

## **Contents for Text ODBC Driver**

[Text ODBC Setup](#)

[Text ODBC Define File](#)

[Text ODBC Parse File](#)

[Text ODBC Date Masks](#)

## Additional Information for Configuring a Text Data Source

**Data Source Name:** A string that identifies this Text data source configuration in ODBC.INI. Examples include "Accounting" or "Text Files."

**Description:** An optional long description of a data source name. For example, "My Accounting Files" or "My Text Files in the Accounting Directory."

**Database Directory:** The directory in which the text files are stored. If none is specified, the current working directory is used.

The following values are optional:

**Rows to Scan:** The number of rows in a text file that the driver scans to determine the data types in the file. If the value is 0, all rows in the file are scanned. The default is 25.

**Default Table Type:** The type of text file: comma-separated, tab-separated, character-separated, or fixed length. This value tells the driver the default type, which is used when creating a new table and opening an undefined table.

**Delimiter Character:** The character used as a delimiter for character-separated files. It can be any printable character. The default is a comma (,).

**Action for Undefined Tables:** Two radio buttons that indicate what action the driver should take when it encounters a file that has not been defined. Set the Prompt for Definition radio button, if you want the driver to prompt the user when it encounters a file whose format is not defined. Otherwise, set the Guess Definition radio button; in this case, the driver guesses the file's format.

**Return Additional Tables:** Select this check box to tell the driver to return files you have defined in functions like SQLTables, SQLColumns, etc. *and* files with a given extension. In Extension List, specify a comma-separated list of the extensions. To have files with no extensions returned, specify NONE. For example, if some of your files have the extensions TXT and CSV and others have no extension, specify TXT,CSV,NONE.

By default, when an application requests a list of tables, only files that have been defined are returned.

**File Open Cache:** A numeric value to specify the maximum number of unused file opens to cache. For example, the value 4 specifies that when a user opens and closes four tables, the tables are not actually closed. The driver keeps them open so that if another query uses one of these tables, the driver does not have to perform another open, which is expensive. The advantage of file open caching is increased performance. The disadvantage is that a user who specifies file locking on open may get a locking conflict even though no one appears to have the file open. The default is 0, which means no file open caching.

**Cache Size:** The amount of memory, in 64K blocks, that the driver uses to cache database records. The higher the number, the better the performance. The maximum number you can set depends on the system memory available. This value must be a multiple of 64. The default is 256K. If the cache size is greater than 0, when browsing backwards, you will not be able to see updates made by other users until you reexecute the Select statement.

**Column Names in First Line:** Select this check box to tell the driver to look for column names in the first line of the file.

**International Sort:** A setting to indicate the order in which records are retrieved when you issue a

Select statement with an Order By clause. Select this check box to use the international sort order as defined by your operating system. International sort order is case-insensitive (*a* precedes *B*); the sorting of accented characters is also affected (see your operating system documentation). Leave this box blank to use the ASCII sort order. ASCII sort order is case-sensitive, where uppercase letter precede lowercase letters (*B* precedes *a*).

**Perform OEM to ANSI Translation:** Select this check box to indicate that the data is stored in the IBM PC character set. If your data is stored in the ANSI character set, do not select this check box. This changes the display format only; the data is not converted.

**Note:** The Rows to Scan, Default Table Type, Delimiter Character, and Column Names in First Line settings apply only to tables *not* previously defined. These fields also determine the attributes of new tables created with the Create Table statement.

## Defining Table Structure

Since text files do not all have the same structure, the driver provides the option of defining the structure of an existing file. Although defining the structure is not mandatory, since the driver can attempt to guess the names and types of the columns, this feature is extremely useful.

In this dialog box, the Database Name field and the File field are informational fields only. The Database Name field displays the name of the database directory that contains the file. The File field displays the name of the file you previously selected.

1. In the Table Information section of this dialog box, specify information about the overall structure of the file:
  - a. In the Table field, enter a table name. Values may be up to 32 characters in length and may not be the same as another defined table in the database. This name is returned by SQLTables. By default, it is the filename without its extension.
  - b. Select the Column Names in First Line check box if the first line of the file contains column names. Otherwise, do not select this check box.
  - c. Open the Table Type box and select a table type: comma, tab, fixed, or character.
  - d. If the table type you select is character-separated, enter the character that separates the values in the Delimiter Character field.
  - e. Select the OEM to ANSI Translation check box to tell the driver that the data is stored in the IBM PC character set. If your data is stored in the ANSI character set, do not select this check box. This changes the display format only; the data is not converted.
2. In the Column Information section, define the types and names of the columns in the table. The box in the upper-left corner of this section lists the defined columns.

If you specified a comma-separated, tab-separated, or character-separated table type, a **Guess** button is displayed in this section. Click **Guess** to tell the driver to guess the fields. The driver then displays what it thinks the fields are. You can then modify the field definitions by clicking **Modify** and completing the Name, Precision, and Scale fields. If you do not want the driver to guess the fields, take the following steps to define the fields:

- a. In the Name field, type the name of the field.
- b. Open the Type box and select the data type of the field. If the field type is date, then you must select a date mask for the field or type one in.
- c. In the Precision field, type the precision of the field.
- d. In the Scale field, type the scale of the field. The precision and scale values determine how numeric data is to be returned.
- e. If you specified a fixed-length table type, you may enter the length and offset in the Length and Offset fields, or you can specify a parse string. The length is the number of bytes the data takes up in storage; the offset is the number of bytes from the start of the table to the start of the field.
- f. Click **Add** to add this field definition to the list box.

To modify the currently selected field in the list box, click **Modify**.

To remove the currently selected field in the list box, click **Remove**.

## **Parsing Fixed Length Files**

This dialog box displays the first line of the file. You must mark where each field begins and ends by enclosing it in brackets. These brackets indicate the position and length of each field value in the record. You must do this for all the fields in the table.

## Date Masks

Date masks tell the driver how a date is stored in a text file. When a value is inserted into the text file, the date is formatted so that it matches the mask. When reading the text file, the driver converts the formatted date into a date data type.

The following table lists the symbols to use when specifying the date mask:

<b>Symbol</b>	<b>Description</b>
m	Output the month's number (1-12).
mm	Output a leading zero if the month number is less than 10.
mmm, Mmm, MMM the	Output the three-letter abbreviation for the month depending on the case of M's (that is, jan, Jan, JAN).
mmmm, Mmmm, MMMM	Output the full month name depending on the case of the M's (that is, january, january, JANUARY).
d	Output the day number (1-31).
dd	Output a leading zero if the day number is less than 10.
ddd, Ddd, DDD	Output the three-letter day abbreviation depending on the case of the D's (that is, mon, Mon, MON).
dddd, Dddd, DDDD	Output the day depending on the case of the D's (that is, monday, Monday, MONDAY).
yy	Output the last two digits of the year.
yyyy	Output the full four digits of the year.
J	Output the Julian value for the date. The Julian value is the number of days since 4712 BC.
\ - . : , (space)	Special characters used to separate the parts of a date.
\	Output the next character. For example, if the mask is mm/dd/yyyy \A\D, the value appears as 10/01/1993 AD in the text file.
"string", 'string'	Output the string in the text file.

Some example date values, masks, and how the date appears in the text file:

<b>Date</b>	<b>Mask</b>	<b>Value</b>
1993-10-01	yyyy-mm-dd	1993-10-01
	m/d/yy	10/1/93
	Ddd, Mmm dd, yyyy	Fri, Oct 01, 1993

