scanned when setting a column's data type based upon existing data.
	A number from 1 to 16 can be entered for the rows to scan. The value defaults to 8; if it is set to 0, all rows are scanned. (A number outside the limit will return an error.)
	For the **Text** driver, you can enter a number from 1 to 32767 for the number of rows to scan; however, the value will always default to 25. (A number outside the limit will return an error.)
	This sets the same option as the Rows to **Scan Control** in the setup dialog box.
PAGETIMEOUT	Specifies the period of time, in tenths of a second, that a page (if not used) remains in the buffer before being removed. For the **Microsoft Access** driver, the default is 5 tenths of a second (0.5 seconds). For the **dBASE**, and **Paradox** drivers, the default is 600 tenths of a second (60 seconds). Note that this option applies to all data sources that use the ODBC driver.
	This sets the same option as the **Page Timeout** control in the setup dialog box.
PARADOXNETPATH	For the **Paradox** driver, the full path of the directory containing a Paradox lock database, because it contains either the PDOXUSRS.net file (in Paradox 4.*x*) or the PARADOX.net file (in Paradox 5.*x*). If the directory does not contain one of these files, the **Paradox** driver creates one. For information about these files, see the Paradox documentation.
	Before a network directory can be selected, a Paradox user name must be entered.
	This sets the same option as the **Select Network** Directory control in the Paradox setup dialog box.
PARADOXNETSTYLE	For the **Paradox** driver, the network access style to use when accessing Paradox data: either "3.x" for Paradox 3.*x* or "4.x" for Paradox 4.*x* or 5.*x*. Can be set to "3.x" or "4.x" if the version is Paradox 4.*x* or 5.*x*; if the version is Paradox 3.*x*, the style must be "3.x".
	This sets the same option as the **Net Style** control in the Paradox setup dialog box.
PARADOXUSERNAME	For the **Paradox** driver, the Paradox user name.
	This sets the same option as the **User Name** control in the Paradox setup dialog box.
PWD	For the **Microsoft Access** or **Paradox** driver, the password.
	For the **Paradox** driver, this is an optional keyword and will never be written to the file by the driver. It is used in a call to **SQLDriverConnect** against password-secured Paradox files.
READONLY	TRUE to make file read-only; FALSE to make file not read-only.

	This sets the same option as the **Read-Only** control in the setup dialog box.
REPAIR_DB	For the **Microsoft Access** driver, <u>repairs a database</u> damaged by a failure that occurs during the commit process.
STATISTICS	For the **dBASE** driver, determines whether table size statistics are approximated. Note that this option applies to all data sources that use the ODBC driver.
	This sets the same option as the **Approximate Row Count** control in the setup dialog box.
SYSTEMDB	For the **Microsoft Access** driver, the path specification to system database file.
	This sets the same option as the **System Database** control in the setup dialog box.
THREADS	The number of background threads for the engine to use. For the **Microsoft Access** driver, this value defaults to 3, but can be changed. For the **dBASE**, **Microsoft Excel**, **Paradox**, and **Text** drivers, this value is 3, and cannot be changed.
	This sets the same option as the **Threads** control in the setup dialog box.
UID	For the **Microsoft Access** driver, the user ID name used for login.
USERCOMMITSYNC	Determines whether the **Microsoft Access** driver will perform user-defined transactions asynchronously. This value is initially set to "Yes", which means that the **Microsoft Access** driver will wait for commits in a user-defined transaction to be completed.
	The value of this option should not be changed without careful consideration of the consequences. For more information about the option, see the the *Microsoft Jet Database Engine Programmer's Guide*.
	This sets the same option as the **UserCommitSync** control in the setup dialog box.

SQLDriverConnect

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**.

Note   If DBQ or DefaultDir is not specified for the **dBASE**, **Microsoft Excel 3.0/4.0**, **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 Excel 3.0 or 4.0	Driver, DriverID	Driver={Microsoft Excel Driver (*.xls)}; DBQ=c:\temp; DriverID=278
Microsoft Excel 5.0/7.0	Driver, DriverID, DBQ	Driver={Microsoft Excel Driver (*.xls)}; DBQ=c:\temp\sample.xls; DriverID=22
Microsoft Excel 97	Driver, DriverID, DBQ	Driver={Microsoft Excel Driver (*.xls)}; DBQ=c:\temp\sample.xls; DriverID=790
Paradox	Driver, DriverID	Driver={Microsoft Paradox Driver (*.db)}; DBQ=c:\temp;DriverID=26
Text	Driver	Driver={Microsoft Text Driver (*.txt;*.csv)}; DefaultDir=c:\temp

SQLGetCursorName

SQLGetCursorName is supported, but can only be used when the Cursor Library is used, because positioned operations are not supported in the driver.

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 the SQL_FILE_USAGE information type. 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, Paradox,** or **Text** driver, because each file is a table.

SQLGetStmtOption

The bookmarks returned by an *fOption* of SQL_GETBOOKMARK are only valid while the query is open and are invalidated when the query is reissued. Persistent bookmarks are not supported.

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.

SQL_ALL_EXCEPT_LIKE will be returned in the SEARCHABLE column for the Byte, Counter, Double, Single, Long, and Short data types. (The LIKE capability can be achieved by converting the value to a character using the ODBC canonical conversion functions, then performing the comparison.)

When the **Microsoft Excel** driver is used, the ODBC type names are returned in the TYPE_NAME column that is returned by **SQLGetTypeInfo**.

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.

Application developers should look for driver-defined columns starting at the end of the result set and proceeding backward.

Column	Comments
COLUMN_TYPE	SQL_PARAM_INPUT or SQL_RESULT_COL
ORDINAL	This is a driver-specific column that is returned at the end of the result set. The SQL type of the column is an integer.

SQLProcedures

SQLProcedures will only return rows for those procedures that have at least one argument. Procedures that have no arguments are treated as views.

Column	Comments
PROCEDURE_QUALIFIER	The path to the database file.
PROCEDURE_OWNER	NULL
PROCEDURE_NAME	Undelimited procedure name
PROCEDURE_TYPE	SQL_PT_PROCEDURE

SQLSetConnectOption

fOption	Comment
SQL_ACCESS_MODE	The SQL_ACCESS_MODE *fOption* can be set to either SQL_MODE_READ_ONLY or SQL_MODE_READ_WRTE. However, the driver does not prevent updates if SQL_ACCESS_MODE is set to SQL_MODE_READ_ONLY.
SQL_AUTOCOMMIT	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(1).
	The **dBASE**, **Microsoft Excel**, **Paradox**, and **Text** drivers only support SQL_AUTOCOMMIT being set to ON (the default state), because they do not support transactions.
SQL_CURRENT_QUALIFIER	Supported.
SQL_LOGIN_TIMEOUT	Not supported.
SQL_OPT_TRACE	Supported.
SQL_OPT_TRACEFILE	Supported.
SQL_PACKET_SIZE	Not supported.
SQL_QUIET_MODE	Not supported.
SQL_TRANSLATE_DLL	Not supported.
SQL_TRANSLATION_OPTION	Not supported.
SQL_TXN_ISOLATION	When the **Microsoft Access** driver is used, SQL_TXN_ISOLATION is always SQL_TXN_READ_COMMITTED. This *fOption* is not supported for the other drivers.

1 Atomic transactions are not supported by the **Microsoft Access** driver. When committing a transaction using the **Microsoft Access** driver, a finite delay exists between the time the transaction is committed and the time the values are written to disk. This delay is determined by a delay inherent in the Microsoft Jet engine. The page timeout will not be less than a minimum value, even if the PageTimeout option is set below that value. As a result, there is no guarantee that committed data is stable, since changes may be made during the delay.

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. It can only be used when the Cursor Library is enabled and the application is using **SQLExtendedFetch**.

SQLSetPos

The bulk-model semantics for **SQLSetPos** calls with the *irow* argument equal to 0 are supported.

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, see the *Microsoft Jet Database Engine Programmer's Guide*.)

SQLSetScrollOptions

Forward and static cursors are supported for SQL_CONCUR_READ_ONLY.

Only keyset-driven cursors are supported for an *fConcurrency* argument of SQL_CONCUR_LOCK.

An *fConcurrency* argument of SQL_CONCUR_ROWVER is not supported.

Dynamic cursors and mixed cursors are not supported.

SQLSetStmtOption

fOption	Comments
SQL_ASYNC_ENABLE	Asynchronous processing is not supported. The SQL_ASYNC_ENABLE *fOption* will return SQLSTATE S1C00 (Driver not capable).
SQL_KEYSET_SIZE	The only valid keyset size is 0, because mixed and dynamic cursors are not supported. If this value is set to any other number, it will be changed to 0 and the call will return SQL_SUCCESS_WITH_INFO and SQLSTATE 01S02 (Option value changed).
SQL_MAX_ROWS	The only valid rowset size is 0, because the Desktop Database Drivers do not support limiting the number of rows that are returned. If this value is set to any other number, it will be changed to 0 and the call will return SQL_SUCCESS_WITH_INFO and SQLSTATE 01S02 (Option value changed).
SQL_QUERY_TIMEOUT	Not supported.
SQL_ROW_NUMBER	Not supported.
SQL_SIMULATE_CURSOR	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_CURROW.

Pattern matching is not supported for either the *szTableQualifier* or *szTableName* argument.

SQLStatistics

Column	Comments
TABLE_QUALIFIER	The path to a database file is returned for Microsoft Access; the path to a directory is returned for dBASE, Microsoft Excel, or Text. Pattern matching is not supported in the *szTableQualifier* argument.
TABLE_OWNER	NULL is returned in this column, since owner name is not supported.
TABLE_NAME	Undelimited table name. Pattern matching is not supported in the *szTableName* argument.
INDEX_QUALIFIER	NULL is always returned.
INDEX_NAME	Index-dependent.
TYPE	Only SQL_TABLE_STAT or SQL_INDEX_OTHER will be returned for TYPE.
SEQ_IN_INDEX	Index-dependent.
COLUMN_NAME	Index-dependent.
COLLATION	Index-dependent.
CARDINALITY	Returned for Microsoft Access only.
PAGES	NULL is always returned.

Filtering is based on uniqueness (the *fUnique* argument). The *fAccuracy* parameter is ignored.

SQLTables

Argument	Comments
szTableOwner	The only valid argument for *szTableOwner* is NULL, since none of the drivers support owner names. With *szTableOwner* set to NULL, all tables are returned. NULL is returned in the TABLE_OWNER column.
szTableQualifier	When the **Microsoft Excel 3.0** or **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.
	In the TABLE_QUALIFIER column, **SQLTables** will return either the path to a database file (for Microsoft Access or Microsoft Excel 5.0, 7.0, or 97) or the path to a directory (dBASE, Microsoft Excel 3.0 or 4.0, Paradox, or Text).
SzTableType	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** or **4.0**, **Paradox**, or **Text** files, "TABLE" is the only table type supported.
	For **Microsoft Excel** 5.0, 7.0, or 97 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, SQL_COMMIT and SQL_ROLLBACK are supported for the *fType* argument in a call to **SQLTransact**.

When the **dBASE**, **Microsoft Excel**, **Paradox**, or **Text** driver is used, this function supports SQL_COMMIT, but not SQL_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 0x04c0000.

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.

When an application performs a Save As command on Microsoft Excel data through the **Microsoft Excel** driver, the application should create a new table and insert the data to be saved into the new table. Inserts result in an append to the table. No other operations can be performed on the table until it is closed and reopened. Once the table is closed, no subsequent insert can be performed, since the table is then a read-only table.

It is possible to update values when using the **Microsoft Excel** driver, but a row cannot be deleted from a table based on a Microsoft Excel spreadsheet, so updates are not considered officially supported by the **Microsoft Excel** driver.

Setup DLL

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

Translation DLLs

Not supported.

Descriptors

A descriptor is a data structure that holds information about either column data or dynamic parameters. **SQLGetDescField** can be used to retrieve the supported descriptors listed below. Implementation Parameter Descriptors (IPD) are not automatically populated, since **SQLDescribeParam** is not supported. Descriptor fields that are not available through Jet (such as SQL_DESC_BASE_TABLE_NAME) are also not supported.

For more information about Jet-supported descriptor fields, see the *Microsoft Jet Database Engine Programmer's Guide*.

For more information about descriptors, see the topics under "Descriptors" in the *ODBC Programmer's Reference*.

Descriptor Fields	Support Level
SQL_DESC_ALLOC_TYPE	Supported
SQL_DESC_ARRAY_SIZE	Supported only for ARD
SQL_DESC_ARRAY_STATUS_PTR	Supported
SQL_DESC_BIND_OFFSET_PTR	Supported
SQL_DESC_BIND_TYPE	Supported
SQL_DESC_COUNT	Supported
SQL_DESC_ROWS_PROCESSED_PTR	Supported only for ARD
SQL_DESC_AUTO_UNIQUE_VALUE	Supported
SQL_DESC_BASE_COLUMN_NAME	Supported (NEW)
SQL_DESC_BASE_TABLE_NAME	Not supported (Refer to the release notes for current status.)
SQL_DESC_CASE_SENSITIVE	Always FALSE
SQL_DESC_CATALOG_NAME	Not supported (Refer to the release notes for current status.)
SQL_DESC_CONCISE_TYPE	Supported
SQL_DESC_DATA_PTR	Supported
SQL_DESC_DATETIME_INTERVAL_CODE	Supported
SQL_DESC_DATETIME_INTERVAL_PRECISION	Supported for INTERVAL C types
SQL_DESC_DISPLAY_SIZE	Supported
SQL_DESC_FIXED_PREC_SCALE	Supported (TRUE for money)
SQL_DESC_INDICATOR_PTR	Supported
SQL_DESC_LABEL	Supported
SQL_DESC_LENGTH	Supported
SQL_DESC_LITERAL_PREFIX	Supported
SQL_DESC_LITERAL_SUFFIX	Supported
SQL_DESC_LOCAL_TYPE_NAME	Not supported (returns EMPTY string)
SQL_DESC_NAME	Supported
SQL_DESC_NULLABLE	Supported **Note** Unsupported in versions preceding Jet 4.0
SQL_DESC_NUM_PREC_RADIX	Supported
SQL_DESC_OCTET_LENGTH	Supported

SQL_DESC_OCTET_LENGTH_PTR	Supported
SQL_DESC_PARAMETER_TYPE	Only input parameters
SQL_DESC_PRECISION	Supported
SQL_DESC_SCALE	Supported
SQL_DESC_SCHEMA_NAME	Not supported
SQL_DESC_SEARCHABLE	Supported
SQL_DESC_TABLE_NAME	Not supported
SQL_DESC_TYPE	Supported
SQL_DESC_TYPE_NAME	Supported
SQL_DESC_UNNAMED	Supported
SQL_DESC_UNSIGNED	Supported
SQL_DESC_UPDATABLE	Supported

Compatibility

Unicode is a method of software character encoding that treats all characters as having a fixed width of two bytes. This method is used as an alternative to Windows ANSI character encoding, which, because it represents characters in one byte, is limited to 256 characters. Because Unicode can represent over 65,000 characters, it accommodates many languages whose characters are not represented in ANSI encoding.

The ODBC 3.5 (or later) Driver Manager is Unicode-enabled. This affects two major areas: function calls and string data types. The Driver Manager maps function string arguments and string data as required by the application and driver, both of which can be either Unicode-enabled or ANSI-enabled.

The ODBC 3.5 (or later) Driver Manager supports the use of a Unicode driver with both a Unicode application and an ANSI application. It also supports the use of an ANSI driver with an ANSI application. The Driver Manager provides limited Unicode-to-ANSI mapping for a Unicode application working with an ANSI driver. This allows access to the Jet 3.5 databases and support of all existing ISAM file types.

When an ANSI application uses the ODBC Desktop Database Driver 4.0 and accesses Microsoft Access 4.0 or higher, the driver exposes the data type as SQL_CHAR, SQL_VARCHAR, or SQL_LONGVARCHAR even though Jet 4.0 supports the wide version. Older versions of Jet do not support SQL_WCHAR, SQL_WVARCHAR, and SQL_WLONGVARCHAR. This restriction also applies in cases where the old formats are used with the Jet 4.0 Database Engine.

For more information concerning Unicode issues with ODBC, see the Unicode topics in "Programming Considerations" of the *ODBC Programmer's Reference*.

Performance Issues

To ensure compatibility with existing ANSI applications, the SQL_WCHAR, SQL_WVARCHAR, and SQL_WLONGVARCHAR data types are exposed as SQL_CHAR, SQL_VARCHAR, and SQL_LONGVARCHAR for Microsoft Access 4.0 or higher data sources. The data sources do not return WIDE CHAR datatypes but the data still must be sent to Jet in Wide Char form. It is important to understand that conversion will take place if a SQL_C_CHAR parameter or result column is bound to a SQL_CHAR data type in an ANSI application.

This conversion can be especially inefficient in terms of memory when a SQL_C_CHAR type is bound to a parameter of type LONGVARCHAR.   Since the Jet 4.0 engine is unable to stream LONGTEXT parameter data, a UNICODE conversion buffer must be allocated that is twice the size of the SQL_C_CHAR ANSI buffer. The most efficient mechanism is for the application to perform the UNICODE conversion and bind the parameter as type SQL_C_WCHAR. When a parameter is marked as data-at-execution and the data is supplied in multiple calls to SQLPutData, a longtext data buffer is grown. One way to avoid the expense of growing this "Put Data" buffer is to supply an optional length via SQL_DATA_AT_EXEC_LEN(x), where x is the expected length of bytes.   This will initialize the size of an internal PutData buffer to x bytes.

Note   An efficient way to insert or update long data can be accomplished using **SQLBulkOperations()** or **SQLSetPos()** and setting the long data to SQL_DATA_AT_EXEC. (EXEC_LEN is ignored in this case.) Data can be streamed in chunks by calling **SQLPutData** multiple times, which will effectively append the data to the table.

Functions Accepting String Parameters

All functions that take string parameters will be converted to Unicode. (The "W" form of the function will be exported.) Count of bytes is converted to count of characters for those applicable ODBC APIs. This applies to the following functions:

- **SQLConnect**
- **SQLDriverConnect**
- **SQLColAttributes**
- **SQLDescribeCol**
- **SQLError** (replaced by **SQLGetDiagField**)
- **SQLExecDirect**
- **SQLGetCursorName**
- **SQLSetCursorName**
- **SQLGetStmtAttr**
- **SQLGetInfo**
- **SQLGetStmtOption** (becomes **SQLGetStmtAttr**)
- **SQLSetStmtOption** (becomes **SQLSetStmtAttr**)
- **SQLGetConnectOption**
- **SQLSetConnectOption**
- **SQLGetTypeInfo**
- **SQLStatistics**
- **SQLTables**
- **SQLNativeSQL**
- **SQLSpecialColumns**
- **ConfigDSNEx**
- **ConfigDSN**

Diagnostics

All errors and warnings not checked or partially checked by the Driver Manager are handled by the driver. The driver also maps native errors, or errors returned by the data source, to SQLSTATEs. Each function listed in the *ODBC Programmer's Reference* contains a "Diagnostics" section that specifies conditions and messages.

Applications call **SQLGetDiagRec** to retrieve SQLSTATE, native error code, and diagnostic messages. Calling **SQLGetDiagField** and specifiying the field retrieves individual diagnostic fields. The support level of the diagnostic identifiers is listed below.

DiagIdentifiers	Support Level
SQL_DIA_DYNAMIC_FUNCTION	Not supported
SQL_DIAG_CLASS_ORIGIN	Supported. Always "ODBC 3.0" for versions 3.0 and later of this driver.
SQL_DIAG_COLUMN_NUMBER	Supported
SQL_DIAG_CURSOR_ROW_COUNT	Not supported
SQL_DIAG_DYNAMIC_FUNCTION_CODE	Not supported
SQL_DIAG_MESSAGE_TEXT	Supported
SQL_DIAG_NATIVE	Supported
SQL_DIAG_NUMBER	Supported
SQL_DIAG_RETURNCODE	Supported but implemented by the Driver Manager
SQL_DIAG_ROW_COUNT	Supported
SQL_DIAG_ROW_NUMBER	Supported
SQL_DIAG_SERVER_NAME	Not supported
SQL_DIAG_SQLSTATE	Supported
SQL_DIAG_SUBCLASS_ORIGIN	Supported

Overview

Microsoft ODBC Desktop Database Drivers 4.0 allow you to open and query a database through the Open Database Connectivity (ODBC) interface. These drivers work with versions of ODBC 3.51 and later.

What's New for the ODBC Desktop Database Drivers

Unicode Support

The Jet Database Engine has changed format to Unicode from ANSI with its upgrade to 4.0. The Microsoft Access 4.0 driver exposes this native Unicode data so that Unicode applications can access the Unicode data directly, and also provides conversion of data from Unicode to ANSI for existing ANSI applications.

 For more information about Unicode, see the Unicode section in Chapter 17 of the *ODBC Programmer's Reference,* "Programming Considerations," or the "Compatibility" section of this document.

Descriptor Support

Descriptors contain header and record fields that completely describe columns or parameters. Though no database operation requires that the application gain direct access to descriptors, access to descriptors can streamline many operations. **SQLGetDescField** is now supported. **SQLDescribeParam** is not supported. Subsequently, neither is automatic population of the Implementation Parameter Descriptor (IPD).

For more information about descriptors, see Chapter 13 of the *ODBC Programmer's Reference*, "Descriptors."

SQLGetDiagField and SQLGetDiagRec Support

Limited support for **SQLGetDiagField** and **SQLGetDiagRec** has been added in Jet.

SQLBulkOperations

With the addition of support, it is now possible to use the Jet Database Engine to return a bookmark on a newly inserted row.

Locale Indicators

The Jet Database Engine connection string has changed to reflect a single LCID rather than separate CodePage and LangID entries. The **ODBC Microsoft Access Driver Setup** dialog box displays this addition. Programmatic changes are listed under Database Compaction (Advanced), Database Repair (Advanced), and Database Create (Advanced).

Data Types

Additional C data types and SQL data types are supported for the Jet Database Engine.

Additional supported data types:

Type	Name	Comments
C data type	SQL_C_WCHAR	
C data type	SQL_C_NUMERIC	
C data type	SQL_C_GUID	
C data type	SQL_C_SBIGINT	
C data type	SQL_C_UBIGINT	
C data type	SQL_C_INTERVAL_*	All
C data type	SQL_WCHAR	Exposed only in Unicode applications using Microsoft

		Access 4.0 or later.
SQL data type	SQL_WVARCHAR	Exposed only in Unicode applications using Microsoft Access 4.0 or later.
SQL data type	SQL_WLONGVARCHAR	Exposed only in Unicode applications using Microsoft Access 4.0 or later.
		The maximum length has been reduced in half to a new value of 1073741824.
SQL data type	SQL_NUMERIC	Max precision 28
SQL data type	SQL_GUID	

Note   For Microsoft Access 4.0 and later, the maximum length for SQL_BINARY and SQL_VARBINARY values has doubled to 510 (from a previous value of 255).

For more information about data types, see Appendix D of the *ODBC Programmer's Reference*, "Data Types."

Unsupported ODBC 3.x features:

- Automatic population of the Implementation Parameter Descriptor
- Arrays of parameters
- **SQLMoreResults** or any of the additions related to batch execution
- Descriptor fields that are not available through Jet (such as BASE_TABLE_NAME)

DATETIME Functions

QUARTER, DAYNAME, and MONTHNAME have been added to the *InfoType* SQL_SQL92_DATETIME_FUNCTIONS of **SQLGetInfo**.

Environment Fields

The ODBC Desktop Database Drivers will support the following ODBC 3.x and ODBC 2.x behaviors for keyset-driven, read-only cursors: variable length bookmarks, X/OPEN SQLSTATE constant values, **SQLSetEnvAttr**, returning SQL_NO_DATA on futile updates, and multirow fetching using **SQLFetch**.

Changing Source of Visual FoxPro Support

The ODBC Desktop Database Drivers pack does not include the IISAM file for Visual FoxPro (Msxbse40.dll). The recommended driver now is the "Visual FoxPro ODBC Driver," which can be found on the Visual FoxPro page under Development Tools on the Microsoft Support Online Web site. Specific documentation accompanies this driver.
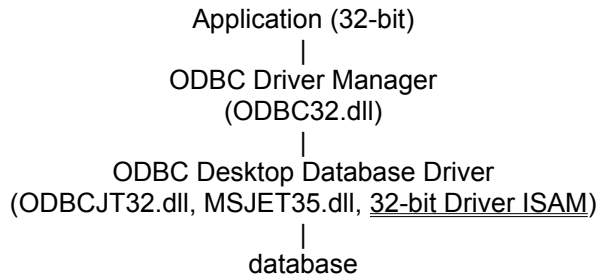
Desktop Database Drivers Architecture

The drivers are designed for use on Microsoft Windows 95 or later, or Windows NT 4.0 or later. Only 32-bit applications are supported on Windows 95 or later; 16-bit and 32-bit applications are supported on Windows NT 4.0 or later.

Note   For information about the version of ODBC to be used with these drivers, refer to the *ODBC Programmer's Reference*, and past and current release notes. Except for noted areas, these drivers conform to the *ODBC Programmer's Reference*.
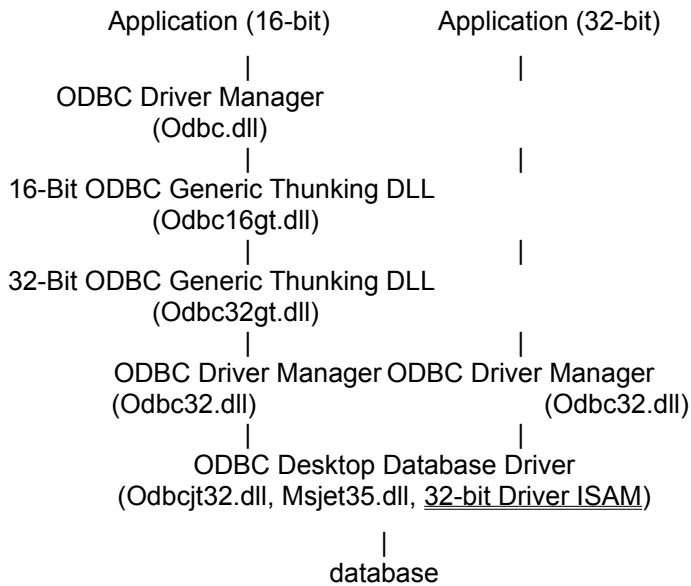
The ODBC Desktop Database Drivers include 32-bit drivers for Microsoft Access, dBASE, Microsoft Excel, Paradox, and Text. No 16-bit drivers are included. (A driver for Microsoft FoxPro is available separately.)

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

<div align="center">

Application (32-bit)
|
ODBC Driver Manager
(ODBC32.dll)
|
ODBC Desktop Database Driver
(ODBCJT32.dll, MSJET35.dll, <u>32-bit Driver ISAM</u>)
|
database

</div>

The use of these drivers by 16-bit applications on Windows 95 is not supported.

The application/driver architecture on Windows NT 4.0 or later is:

<div align="center">

Application (16-bit)   Application (32-bit)
| |
ODBC Driver Manager
(Odbc.dll)
| |
16-Bit ODBC Generic Thunking DLL
(Odbc16gt.dll)
| |
32-Bit ODBC Generic Thunking DLL
(Odbc32gt.dll)
| |
ODBC Driver Manager   ODBC Driver Manager
(Odbc32.dll)   (Odbc32.dll)
| |
ODBC Desktop Database Driver
(Odbcjt32.dll, Msjet35.dll, <u>32-bit Driver ISAM</u>)

|
database

</div>

Driver ISAM Files

ODBC Desktop Database Drivers 4.0 include ISAM files. These files are as follows:

Driver	32-bit ISAM	Versions
Microsoft Access	None (included in Msjet40.dll)	Microsoft Access versions 7.0 and 97
Microsoft Access	Msrd2x40.dll	Microsoft Access version 2.0
Microsoft Access	Msrd3x40.dll	Microsoft Access version 3.0
dBase	Msxbde40.dll	dBase versions III, IV, and V (dBASE for Windows)
Microsoft Excel	Msexcl40.dll	Microsoft Excel versions 3.0, 4.0, 5.0, 7.0, 97 (1)
Paradox	Mspbde40.dll	Paradox versions 3.5, 4.0, 4.5, 5.*x* (Paradox for Windows)
Text	Mstext40.dll	ASCII Text

1   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 3.0, 4.0, 5.0, 7.0, or 97 format that the driver is able to read.

Microsoft FoxPro

The ODBC Desktop Database Drivers do not include a driver for Microsoft FoxPro. The Microsoft FoxPro drivers can be obtained from the Microsoft Support Online Web site.

Registry Entries

IISAM registry settings are initially written when the **DLLRegisterServer** function call is called on each respective IISAM DLL, and are removed when **DLLUnregisterServer** is called. This **DLLRegisterServer** function is called by the Jet ODBC function dynamically for each IISAM DLL. For example, if a connection is attempted using the Paradox IISAM and it fails initially, the driver will attempt to initialize this IISAM (calling **DLLRegisterServer**) and retry the connection.

ODBC Component Files

The following table lists the files required by each component of ODBC Desktop Database Drivers 4.0. 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 applications.

The following files are common for each of the ODBC Desktop Database Drivers 4.0:

Ds16gt.dll	Odbc32.dll	Odbcjet.cnt
Ds32gt.dll	Odbc32gt.dll	Odbcji32.dll
Expsrv.dll	Odbccp32.cpl	Odbcjt32.dll
Msjint40.dll	Odbccp32.dll	Odbcjtnw.hlp
Msjet40.dll	Odbccr32.dll	Odbcjtnw.cnt
Msjter40.dll	Odbcinst.cnt	Odbctl32.dll
Msjtes40.dll	Odbcinst.hlp	Vbajet32.dll
Msvcrt40.dll	Odbcint.dll	Odbctrac.dll
Odbc16gt.dll	Odbcjet.hlp	

The following files are unique to each driver:

Driver	File
Microsoft Access	Msrd2x40.dll
	Msrd3x40.dll
dBASE	Msxbse40.dll
	Oddbse32.dll
Microsoft Excel	Msexcl40.dll
	Odexl32.dll
Paradox	Mspdox40.dll
	Odpdx32.dll
Text	Mstext40.dll
	Odtext32.dll

Hardware and Software Requirements

Hardware Requirements

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

- An IBM-compatible personal computer.
- A hard disk with 6 MB of free disk space.
- At least 16 MB 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 3.51 or later (Odbc32.dll).
- Microsoft Windows 95 or later, or Windows NT 4.0 or later.
- The stack size for an application using a Microsoft ODBC driver should be at least 20K.

When using Microsoft Windows NT 4.0 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. All access to the Jet ISAM layer will be single-threaded for all applications using the Microsoft Jet engine.

When running multiple 16-bit applications on Windows on Windows (WOW) on Microsoft Windows NT 4.0, 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, select the application's icon in the Microsoft Windows NT Program Manager, choose the File-Properties command, and then choose Run In Separate Memory Space.

The use of these drivers by 16-bit applications on Windows 95 is not supported.

Driver-Specific Hardware and Software Requirements

- The **Microsoft Access** and **dBASE** 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 on Microsoft Windows 95/98 (not on Microsoft Windows NT) and do not have Microsoft Access installed on your computer, add the line "FILES= 50" to your Config.sys file.

Setting Up Drivers

1 Double-click the Control Panel icon. In the Control Panel window, double-click the ODBC Data Sources icon (or open the ODBC Administrator directly).

2 Click the ODBC Administrator **Drivers** tab.

Drivers can no longer be added or deleted from the Control Panel or Administrator. The **Add** and **Delete** buttons have been removed from the ODBC Administrator **Drivers** tab. Adding or deleting drivers is supported only during installation of the ODBC Desktop Database Drivers when the driver's own setup program executes.

Adding a Data Source

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

2 Click the **User DSN**, **System DSN**, or **File DSN** tab.

3 Click **Add**.

4 In the **Create New Data Source** dialog box, select a driver name and follow the instructions of the wizard.

Modifying a Data Source

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

2 Click the **User DSN**, **System DSN**, or **File DSN** tab, depending on the type of the data source to be modified.

3 Select the data source from the list. If modifying a file DSN, find the appropriate directory in the **Look In** box (using the Up control if necessary), then select the data source from the directory.

4 Click **Configure** and set up the data source as necessary.

Deleting a Data Source

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

2 Click the **User DSN**, **System DSN**, or **File DSN** tab, depending on the type of the data source to be modified.

3 Select the data source from the list. If modifying a file DSN, find the appropriate directory in the **Look In** box (using the Up control if necessary), then select the data source from the directory.

4 Click **Remove**, and then click **Yes** 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 4.0 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

You can call the **SQLConfigDataSource** 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

Product support for ODBC is provided by Microsoft Technical Support (MTS). Because many Microsoft products utilize ODBC as a core component, and also redistribute ODBC drivers, MTS considers ODBC to be a part of whatever product the user has that incorporates ODBC as a component. Support for ODBC is provided in accordance with the support agreement of that product. Each of these Microsoft products has its own support offering, as described in the documentation that comes with the product. Please refer to this documentation to determine which support options are available.

ODBC Setup Dialog Box

The **ODBC Setup** dialog box contains the following fields. Most of these options can also be set up dynamically by calling **SQLConfigDataSource** using keywords.

- Approximate Row Count
- Buffer Size
- Collating Sequence
- Data Source Name
- Database
- Define Format
- Description
- Directory
- Encryption
- Exclusive
- Extensions List
- ImplicitCommitSync
- Locale
- Net Style
- Page Timeout
- Read Only
- Rows to Scan
- Select Directory
- Select Indexes
- Select Network Directory
- SelectWorkbook
- Show Deleted Rows
- System Database
- Threads
- Use Current Directory
- UserCommitSync
- User Name
- Version

Approximate Row Count

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

Determines whether table size statistics are approximated. This option applies to all data sources that use the ODBC driver. To set this option dynamically, use the **STATISTICS** keyword in a call to **SQLConfigDataSource**.

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 2048K (displayed as 2048). Any integer value divisible by 256 can be entered. To set this option dynamically, use the MAXBUFFERSIZE keyword in a call to **SQLConfigDataSource**.

Collating Sequence

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

The sequence in which the fields are sorted.

When the **dBASE** driver is used, the sequence can be: ASCII (the default) or International.

When the **Paradox** driver is used, the sequence can be: ASCII (the default), International, Swedish-Finnish, or Norwegian-Danish.

To set this option dynamically, use the **COLLATINGSEQUENCE** keyword in a call to **SQLConfigDataSource**.

Data Source Name

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

To set this option dynamically, use the **DSN** keyword in a call to **SQLConfigDataSource**.

Database

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

The full path of the Microsoft Access database you want to access. Use the <u>Select</u> button to select an existing database. Use the <u>Create</u> 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 <u>Repair</u> button to repair a damaged database. Use the <u>Compact</u> button to compact a database.

To set this option dynamically, use the **DBQ** keyword in a call to **SQLConfigDataSource**.

Define Format

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

Displays the **Define Text Format** dialog box and enables you to specify the schema for individual tables in the data source directory.

This option cannot be set dynamically by a call to **SQLConfigDataSource**.

Description

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

To set this option dynamically, use the **DESCRIPTION** keyword in a call to **SQLConfigDataSource**.

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, 7.0, or 97 files, the path display is labeled "Workbook".

To set this option dynamically, use the **DEFAULTDIR** keyword in a call to **SQLConfigDataSource**.

Encryption

(This control is displayed only for the **Microsoft Access** driver and is available only when Version 4.x format is selected.)

With the release of the 4.0 version of the Microsoft Jet Database Engine, database encryption is allowed when creating databases. See the *Jet Database Engine Programmer's Reference* for additional information.

Exclusive

(This control is displayed only for the **dBASE**, **Microsoft Access**, 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.

To set this option dynamically, use the **EXCLUSIVE** keyword in a call to **SQLConfigDataSource**.

Extensions List

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

Lists the file name 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. When **Default** (*.*) is unchecked, the list of extensions defaults to *.asc, *.csv, *.tab, and *.txt.

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

When the **Text** driver is used, a file with no extension is created when the CREATE TABLE statement is executed with a name that has no extension. Other drivers create a file with a default extension when no extension is provided. To create a file with a .TXT extension, the extension must be included in the name. To display files without extensions in the **Define Text Format** dialog box, "*." must be added to the Extensions List.

To set this option dynamically, use the **EXTENSIONS** keyword in a call to **SQLConfigDataSource**.

ImplicitCommitSync

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

Determines how changes made outside of a transaction are written to the database. This value is initially set to "Yes", which means that the **Microsoft Access** driver will wait for commits in an internal/implicit transaction to be completed.

This option is included in the **Advanced Options** dialog box for the **Microsoft Access** driver.

Net Style

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

The network access style to use when accessing Paradox data: either "3.*x*" for Paradox 3.*x* or "4.*x*" for Paradox 4.*x* or 5.*x*. Can be set to "3.*x*" or "4.*x*" if the version is Paradox 4.*x* or 5.*x*; if the version is Paradox 3.*x*, the style must be "3.*x*".

To set this option dynamically, use the **PARADOXNETSTYLE** keyword in a call to **SQLConfigDataSource**.

Page Timeout

(This control is displayed only for the **dBASE**, **Microsoft Access**, 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. For the **Microsoft Access** driver, the default is five-tenths of a second (0.5 seconds). For the **dBASE** and **Paradox** drivers, the default is 600 tenths of a second (60 seconds). This option applies to all data sources that use the ODBC driver.

The page timeout cannot be 0 because of an inherent delay. The page timeout cannot be less than the inherent delay, even if the page timeout option is set below that value.

To set this option dynamically, use the **PAGETIMEOUT** keyword in a call to **SQLConfigDataSource**.

Read-Only

Designates the database as read-only.

To set this option dynamically, use the **READONLY** keyword in a call to **SQLConfigDataSource**.

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. The value defaults to 8; if it is set to 0, all rows are scanned. (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 25. (A number outside the limit will return an error.)

To set this option dynamically, use the **MAXSCANROWS** keyword in a call to **SQLConfigDataSource**.

Select Directory

(This control is displayed only for the **dBASE**, **Microsoft Excel 3.0/4.0**, **Paradox**, and **Text** drivers.)

Displays a dialog box where you can select a directory containing the files you want to access.

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 file names 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.

When using the **Microsoft Access** driver, you can use the **DefaultDir** setting in the **Advanced Option** dialog box to define a default directory, since you can create a data source without a corresponding .mdb file.

Directory/Workbook Path Displays (Microsoft Excel)

For Microsoft Excel 3.0 or 4.0 files, the path display is labeled "Directory", and the path selection button is labeled "Select Directory". For Microsoft Excel 5.0, 7.0, or 97 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 3.0/4.0, or the directory where the workbook file is located for Microsoft Excel 5.0, 7.0, or 97. The **Use Current Directory** control is disabled for Microsoft Excel 5.0, 7.0, or 97.

To set this option dynamically, use the **DEFAULTDIR** keyword in a call to **SQLConfigDataSource**.

Select Indexes

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

Displays the **Select Indexes** dialog box where you can associate dBASE 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 Click **OK** to save the table/index associations.

Note   dBASE III indexes must be assigned using this dialog box for the driver to recognize them.

Note   The ODBC dBASE driver does not support Clipper .ntx index files.

This option cannot be set dynamically by a call to **SQLConfigDataSource**.

Select Network Directory

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

The full path of the directory containing a Paradox lock database, because it contains either the Pdoxusrs.net file (in Paradox 4.*x*) or the Paradox.net file (in Paradox 5.*x*). If the directory does not contain one of these files, the Paradox driver creates one. For information about these files, see the Paradox documentation.

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

To set this option dynamically, use the **PARADOXNETPATH** keyword in a call to **SQLConfigDataSource**.

Select Workbook

(This control is displayed only for **Microsoft Excel** 5.0, 7.0, or 97.)

For **Microsoft Excel** 5.0, 7.0, or 97 files, this option displays the currently selected workbook. Before you add the data source, you must click **Select Workbook** to select a directory.

Worksheets within workbooks are treated as SYSTEM TABLES. Defined or named ranges within a worksheet are treated as TABLES.

Show Deleted Rows

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

Specifies whether or not rows that have been marked as deleted can be retrieved or positioned on. If unchecked, deleted rows are not displayed; if checked, deleted rows are treated the same as non-deleted rows. The default is unchecked.

To set this option dynamically, use the **DELETED** keyword in a call to **SQLConfigDataSource**.

Locale

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

Sets locale (LCID) for the database. Can be:

General (English, French, German, Portuguese, Italian), traditional Spanish, Dutch, Swedish/Finnish, Norwegian/Danish, Icelandic, Czech, Hungarian, Polish, Russian, Turkish, Arabic, Hebrew, Greek, Japanese, Korean, Taiwanese, or PRC (People's Republic of China).

This option cannot be set dynamically by a call to **SQLConfigDriver**.

System Database

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

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

Select the **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.

To set this option dynamically, use the **SYSTEMDB** keyword in a call to **SQLConfigDataSource**.

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

Threads

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

The number of background threads for the engine to use. For the **Microsoft Access** driver, this value defaults to 3, but can be changed. The user may want to increase the number of threads if there is a large amount of activity in the database.

This option is included in the **Advanced Options** dialog box for the **Microsoft Access** driver.

To set this option dynamically, use the **THREADS** keyword in a call to **SQLConfigDataSource**.

Use Current Directory

(This control is displayed only for the **dBASE**, **Microsoft Excel**, **Paradox**, and **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 disabled for **Microsoft Excel** 5.0, 7.0, or 97.

UserCommitSync

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

Determines whether the **Microsoft Access** driver will perform an explicit user-defined transactions asynchronously. This value is initially set to "Yes", which means that the **Microsoft Access** driver will wait for commits in a user-defined transaction to be completed.

Setting this option to False can have unpredictable consequences in a multi-user environment.

This option is included in the **Advanced Options** dialog box for the **Microsoft Access** driver.

To set this option dynamically, use the **USERCOMMITSYNC** keyword in a call to **SQLConfigDataSource**.

User Name

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

The Paradox user name. This is the name displayed to other users of Paradox files when a lock is encountered.

To set this option dynamically, use the **PARADOXUSERNAME** keyword in a call to **SQLConfigDataSource**.

Version

A pull-down control that enables you to select the version of the files used, for example, dBASE III, dBASE IV, or dBASE 5.0 for the dBASE driver.

For **Microsoft Access**, you can select the database format as expressed by the version numbers. If you select Version 4.x, database encryption is an option.

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.

Defining Text Format

When the **Text** driver is used, you can use the **Define Text Format** dialog box 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. The **Text** driver does not change the format of an existing text file to match the format defined in this dialog box, but returns an error when it uses the format, such as when it attempts to retrieve data from the text file.

Control	Information
Add	Adds a column using the values in the Data Type, Name, and Width fields from the dialog box, and if applicable, the Date Separator value from Schema.ini.
Characters	ANSI or OEM. OEM specifies a non-ANSI character set. This defaults to OEM if the format of the item selected in the Tables list has not been previously defined by this dialog box.
Column NameHeader	Indicates whether the columns of the first row of the selected table are to be used as column names. Either TRUE or FALSE. This defaults to FALSE if the format of the item selected in the Tables list has not been previously defined by this dialog box.
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. This list is enabled if a file has been selected in the Tables list.
Data Type	Can be BIT, BYTE, CHAR, CURRENCY, DATE, FLOAT, INTEGER, LONGCHAR, SHORT, or SINGLE. 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 Delimited" format is selected. The delimiter can only be one character in length, and double quotations marks (") cannot be used as the delimiter character. (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). This defaults to CSV Delimited if the format of the item selected in the Tables list has not been previously defined by this dialog box.
	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 selected, column names are generated automatically as "F1", "F2", and so on. No default value is shown in the Data Type box.
	This functionality only works on columns that are less than 64,513 bytes.
Modify	Modifies the selected column using the values in the Data Type, Name, and Width fields.
Name	Displays the name of the selected column. Can 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.
Remove	Deletes the selected column.
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 can enter a number from 1 to 32767 for the number of rows to scan. This defaults to 25 if the format of the item selected in the Tables list has not been previously defined by this dialog box. (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.
	When <default> is selected, and one of the following is true, then the values of the table attributes in the Tables group are written to Schema.ini. No other entries in Schema.ini are touched.
	• There is no Schema.ini in the specified directory.
	• The Schema.ini file exists, but there is no section in Schema.ini for one of the Text files (with the specified extension) in the directory.
	• The section for a Text file exists in Schema.ini, but the body is empty.
	When <default> is selected, the "Columns" group is disabled.
Width	The width of the column can be changed for CHAR or LONGCHAR columns. The width defaults to 1 if the format of the item selected in the Tables list has not been previously defined by this dialog box.
	For other data types, the width control is disabled, and no value is displayed.

Schema

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

Schema.ini File

When the **Text** driver is used, the format of the text file is determined by using a schema information file. The schema information file, which is always named Schema.ini and always kept in the same directory as the text data source, provides the IISAM with information about the general format of the file, the column name and data type information, and a number of other data characteristics. A Schema.ini file is always required for accessing fixed-length data; you should use a Schema.ini file when your text table contains DateTime, Currency, or Decimal data or any time you want more control over the handling of the data in the table.

Note   The Text ISAM will obtain initial values from the registry, not from Schema.ini. 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. If the values in the registry are different from the values in Schema.ini, the values in the registry will be overwritten by the values from Schema.ini.

Understanding Schema.ini Files

Schema.ini files provide schema information about the records in a text file. Each Schema.ini entry specifies one of five characteristics of the table:

- The text file name
- The file format
- The field names, widths, and types
- The character set
- Special data type conversions

The following sections discuss these characteristics.

Specifying the File Name

The first entry in Schema.ini is always the name of the text source file enclosed in square brackets. The following example illustrates the entry for the file Sample.txt:

```
[Sample.txt]
```

Specifying the File Format

The **Format** option in Schema.ini specifies the format of the text file. The Text IISAM can read the format automatically from most character-delimited files. You can use any single character as a delimiter in the file except the double quotation mark ("). The **Format** setting in Schema.ini overrides the setting in the Windows Registry on a file-by-file basis. The following table lists the valid values for the **Format** option.

Format specifier	Table format	Schema.ini Format statement
Tab Delimited	Fields in the file are delimited by tabs.	`Format=TabDelimited`
CSV Delimited	Fields in the file are delimited by commas (comma-separated values).	`Format=CSVDelimited`
Custom Delimited	Fields in the file are delimited by any character you choose to input into the dialog box. All	`Format=Delimited(custom character)` -or-

	except the double quote (") are allowed, including blank.	With no delimiter specified:
Fixed Length	Fields in the file are of a fixed length.	`Format=Delimited( )` `Format=FixedLength`

Specifying the Fields

You can specify field names in a character-delimited text file in two ways:

- Include the field names in the first row of the table and set **ColNameHeader** to **True.**
- Specify each column by number and designate the column name and data type.

You must specify each column by number and designate the column name, data type, and width for fixed-length files.

Note   The **ColNameHeader** setting in Schema.ini overrides the **FirstRowHasNames** setting in the Windows Registry on a file-by-file basis.

The data types of the fields can also be determined. Use the **MaxScanRows** option to indicate how many rows should be scanned when determining the column types. If you set **MaxScanRows** to 0, the entire file is scanned. The **MaxScanRows** setting in Schema.ini overrides the setting in the Windows Registry on a file-by-file basis.

The following entry indicates that Microsoft Jet should use the data in the first row of the table to determine field names and should examine the entire file to determine the data types used:

```
ColNameHeader=True
MaxScanRows=0
```

The next entry designates fields in a table by using the column number (**Col***n*) option, which is optional for character-delimited files and required for fixed-length files. The example shows the Schema.ini entries for two fields, a 10-character CustomerNumber text field and a 30-character CustomerName text field:

```
Col1=CustomerNumber Text Width 10
Col2=CustomerName Text Width 30
```

The syntax of **Col***n* is:

Col*n=ColumnName type* [**Width #**]

The following table describes each part of the **Col***n* entry.

Parameter	Description
ColumnName	The text name of the column. If the column name contains embedded spaces, you must enclose it in double quotation marks.
type	Data types are:
	Microsoft Jet data types
	Bit
	Byte
	Short
	Long
	Currency
	Single
	Double
	DateTime
	Text
	Memo
	ODBC data types

Char (same as Text)
Float (same as Double)
Integer (same as Short)
LongChar (same as Memo)
Date *date format*

Width	The literal string value `Width`. Indicates that the following number designates the width of the column (optional for character-delimited files, required for fixed-length files).
#	The integer value that designates the width of the column (required if **Width** is specified).

Selecting a Character Set

You can select from two character sets: ANSI and OEM. The **CharacterSet** setting in Schema.ini overrides the setting in the Windows Registry on a file-by-file basis. The following example shows the Schema.ini entry that sets the character set to ANSI:

```
CharacterSet=ANSI
```

Specifying Data Type Formats and Conversions

The Schema.ini file contains a number of options that you can use to specify how data is converted or displayed. The following table lists each of these options.

Option	Description
DateTimeFormat	Can be set to a format string indicating dates and times. You should specify this entry if all date/time fields in the import/export are handled with the same format. All of the Microsoft Jet formats except A.M. and P.M. are supported. In the absence of a format string, the Windows Control Panel short date picture and time options are used.
DecimalSymbol	Can be set to any single character that is used to separate the integer from the fractional part of a number.
NumberDigits	Indicates the number of decimal digits in the fractional portion of a number.
NumberLeadingZeros	Specifies whether a decimal value less than 1 and greater than –1 should contain leading zeros; this value can either be **False** (no leading zeros) or **True**.
CurrencySymbol	Indicates the currency symbol to be used for currency values in the text file. Examples include the dollar sign ($) and Dm.
CurrencyPosFormat	Can be set to any of the following values: • Currency symbol prefix with no separation ($1) • Currency symbol suffix with no separation (1$) • Currency symbol prefix with one

	character separation ($ 1)
	• Currency symbol suffix with one character separation (1 $)
CurrencyDigits	Specifies the number of digits used for the fractional part of a currency amount.
CurrencyNegFormat	Can be one of the following values: • ($1) • –$1 • $–1 • $1– • (1$) • –1$ • 1–$ • 1$– • –1 $ • –$ 1 • 1 $– • $ 1– • $ –1 • 1– $ • ($ 1) • (1 $) This example shows the dollar sign, but you should replace it with the appropriate **CurrencySymbol** value in the actual program.
CurrencyThousandSymbol	Indicates the single-character symbol to be used for separating currency values in the text file by thousands.
CurrencyDecimalSymbol	Can be set to any single character that is used to separate the whole from the fractional part of a currency amount.

Note   If you omit an entry, the default value in the Windows Control Panel is used.

Text File Format (Advanced)

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 (""). No blanks can occur before or after delimited values.

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*
Note 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
Note 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 ::=
Note For delimited files, a NULL is represented by no data between two delimiters.

fixed-width-null ::= <SPACE>...
Note For fixed-width files, a NULL is represented by spaces.

Select Unique Record Identifier

Select the column or columns 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**, **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.

For **Microsoft Excel** 5.0, 7.0, or 97, 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

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 might 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.

When the **Paradox** driver is used, a password is only supported if entered in the connection string. That password is valid whenever a table is opened. If no password is passed in the connection string, no password is established for a table. If tables have different passwords, they cannot both be opened in the same session, nor can the tables be joined.

Heterogeneous Joins/Attached Tables

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

Creating links to attached tables is not 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.

The use of the ODBC Desktop Database Drivers from within Data Access Objects (DAO) code is not supported.

Opening Microsoft Excel Tables

The **Microsoft Excel** driver is not able to read Microsoft Excel files earlier than version 3.0. These files can be opened in a later release of Microsoft Excel and saved in version 3.0, 4.0, 5.0, 7.0, or 97 format that the driver is able to read.

A BIFF3-version table is created if the version of the data source through which the connection was made is "Excel", which means either Microsoft Excel 3.0 or 4.0. A worksheet is created in the workbook that is connected to if the version of the data source was "Excel 5.0", "Excel 7.0", or "Excel 97".

Microsoft Excel 3.0 and Microsoft Excel 4.0 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 by default, and can be opened by only one user at a time. The user must explicitly choose to open Microsoft Excel tables as 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.

Failing after Repeated Connections

If an ODBC application working on Microsoft Windows 95/98 connects to and disconnects from the server repeatedly (over 50-60 times), the connection might fail, with the driver returning SQLSTATE 01000 (General warning) and the error message "Failed to get the expression service." This might particularly be a problem for Internet Web server applications. This failure is resolved by updating the Rpcrt4.dll file in the \Windows\System directory to release QFE 324. For more information, contact Microsoft Technical Support.

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 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. For more information, see the Visual Basic documentation.

Connection Strings (Advanced)

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

Keyword	Description
DSN	Name of the data source
DBQ	Name of the directory
DRIVERID	An integer ID for the driver
FIL	File type (MS Access for Microsoft Access; dBase III, dBase IV, or dBase5; Excel 3.0, Excel 4.0, Excel 5.0, Excel 7.0, or Excel 97 for Microsoft Excel; Paradox 3.x, 4.x, or 5.x; or Text) This keyword is supported only for backward compatibility.

Driver-Specific Keywords

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

Keyword	Description
UID	The user login ID (Microsoft Access only). Initially set to "Admin". For more information about the Admin user ID, see "Creating a New Admin User" in Chapter 10 of the Microsoft Jet Database Engine Programmer's Guide, "Managing Security."
PWD	The user-specified password.

Other keywords are supported for use in connection strings, in addition to the ones documented above. For more information about these keys, see **SQLConfigDataSource**. The **Microsoft Access**-specific keywords compacting a database (COMPACT_DB) and creating a database (CREATE_DB) are not supported in connection strings.

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**).

Connection strings are also used when programatically attaching a **Microsoft Access** database using **SQLConfigDriver**. Use the syntax "JET_ATTACH <source db> <connection string>" in the list of arguments (*lpszArgs*) of the **SQLConfigDriver** function. In this function, <source db> is the full path to the Microsoft Access database to be attached and <connection string> follows conventions presented in this section.

Note   Connection strings used to connect to any database using the Jet 4.0 or later database engine should contain a single LCID rather than separate CodePage and LangID entries.

Select Database (Advanced)

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

The **Open Database** dialog box contains the following fields:

Control	Description
Database 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 can be pressed to create a new database.

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

The **New Database** dialog box contains the following fields:

Control	Description
Database Name	Name of the database file with an .mdb extension.
Format	Format of the Microsoft Jet Database Engine. Can be: Version 4.x (most recent, Unicode format), Version 3.x (ANSI format), or Version 2.x (ANSI format).
Locale	Identifies the LCID. Replaces the Sort Order control, but most of the default sort order rules apply. Can be:
	General (English, French, German, Portuguese, Italian), traditional Spanish, Dutch, Swedish/Finnish, Norwegian/Danish, Icelandic, Czech, Hungarian, Polish, Russian, Turkish, Arabic, Hebrew, Greek, Japanese, Korean, Taiwanese, or PRC (People's Republic of China).
Options	System Database: Creates a system database.
	Encryption: Microsoft Jet Database Engine 4.0 supports encrypted passwords.

Database Creation Using SQLConfigDriver

To create a database programmatically using **SQLConfigDriver**, include "CREATE_DB=<path_name><optional_sort-order><optional_ENCRYPT keyword>" in the list of arguments (*lpszArgs*) of the **SQLConfigDriver** function. In this function, <path_name> is the full path to the Microsoft Access database to be compacted, and the optional ENCRYPT keyword allows you to add security to the Microsoft Access database.

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 might be corrupted and require repairs. Repairs can be performed either upon setup through the Microsoft Access driver setup dialog box, or programmatically by using the **SQLConfigDataSource** function. Starting with version 4.0 of the Jet Database Engine, the Repair functionality has been included in the Compaction routine. For backward compatibility, the Desktop Database Drivers map the Repair command to the Compact commmand.

Using the Microsoft Access driver setup dialog box

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

Using SQLConfigDataSource

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

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

Using SQLConfigDriver

To perform repair operations on a database file programmatically using **SQLConfigDriver**, include " REPAIR_DB=<path_name>" in the list of arguments (*lpszArgs*) of the **SQLConfigDriver** function. In this function, <path_name> is the full path to the Microsoft Access database to be repaired.

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.

A database indicated as corrupted may in fact not be corrupted. The indication might be prompted by an improper shutdown that does not clear the appropriate bits. In this and other cases, a programmer might 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 using one of three methods: upon setup through the Microsoft Access driver setup dialog box, or programmatically by using either the **SQLConfigDataSource** function or the **SQLConfigDriver** function. Microsoft Access databases should be compacted on a regular basis.

Using the Microsoft Access driver setup dialog box

To perform data compaction on a database file through the Microsoft Access driver setup dialog box, click **Compact**. 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 (which can be changed for the destination database), and then click **OK**. The name of the file to compact into can be the same as the file to be compacted. An error is returned if the name of the file to compact into is the same as an existing file.

Using SQLConfigDataSource

To perform data compaction on a database file programmatically, include "COMPACT_DB=<path_name><optional_sort_order><optional ENCRYPT keyword>" in the list of attributes (*lpszAttributes*) of the **SQLConfigDataSource** function. In this function, <path_name> is the full path to the Microsoft Access database to be compacted.

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. Compacting password-protected files may have unpredicted results.

Using SQLConfigDriver

To perform data compaction on a database file programmatically using **SQLConfigDriver**, include "COMPACT_DB=<path_name><optionaL_sort_order><optional ENCRYPT keyword>" in the list of arguments (*lpszArgs*) of the **SQLConfigDriver** function. In this function, <path_name> is the full path to the Microsoft Access database to be compacted.

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.