GO.VariableExport V1.6.4

User guide
2018

Table of contents

Introduction

GO.VariableExport allows the user to export ranges of Common Variables to a file in the Open XML spreadsheet format. This spreadsheet can be opened in Products like Microsoft Excel, or Open Office Spreadsheets. No other software is required on the machine to generate the spreadsheet.

Installation

Installation

Prerequisites

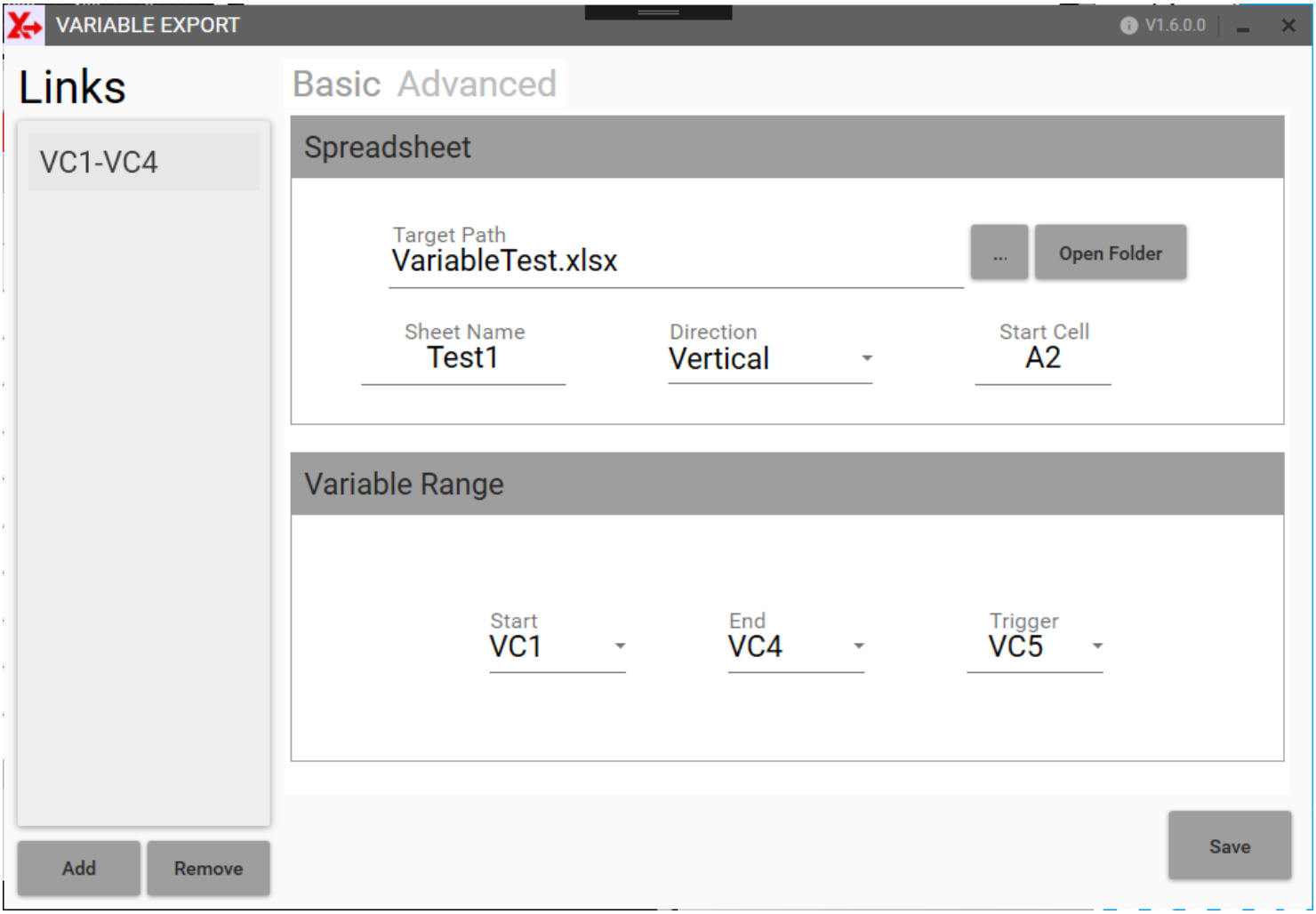
OSP P200A or Above
Windows 7
.Net Framework 4.5.2
THINC API V1.18.0

Installer

To Install GO.VariableExport run the setup file and follow the instructions. By default the application will be installed @ D:\Program Files\Gosiger. GO.VariableExport will also automatically register itself with the Thinc Startup Service.

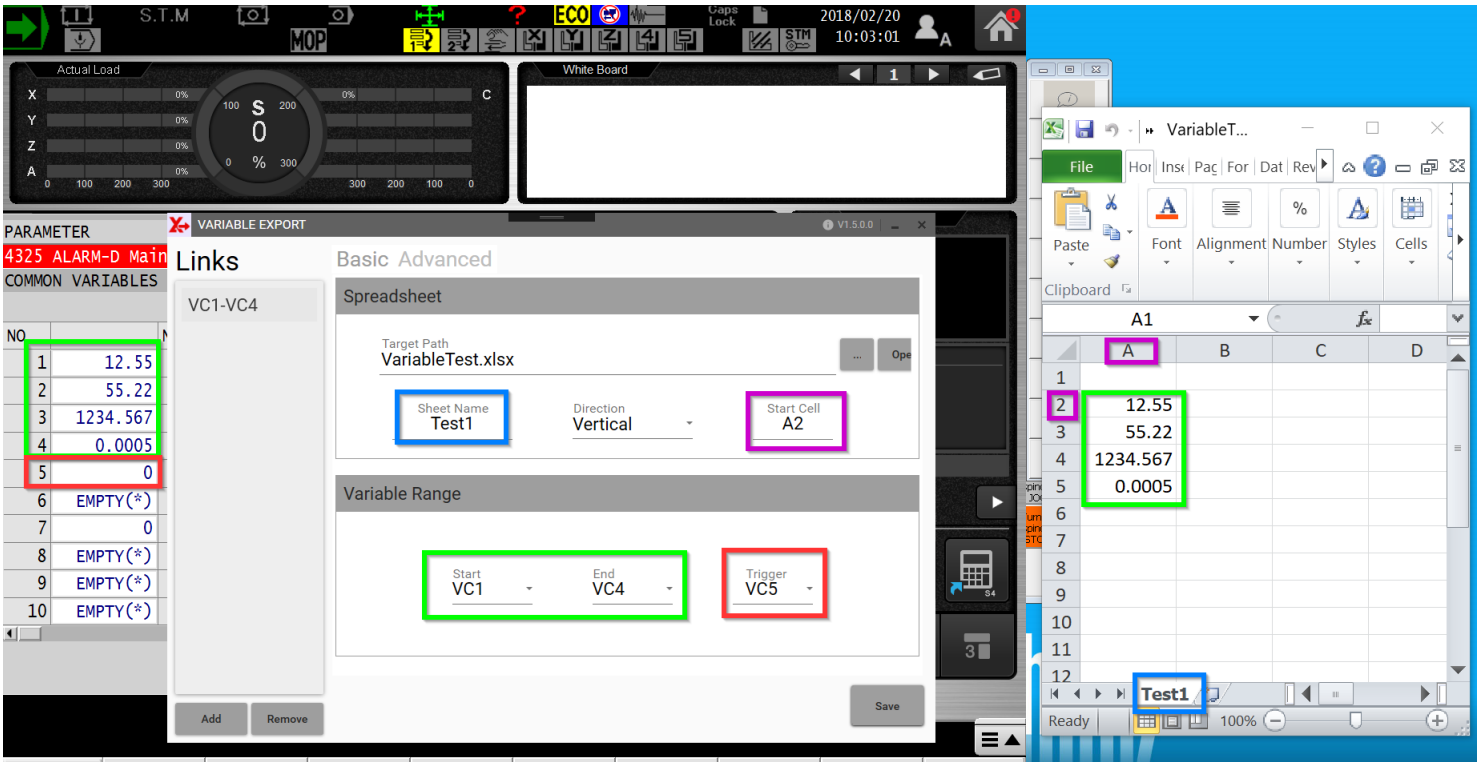
Operation

GO.VariableExport allows the user to specify a range of variables to export and the name of the spreadsheet file. You can also specify the name of the sheet within the workbook and the address of the cell to write to. When the specified trigger variable (VC5 in the example below) is set to 100 the values of the specified range are used to create a spreadsheet file in the Open XML format. The spreadsheet file can be an existing template file. You can also specify another spreadsheet file to use as a template.



Below is an an example of a simple variable range transfer.

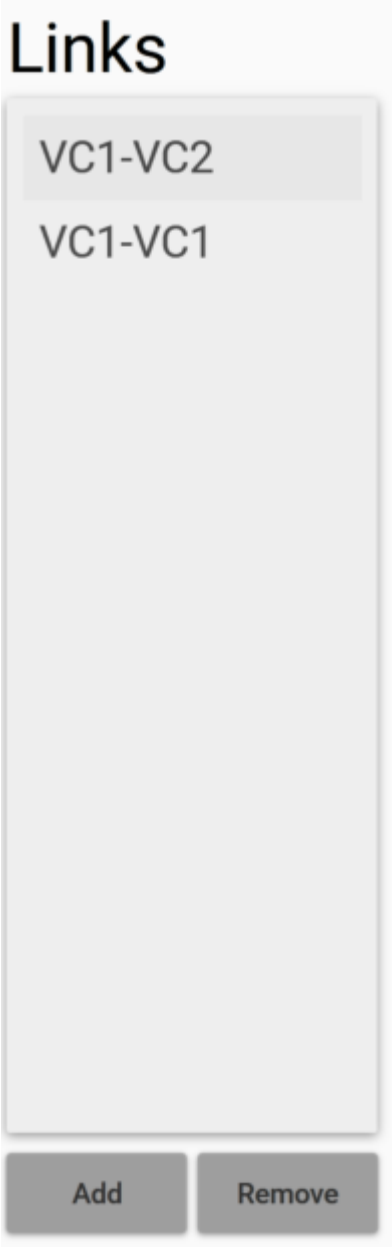
As soon as the machine sets the **VC5** trigger to **100** the range of variables **VC1-VC4** will be written to the Spreadsheet vertically beginning with **A2.** When the operation is complete the trigger variable returns to 0.


S.T.M
MOP
ECO
Caps Lock
2018/02/20
10:03:01
Actual Load
White Board
PARAMETER
4325 ALARM-D Main
COMMON VARIABLES
NO
1 12.55
2 55.22
3 1234.567
4 0.0005
5 0
6 EMPTY(*)
7 0
8 EMPTY(*)
9 EMPTY(*)
10 EMPTY(*)
VARIABLE EXPORT
Links
VC1-VC4
Add
Remove
Basic Advanced
Spreadsheet
Target Path
VariableTest.xlsx
Sheet Name
Test1
Direction
Vertical
Start Cell
A2
Variable Range
Start
VC1
End
VC4
Trigger
VC5
Save
VariableT...
File
Paste
Font
Alignment
Number
Styles
Cells
Clipboard
A1
A
B
C
D
12.55
55.22
1234.567
0.0005
Test1
Ready
100%

GUI

Link List

The Links list shows all of the defined variable exports. Using the Add button you can define new ranges of variable to export using the same or a different variable trigger.

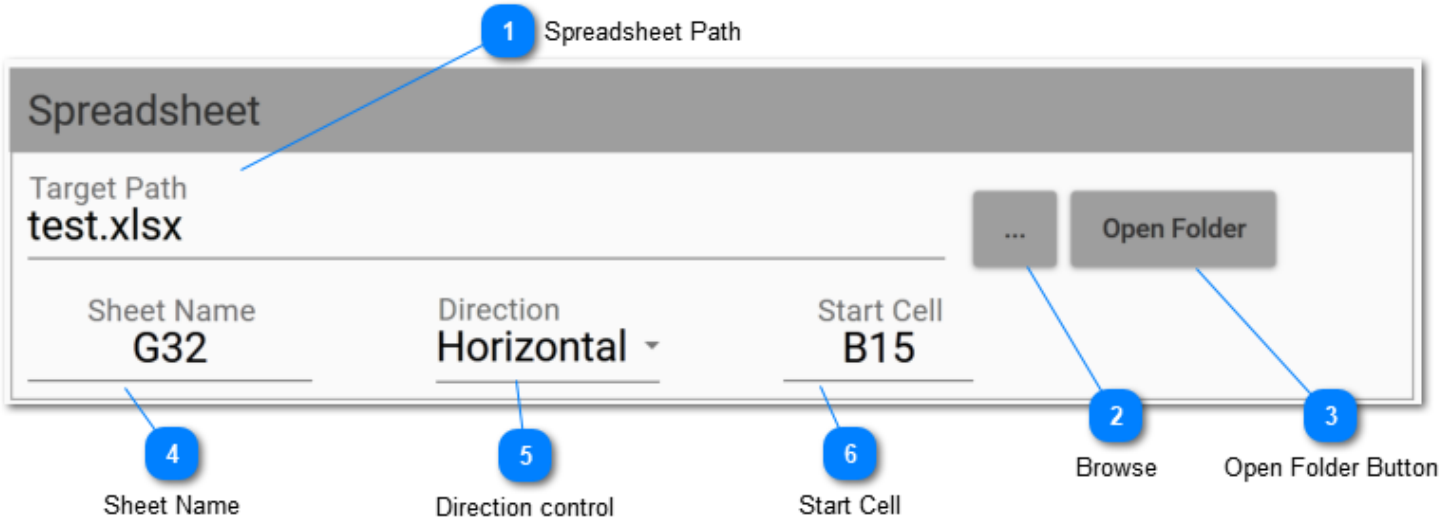


Basic

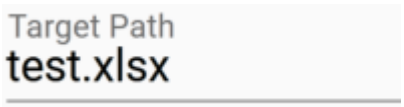
The basic settings are all that are required for a simple scenario

Spreadsheet

The Spreadsheet section is used to describe the target spreadsheet that will be created, or added to.



1 Spreadsheet Path

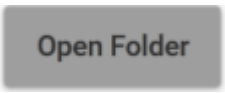


The path to the target spreadsheet. If an existing file is selected the variable range will be added to it, otherwise the file will be created.

2 Browse

Browse the path to a new or existing workbook file.

3 Open Folder Button



This will open the folder of the target file. This is useful if you want to quickly navigate to the target directory.

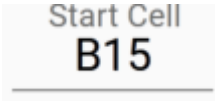
4 Sheet Name

The name of the sheet used to record the Variable Range Values. If the sheet does not exist it will be created. It is also possible to prompt the operator to enter the name of the sheet (see Advanced Automation)

5 Direction control

Select which direction the series of variable values should be recorded.

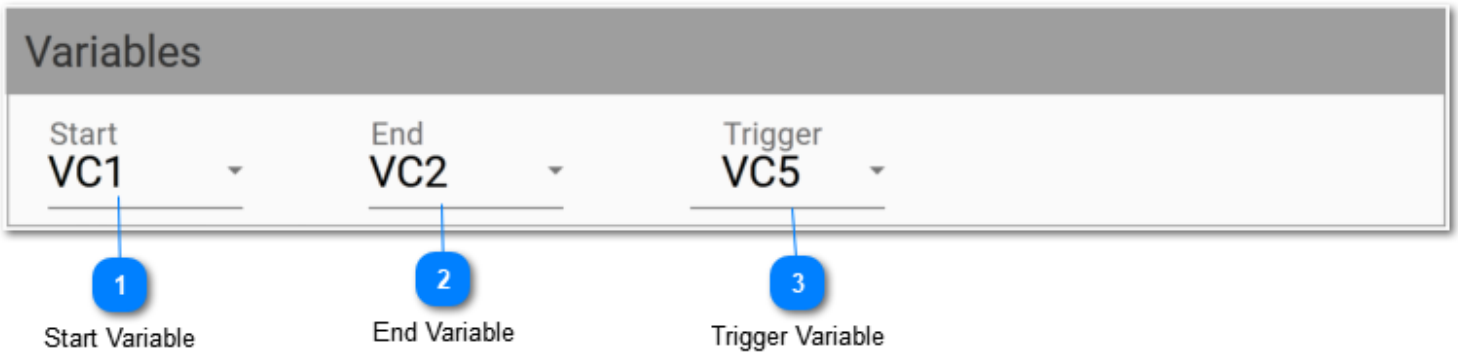
6 **Start Cell**



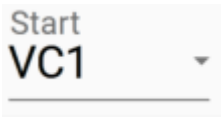
The cell to start with when recording the series of values from the common variables.

Variables

The Variables section describes the range of variables to record and the variable to use as the Trigger.



1 Start Variable

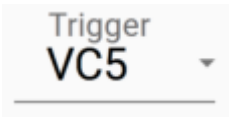


This is the start of the range. It must be less than or equal to the End Variable.

2 End Variable

The is the end of the range. It must be greater than or equal to the Start Variable.

3 Trigger Variable



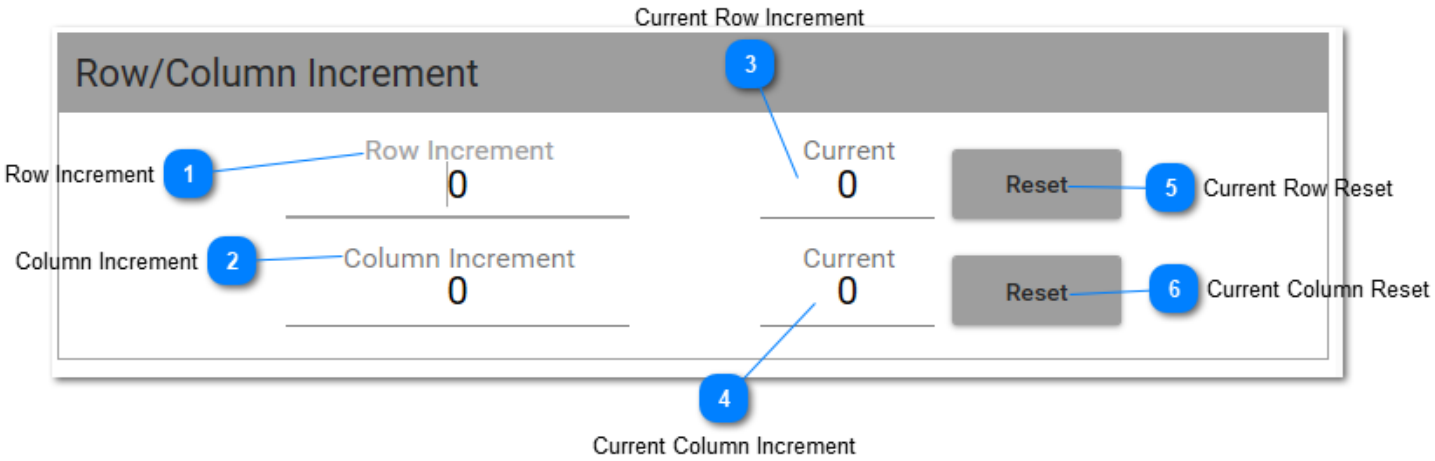
The trigger variable is the variable that when set to 100 will initiate the recording of the range to the spreadsheet. The trigger variable is also used for other functions in advanced scenarios (see Advanced Automation)

Advanced

The advanced section allows you to configure various settings which control the way the variable values are recorded in the spreadsheet.

Row/Column Increment

The row/column increment settings will allow you to add a row or column of data to the current sheet every time the trigger value is set.



1 Row Increment

The number of rows to increment at each successive trigger. Can also be set programmatically (see Advanced Automation)

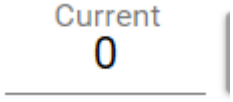
2 Column Increment

The number of Columns to increment at each successive trigger. Can also be set programmatically (see Advanced Automation)

3 Current Row Increment

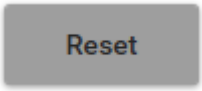
The number of rows currently skipped.

4 Current Column Increment



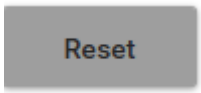
The number of columns currently skipped

5 Current Row Reset



Reset the current row counter. Can also be reset programmatically (see Advanced Automation)

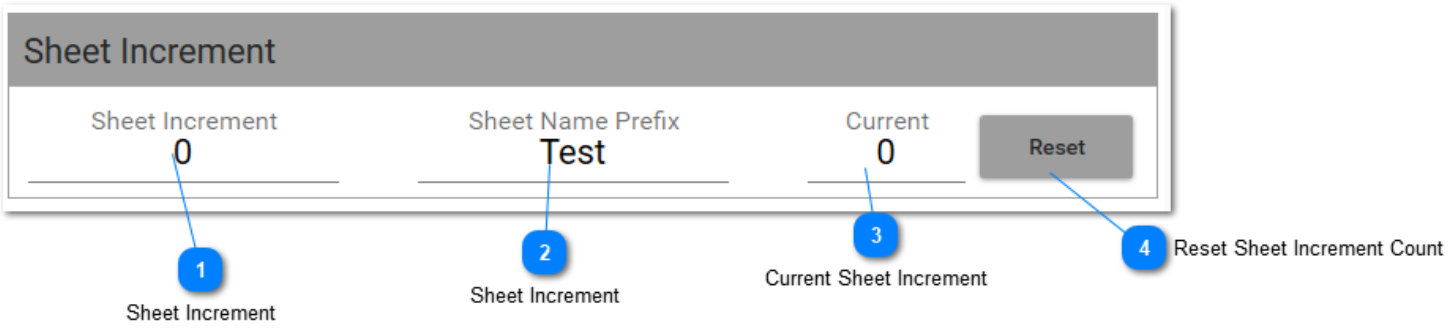
6 Current Column Reset



Reset the current column counter. Can also be reset programmatically (see Advanced Automation)

Sheet Increment

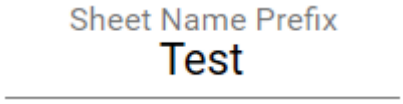
The sheet increment settings will allow you to generate a new sheet every time the trigger value is set.



1 Sheet Increment

<TODO>: Insert description text here...

2 Sheet Increment

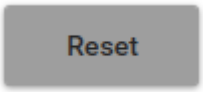


Sets the text used to create the incremental sheets. For example if the Sheet Name Prefix is 'Test' and the Current Sheet Increment is 4 the next time the trigger variable is set a new sheet called 'Test4' will be created.

3 Current Sheet Increment

The current number added to the sheet name

4 Reset Sheet Increment Count



Resets the counter

Sheet Template

If you specify the path to an existing workbook every time a sheet is created the first sheet of the specified template will be used to create the new sheet.

Advanced Automation

Setting specific values in the Trigger Variable will perform different actions

For example if you have a Variable Range Link that uses VC5 as the trigger variable you can:

Set VC5 = 100 is the default that will command the values to be written as programmed in the app. (this is the default mode).

Set VC5=-1 to reset the row increment (same as pushing Row Counter Reset)

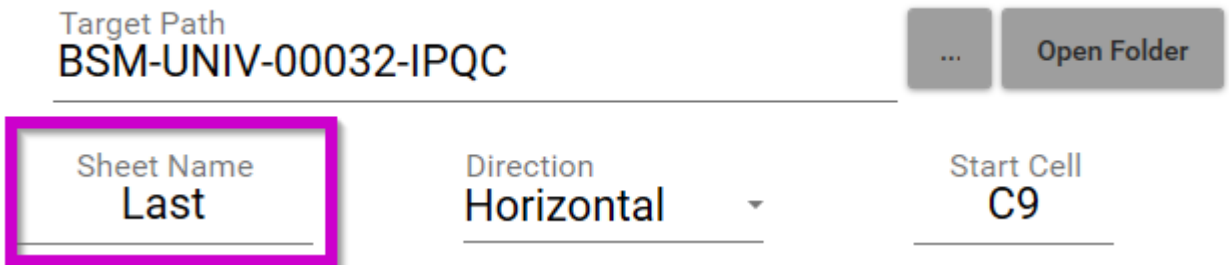
Set VC5 =-2 to reset the column increment (same as pushing Column Counter Reset)

Set VC5 = -3 to reset the sheet increment (as as pushing Sheet Increment Reset)

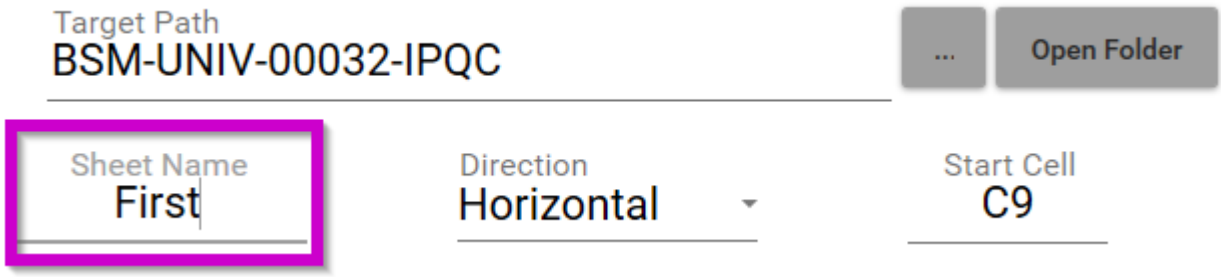
Set VC5 = 10x where x is the number of lines to skip before writing e.g. VC1=112 would skip 12 lines before writing the specified variable range

Set VC5 = 200 to prompt the operator to input the Sheet Name to write to (same as entering text in the Sheet Name)

Set the sheet name of a link to 'Last' in order to write to the last sheet of the workbook



Set the sheet name to 'First' to write to the first sheet of the workbook



Simple Program Example

```
VC5 = -1 (RESET ROW INCREMENT)
(ADD VALUES TO VARS FOR TEST)
VC1=VC1+.01
VC2=VC2+.01
VC3=VC3+.01
VC4=VC4+.01
VC5=100 (SET TRIGGER)
M2
```

Advanced Program Example

This program demonstrates the following automatable features:

Reset row increment counter (Trigger variable = -1)

Prompt operator for sheet name (Trigger Variable = 200)

print out a series of variables (using VC5 as a trigger)

skip 8 lines every 5 triggers (Trigger variable = 100 + number to skip)

```
(START WITH BLOCK SKIP ON TO LOOP)
(VC5 IS THE TRIGGER VAR)
VC5=-1 (CLEAR OUT THE ROW INC COUNTER)
CALL OSSN (PROMPT OPERATOR FOR SHEET NAME)
CALL OLAC (LOOK AHEAD CANCEL)
CNT = 0 (INIT COUNTER)
NLP10 (LOOP)
CALL OLAC
CNT=CNT+1
CALL OLAC
VC1=VC1+.01
VC2=VC2+.01
VC3=VC3+.01
VC4=VC4+.01
CALL OLAC
IF[CNT GT 5]NJMP10
CALL OSTGR (TRIGGER)
/GOTO NEND
GOTO NLP10
NJMP10
CALL OSTGR PS=8 (TRIGGER SKIP 8 LINES)
CNT=0 (RESET COUNTER)
/GOTO NEND
GOTO NLP10
NEND
M2

OSTGR
(SET TRIGGER)
(INPUT PS = SKIP)
CALL OLAC
VC5 = 100 + PS
G4 F1. (ALLOW TIME TO WRITE)
NLP10
CALL OLAC
IF[VC5 NE 0]NLP10 (LOOP TILL 0)
G4 F1.
RTS

OSSN
(SET SHEET NAME)
VC5=200
NLP10
CALL OLAC
```

```
IF[VC5 NE 0]NLP10 (LOOP TILL 0)
G4 F1.
RTS


OLAC
(LOOK AHEAD CANCEL)
VDOUT[14]=0
PDMY=VDIN[1000]
PDMY=VDIN[1000]
RTS
```

26	318.36	1263.36	50.865	51.3852
27	318.37	1263.37	50.875	51.3952
28	318.38	1263.38	50.885	51.4052
29	318.39	1263.39	50.895	51.4152
30	318.4	1263.4	50.905	51.4252
31				
32				
33				
34				
35				
36				
37				
38				
39	318.41	1263.41	50.915	51.4352
40	318.42	1263.42	50.925	51.4452
41	318.43	1263.43	50.935	51.4552
42	318.44	1263.44	50.945	51.4652
43	318.45	1263.45	50.955	51.4752
44	318.46	1263.46	50.965	51.4852
45				
46				
47				
48				
49				
50				
51				
52				
53	318.47	1263.47	50.975	51.4952
54	318.48	1263.48	50.985	51.5052

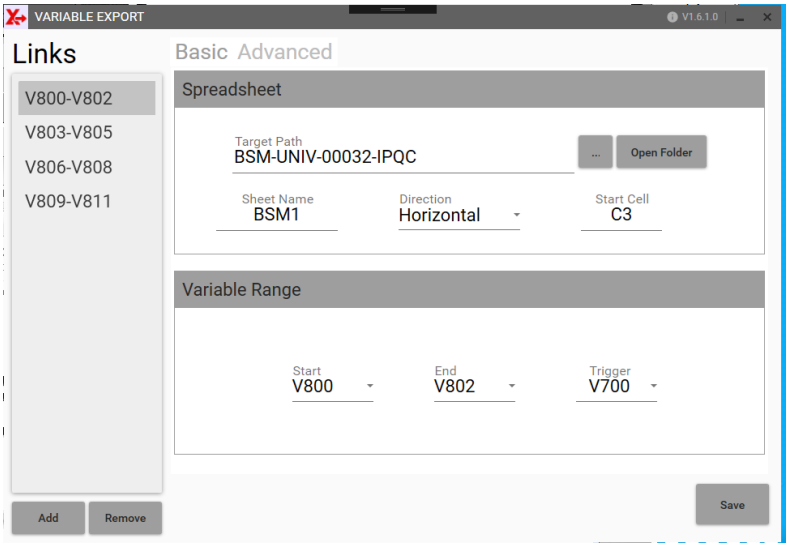
Advance Program Example 2

The following example will write 4 ranges of variables to a sheet and then increment to the next sheet the next time it run.

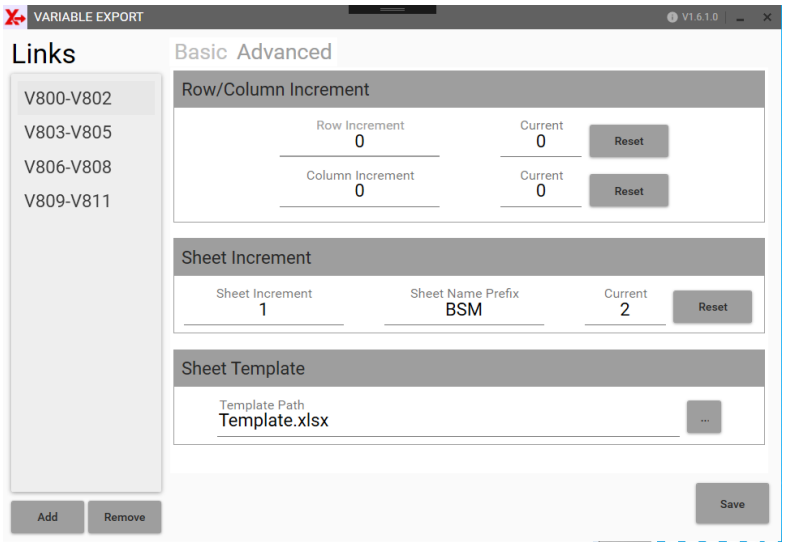
Link 1

Link 1 is setup to write the first series of 3 variables 800-802 to the workbook. The sheet increment is set to 1 so every time the trigger is set to 100 a new sheet will be created and Variables 800-802 will be written horizontally starting at C3.

Basic

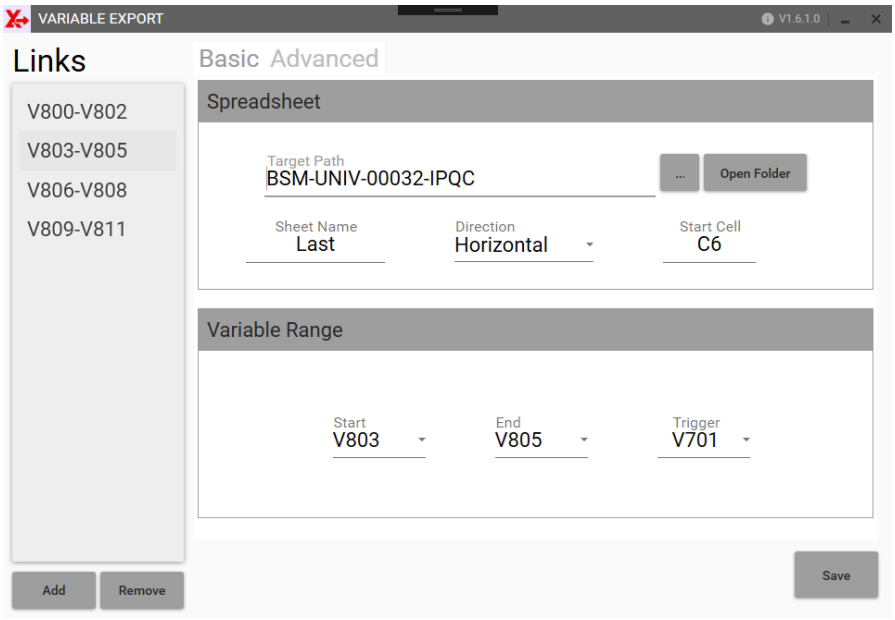


Advanced



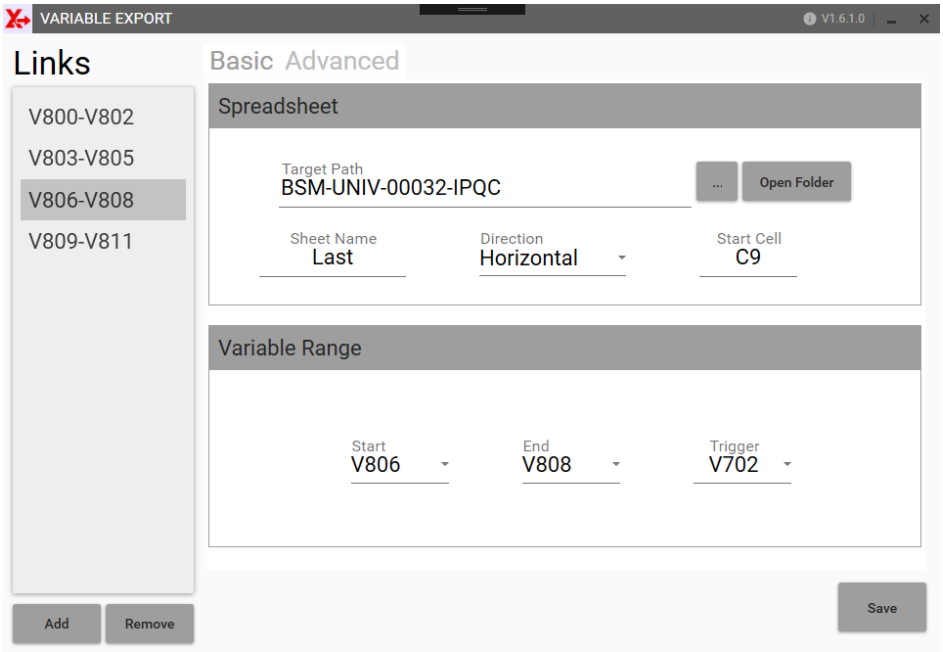
Link 2

Link 2 is set to record variables 803-805 to the last sheet of the workbook starting at cell C6. Link 2 is triggered with V701



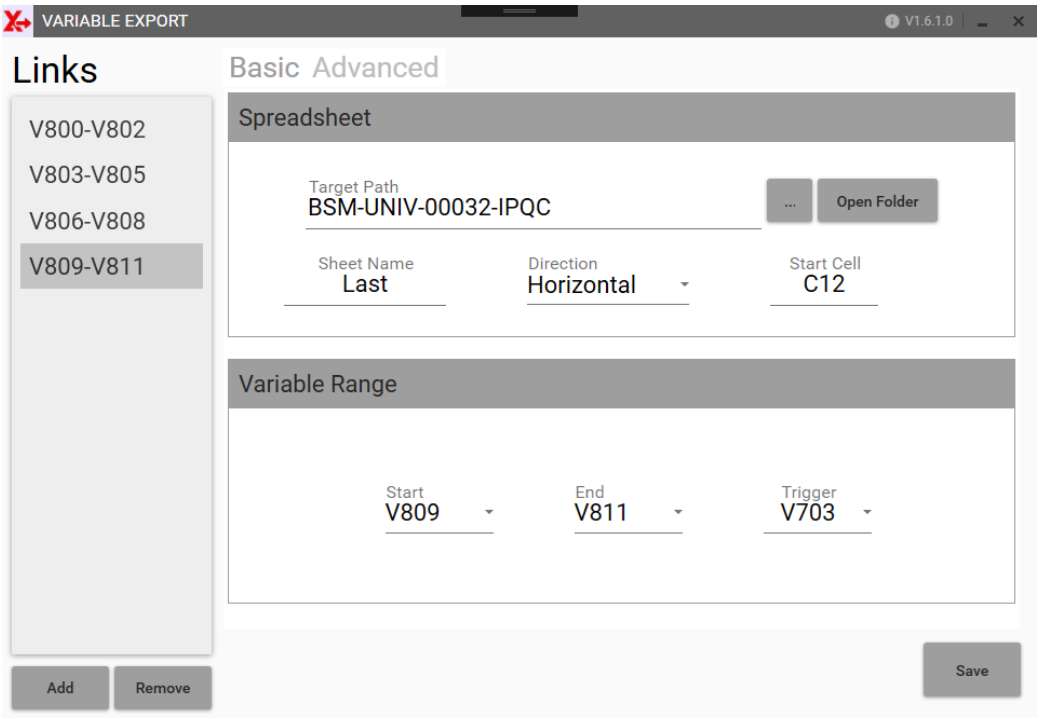
Link 3

Link 3 is set to record variables 806 to 808 to the last sheet of the workbook starting at cell C9. It uses V702 as the trigger



Link 4

Link 4 is set to record variable 809 to 811 to the last sheet of the workbook starting at cell C12



The program used to generate the workbook looks like this:

G13
G140

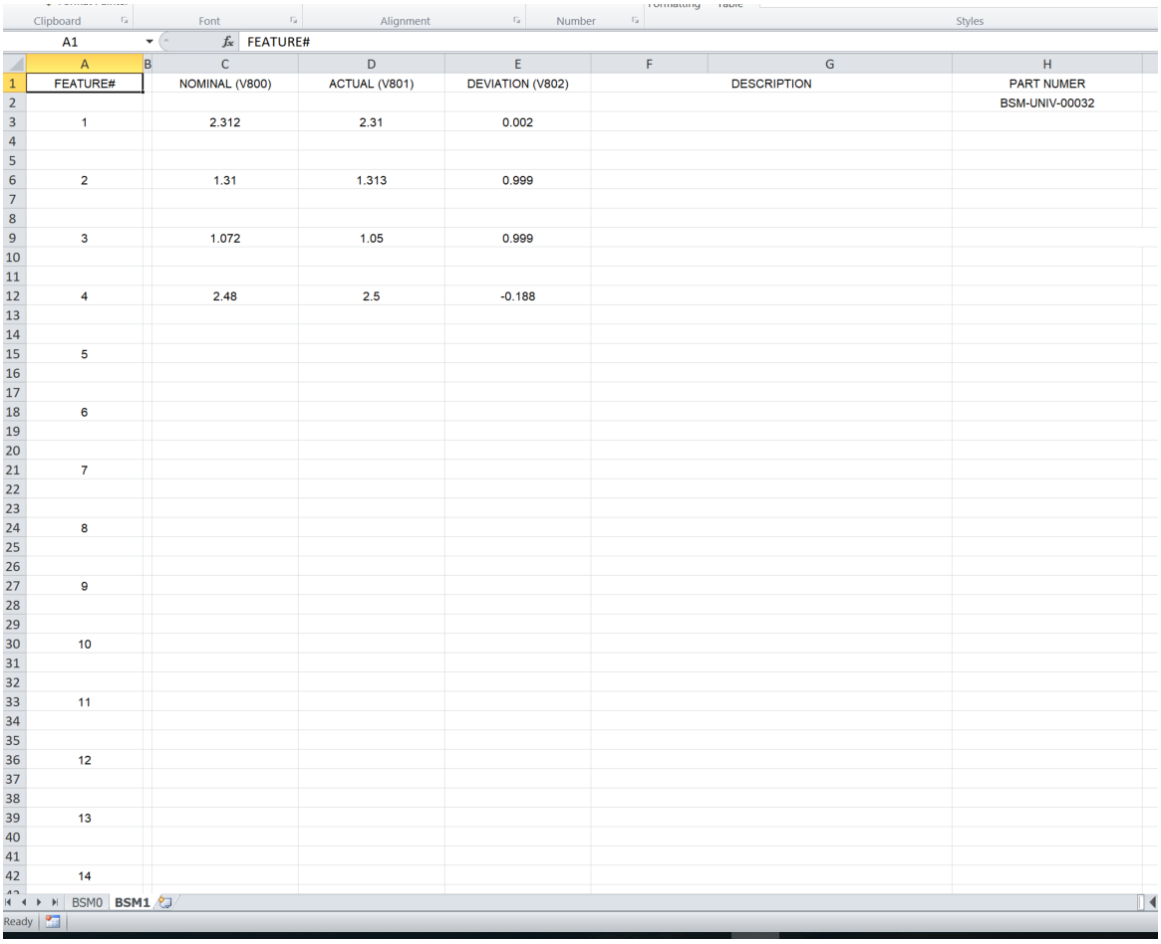
(LINK 1)
V800=2.3120 (ACTUAL PRINT DIMENSION)
V801=V138 (ACTUAL MEASURED DIMENSION)
V802=V143 (DEVIATION)
V700=100 (CREATE SHEET AND WRITE FIRST 3 POINTS)
M1

(LINK 2)
V803=1.310 (ACTUAL PRINT DIMENSION)
V804=V138 (ACTUAL MEASURED DIMENSION)
V805=V143 (DEVIATION)
V701=100 (WRITE NEXT 3 TO LAST SHEET)
M1

(LINK 3)
V806=1.072 (ACTUAL PRINT DIMENSION)
V807=V137 (ACTUAL MEASURED DIMENSION)
V808=V142 (DEVIATION)
V702=100 (WRITE NEXT 3 TO LAST SHEET)
M1

(LINK 4)
V809=2.480 (ACTUAL PRINT DIMENSION)
V810=V138 (ACTUAL MEASURED DIMENSION)
V811=V143 (DEVIATION)
V703=100 (WRITE LAST 3 POINTS TO LAST SHEET)
G14
M30

After running the program 2 times the output workbook looks like this:



FEATURE#	NOMINAL (V800)	ACTUAL (V801)	DEVIATION (V802)	DESCRIPTION	PART NUMER
					BSM-UNIV-00032
1	2.312	2.31	0.002		
2	1.31	1.313	0.999		
3	1.072	1.05	0.999		
4	2.48	2.5	-0.188		
5					
6					
7					
8					
9					
10					
11					
12					
13					
14					

Configuration

The configuration file for the application is stored @ D:\AppData\Gosiger\Go.VariableExport\Config. The configuration file for the defined variable ranges is located @ D:\AppData\Gosiger\Go.VariableExport \VariableRanges.json. Both of these file are portable and can be used to match configuration between machines.