



**The Google Sheets plugin  
PRINTED MANUAL**

# Google Sheets plugin

© 2020-2021 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: 6/22/2021

## **Publisher**

*AGG Software*

## **Production**

© 2020-2021 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 Google Sheets</b>	<b>2</b>
<b>Part 4 Glossary</b>	<b>3</b>
<b>Part 5 User guide</b>	<b>3</b>
1 Configuring Google Sheets connection parameters .....	3
2 How to get API keys .....	6
3 Data presentation formats .....	8
4 Binding .....	10

## 1 Introduction

This data export plugin for our data logger (e.g., Advanced Serial Data Logger) allows you to write real-time data to Google Sheets. The plugin uses the Google API interface to connect to Google Apps. It does not use any local files and sends data directly to your cloud storage.

This module has the following features:

- Data can be written to Google Sheets by rows or columns.
- You may specify any start point (e.g., B6) for each data item.
- The plugin can fill data using several methods (e.g., move old data using the FIFO method),
- The plugin can write several parser variables to different columns or rows at the same time,
- The plugin can automatically create new workbooks or worksheets by date/time,
- The plugin can use a template for newly created files.

Thanks to these possibilities, you can:

- Watch in real-time data in Google Sheets in your browser on any computer.
- Get live charts and diagrams based on captured data.
- Easily publish your data to the Internet.

## 2 System requirements

The following requirements must be met for "Google Sheets" 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.

### Notes for Microsoft Vista and above:

Since our software saves data to the registry and installs to the Program Files folder, the following requirements must be met:

1. You need Administrator rights to run and install our software
2. The shortcut icon of our software will be located on the desktop;
3. Windows Vista will ask for your confirmation to continue the installation.

NOTE: You can configure the user account only once in order not to see the above dialog box any more. Search Google for the solution to this problem.

### 3 Installing Google Sheets

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.

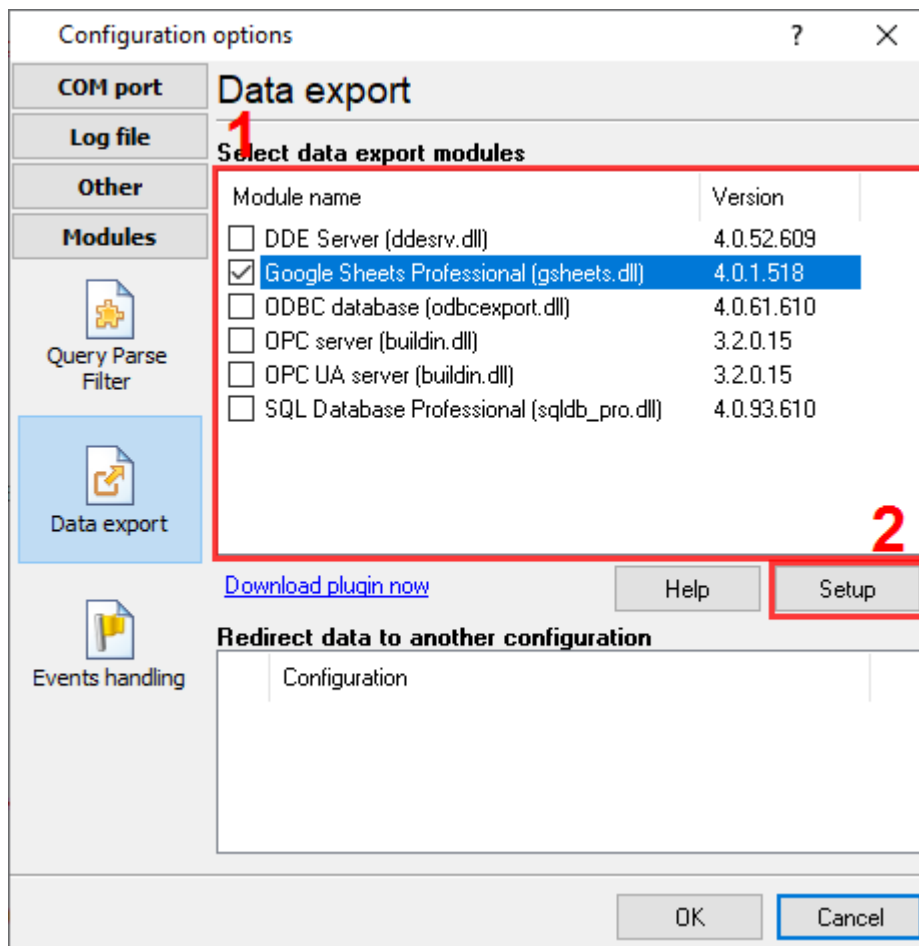


Fig.1. Example of installed module

## 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 User guide

### 5.1 Configuring Google Sheets connection parameters

To enable data export, you should configure a connection with your Google Sheets.

1. Start the main program if it is not running yet (for example, Advanced Serial Data Logger).
2. Open the "Modules - Data export" tab in the program options.
3. Select the "Google Sheets Professional" module in the list of data export modules on this tab. If the module with this name does not exist, make sure the module is installed correctly.
4. Click the "Setup" button.
5. Select the "Connection" tab in the dialog box that appears on the screen (fig. 1).

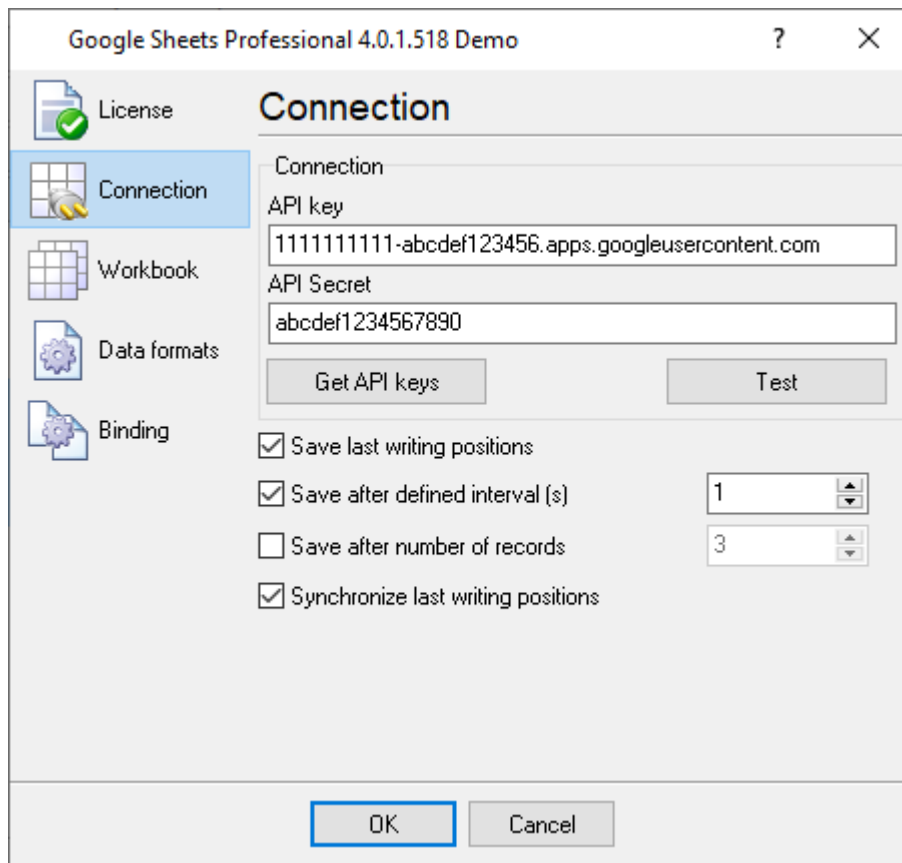


Fig. 1 Google Sheets connection parameters

**API key, API secret** - you should [get these keys](#) in your Google account.

**Test** - it allows you to test a connection with your Google Sheets. The program will ask you to authorize our program in your account to use Google Sheets API if you test the connection the first time or changed API keys.

**Save last writing positions** - the program saves the position of the last cell with a value for the last opened workbook and continues writing from that position when you restart the program.

**Save after defined interval (s), Save after number of records** - these options allow you to optimize writing operations and minimize the number of API calls. The program collects data in the internal buffer for the defined interval and sends a bunch of records at once when it reaches the defined limit. If you receive one record per few seconds, you may use default settings.

**Synchronize last writing position** - if this option is enabled, the program writes all values defined on the "Binding" page in one row even if some values are absent.

## Workbook

After you specify how to connect to Google Sheets, it is time to specify how a workbook and a worksheet will be created and used. To do it, switch to the "Workbook" tab (fig. 2).

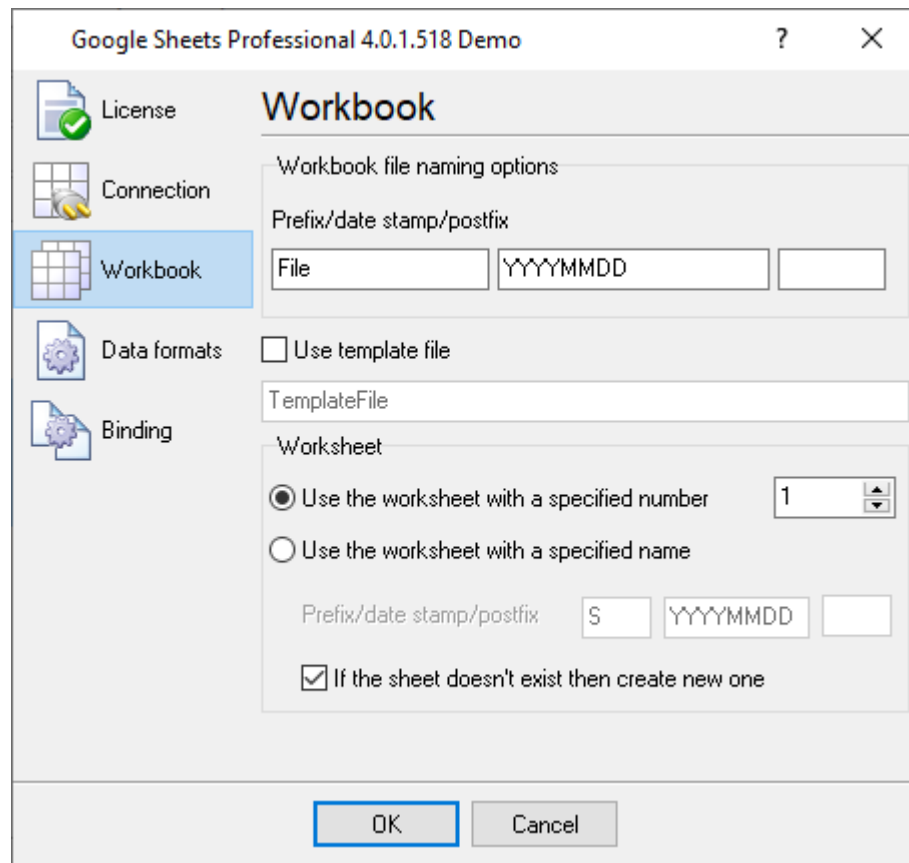


Fig. 2 Workbook options

There are several variants to create a workbook:

**Prefix** - allows you to specify the prefix for the filename.

**Date** - allows you to include the current date and/or time in the filename. It allows you to create a new file each hour, day, month, etc. Use this field to specify the date/time template. The template format is the same as the one used in the main program for log files (see the help file of the main program). If you leave the template empty, neither date nor time will be added to the filename.

**Postfix** - the optional suffix of the filename.

**Use template file** - if this option is enabled and the plugin cannot find a remote file with the name got from the prefix, timestamp, and postfix, the plugin copies the specified template with a new name and use the copied file for writing data.

## Worksheet

Operations with the worksheet are configured in the same way (fig. 2). You can also:

1. Use sheet with specified number - this option instructs the plugin to use the worksheet with the specified index in the opened workbook. The first sheet has index 1.
2. Use the worksheet with the specified name - the plugin uses **Prefix**, **Date stamp**, and **Postfix** to get a sheet name. Then the plugin tries to find a sheet with this name in the opened workbook.



---

If the worksheet with the specified name or number does not exist, the selected "**If sheet doesn't exist then create new one**" checkbox allows you to try to create a new worksheet.

If the attempt to create a new worksheet fails, the corresponding message will be written to the program's log, and sending data to Google Sheets will be stopped until the module configuration is changed.

## 5.2 How to get API keys

1. Create a Google account if you don't have it yet.
2. Go to the "[Google API](#)" dashboard.
3. Go to the "Credentials" page.
4. Click the "Create credentials - Create OAuth client ID" button and select the following options (fig. 3):
  - Web Application.
  - Use any name
  - In the "Authorized redirect URIs" add the following exact link: <http://localhost:8888>
  - Click the "Create" or "Save" button.

**API** APIs & Services

- Dashboard
- Library
- Credentials**
- OAuth consent screen
- Domain verification
- Page usage agreements

### ← Create OAuth client ID

A client ID is used to identify a single app to Google's OAuth servers. If your app runs on multiple platforms, each will need its own client ID. See [Setting up OAuth 2.0](#) for more information.

**Application type \***  
Web application

[Learn more](#) about OAuth client types

**Name \***  
My Google Sheets

The name of your OAuth 2.0 client. This name is only used to identify the client in the console and will not be shown to end users.

**i** The domains of the URIs you add below will be automatically added to your [OAuth consent screen](#) as [authorized domains](#).

#### Authorized JavaScript origins ?

For use with requests from a browser

+ ADD URI

#### Authorized redirect URIs ?

For use with requests from a web server

**URIs \***

http://localhost:8888

+ ADD URI

**CREATE** CANCEL

**Fig. 3 Google API credentials**

When you created credentials, you can copy the "Client ID" value to the "API key" field and "Client secret" value to the "API secret" field (fig. 1)

5. Go to the "[Google API](#)" dashboard and search for "Google Sheets API" using the search bar. Then navigate to the suggested page and enable the corresponding API (fig. 4).

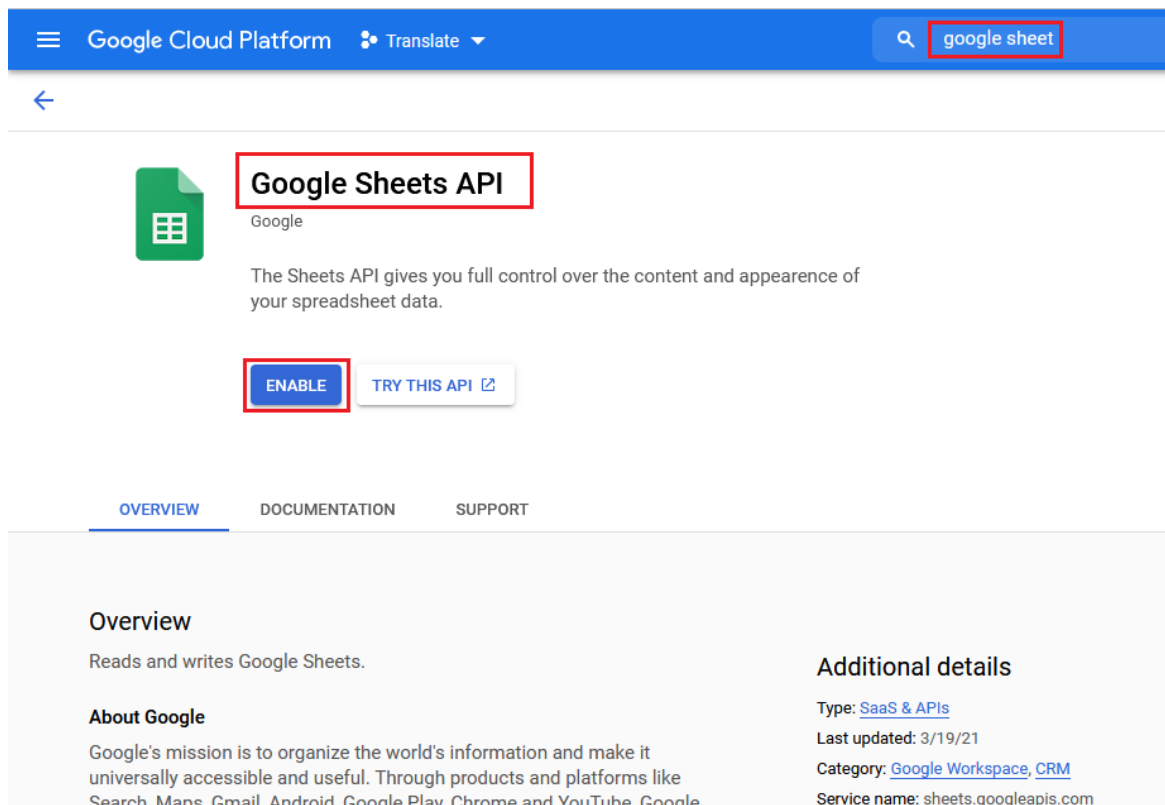


Fig. 4 Activating Google Sheets API

6. You should do the same actions for Google Drive API too. This API allows our plugin to use templates from your cloud storage.

7. Click the "Test" button in the plugin window (fig. 1). The plugin will open a web browser and navigate you to a login page for your account.

8. Log in to your Google account.

9. Grant access for these Google Sheets API and Google Drive API for this computer. Please, note our software does not store your Google account name and password. It only gets and stores an authorization token provided by Google API. At any time, you can disable these APIs or revoke granted access rights.

## 5.3 Data presentation formats

These settings allow you to convert numbers, dates, or times to the necessary format before writing to Google Sheets (fig. 5).

Please, note that "System" means your Windows OS, not Google Sheets.

The "**User format**" option allows you to define a custom format.

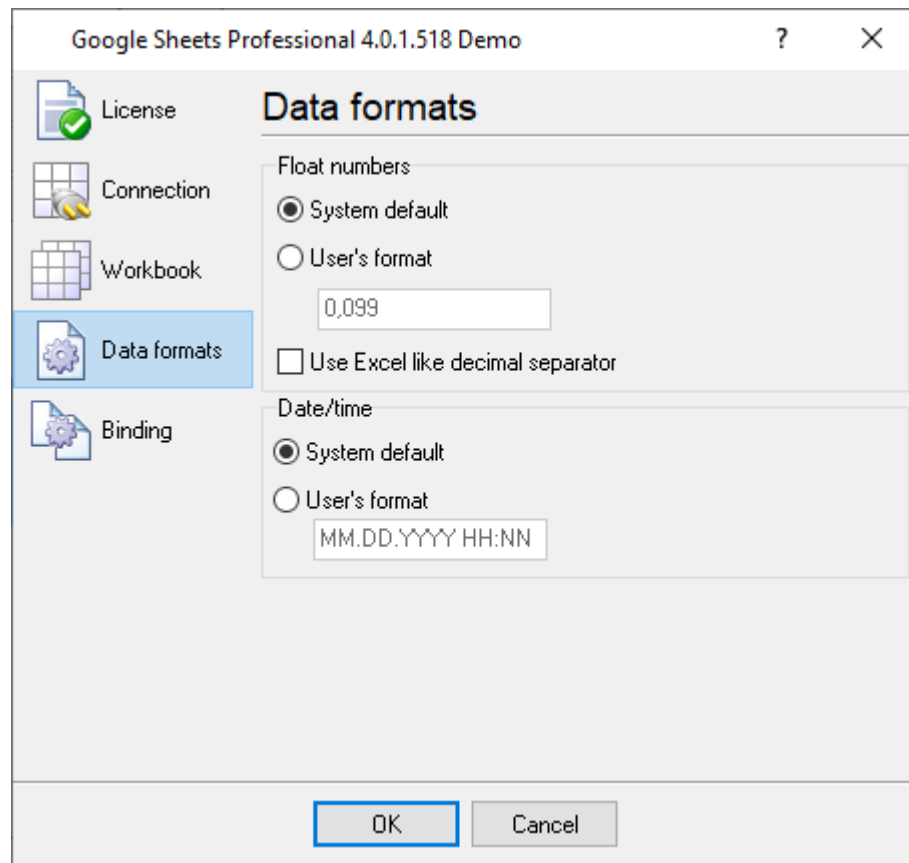


Fig. 5 Data format

## Date and time format

You can specify the custom date and time representation format in the following way:

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

**Example:** The following format is specified on fig. 5: MM.DD.YYYY HH:NN. The date looks like this according to this format: 10.29.2003 23:59.

## Number format

You can use the following special characters to specify the custom representation format for float numbers:

**0** - number position. If the number being formatted by the mask contains a number in this position of the format mask, this number will be copied to the result. If there is no number in this position of the number being formatted, this position will be replaced with 0 in the result.

**#** - number position. If the number being formatted by the mask contains a number in this position of the format mask, this number will be copied to the result. If there is no number in this position of the number being formatted, this position will be empty in the result.

**.** - decimal symbol. The first '.' character will be considered as a decimal symbol. The rest of characters in the format mask will be ignored. In the result, the '.' character will be replaced with the system value of the variable defining the decimal symbol. The default value of this symbol is set in the Windows Control Panel under "Regional Options" - "Number"

**,** - digit grouping symbol. If the format mask contain one or more ',' characters, this digit grouping symbol will be displayed in the specified positions of the format mask in the result. The default value of this symbol is set in the Windows Control Panel under "Regional Options" - "Number".

**E+** - scientific representation. If any of these strings 'E+', 'E-', 'e+', or 'e-' is present in the format mask, this number will be represented in the scientific format in the result, where a positive exponent will be displayed after the 'E+' or 'e+' string and the negative exponent will be displayed after the 'E-' or 'e-' string.

**'xx'/"xx"** - a sequence of characters enclosed in single or double quotation marks will be displayed in the result without any changes.

## 5.4 Binding

Binding will allow you to specify what operations the module should perform with this or that variable passed from the parser to the data export module.

To configure it, you should switch to the "Binding" tab that contains the list of the variables being exported and the export parameters of each variable, as well as the buttons that you can use to add or remove variables being exported (fig. 6).

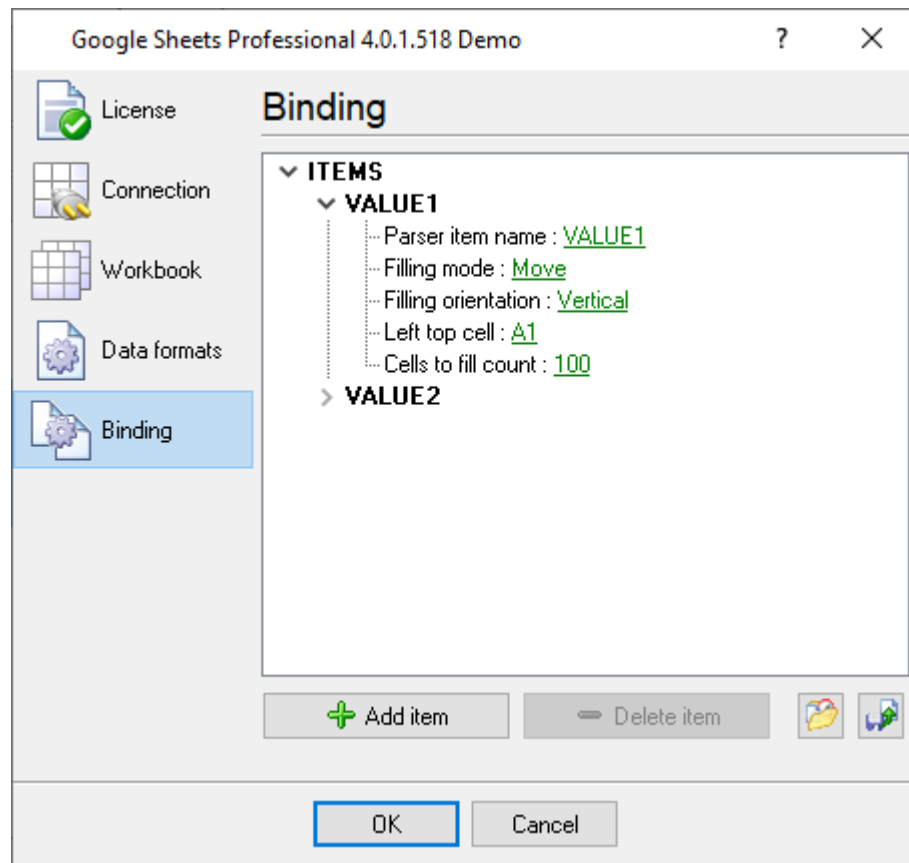


Fig. 6 List of variables being exported

The window represents the current configuration in the form of a tree. Where the top level of the tree contains the variables being exported. Each variable has several options that are located one level lower in the tree.

To add a new exported variable, you should click the "Add item" button or select the corresponding item in the context menu. After you click the button, you will see the dialog box that will offer you to enter the description of the variable.

After you enter its name and click "OK", an item with the default parameters will be created in the tree. Click the green link (see fig. 6) in the configuration window, and you will be able to change the value of the corresponding parameter the way you need it. See the detailed description of variable parameters below:

- **Parser's item name** - here, you should specify the name of the parser item that will be exported. The dialog box that appears after a click on the link offers you to select the variable name from the list. If the parser is not configured or if it is configured incorrectly, the list of available variables will be empty. If you select the name of a variable from the list, you will let the module know that the data of this very variable should be exported.
- **Filling mode** - here, you can select one of the following values:
  - Clear - if the number of cells for filling is exceeded, the plugin clears all filled cells in the destination column or row.
  - Move - in this mode, the plugin shift filled cells up and writes the new values to the last cell.
  - Reversed move - in this mode, the plugin shift filled cells down and writes the new value to the first cell.

- **Stop** - the plugin stops writing new data to Google Sheets until a new workbook or worksheet is created.
- **Next column or row** - the plugin starts filling the next column or row.
- **Filling orientation** - here, you can select your data orientation in Google Sheets, by columns (Vertically) or by rows (Horizontally).
- **Left top cell** - here, you should specify the coordinates of the upper-left cell starting from which data will be sent to Google Sheets. The format the coordinates are specified in should comply with the standard accepted in Microsoft Excel. For example, a record like A1 will mean the coordinates of the top-left cell on the worksheet.

You may use macros in the "Left top cell" field to define the left-top cell dynamically. Macros are strings between the { } characters. When processing data, the program will evaluate the specified expression and expects to receive the value with the cell coordinates like A1. You may use any parser variable in the expression. Also, you may define custom variables in the "Expression" plugin.

Example:

A{DEVICE\_ID} - may return A1 - A10 if data contains the "DEVICE\_ID" parser value with numbers 1-10.

- **Cells to fill count** - here, you can specify the maximum number of cells in a column or in a row that will be filled when data is sent to Google Sheets. Please, consider to limit this number to 65000 rows or 1024 columns, and create new workbooks or worksheets periodically.

To remove an unnecessary item from the configuration, you should select the name of this item in the tree and click the "Delete item" button or use the corresponding item in the context menu.

Also, the window has two buttons that you can use to save the current configuration to a file and later load it on this computer or on another computer.

After you finish configuring the module, you should close the dialog box with a click on the "OK" button. After that, the current configuration will be saved.

If you want to cancel ALL changes, click the "Cancel" button in the dialog box.