

Getting Started with ProSimPlus

Use Case 3: Use of ProSimPlus Console with Excel

Software & Services In Process Simulation

We guide You to efficiency



ProSim

Introduction

The main objective of this document is to introduce the use of ProSimPlus Console "PSPC.exe" with Microsoft Excel®.

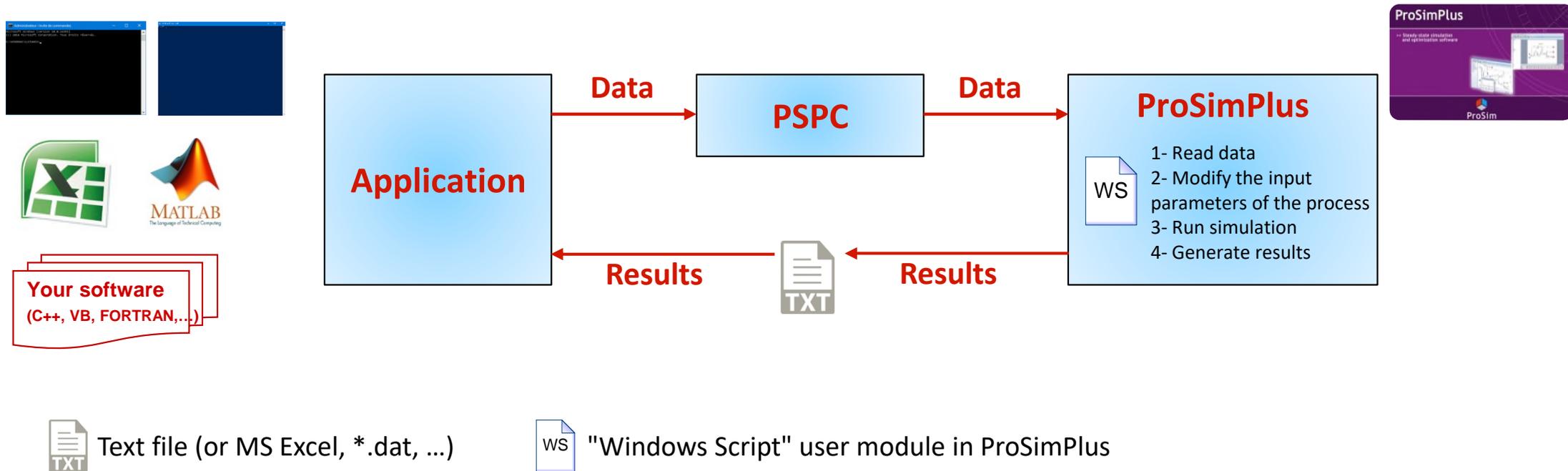
PSPC is the command line version of ProSimPlus running without graphical user interface (GUI).

This tool allows to use ProSimPlus through another application.

The example presented in this document explains how to use ProSimPlus in the Microsoft Excel® environment.

Introduction

The classical procedure for the PSPC call is the following:



The steps are as follows:

1. From the application, generate data and send it to PSPC
2. PSPC runs ProSimPlus: using a "Windows Script" in ProSimPlus, data are read. These data are used to modify the process input parameters. Once the simulation reaches the convergence, the results are generated in a text file (or MS Excel)
3. The results file is read by the application to get the simulation results

Introduction

This document presents the use of PSPC with Excel based on the example named "[Simple-Example](#)" which is delivered during the installation of ProSimPlus.

ProSim has created an Excel template file to import or export data from or to ProSimPlus using PSPC.

This file allows the user to get started with the use of PSPC with Excel.

The "template" file is named: "[PSPC_GS03_EN-ProSimPlusConsole.xlsx](#)"

Note: You can get more help on the use of PSPC by referring to the help available in the ProSimPlus installation directory: "PSPC Command line.pdf".

The "template" file uses Visual Basic for Applications (VBA) macros to call PSPC tool.

1- Configure the Excel file (data)

The 1st step is the configuration of the data sent to ProSimPlus via PSPC.

1. Specify which ProSimPlus file to use
2. Define the number of data to be sent
3. Define the number of data sets to be tested

In this example, we will vary 2 parameters (2 tags) on the studied process.

We will do a sensitivity study on 5 operating points (5 sets).

Use of PSPC to export or/and import data from/to Excel

Select the ProSimPlus file:



ProSimPlus file

C:\XXXXXXXXXXXX\IPSPS_EX_EN-Simple-Example.pmp3

Export Data from Excel to ProSimPlus

Tags number

2

Sets number

5

		Tags			
		Temp	Pres		
		°C	atm		
Sets	Set 1	-60	100		
	Set 2	-50	110		
	Set 3	-40	120		
	Set 4	-30	130		
	Set 5	-20	140		
	...				
	...				
	...				
	...				
	...				
...					



The icon  displays a dialog box for an easy search of the ProSimPlus file.

It is also possible to directly fill the file path without using the dialog box.

1- Configure the Excel file (data)

4. Then, the user has to define the "tags" for the data sent to ProSimPlus (a "tag" is a name assigned to each data)
5. Define the values for the different sets of parameters
6. Optional: It is possible to add a description or a unit in the cell under the "tag" of the parameter

Export Data from Excel to ProSimPlus					
Tags number		2			
Sets number		5			
		Tags			
		Temp	Pres		
		°C	atm		
Sets	Set 1	-60	100		
	Set 2	-50	110		
	Set 3	-40	120		
	Set 4	-30	130		
	Set 5	-20	140		
	...				
	...				
	...				
	...				
	...				

For the studied example, the 2 parameters modified in ProSimPlus are the outlet temperature of the exchanger "E101" and the pressure of the feed named "FEED". The tags (data names) are "Temp" for the exchanger temperature and "Pres" for the feed pressure.

We test 5 operating points: the temperature varies from -60 to -20°C and the pressure from 100 to 140 atm.

2- Configure the "Windows Script" in ProSimPlus

The Excel file is configured for data sent to ProSimPlus. It is now necessary to retrieve these data in ProSimPlus and to assign them to the parameters that we wish to modify.

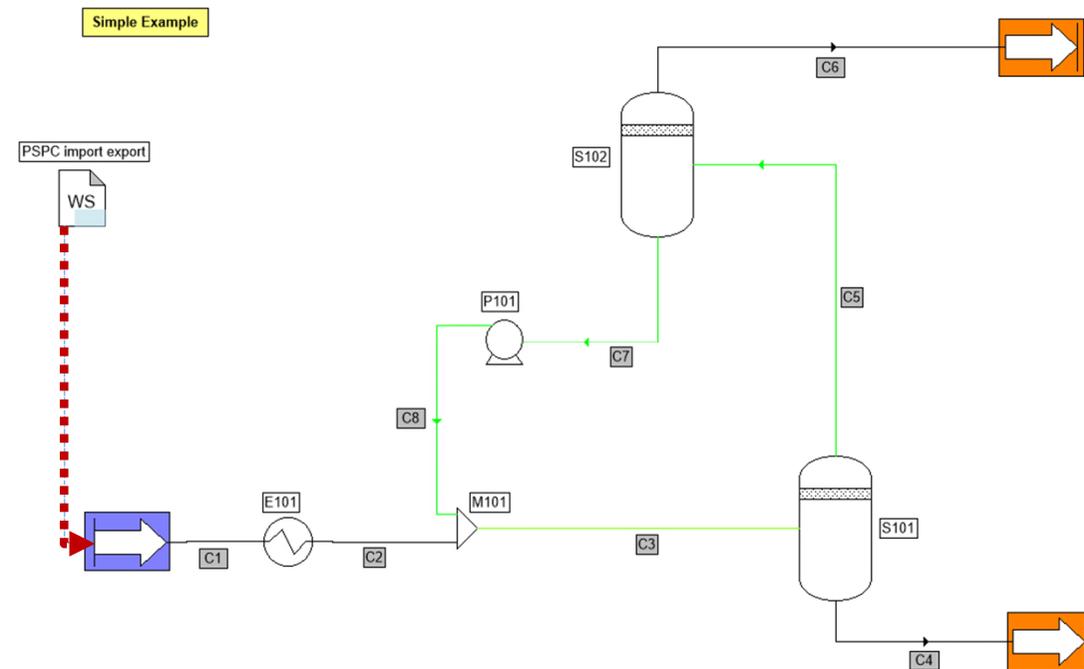
To do this, it is necessary to add a "Windows Script" user module that will be used for importing data from Excel but also for exporting data (results) to the Excel file once the simulation is complete.



This module has to be the first calculated module in the calculation sequence.

This is the only way to ensure that the data imported from Excel will be the data used in unit operations.

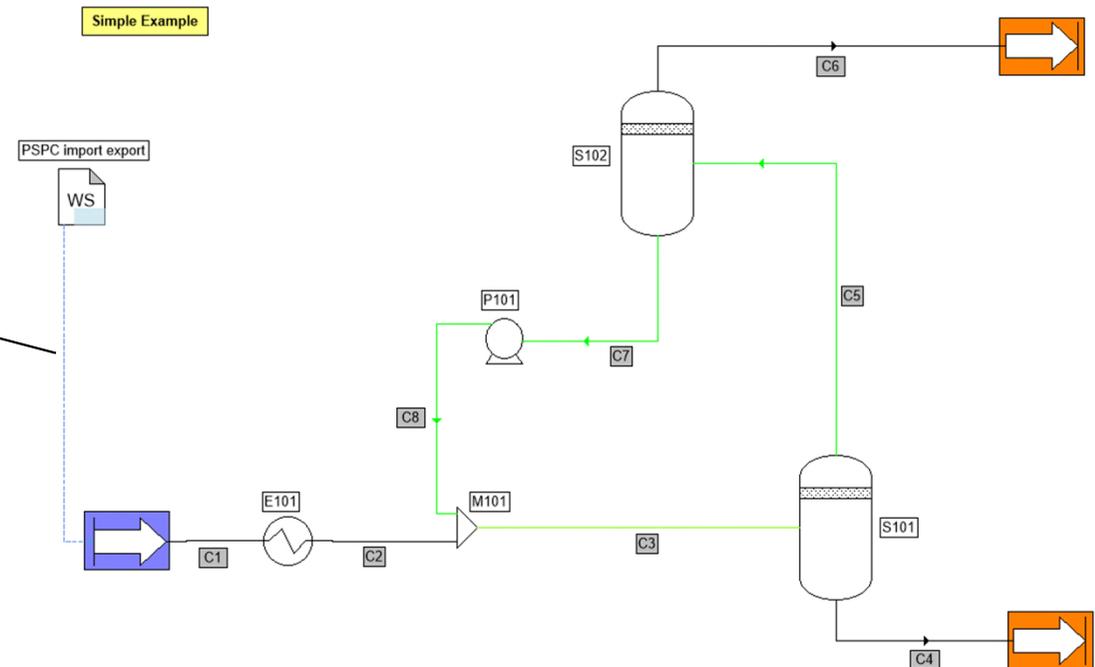
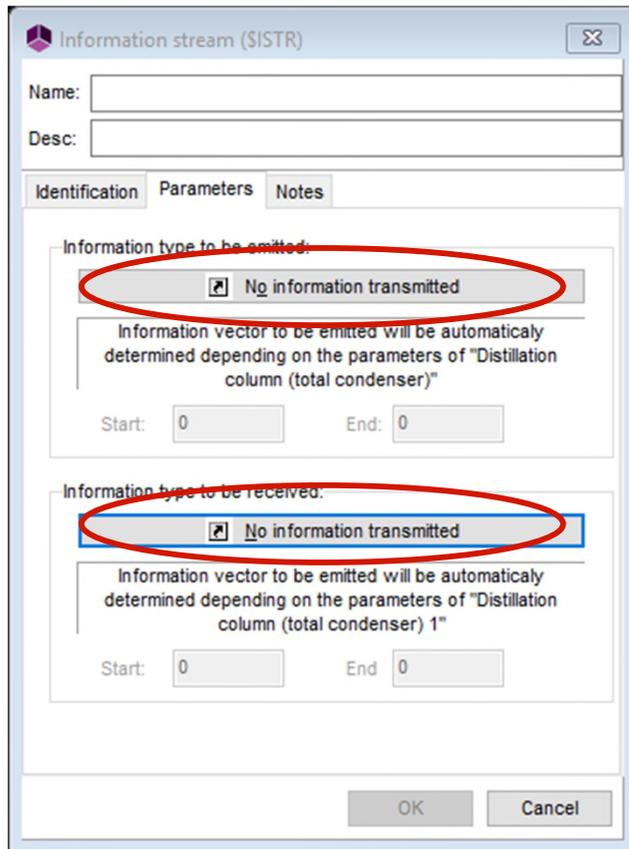
In the studied example, it is necessary to impose the calculation sequence by adding an information stream from the "Windows Script" module to the feed of the process "FEED".



2- Configure the "Windows Script" in ProSimPlus

Adding an information stream from the "Windows Script" to the "FEED" module imposes the calculation sequence on the software. The "Windows Script" module has to be calculated before the "FEED" module.

This information stream is a "fictive" information stream because it is only used to impose the calculation sequence (no information transmitted).



2- Configure the "Windows Script" in ProSimPlus

The script (calculation code) of the "Windows Script" is divided into several functions:

1. The "OnCalculation » function: this function is used to write the calculation code of the "Windows Script".
In this example, this function is used to retrieve the values sent by PSPC, which are imported from Excel. This routine is called when the "Windows Script" is calculated and allows to assign the data in the different unit operations.
2. The "OnPrintResults » routine: this routine is used to export the data (results) after convergence of the simulation.

2- Configure the "Windows Script" in ProSimPlus

The following code is a "template" code that can be used with "PSPC_GS03_EN-ProSimPlusConsole.xlsm" Excel file:

```
with CreateObject("Scripting.FileSystemObject")  
    ExecuteGlobal .OpenTextFile(Project.ApplicationPath & "Scripts\UnitConversion.vbs", 1).ReadAll()  
    ExecuteGlobal .OpenTextFile(Project.ApplicationPath & "Scripts\PSPCUtills.vbs", 1).ReadAll()  
end with
```

```
Function OnCalculation()  
,
```

```
TemperatureFromExcel = Project.UserValues("Temp")  
PressureFromExcel = Project.UserValues("Pres")  
,  
Project.Modules("E101").TemperatureSpecValue = ConvertToProSim("Temperature", TemperatureFromExcel, "°C")  
Project.Modules("FEED").OutputStreamPressureSpecValue = ConvertToProSim("Pressure", PressureFromExcel, "atm")  
,
```

```
OnCalculation = true
```

```
End Function
```

```
Sub OnPrintResults()  
,
```

```
dim Array
```

```
redim Array(2)
```

```
Array(0) = Project.Streams("C5").MassFlowrateIn("kg/h")  
Array(1) = Project.Streams("C6").MolarFlowrateIn("Nm3/h")  
Array(2) = Project.Modules("P101").UsefulPower
```

```
call SavePSPCResults(Array)
```

```
End Sub
```

Code to modify (add or delete parameters)

2- Configure the "Windows Script" in ProSimPlus

Configure the "OnCalculation" function:

```
Function OnCalculation()  
,
```

```
TemperatureFromExcel = Project.UserValues("Temp")
```

```
PressureFromExcel = Project.UserValues("Pres")  
,
```

```
Project.Modules("E101").TemperatureSpecValue = ConvertToProSim("Temperature", TemperatureFromExcel, "°C")
```

```
Project.Modules("FEED").OutputStreamPressureSpecValue = ConvertToProSim("Pressure", PressureFromExcel, "atm")  
,
```

```
OnCalculation = true
```

```
End Function
```

Retrieving tag values defined in Excel

		Tags			
		Temp	Pres		
		°C	atm		
Set 1		-60	100		
Set 2		-50	110		

The values extracted from the tags are saved in 2 intermediate variables "TemperatureFromExcel" and "PressureFromExcel".

These values are then assigned to the desired operating parameters: the outlet temperature of the exchanger "E101" and the pressure of the feed "FEED".

2- Configure the "Windows Script" in ProSimPlus

Configure the "OnCalculation" function:

Using the script, it is possible to access all the parameters of the simulation in order to assign the value imported from Excel to any parameter of the simulation.

In the "Script" tab of the module, you can access the different parameters:

```
Function OnCalculation()  
,
```

```
TemperatureFromExcel = Project.UserValues("Temp")
```

```
PressureFromExcel = Project.UserValues("Pres")
```

```
Project.Modules("E101").TemperatureSpecValue = ConvertToProSim("Temperature", TemperatureFromExcel, "°C")
```

```
Project.Modules("FEED").OutputStreamPressureSpecValue = ConvertToProSim("Pressure", PressureFromExcel, "atm")
```

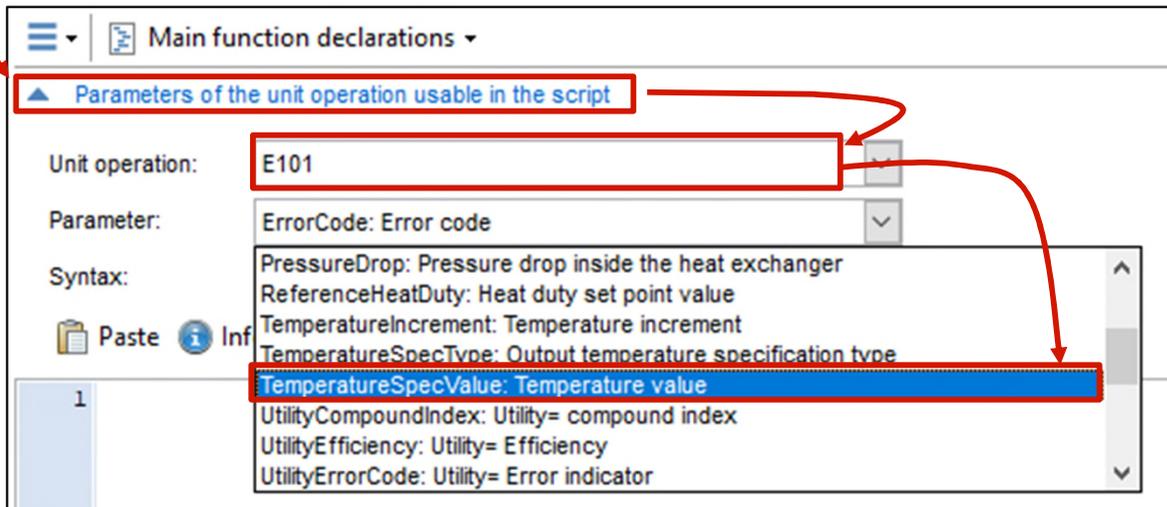
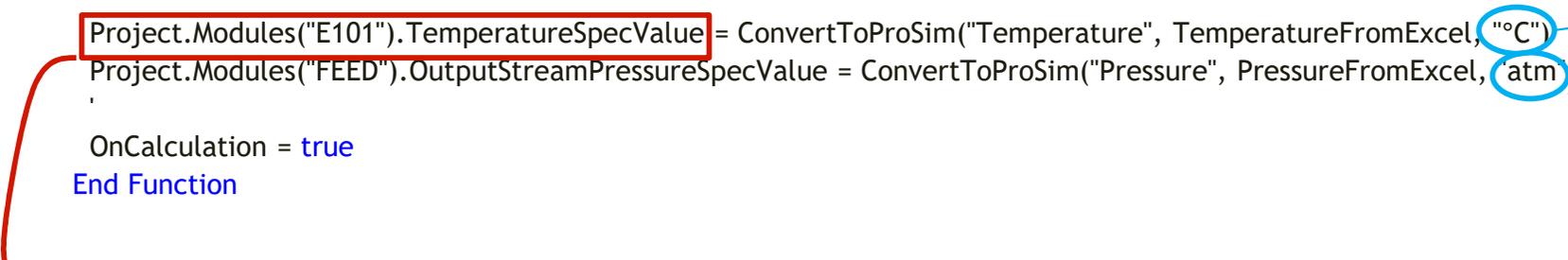
```
OnCalculation = true
```

```
End Function
```

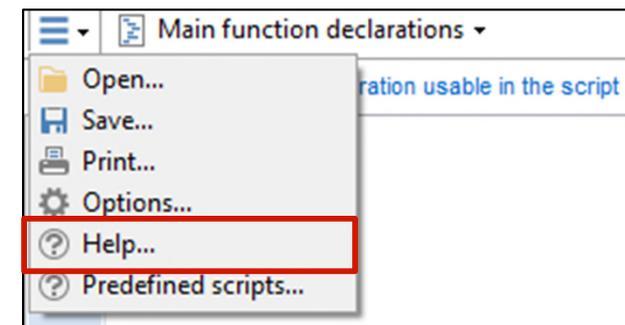


The "ConvertToProSim" function converts the values in the ProSimPlus units

Temp	Pres
°C	atm



Help on using the script is also available



2- Configure the "Windows Script" in ProSimPlus

Configure the "OnPrintResults" routine:

When the results are printed, the module sends the results to Excel.

It is necessary to modify the dimension of the "Array" vector according to the number of results transmitted to Excel.

For the studied example, we decide to export the mass flowrate of the stream "C5", the molar flowrate of the stream "C6" and the electric power of the pump "P101".

```
Sub OnPrintResults()  
,  
  
    dim Array  
    redim Array(2)  
    Array(0) = Project.Streams("C5").MassFlowrateIn("kg/h")  
    Array(1) = Project.Streams("C6").MolarFlowrateIn("Nm3/h")  
    Array(2) = Project.Modules("P101").UsefulPower  
,  
  
    call SavePSPCResults(Array)  
End Sub
```

To add new data to export, simply modify the dimension of the "Array" vector (ex: redim Array (3)) and add the new parameter

(ex: Array (3) = Project.Streams("C8").MassFlowrateIn ("kg/s"))

3- Configure the Excel file (results)

The results are saved in a text file (".txt"). This file is saved in the same folder as the ProSimPlus file with the extension "-out.txt". In our case, the output file is "PSPS_EX_EN_Simple-Example-out.txt".

Import Data from ProSimPlus to Excel

This file is read by Excel and the results are written in the "Results" section of the Excel file for each "set" of data (each operating point)

		Results				
		Property N°1	Property N°2	Property N°3	Property N°4	Property N°5
Sets	Set 1					
	Set 2					
	Set 3					
	Set 4					
	Set 5					
	...					
	...					
	...					
	...					
	...					
	...					
	...					

3- Configure the Excel file (results)

The columns of the results table are displayed in the order imposed in the "OnPrintResults" routine of ProSimPlus "Windows Script".

The column named "Property n°1" will display the values of "Array (0)". The column named "Property n°2" will display the values of "Array (1)"

For this example:

1st column (Property n°1): Mass flowrate of the stream "C5"

2nd column: Molar flowrate of stream "C6"

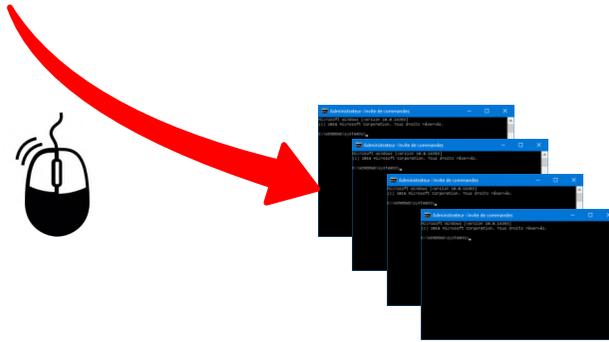
3rd column: Electric power of the pump "P101".

Import Data from ProSimPlus to Excel

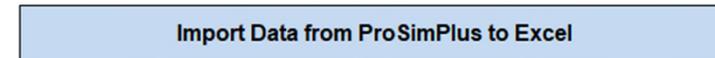
		Results				
		Property N°1	Property N°2	Property N°3	Property N°4	Property N°5
Sets	Set 1					
	Set 2					
	Set 3					
	Set 4					
	Set 5					
	...					
	...					
	...					
	...					
	...					

4- Utilisation de PSPC

To run the simulation with PSPC, just click on the "PSPC" button



Depending on the number of data sets, one or more command line windows appear on the screen



At the end of the calculation, the results are displayed in the results table

		Results			
		Property n°1	Property n°2	Property n°3	
		kg/hr	Nm3/h	kcal/hr	
Sets	Set 1	242,99	263,28	0,65	
	Set 2	326,65	358,98	2,02	
	Set 3	405,77	449,60	5,81	
	Set 4	479,43	532,20	14,96	
	Set 5	547,85	605,27	33,75	
	...				
	...				
	...				
	...				
	...				
	...				
	...				

4- Use of PSPC

It is possible to change the fields in green to define the type of each property and its unit:

	Results		
	Property n°1	Property n°2	Property n°3
	kg/hr	Nm3/h	kcal/hr
Set 1	242,99	263,28	0,65
Set 2	326,65	358,98	2,02
Set 3	405,77	449,60	5,81
Set 4	479,43	532,20	14,96
Set 5	547,85	605,27	33,75



	Results		
	Stream C5	Stream C6	Power P101
	kg/hr	Nm3/h	kcal/hr
Set 1	242,99	263,28	0,65
Set 2	326,65	358,98	2,02
Set 3	405,77	449,60	5,81
Set 4	479,43	532,20	14,96
Set 5	547,85	605,27	33,75

By default, the parameters of the modules (unit operations) or streams are saved in "ProSim" units (ProSim unit system).

It is possible to convert the value in the unit specified by the report with the function "ProSimToReport" in the "Windows Script". Example: Conversion of pump power into report unit:

```
Sub OnPrintResults()
    '
    dim Array
    redim Array(2)
    Array(0) = Project.Streams("C5").MassFlowrateIn("kg/h")
    Array(1) = Project.Streams("C6").MolarFlowrateIn("Nm3/h")
    Array(2) = ProSimToReport("Power", Project.Modules("P101").UsefulPower)
    '
    call SavePSPCResults(Array)
End Sub
```

	Results		
	Stream C5	Stream C6	Power P101
	kg/hr	Nm3/h	W
Set 1	242,99	263,28	0,76
Set 2	326,65	358,98	2,35
Set 3	405,77	449,60	6,75
Set 4	479,43	532,20	17,39
Set 5	547,85	605,27	39,23

4- Use of PSPC

Usage tip for another case study:

Before using the templates (Excel file and "script" code for "Windows Script"), it is important for the user to test the various operating points (operating point with the minimum and maximum values of the various modified parameters) in order to ensure the reliability and the convergence of the simulation before using PSPC.

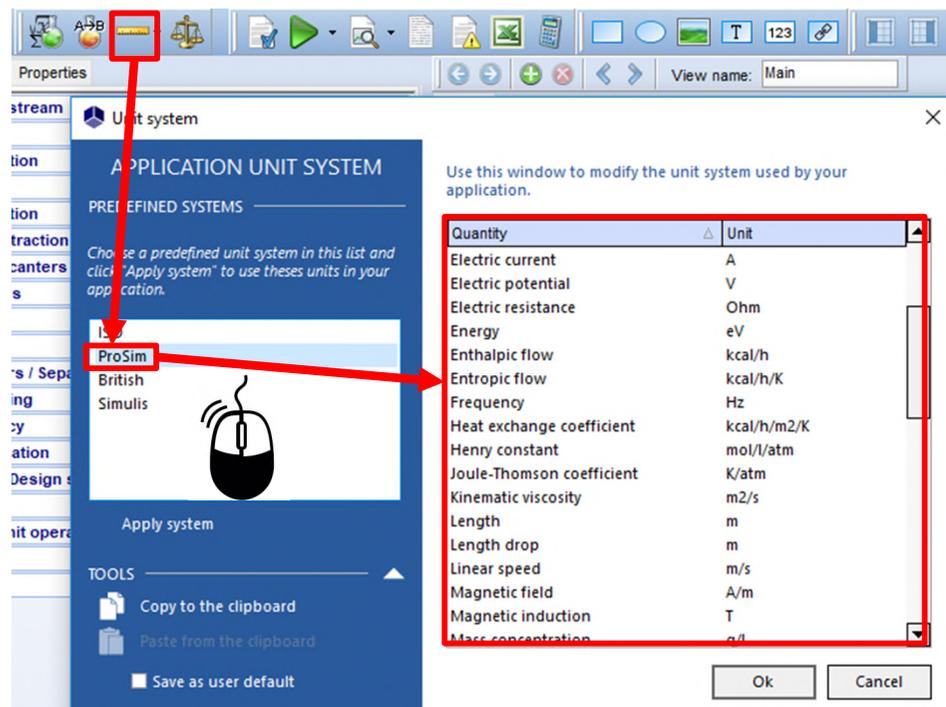
The call to ProSimPlus by PSPC has the advantage of avoiding the ProSimPlus interface.

The main drawback is the lack of information in case of non-convergence (because there is no interface). In this case, no result will be displayed and the reason for the non-convergence will not be known easily (need to open the ProSimPlus interface and run the simulation for the operating point that does not work).

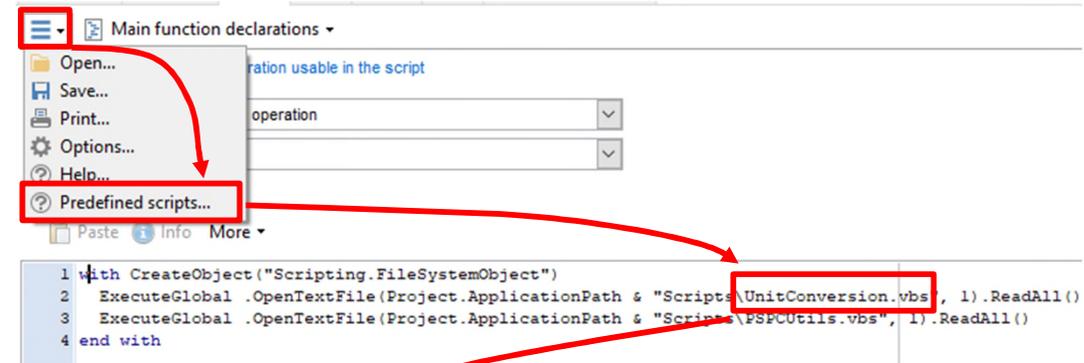
4- Use of PSPC

For more information:

About "ProSim" units and quantities



About functions and predefined scripts



UnitConversion.vbs

This source contains routines dedicated to conversion unit (get unit and convert in to another unit)

Global Procedure Summary

[Convert](#)(QuantityName, Value, FromUnit, ToUnit)

To convert a value from one unit into another.

[ConvertFromProSim](#)(QuantityName, Value, ToUnit)

To convert a value from the ProSim unit into another unit.

[ConvertToProSim](#)(QuantityName, Value, FromUnit)

To convert a value from one unit into the ProSim unit.

[ProSimToReport](#)(QuantityName, Value)

To convert values from ProSim to Report unit systems.

[ProSimUnit](#)(QuantityName)

To get ProSim unit corresponding to a given quantity.

[ReportUnit](#)(Quantityname)

To get Report unit corresponding to a given quantity.



ProSim SA
51, rue Ampère
Immeuble Stratège A
F-31670 Labège
France

☎: +33 (0) 5 62 88 24 30



ProSim

Software & Services In Process Simulation

www.prosim.net
info@prosim.net



ProSim, Inc.
325 Chestnut Street, Suite 800
Philadelphia, PA 19106
U.S.A.

☎: +1 215 600 3759