DataDirect ODBC Text Driver

About the Text Driver

The Text driver supports ASCII text files. These files can be printed directly or edited with text editors or word processors, since none of the data is stored in a binary format.

The driver filename is IVTXT*nn*.DLL.

The Text driver executes SQL statements directly on the text files. The driver supports Insert statements, and inserts the record at the end of the file. However, you may not execute Update or Delete statements.

Common Text File Formats

Some common formats for text files are listed in the following table:

Format	Description
Comma-separated values	Commas separate column values, and each line is a separate record. Column values can vary in length. These files often have the .CSV extension.
Tab-separated values	Tabs separate column values, and each line is a separate record. Column values can vary in length.
Character-separated values	Any printable character except single or double quotation marks can separate column values, and each line is a separate record. Column values can vary in length.
Fixed	No character separates column values. Instead, values start at the same position and have the same length in each line. The values appear in fixed columns if you display the file. Each line is a separate record.
Stream	No character separates column values nor records. The table is one long stream of bytes.

Comma-, tab-, and character-separated files are called character-delimited files, since values are separated by a special character.

Configuring Data Sources

To configure a Text data source, do the following:

1 Start the ODBC Administrator. A list of data sources appears.

2 If you are configuring a new data source, click **Add**. A list of installed drivers appears. Select INTERSOLV TextFile and click **OK**.

If you are configuring an existing data source, select the data source name and click **Setup**.

The Text ODBC Setup dialog box appears.

3 Specify a data source name, a database directory, and optionally, a description. a default table type, a delimiter character, and whether to use column names in the first line.

4 Click the **Advanced** button to configure optional data source settings, such as server list and database name.

The ODBC Advanced Driver Setup dialog box appears.

5 Click **Translate** to perform a translation of your data from one character set to another. The Select Translator dialog box appears in which you can select a translator to peform the translation. INTERSOLV provides a translator named INTERSOLV OEM ANSI that translates your data from the IBM PC character set to the ANSI character set. Click **OK** to leave this dialog box and perform the translation.

The translators that are listed in this dialog box are determined by the values listed in the ODBC Translators section of your ODBCINST.INI file.

6 Click **OK** to write these values to ODBC.INI. These values are now the defaults when you connect to the data source. You can change the defaults by configuring your data source again. You can override the defaults by connecting to the data source using a connection string with alternate values.

Click **Define** in this dialog box to define the structure of your text files. Defining Table Structure discusses how to define the column names and data types in a table.

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.

Define the structure of a file as follows:

1. Click **Define** in the ODBC Text Driver Setup dialog box, which you can access through the ODBC Administrator. The Define File dialog box appears.
2. Select the correct file and click **OK** to define the file's structure using the Define Table dialog box.

   In the Define Table dialog box, Database Name and File display the name of the database directory that contains the file and the name of the file you previously selected, respectively.
3. In the Table Information section of this dialog box, specify information about the overall structure of the file:
   - **a** In the Table box, type 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, character or stream.
   - **d** If the table type is character-separated, type the character that separates the values in the Delimiter Character box.
   - **e** If you are using Windows, 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.
4. 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, the **Guess** button appears 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 specifying values in the Name, Type, Mask, Precision, and Scale boxes (as appropriate) and clicking **Modify**. If you do not want the driver to guess the fields, take the following steps to define the fields:
   - **a** In the Name box, type the name of the field.
   - **b** Open the Type drop-down list 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. See the following section, "Date Masks," for more information.
   - **c** In the Precision box, type the precision of the field.
   - **d** In the Scale box, 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 specify the length and offset in the Length and Offset boxes, 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 selected field in the list box, click **Modify**.

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

If you specified a fixed-length table type, the **Parse** button is displayed. If you have not entered values in the length and offset boxes, click **Parse** to define the columns of the table. The Parse Table dialog box appears.

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. Click **OK** when you are done.

5 Click **OK** to define the table.

Date Masks

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

Symbol	Description
m	Output the month's number (1-12).
mm	Output a leading zero if the month number is less than 10.
mmm, Mmm, MMM	Output the three-letter abbreviation for the month depending on the case of the 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.

The following table shows some example date values, masks, and how the date appears in the text file.

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

Connecting to a Text Database Using a Connection String

If your application requires a connection string to connect to a data source, you must specify the data source name that tells the driver which ODBC.INI section to use for the default connection information. Optionally, you may specify *attribute=value* pairs in the connection string to override the default values stored in ODBC.INI. These values are not written to ODBC.INI.

You can specify either long or short names in the connection string. The connection string has the form:
`DSN=data_source_name[;attribute=value[;attribute=value]...]`

An example of a connection string for text files is
`DSN=TEXT FILES;TT=CHARACTER;DC=&`

The following table gives the long and short names for each attribute, as well as a description.

The defaults listed in the table are initial defaults that apply when no value is specified in either the connection string or in the data source definition in ODBC.INI. If you specified a value for the attribute when configuring the data source, that value is your default.

Attribute	Description
DataSourceName (DSN)	A string that identifies a Text data source configuration in ODBC.INI. Examples include "Accounting" or "Text Files."
Database (DB)	The directory in which the text files are stored.
ScanRows (SR)	The number of rows in a text file that the driver scans to determine the column types in the file. If the value is 0, all rows in the file are scanned. The initial default is 25.
TableType (TT)	TableType={Comma | Tab | Character | Fixed | Stream} The Text driver supports five types: comma-separated, tab-separated, character-separated, fixed length, and stream. Setting this value tells the driver the default type, which is used when creating a new table and opening an undefined table.
Delimiter (DC)	The character used as a delimiter for character-separated files. It can be any printable character except a single or double quote. The initial default is a comma (,).
UndefinedTable (UT)	The Text driver can perform two operations when it encounters a file that has not been defined. UndefinedTable=PROMPT tells the driver to display a dialog box that allows the user to describe the file's format. UndefinedTable=GUESS tells the driver to guess the file's format. This is the initial default.
ExtraExtensions (EE)	A list of additional filename extensions to be recognized as text tables. When an application requests a list of tables, only files that have been defined are returned. To have the driver also return names of undefined files, specify a comma-separated list of file extensions. To specify files with no extension, use the keyword NONE.
FileOpenCache	The maximum number of unused file opens to

(FOC)	cache. For example, when FileOpenCache=4, and a user opens and closes four files, the files are not actually closed. The driver keeps them open so that if another query uses one of these files, the driver does not have to perform another open, which is expensive. The advantage of using file open caching is increased performance. The disadvantage is that a user who tries to open the file exclusively may get a locking conflict even though no one appears to have the file open. The initial default is 0.
CacheSize (CSZ)	The number of 64K blocks the driver uses to cache database records. The higher the number, the better the performance. The maximum number of blocks you can set depends on the system memory available. 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. The default is 4.
FirstLineNames (FLN)	FirstLineNames={0 | 1}. A value that determines whether the driver looks for column names in the first line of the file. If FirstLineNames=1, the driver looks for column names in the first line of the file. If FirstLineNames=0 (the initial default), the first line is interpreted as the first record in the file.
DataFileExtension (DFE).	String of three or fewer characters that specifies the file extension to use for data files. The default DataFileExtension value is TXT. The DataFileExtension value is used for all CREATE TABLE statements. Sending a CREATE TABLE using an extension other than the DataFileExtension value causes an error. In other SQL statements such as SELECT or INSERT, users can specify an extension other than the DataFileExtension value. The DataFileExtension value is used when no extension is specified.
UseLongQualifiers (ULQ)	UseLongQualifiers={0 | 1}. This attribute specifies whether the driver uses long pathnames as table qualifiers. When you set this check box, pathnames can be up to 255 characters. The default length for pathnames is 128 characters.
IntlSort (IS)	IntlSort={0 | 1}. A value that determines the order that records are retrieved when you issue a Select statement with an Order By clause. If IntlSort=1, the driver uses the international sort order as defined by your operating system. The sort is case-insensitive (*a* precedes *B*); the sorting of accented characters is also affected (see your operating system documentation). If IntlSort=0 (the initial default), the driver uses the ASCII sort order, where uppercase letters precede lowercase

letters (*B* precedes *a*).

Note: The ScanRows, TableType, Delimiter, FirstLineNames, and Charset attributes apply to tables that have *not* been defined. These attributes also determine the characteristics of new tables created with the Create Table statement.

Data Types

The text file data types are mapped to the standard ODBC data types as follows:

Text	ODBC
Numeric	SQL_NUMERIC
Date	SQL_DATE
Varchar	SQL_VARCHAR

SQL Statements for the Text Driver

Select Statements

You use the SQL Select statement to specify the columns and records to be read. The driver supports all Select statement clauses as described in SQL for Flat-File Drivers.

Alter Table Statement

The Text driver supports the Alter Table statement to add one or more columns to the table, or to delete (drop) a single column.

The Alter Table statement has the form:

```
ALTER TABLE table_name {ADD column_name data_type | ADD (column_name data_type [, column_name
data_type] . . . ) |
DROP [COLUMN] column_name}
```

table_name is the name of the table for which you are adding or dropping columns.

column_name assigns a name to the column you are adding or specifies the column you are dropping.

data_type specifies the native data type of each column you add. See Text Data Types for more information.

For example, to add two columns to the emp table,

```
ALTER TABLE emp {ADD startdate date, dept char 10)
```

You cannot add and drop columns in a single statement, and you can drop only one column at a time. For example, to drop a column,

```
ALTER TABLE emp DROP startdate
```

ODBC Conformance Level

The Text driver supports the Core, Level 1, and Level 2 API functions listed in Supported ODBC Functions. It also supports backward and random fetching in SQLExtendedFetch.

The driver supports the minimum SQL grammar.

Number of Connections and Statements Supported

Text files support multiple connections and multiple statements per connection.

Text ODBC Setup Dialog

Use the Text ODBC Setup dialog to create new data sources or configure existing data sources.

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.

Default Table Type: The type of text file: comma-separated, tab-separated, character-separated, fixed length or stream. 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 (,).

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.

Note: The 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.

Advanced

Displays the ODBC Advanced Driver Setup dialog box to configure optional data source settings, such as server name.

OK

Creates or modifies the current data source using the options you specify.

Cancel

Exits the ODBC Setup dialog without creating or modifying a data source.

Advanced Dialog

To configure optional settings for a Text data source, specify values as follows:

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.

Note: The Rows to Scan setting applies only to tables *not* previously defined. These fields also determine the attributes of new tables created with the Create Table statement.

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.

Data File Extension: Specifies the file extension to use for data files. The default Data File Extension setting is TXT. The Data File Extension setting cannot be greater than three characters. The Data File Extension setting is used for all CREATE TABLE statements. Sending a CREATE TABLE using an extension other than the Data File Extension setting causes an error.

In other SQL statements such as SELECT or INSERT, users can specify an extension other than the Data File Extension setting. The Data File Extension setting is used when no extension is specified.

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*).

Use Long Qualifiers: Set this check box to use long pathnames as table qualifiers. When you set this check box, pathnames can be up to 255 characters. The default length for pathnames is 128 characters.

Translate

Displays the Select Translator dialog box to allow you to perform a translation of your data from one character set to another. Choose the INTERSOLV OEM ANSI translator to translate your data from the IBM PC character set to the ANSI character set.

Close

Returns to the ODBC Text ODBC Setup dialog box where you can click the **OK** button to write these settings to the ODBC.INI file.

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, character, or stream.
   d. If the table type you select is character-separated, enter the character that separates the values in the Delimiter Character field.
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:

Symbol	Description
m	Output the month's number (1-12).
mm	Output a leading zero if the month number is less than 10.
mmm, Mmm, MMM	Output the three-letter abbreviation for the month depending on the case of the 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:

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

The *stream* table type has no delimiter characters between columns nor between records. The table is one long, continuous stream of data.