ODBC Btrieve Driver

For All Users

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

For Advanced Users

The following topics discuss how to use the ODBC Btrieve driver directly.

For Programmers

The following topics provide programming information on the ODBC Btrieve driver. They are intended for application programmers and require knowledge of the Open Database Connectivity (ODBC) application programming interface (API).

Overview

The ODBC Btrieve driver allows you to open and query a Btrieve file through the Open Database Connectivity (ODBC) interface.

The application/driver architecture is:
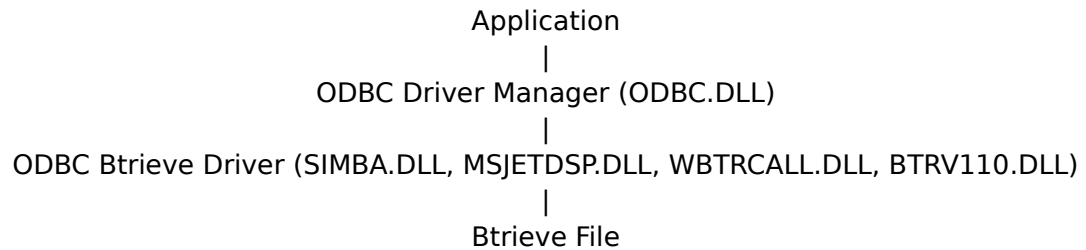
Application
|
ODBC Driver Manager (ODBC.DLL)
|
ODBC Btrieve Driver (SIMBA.DLL, MSJETDSP.DLL, WBTRCALL.DLL, BTRV110.DLL)
|
Btrieve File

See Also

For All Users

Hardware and Software Requirements

To access Btrieve data, you must have:

- The ODBC Btrieve driver.
- The ODBC Driver Manager 1.0 (ODBC.DLL).
- Novell Btrieve for Windows (WBTRCALL.DLL).
- A computer running MS-DOS 3.3 or later.
- Microsoft Windows 3.0a or later.

The ODBC Btrieve driver requires the stand-alone Btrieve for Windows dynamic-link library (DLL), WBTRCALL.DLL.   This file must be in the directory that contains Windows system DLLs (usually the C:\WINDOWS\SYSTEM directory).   To check if you already have a copy of this file on your hard drive:

1 In the Main group, choose the File Manager icon.

2 From the File menu, choose Search.

   The Search dialog box appears.

3 In the Search For box, type WBTRCALL.DLL.

4 In the Start From box, type C:\.

5 Choose the OK button.

If this file exists, copy it to the directory that contains Windows system DLLs (usually the C:\WINDOWS\SYSTEM directory).

Important   If you already have a copy of WBTRCALL.DLL in your C:\WINDOWS\SYSTEM directory, contact your system administrator before copying over it.   This file may be used by other programs on your computer.

If you don't have a copy of this file, contact Novell, Inc. at (800) 453-1267 or your local software dealer.

For more information about Btrieve files, see your Btrieve documentation.

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

See Also
 For All Users
   <u>Setting Up the ODBC Btrieve Driver</u>

Setting Up the ODBC Btrieve Driver

To set up the ODBC Btrieve driver

1 In the Main group in the Program Manager window, double-click the Control Panel icon. In the Control Panel window, double-click the ODBC icon.

Note   For Microsoft Windows 3.0a, start the ODBC Administrator by double-clicking the ODBC Administrator icon in the Microsoft ODBC group.

---

2 In the Data Sources dialog box, choose the Drivers button.
3 In the Drivers dialog box, choose the Add button.
4 In the Add Driver dialog box, enter the name of the drive and directory containing the ODBC Btrieve driver in the text box.   Or choose the Browse button to select a drive and directory name.
5 Choose the OK button.
6 In the Install Drivers dialog box, choose Btrieve from the Available ODBC Drivers list.
7 Choose the OK button to install the driver.

Note   The ODBC Btrieve driver may share some of the same dynamic link libraries (DLLs) with other drivers installed on your computer.   If so, you will be asked to overwrite the ODBC Btrieve driver, regardless of whether it has been installed.   Choose the Yes button to install the driver.

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

To delete the ODBC Btrieve driver

1 In the Main group in the Program Manager window, double-click the Control Panel icon. In the Control Panel window, double-click the ODBC icon.

Note   For Windows 3.0a, start the ODBC Administrator by double-clicking the Microsoft ODBC Administrator icon in the Microsoft ODBC group.

---

2 In the Data Sources dialog box, choose the Drivers button.
3 In the Drivers dialog box, select the ODBC Btrieve driver from the Installed ODBC Drivers list.
4 Choose the Delete button.
   The ODBC Btrieve setup program asks if you want to remove the driver and all the data sources that use the driver.

Note   Deleting the the ODBC Btrieve driver will not delete the Btrieve for Windows dynamic-link library (WBTRCALL.DLL), since it may be used by programs that do not use ODBC.

---

5 Choose the Yes button.

See Also

For All Users

Hardware and Software Requirements

Adding, Modifying, and Deleting a Btrieve Data Source

Before you can access data with the ODBC Btrieve driver, you must add a data source for it.   You can change or delete a data source at any time.

To add a Btrieve data source

1 In the Main group in the Program Manager window, double-click the Control Panel icon. In the Control Panel window, double-click the ODBC icon.

Note   For Windows 3.0a, start the ODBC Administrator by double-clicking the Microsoft ODBC Administrator icon in the Microsoft ODBC group.

2 In the Data Sources dialog box, choose the Add button.

3 In the Add Data Source dialog box, select Btrieve Data (file.ddf) from the Installed ODBC Drivers list and choose OK.

4 In the ODBC Btrieve Setup dialog box, enter information to set up the data source.

To modify a Btrieve data source

1 In the Main group in the Program Manager window, double-click the Control Panel icon. In the Control Panel window, double-click the ODBC icon.

Note   For Windows 3.0a, start the ODBC Administrator by double-clicking the Microsoft ODBC Administrator icon in the Microsoft ODBC group.

2 In the Data Sources dialog box, select the data source from the Data Sources list.

3 Choose the Setup button.

4 In the ODBC Btrieve Setup dialog box, enter information to set up the data source.

To delete a Btrieve data source

1 In the Main group in the Program Manager window, double-click the Control Panel icon. In the Control Panel window, double-click the ODBC icon.

Note   For Windows 3.0a, start the ODBC Administrator by double-clicking the Microsoft ODBC Administrator icon in the Microsoft ODBC group.

2 In the Data Sources dialog box, select the data source from the Data Sources list.

3 Choose the Delete button, and then choose the Yes button to confirm the deletion.

See Also

For All Users

Connecting to a Btrieve Data Source

Setting Up the ODBC Btrieve Driver

Btrieve data source

A Btrieve data source specifies the default data directory in which the ODBC Btrieve driver searches for Btrieve files you want to access, as well as other database and driver information.   The ODBC Btrieve driver uses the information you enter to access data.

ODBC Btrieve Setup Dialog Box

The ODBC Btrieve Setup dialog box contains the following fields:

Data Source Name

The name of the data source.   This is usually something descriptive, such as Payroll or Personnel.

Description

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

Directory

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

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

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

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

```
USE C:\MYDIR
```

Select Directory

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

Use Current Directory

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.

Options

Displays the following options:

Caution   Except for the Exclusive option, these options apply to all data sources that use the ODBC Btrieve driver.

Collating Sequence

The sequence in which the fields are sorted: ASCII or International.

Page Timeout

Specifies the period of time, in tenths of a second, that a page (if not used) remains in the buffer before being removed.   The default is 600 tenths of a second (60 seconds).

Exclusive

If the Exclusive box is selected, Btrieve files will be opened in Exclusive mode and can be accessed by only one user at a time.   If the Exclusive box is cleared, Btrieve files will be opened in Shared mode and can be accessed by more than one user at a time. Performance is enhanced when running in Exclusive mode.

See Also

For All Users

Connecting to a Btrieve Data Source

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

See Also

For All Users

Adding, Modifying, and Deleting a Btrieve Data Source
Using the ODBC Btrieve Driver

For Advanced Users

Connection Strings (Advanced)

Using the ODBC Btrieve Driver

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

Column and Table Names

If column or table names contain any characters except letters, numbers, and underscores, they must be delimited.   To delimit a column or table name, enclose the name in double quotes(").

Character Sets

The ODBC Btrieve driver supports ASCII and International character sets.

.DDF Files

If .DDF files do not exist, they are created automatically at connection time and contain information about tables created after the .DDF files are created (not existing tables). When you set up a Btrieve data source, make sure that .DDF files for existing tables are in the data source directory you specify before connecting to the data source.

Passwords

The ODBC Btrieve driver cannot access data in password-protected Btrieve files.

Using Reserved Words

Do not use reserved words listed in the SQL grammar in Appendix C of the *Microsoft ODBC Programmers Reference* as identifiers (that is, table or column names), unless you surround the word in double quotation marks (for example, "DATE").

Connection Strings (Advanced)

The connection string for the ODBC Btrieve driver uses the following keywords:

Keyword	Description
DSN	Name of the Btrieve data source.
DBQ	The Btrieve database directory.
FIL	File type (Btrieve).

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

```
DSN=Accounting;DBQ=C:\BTRIEVE\DATA;FIL=BTRIEVE
```

See Also

For All Users

SQL Statements (Advanced)

The ODBC Btrieve driver supports all SQL statements and clauses in the ODBC minimum grammar.

For information about ODBC SQL grammar limitations, and additional and driver-specific grammar supported, see the following topics:

For Advanced Users

Additional Supported ODBC SQL Grammar (Advanced)

Driver-specific ODBC SQL Grammar (Advanced)

Limitations to ODBC SQL Grammar (Advanced)

See Also

For Advanced Users

[Data Types (Advanced)](#)

Additional Supported ODBC SQL Grammar (Advanced)

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

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

See Also

For Advanced Users

Driver-specific ODBC SQL Grammar (Advanced)

The ODBC Btrieve driver supports the following driver-specific ODBC SQL grammar:

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

example, EMP FROM C:\ABC\EMP).

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

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

USE [*drive*:]*dir*   Sets the current database directory. *drive* is a valid drive name and *dir* is any valid MS-DOS directory name.

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

```
USE C:\DBDIR
```

See Also

For Advanced Users

Additional Supported ODBC SQL Grammar (Advanced)

Limitations to ODBC SQL Grammar (Advanced)

Outer Joins (Advanced)

The ODBC Btrieve driver extends the OUTER JOIN syntax to support nested outer joins. The OUTER JOIN syntax is:

left-outer-join ::=
    table-reference **LEFT OUTER JOIN** *table-reference*
    ON *search-condition*

table-reference ::=
    table-name | [**(**] *left-outer-join* [**)**]

where *table-name* can be a table name or a table name followed by a correlation name. For example, the following statement uses a three-way outer join to create a list of sales orders.   For each sales order, all line numbers (if any) are listed, and for each line number, the part and description (if any) are listed.

```
SELECT Order.SONum,
   Line.LineNum,
   Part.PartNum,
   Part.Description
FROM Order LEFT OUTER JOIN
   (Line LEFT OUTER JOIN Part
   ON Line.PartNum=Part.PartNum)
   ON Order.SONum=Line.SONum
```

Note   The rightmost ON corresponds to the leftmost LEFT OUTER JOIN.

See Also

For Advanced Users

Additional Supported ODBC SQL Grammar (Advanced)
Limitations to ODBC SQL Grammar (Advanced)

Limitations to ODBC SQL Grammar (Advanced)

The ODBC Btrieve driver and Btrieve impose the following limitations on the ODBC SQL grammar.

Grammar	Limitation
Character string literals	Any ANSI character (1 - 255 decimal). Use two consecutive single quotation marks ('') to represent one single quotation mark (')
Column names	Maximum length is 20 characters.
Columns	Maximum number is 255 when a table is created.
Comparison predicate	For SQL_BIT data, comparisons can be made for equality and inequality only (= and <> operators).
LIKE Predicate	If data in a column is longer than 255 characters, the LIKE comparison will be based only on the first 255 characters.
Literals	Maximum length is 1000 characters.
Predicates	Maximum number is 300.
Row Length	The maximum length of a row in a table must be less than or equal to 4096.
Set Functions	AVG, MAX, MIN, and SUM do not support the DISTINCT keyword.
Sort Keys	The maximum length of a sort key in a GROUP BY clause, ORDER BY

clause, SELECT DISTINCT statement, or outer join is 255 bytes; the maximum length of all sort keys in a sort row is 65,500 bytes.

If the length of the data in a column is greater than 255 characters, sorting will be based on the first 255 characters.

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

Data Types (Advanced)

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

Btrieve data type	ODBC data type
DATE	SQL_DATE
FLOAT4	SQL_REAL
FLOAT8	SQL_DOUBLE
INTEGER1	SQL_TINYINT
INTEGER2	SQL_SMALLINT
INTEGER4	SQL_INTEGER
LOGICAL1	SQL_BIT
MONEY	SQL_NUMERIC
NOTE	SQL_LONGVARCHAR
STRING	SQL_CHAR

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

Limitations

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

- The maximum length of a NOTE column is 4,000 bytes.
- The driver does not allow indexes on NOTE or INTEGER1 fields.
- When converting Btrieve data to the C data type SQL_C_TINYINT, numbers from 0 to 127 are converted correctly.   Numbers from 128 to 255 are converted to numbers from -128 to -1.   Numbers less than 0 or greater than 255 cannot be converted.
  When converting data from the C data type SQL_C_TINYINT to Btrieve data, numbers from 0 to 127 are converted correctly.   Numbers from -128 to -1 are converted to numbers from 128 to 255.
  This occurs because SQL_C_TINYINT is signed, but the ODBC Btrieve driver uses unsigned single-byte integers.

See Also

For Advanced Users

[SQL Statements (Advanced)](#)

Error Messages (Advanced)

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

Native Error

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

SQLSTATE

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

Error Message

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

Error messages have the following format:

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

where the prefixes in brackets ([ ]) identify the location of the error.   When the error occurs in the Driver Manager or Simba driver, *data-source* is not given.   When the error occurs in the data source, the [*vendor*] and [*ODBC-component*] prefixes identify the vendor and name of the ODBC component that received the error from the data source.

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

Error message	Error location
[Microsoft][ODBC DLL]*message-text*	Driver Manager (ODBC.DLL)
[Microsoft][ODBC Single-Tier Driver]*message-text*	Simba Driver (SIMBA.DLL)
[Microsoft][ODBC Single-Tier Driver][ODBC File Library]*message-text*	Btrieve ISAM (BTRV110.DLL)

SQLGetInfo Return Values (Programming)

The following table lists the C language #defines for the *fInfoType* argument and the corresponding values returned by **SQLGetInfo**.   This information can be retrieved by passing the listed C language #defines to **SQLGetInfo** in the *fInfoType* argument.   Where **SQLGetInfo** returns a 32-bit bitmask, a vertical bar (|) represents a bitwise OR.   For more information about the values return by **SQLGetInfo**, see the *Microsoft ODBC SDK Programmer's Reference, Version 1.0*.

fInfoType Value (#define)	Returned value
SQL_ACCESSIBLE_PROCEDURES	"N"
SQL_ACCESSIBLE_TABLES	"N"
SQL_ACTIVE_CONNECTIONS	0
SQL_ACTIVE_STATEMENTS	0
SQL_CONCAT_NULL_BEHAVIOR	1
SQL_CONVERT_BIGINT	0
SQL_CONVERT_BINARY	0
SQL_CONVERT_BIT	SQL_CVT_BIT | SQL_CVT_CHAR | SQL_CVT_DOUBLE | SQL_CVT_INTEGER | SQL_CVT_LONGVARCHAR | SQL_CVT_NUMERIC | SQL_CVT_REAL | SQL_CVT_SMALLINT | SQL_CVT_TINYINT
SQL_CONVERT_CHAR	SQL_CVT_BIT | SQL_CVT_CHAR | SQL_CVT_DATE | SQL_CVT_DOUBLE | SQL_CVT_INTEGER | SQL_CVT_LONGVARCHAR | SQL_CVT_REAL | SQL_CVT_SMALLINT | SQL_CVT_TINYINT
SQL_CONVERT_DATE	SQL_CVT_CHAR | SQL_CVT_DATE
SQL_CONVERT_DECIMAL	0
SQL_CONVERT_DOUBLE	SQL_CVT_BIT | SQL_CVT_CHAR | SQL_CVT_DOUBLE | SQL_CVT_INTEGER | SQL_CVT_LONGVARCHAR | SQL_CVT_NUMERIC | SQL_CVT_REAL | SQL_CVT_SMALLINT | SQL_CVT_TINYINT
SQL_CONVERT_FLOAT	SQL_CVT_BIT | SQL_CVT_CHAR | SQL_CVT_DOUBLE | SQL_CVT_INTEGER | SQL_CVT_LONGVARCHAR | SQL_CVT_NUMERIC | SQL_CVT_REAL | SQL_CVT_SMALLINT | SQL_CVT_TINYINT

SQL_CONVERT_FUNCTIONS	SQL_FN_CVT_CONVERT
SQL_CONVERT_INTEGER	SQL_CVT_BIT | SQL_CVT_CHAR | SQL_CVT_DOUBLE | SQL_CVT_INTEGER | SQL_CVT_LONGVARCHAR | SQL_CVT_NUMERIC | SQL_CVT_REAL | SQL_CVT_SMALLINT | SQL_CVT_TINYINT
SQL_CONVERT_LONGVARBINARY	0
SQL_CONVERT_LONGVARCHAR	SQL_CVT_CHAR | SQL_CVT_LONGVARCHAR
SQL_CONVERT_NUMERIC	SQL_CVT_BIT | SQL_CVT_CHAR | SQL_CVT_DOUBLE | SQL_CVT_INTEGER | SQL_CVT_LONGVARCHAR | SQL_CVT_NUMERIC | SQL_CVT_REAL | SQL_CVT_SMALLINT | SQL_CVT_TINYINT
SQL_CONVERT_REAL	SQL_CVT_BIT | SQL_CVT_CHAR | SQL_CVT_DOUBLE | SQL_CVT_INTEGER | SQL_CVT_LONGVARCHAR | SQL_CVT_NUMERIC | SQL_CVT_REAL | SQL_CVT_SMALLINT | SQL_CVT_TINYINT
SQL_CONVERT_SMALLINT	SQL_CVT_BIT | SQL_CVT_CHAR | SQL_CVT_DOUBLE | SQL_CVT_INTEGER | SQL_CVT_LONGVARCHAR | SQL_CVT_NUMERIC | SQL_CVT_REAL | SQL_CVT_SMALLINT | SQL_CVT_TINYINT
SQL_CONVERT_TIME	SQL_CVT_CHAR
SQL_CONVERT_TIMESTAMP	SQL_CVT_CHAR | SQL_CVT_DATE
SQL_CONVERT_TINYINT	SQL_CVT_BIT | SQL_CVT_CHAR | SQL_CVT_DOUBLE | SQL_CVT_INTEGER | SQL_CVT_LONGVARCHAR | SQL_CVT_NUMERIC | SQL_CVT_REAL | SQL_CVT_SMALLINT | SQL_CVT_TINYINT
SQL_CONVERT_VARBINARY	0
SQL_CONVERT_VARCHAR	0
SQL_CORRELATION_NAME	2
SQL_CURSOR_COMMIT_BEHAVIOR	2

SQL_CURSOR_ROLLBACK_BEHAVIOR	0
SQL_DATA_SOURCE_READ_ONLY	"N" (The driver does not check to see whether the disk drive is read-only.)
SQL_DBMS_NAME	"BTRIEVE"
SQL_DBMS_VER	"5.1"
SQL_DEFAULT_TXN_ISOLATION	0
SQL_DRIVER_NAME	"SIMBA.DLL"
SQL_DRIVER_VER	" 1.01.*nnnn*" (*nnnn* specifies the build date.)
SQL_EXPRESSIONS_IN_ORDERBY	"Y"
SQL_FETCH_DIRECTION	SQL_FD_FETCH_NEXT
SQL_IDENTIFIER_CASE	4
SQL_IDENTIFIER_QUOTE_CHAR	""" (double quotation mark).
SQL_MAX_COLUMN_NAME_LEN	20
SQL_MAX_CURSOR_NAME_LEN	18
SQL_MAX_OWNER_NAME_LEN	0
SQL_MAX_PROCEDURE_NAME_LEN	0
SQL_MAX_QUALIFIER_NAME_LEN	66
SQL_MAX_TABLE_NAME_LEN	20
SQL_MULT_RESULT_SETS	"N"
SQL_MULTIPLE_ACTIVE_TXN	"N"
SQL_NON_NULLABLE_COLUMNS	1
SQL_NUMERIC_FUNCTIONS	SQL_FN_NUM_MOD
SQL_ODBC_API_CONFORMANCE	1
SQL_ODBC_SAG_CLI_CONFORMANCE	1
SQL_ODBC_SQL_CONFORMANCE	0
SQL_ODBC_SQL_OPT_IEF	"N"
SQL_OUTER_JOINS	"Y"
SQL_OWNER_TERM	""
SQL_PROCEDURE_TERM	""
SQL_PROCEDURES	"N"
SQL_QUALIFIER_NAME_SEPARATOR	"\" (backslash)
SQL_QUALIFIER_TERM	"DIRECTORY"
SQL_ROW_UPDATES	"Y"
SQL_SCROLL_CONCURRENCY	SQL_SCCO_READ_ONLY
SQL_SCROLL_OPTIONS	SQL_SO_FORWARD_ONLY
SQL_SEARCH_PATTERN_ESCAPE	"\" (backslash)
SQL_SERVER_NAME	"BTRIEVE"
SQL_STRING_FUNCTIONS	SQL_FN_STR_CONCAT | SQL_FN_STR_LCASE | SQL_FN_STR_LEFT | SQL_FN_STR_LENGTH | SQL_FN_STR_LOCATE | SQL_FN_STR_LTRIM | SQL_FN_STR_RIGHT | SQL_FN_STR_RTRIM | SQL_FN_STR_SUBSTRING | SQL_FN_STR_UCASE
SQL_SYSTEM_FUNCTIONS	SQL_FN_SYS_DBNAME |

	SQL_FN_SYS_USERNAME
SQL_TABLE_TERM	"TABLE"
SQL_TIMEDATE_FUNCTIONS	SQL_FN_TD_CURDATE | SQL_FN_TD_CURTIME | SQL_FN_TD_DAYOFMONTH | SQL_FN_TD_DAYOFWEEK | SQL_FN_TD_MONTH | SQL_FN_TD_YEAR
SQL_TXN_CAPABLE	0
SQL_TXN_ISOLATION_OPTIONS	0

See Also
For Advanced Users
    SQL Statements (Advanced)
    Data Types (Advanced)

For Programmers
    Scalar Functions (Programming)

Scalar Functions (Programming)

The ODBC Btrieve driver supports the following scalar functions:

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

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

ODBC API Functions (Programming)

The ODBC Btrieve driver supports all Core and Level 1 functions and the following Level 2 functions:

- SQLDataSources
- SQLMoreResults

These ODBC API functions have the following implementations with the ODBC Btrieve driver:

Function	Description
SQLDriverConnect	The following keywords are supported in the <u>connection string</u>: **DSN**, **DBQ**, and **FIL**.
SQLGetConnectOption and **SQLSetConnectOption**	These functions support the SQL_ACCESS_MODE, SQL_CURRENT_QUALIFIER, SQL_OPT_TRACE, and SQL_OPT_TRACEFILE connection options. **SQLGetConnectOption** also supports the SQL_AUTOCOMMIT option.
SQLGetCursorName and **SQLSetCursorName**	These functions are supported, but cannot be used for positioned updates or deletes (for example, WHERE CURRENT OF *cursorname*).
SQLGetData	This function can retrieve data from any column, whether or not there are bound columns after it and regardless of the order in which the columns are retrieved.
SQLGetInfo	**SQLGetInfo** supports a driver-specific information type, SQL_FILE_USAGE (65002). The returned value is a 16-bit integer that indicates how the driver directly treats files in a data source:
	0 (SQL_FILE_NOT_SUPPORTED) = The driver is not a single-tier driver.
	1 (SQL_FILE_TABLE) = A single-tier driver treats files in a data source as tables.
	3 (SQL_FILE_QUALIFIER) = A single-tier driver treats files in a data source as a qualifier.
	The ODBC Btrieve driver returns 3, since the FILE.DDF file provides information about a complete Btrieve database.
SQLGetStmtOption and **SQLSetStmtOption**	These functions support the SQL_MAX_LENGTH, SQL_MAX_ROWS, and

	SQL_NOSCAN statement options.
	They also support a driver-specific statement option, SQL_LOCK_TABLES.
SQLGetTypeInfo	Only data type names returned by **SQLGetTypeInfo** can be used with CREATE statements.
SQLMoreResults	**SQLMoreResults** always returns SQL_NO_DATA_FOUND. It cannot return additional results.
SQLTransact	The ODBC Btrieve driver does not support transactions; all SQL statements that modify data are always committed. Therefore, **SQLTransact** always returns SQL_SUCCESS when *fType* is SQL_COMMIT and always returns SQL_ERROR when *fType* is SQL_ROLLBACK.

See Also

For Advanced Users

Error Messages (Advanced)

For Programmers

Implementation Issues (Programming)

SQL_LOCK_TABLES Statement Option (Programming)

The ODBC Btrieve driver supports a driver-specific statement option, SQL_LOCK_TABLES, that supports the values DEFLOCK and XLOCK.   When the value of the option is DEFLOCK, tables used by the *hstmt* are subject only to the default locking mechanisms of the ODBC Btrieve driver.   This is the default setting.

When the value of the option is XLOCK, all tables used by the *hstmt* are exclusively locked when a **SELECT**, **UPDATE**, **INSERT**, **DELETE**, or **CREATE INDEX** statement is executed on the *hstmt*.   The tables remain locked until the *hstmt* is dropped (by calling **SQLFreeStmt** with the SQL_DROP option) or the option is set to DEFLOCK and the *hstmt* is reexecuted.

Locked tables can only be used by the locking *hstmt*; they cannot be used by any other *hstmts*.   For example, the last function call in the following code generates an access violation because it uses a different *hstmt*:

```
SQLSetStmtOption(hstmt1,SQL_LOCK_TABLES,XLOCK);
SQLExecDirect(hstmt1,"SELECT * FROM EMP",SQL_NTS);
SQLExecDirect(hstmt1,"UPDATE EMP SET DEPT=12",SQL_NTS);
/* This call generates an access violation */
SQLExecDirect(hstmt2,"SELECT * FROM EMP",SQL_NTS);
```

Furthermore, **SQLColumns**, **SQLSpecialColumns**, and **SQLStatistics** cannot retrieve information about a table locked on another *hstmt*.

Caution   Be careful to avoid locking more files than necessary.   If the same *hstmt* is used to execute statements for many different tables, all the tables will remain locked until the *hstmt* is dropped or locking is turned off.

The SQL_LOCK_TABLES statement option uses the following #defines:

```
#define SQL_LOCK_TABLES 1153
#define DEFLOCK 0
#define XLOCK 1
```

The SQL_LOCK_TABLES statement option cannot be used with **SQLSetConnectOption**.

Implementation Issues (Programming)

The following implementation-specific issues might affect the use of the ODBC Btrieve driver.

Arithmetic Errors

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

Closing Open Tables (Files)

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

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

```
SQLPrepare(hStmt,"SELECT * FROM emp,dept
WHERE emp.dept = dept.dept_id",SQL_NTS);

SQLExecute(hStmt);

/*.SQLFetch until NO_DATA_FOUND */

SQLFreeStmt(hStmt,SQL_CLOSE);

SQLPrepare(hStmt,"SELECT * FROM emp",SQL_NTS);
```

Note   Each file used by the ODBC Btrieve driver requires a file handle.   Because tables (files) remain open until **SQLFreeStmt** is called with the SQL_DROP option, reusing an *hstmt* for different tables without dropping it can result in an error caused by attempting to open too many files.

Sorting with DISTINCT, GROUP BY, or ORDER BY

DISTINCT, GROUP BY, and ORDER BY always result in a sort.   If indexes are found, data is dynamically fetched and the sort is based on those indexes.   If indexes are not found, a temporary table called a snapshot is taken of the data and the sort occurs on the temporary table.   This type of sort is not based on dynamic data since the temporary table is built from data found in the original data file at **SQLExecute** time.

API

Application programming interface.   A set of routines that an application, such as Microsoft Access, uses to request and carry out lower-level services.

character set

A character set is a set of 256 letters, numbers, and symbols specific to a country or language. Each character set is defined by a table called a code page. An OEM (Original Equipment Manufacturer) character set is any character set except the ANSI character set. The ANSI character set (code page 1007) is the character set used by Microsoft Windows.

conformance level

Some applications can use only drivers that support certain levels of functionality, or conformance levels.   For example, an application might require that drivers be able to prompt the user for the password for a data source.   This ability is part of the Level 1 conformance level for the application programming interface (API).

Every ODBC driver conforms to one of three API levels (Core, Level 1, or Level 2) and one of three SQL grammar levels (Minimum, Core, or Extended).   Drivers may support some of the functionality in levels above their stated level.

For detailed information about conformance levels, programmers should see the *Microsoft ODBC SDK Programmer's Reference*.

data source

A data source includes the data a user wants to access and the information needed to get to that data.   Examples of data sources are:

- A SQL Server database, the server on which it resides, and the network used to access that server.
- A directory containing a set of dBASE files you want to access.

DBMS

Database management system.   The software used to organize, analyze, search for, update, and retrieve data.

DDL

Data definition language.   Any SQL statement that can be used to define data objects and their attributes.   Examples include CREATE TABLE, DROP VIEW, and GRANT statements.

DLL

Dynamic-link library.   A set of routines that one or more applications can use to perform common tasks.   The ODBC drivers are DLLs.

DML

Data manipulation language.   Any SQL statement that can be used to manipulate data.
Examples include UPDATE, INSERT, and DELETE statements.

ODBC

Open Database Connectivity.   A Driver Manager and a set of ODBC drivers that enable applications to access data using SQL as a standard language.

ODBC Driver Manager
  A dynamic-link library (DLL) that provides access to ODBC drivers.

ODBC driver

A dynamic-link library (DLL) that an ODBC-enabled application, such as Microsoft Excel, can use to gain access to a particular data source.   Each database management system (DBMS), such as Microsoft SQL Server, requires a different driver.

SQL

Structured Query Language.   A language used for retrieving, updating, and managing data.

SQL statement

A command written in Structured Query Language (SQL); also known as a query. An SQL statement specifies an operation to perform, such as SELECT, DELETE, or CREATE TABLE; the tables and columns on which to perform that operation; and any constraints to that operation.

translation option

An option that specifies how a translator translates data.   For example, a translation option might specify the character sets between which a translator translates character data.   It might also provide a key for encryption and decryption.

translator

A dynamic-link library (DLL) that translates all data passing between an application, such as Microsoft Access, and a data source.   The most common use of a translator is to translate character data between different character sets.   A translator can also perform tasks such as encryption and decryption or compression and expansion.