



**The Local Database plugin  
PRINTED MANUAL**

# Local Database plugin

© 1999-2025 AGG Software

All rights reserved. No parts of this work may be reproduced in any form or by any means - graphic, electronic, or mechanical, including photocopying, recording, taping, or information storage and retrieval systems - without the written permission of the publisher.

Products that are referred to in this document may be either trademarks and/or registered trademarks of the respective owners. The publisher and the author make no claim to these trademarks.

While every precaution has been taken in the preparation of this document, the publisher and the author assume no responsibility for errors or omissions, or for damages resulting from the use of information contained in this document or from the use of programs and source code that may accompany it. In no event shall the publisher and the author be liable for any loss of profit or any other commercial damage caused or alleged to have been caused directly or indirectly by this document.

Printed: 1/7/2025

## **Publisher**

*AGG Software*

## **Production**

© 1999-2025 AGG Software

*<http://www.aggsoft.com>*

---

# Table of Contents

<b>Part 1 Introduction</b>	<b>1</b>
<b>Part 2 System requirements</b>	<b>1</b>
<b>Part 3 Installing Local Database</b>	<b>1</b>
<b>Part 4 Glossary</b>	<b>2</b>
<b>Part 5 Principles of work. General provisions</b>	<b>3</b>
<b>Part 6 File creation mode</b>	<b>3</b>
<b>Part 7 Export options</b>	<b>7</b>
<b>Part 8 Columns order</b>	<b>8</b>

## 1 Introduction

The "Local database" data export module for our data logger contains a set of tools for exporting parsed data to different popular formats, such as Microsoft Excel, RTF, HTML, CSV, DIFF, SYLK, Plain text, Windows Clipboard, XML, DBF, PDF, Lotus 1-2-3 and Quattro format. The plugin creates these files directly without using third-party components (e.g., DDE, OLE) that guarantees high-speed operations.

The module has the features:

- You may customize export options specific to the selected file format.
- You may define a custom data format for standard data types (date and time, numbers).
- You may configure the order of columns in the resulting file.
- The plugin can cache high-speed data flow and export a bunch of records at once.

These features allow you to get ready-to-use files on a local or network drive.

## 2 System requirements

The following requirements must be met for "Local Database" to be installed:

**Operating system:** Windows 2000 SP4 and above, including both x86 and x64 workstations and servers. The latest service pack for the corresponding OS is required.

**Free disk space:** Not less than 5 MB of free disk space is recommended.

**Special access requirements:** You should log on as a user with Administrator rights in order to install this module.

The main application (core) must be installed, for example, Advanced Serial Data Logger.

## 3 Installing Local Database

1. Close the main application (for example, Advanced Serial Data Logger) if it is running;
2. Copy the program to your hard drive;
3. Run the module installation file with a double click on the file name in Windows Explorer;
4. Follow the instructions of the installation software. Usually, it is enough just to click the "Next" button several times;
5. Start the main application. The name of the module will appear on the "Modules" tab of the "Settings" window if it is successfully installed.

If the module is compatible with the program, its name and version will be displayed in the module list. You can see examples of installed modules on fig.1-2. Some types of modules require additional configuration. To do it, just select a module from the list and click the "Setup" button next to the list. The configuration of the module is described below.

You can see some types of modules on the "Log file" tab. To configure such a module, you should select it from the "File type" list and click the "Advanced" button.

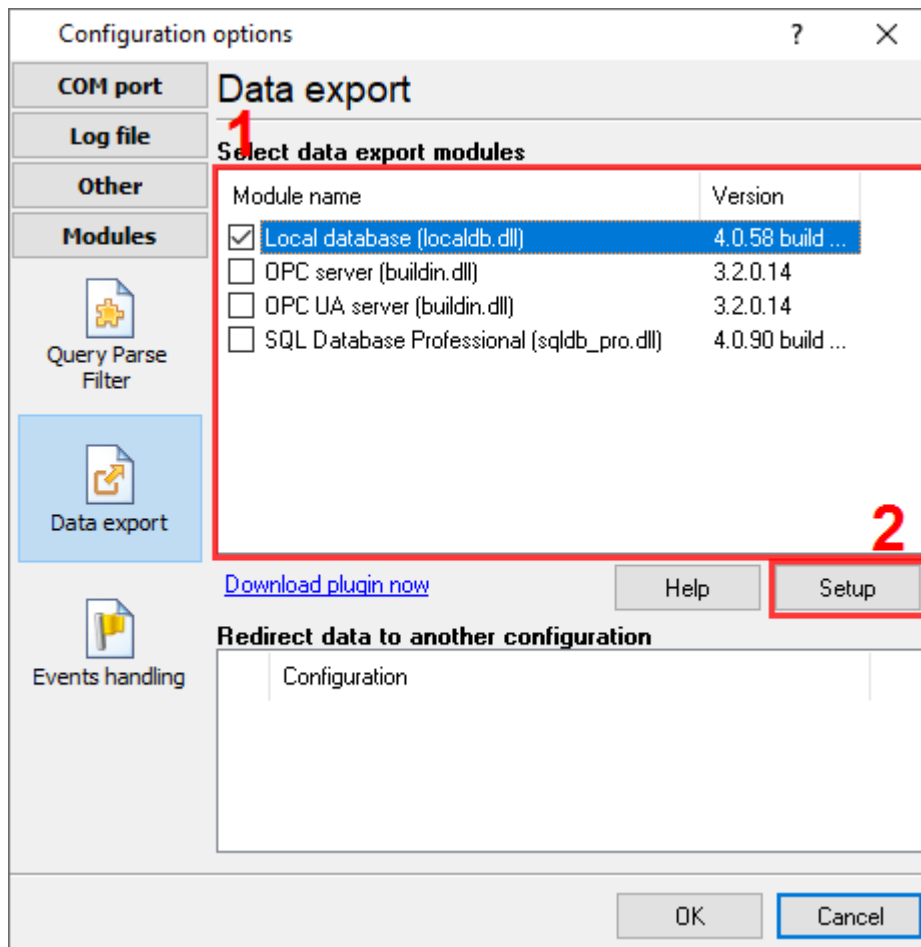


Fig. 1. Example of installed module

## 4 Glossary

**Main program** - it is the main executable of the application, for example, Advanced Serial Data Logger and asdlog.exe. It allows you to create several configurations with different settings and use different plugins.

**Plugin** - it is the additional plugin module for the main program. The plugin module extends the functionality of the main program.

**Parser** - it is the plugin module that processes the data flow, singling out data packets from it, and then variables from data packets. These variables are used in data export modules after that.

**Core** - see "Main program."

## 5 Principles of work. General provisions

The module waits for data from a configured parser and places data to a temporary buffer in the computer's memory optimizing writing operations. After the defined time or a number of records, it writes all buffered data to a file using the [specified format](#).

The plugin detects the presence of a target file immediately after starting. If the selected file format does not support appending, the plugin renames the existing file and creates a new one.

Please, note that the file format does not support data appending, then the module should store all data of that file in memory. For example, if you create a new file once per month, the program may require a lot of memory and utilizes many system resources when it dumps all data from memory to a file.

## 6 File creation mode

You should select the format of a target file in the "Database type" list (fig. 1).

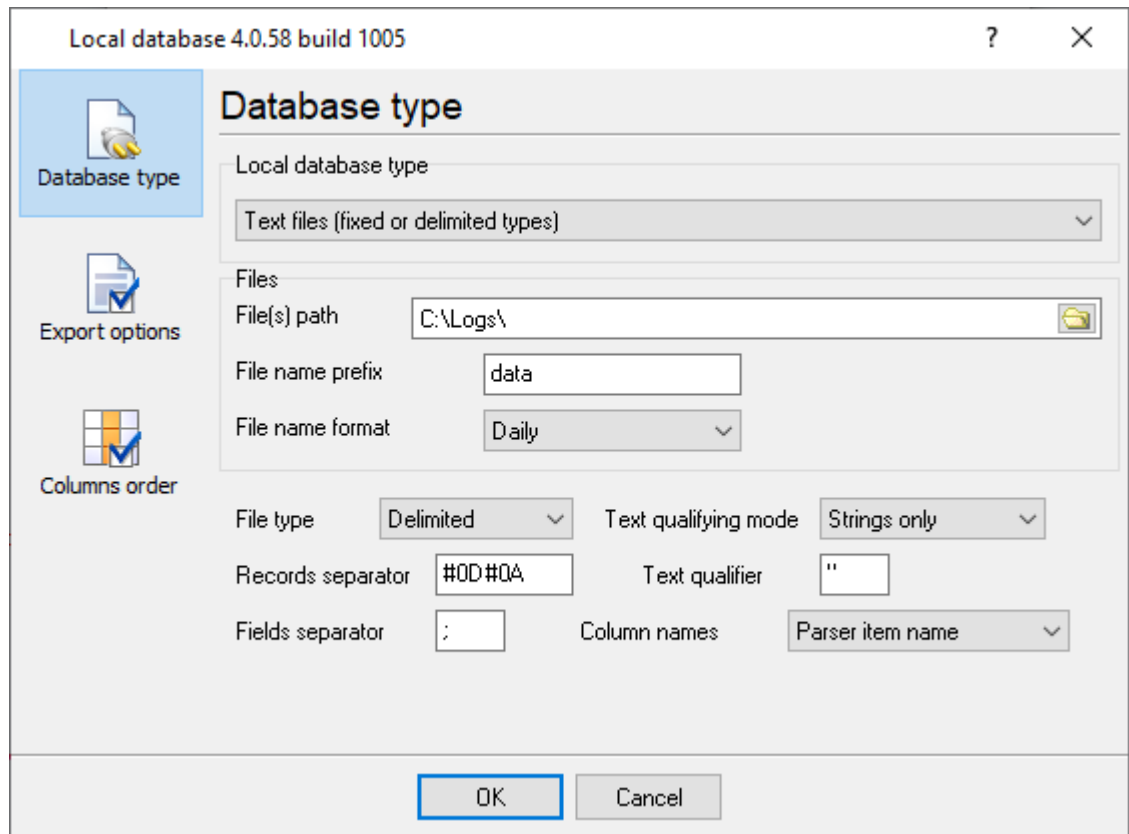


Fig. 1 File format options

After that, you should configure file parameters:

**File path.** It is the destination of created files. The program does not create a target directory, and you should create it yourself. The plugin is designed to create files with data on a local computer. You can also create files on a mapped network drive, but these operations are not optimized, and we recommend frequently creating smaller files.

**Filename prefix.** It is a fixed part of a file name. You may type any valid characters there. The prefix may include special placeholders like {SAMPLE\_ID} that the plugin can replace with the corresponding value. If the prefix contains placeholders, the plugin will reassemble the file name for each data row. The process may be slow if you export many records.

Example #1:

Prefix: data\_{SAMPLE\_ID}\_

Parser variable SAMPLE\_ID: A123

The resulting value will be: data\_A123\_

Example #2:

Prefix: data\_{SAMPLE\_ID2}\_

Parser variable SAMPLE\_ID: absent

The resulting value will be: data\_\_

**Filename format.** It is a variable part of a file name and contains a timestamp. The format for each mode is shown below:

- **Hourly.** YYYYMMDDHH, Example: 2020123123
- **Daily.** YYYYMMDD, Example: 20201231
- **Monthly.** YYYYMM, Example: 202012
- **User format.** You may define a custom format using placeholders like YYYY, MM, DD, etc. Example: YYYY-MM-DD => 2020-12-31

**Filename extension.** It depends on the file format, and the program appends it to a file name automatically:

- MS Excel directly without OLE - XLS.
- RichText format (directly without OLE/DDE/RichText.dll) - RTF.
- [Text files](#) (fixed or delimited types) - CSV for files with a divider and TXT for files with a fixed position of values.
- [XML](#) file - XML.
- Lotus 1-2-3 directly without OLE - WK1.
- QuattroPro directly without OLE - WQ1.
- SYLK (Symbolic Link) - SLK.
- DIF (Data Interchange Format) - DIF.
- [LDAP](#) Data Interchange Format (directory entries) - LDIF.
- SPSS file for statistical software and analysis - SAV;
- DBase (DBF) tables (without BDE) - DBF.
- MS Windows clipboard with separated fields/columns - files are not created.
- [HTML](#) files - HTM.
- MS Access - MDB. The plugin can create Access file without Microsoft Access installed, but features are limited. Please, consider using the ODBC Database plugin instead.
- Adobe Acrobat PDF document - PDF.

You can also define format-specific parameters for some file types.

## Text files

You need to choose a data placement type in a text file: a fixed position or using a delimiter in the "File type" field (fig. 1).

Each column in a file with a fixed position has a fixed width, depending on the column's type of data. For example, for columns with the "string" data type, you may define a width from 1 to 255.

The text file with delimited data contains columns with a variable size, but all columns are divided using a special symbol from the "**Fields separator**" field. You can type any character there or specify a character's code, using an expression like #EE, where "EE" is the hexadecimal code of the symbol you need. This mode can be used to create files in the CSV format.

**Records separator.** It is a line separator in a text file. Usually, it is CR, LF, or CR+LF characters. Therefore, you should specify #0D, #0A, or #0D#0A respectively.

**Text qualifier.** It is a special framing character for columns with string values because they may contain a column delimiter. You may disable this feature at all if you specify #00 in this field.

**Text qualifying mode.** This option specifies how the plugin should add a text qualifier to data.

- Any - a text qualifier is added to all data.
- Any, without numbers - a text qualifier is added to all data, except numbers (e.g., timestamps, boolean values).
- Strings only - a text qualifier is added to values with the "string" data type.

## XML files

Fig. 2. XML file options

**XML file type.** This option allows you to select the XML file format (fig. 2).

- **All rows are aligned using spaces** - it is a user-friendly and readable format. But a file requires allocates more space on your disk.
- **All rows are aligned using spaces, but the total number of spaces is limited** - it is a compact format without optional formatting spaces.

**Quote character** - this character is used for XML attributes. The default value is a double quote.

**Records tag** - it is the name of a root node for all records in a file.

**One row tag** - it is the name of a record node. It is a child node of the root node.

**One column tag** - it is the name of a single value inside a record. It is a child node of the row node.



## LDAP Data Interchange Format

The only value you need to set is the class name.

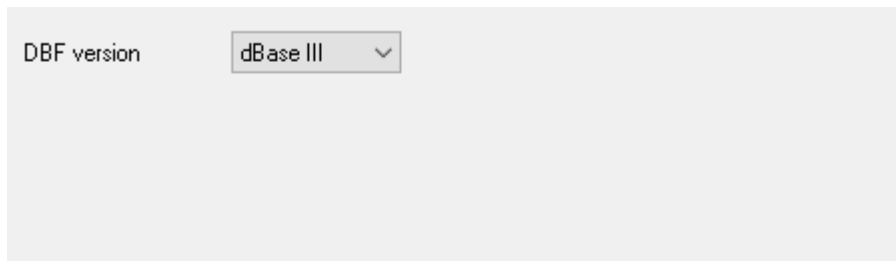


A screenshot of a software interface showing the 'LDAP file options' section. It features a label 'Object class' followed by a text input field containing the word 'data'.

Fig. 3. LDAP file options

## DBase (DBF) tables

You can choose from FoxPro or dBase III formats.



A screenshot of a software interface showing the 'DBF file options' section. It features a label 'DBF version' followed by a dropdown menu currently displaying 'dBase III'.

Fig. 4. DBF file options

## HTML files

You can specify a page template for your needs. You can prepare a template in any 3rd-party HTML editor, copy and paste it in the "HTML template" field. The HTML template may include any HTML compatible tags and formatting. The table with generated data will be inserted immediately after the opening "<body>" tag.



A screenshot of a software interface showing the 'HTML file options' section. It features a label 'HTML template' above a large, empty text area with a vertical scrollbar on the right and horizontal scrollbars at the bottom.

Fig. 5. HTML file options

## Clipboard

The settings are identical to the text file.

## Other

Other file types have no additional parameters necessary to adjust.

## 7 Export options

In this section, you can customize data format options for basic data types. For example, you can set the date and time format.

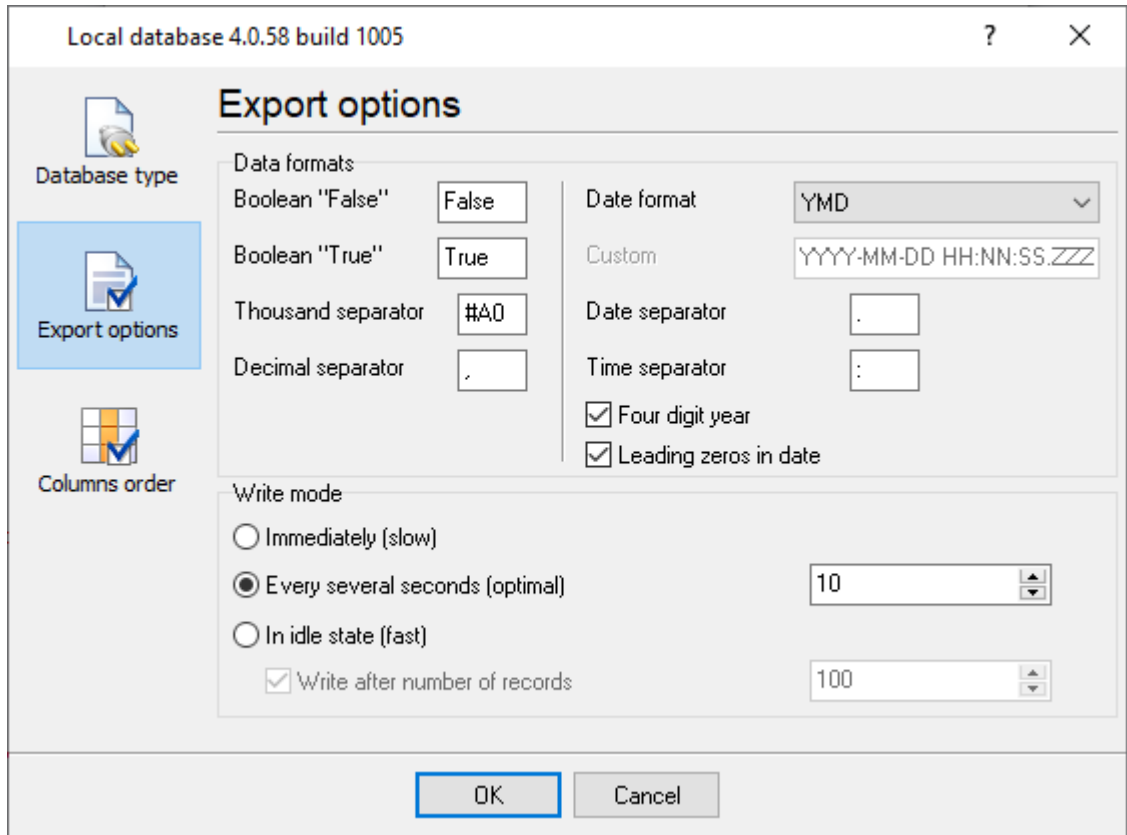


Fig. 6. File format options

**Boolean False or True.** Here you can define a text representation for logical values.

**Thousand separator.** This option allows you to define a separator for groups of digits for large numbers. You may specify #00 and disable this separator.

**Decimal separator** - this symbol is used to separate the integer part from the fractional part of a number written in decimal form. It is applicable for "float", "double", and "currency" data types.

**Date format** - you may select from pre-defined date formats or specify your custom format using the following placeholders.

**Time and date separators** - these separators are used in pre-defined formats of date.

**Four digit year** - if this option is active, the generated date value contains the four-digit year, otherwise two-digit.

**Leading zeros in date** - changes the format between 1 and 01.

Placeholders:

<b>d</b>	- day, without a leading zero (1-31).
<b>dd</b>	- day, with a leading zero (01-31).
<b>ddd</b>	- weekday, in the short text format (Mon-Sun) specified on this computer.
<b>dddd</b>	- weekday, in the full text format (Monday-Saturday) specified on this computer.
<b>m</b>	- month, without a leading zero (1-12).
<b>mm</b>	- month, with a leading zero (01-12).
<b>mmm</b>	- month, in the short text format (Jan-Dec) specified on this computer.
<b>mmm</b>	- month, in the full text format (January-December) specified on this computer.
<b>yy</b>	- year, two last digits (00-99)
<b>yyyy</b>	- year, four digits (0000-9999).
<b>h</b>	- hours, without a leading zero (0-23).
<b>hh</b>	- hours, with a leading zero (00-23).
<b>n</b>	- minutes, without a leading zero (0-59).
<b>nn</b>	- minutes, with a leading zero (00-59).
<b>s</b>	- seconds, without a leading zero (0-59).
<b>ss</b>	- seconds, with a leading zero (00-59).

**Example:** The user specifies HHDDMMYYYY in the files for the date format. Now the date / time field will be created as a string similar to the following: 0101122003.

## Write mode

This group of parameters allows you to select how the plugin outputs data to a file.

**Immediately (slow)** - in this mode, the plugin immediately writes every record to a file. This mode is robust but very slow, especially for file types that do not allow appending data.

**Every several seconds** - the plugin buffers data in memory and outputs to a file after the defined timeout.

**In idle state (fast)** - the plugin buffers data in memory and saves it to a file only when new data does not appear for ten seconds or after the specified number of records in the internal buffer.

## 8 Columns order

Here you can define columns in your file and their order (fig. 7). If this list is empty, the plugin export all received data.

If you want, you can import all parser items to the list by clicking the "Create all" button and edit it for your need.

Also, you can add all the necessary items manually. In this case, the plugin will ask you about the item name.

**Parser item name.** It is the parser's variable name. You may select it from a list. Or you may manually type a name if you defined it in scripts or filter plugins.

**Column order.** This value defines the column position in a file. The leftmost column has the number 1. If several columns have identical value, they will be outputted as they are defined in the configuration.

**Column length.** This parameter defined the maximum column size for text files with a fixed position and column size.

**Data type.** It determines the column data type. The plugin uses this data type to format incoming data before writing it to a file.

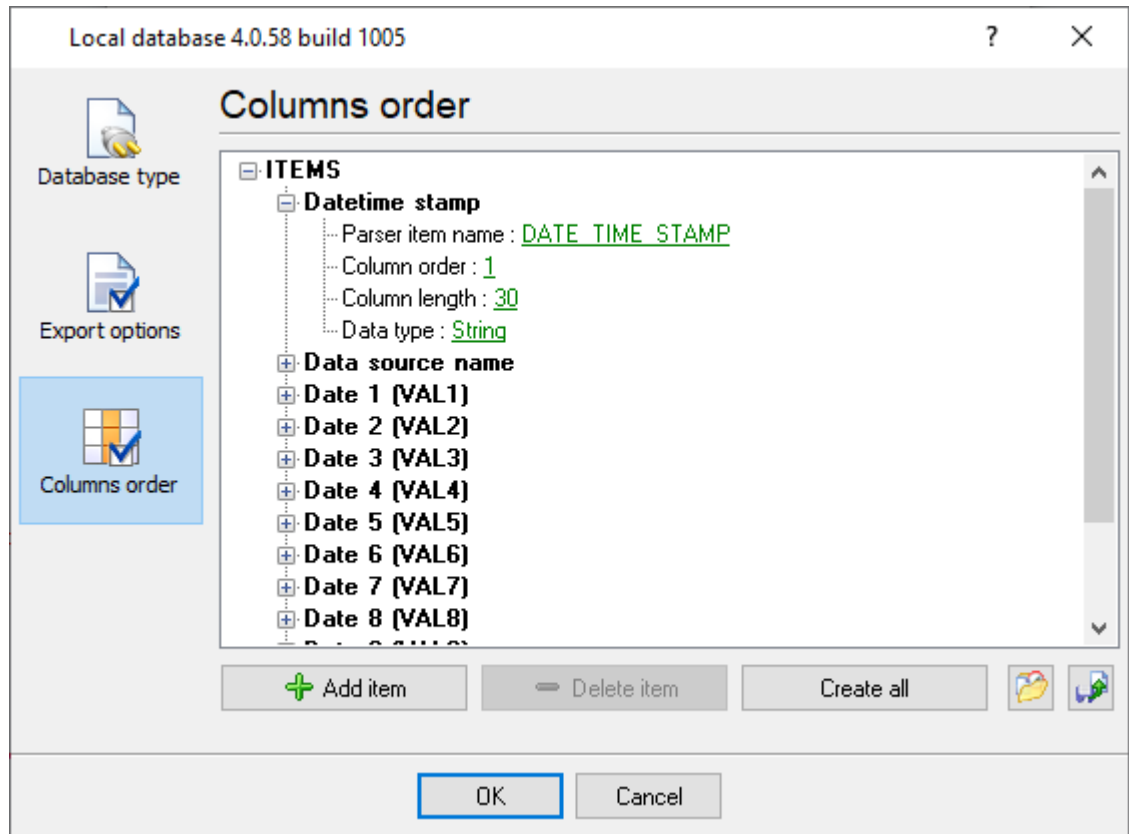


Fig. 7. Columns order