

Démarrer rapidement avec ProSimPlus

Cas 3 : Utilisation de ProSimPlus Console avec Excel

Software & Services In Process Simulation

We guide You to efficiency



ProSim

Introduction

L'objectif de ce document est de présenter l'utilisation de ProSimPlus Console « PSPC.exe » avec Microsoft Excel®.

PSPC est une version de ProSimPlus en ligne de commandes et sans aucune interface graphique.

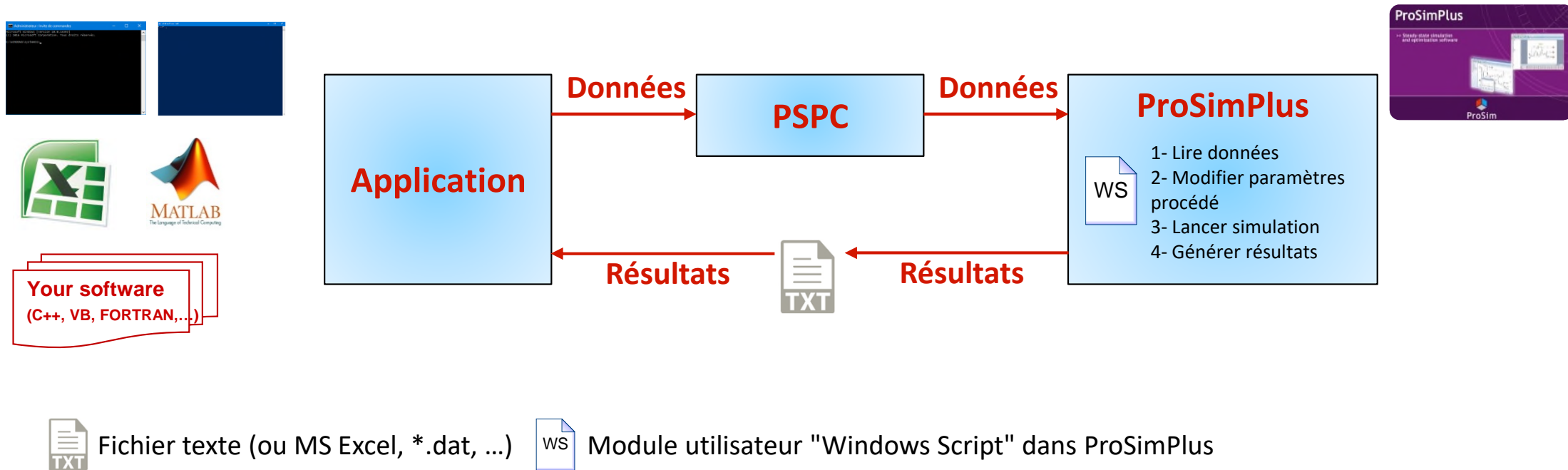
Cet outil permet d'utiliser ProSimPlus à travers une autre application.

L'exemple étudié dans ce document permet d'utiliser ProSimPlus dans l'environnement de Microsoft Excel®.

Introduction

3

Le schéma classique d'appel à PSPC est le suivant :



Les étapes sont les suivantes :

1. Depuis l'application, générer des données et les envoyer à PSPC
2. PSPC lance ProSimPlus : par le biais d'un "Windows Script" dans ProSimPlus, les données sont lues. Ces données sont utilisées pour modifier les paramètres du procédé. Une fois la simulation convergée, les résultats sont générés dans un fichier texte (ou MS Excel)
3. Le fichier de résultats est lu par l'application pour récupérer les résultats de simulation

Introduction

Ce document présente l'utilisation de PSPC avec Excel en s'appuyant sur l'exemple « [Exemple-Simple](#) » disponible dans les exemples livrés lors de l'installation de ProSimPlus.

ProSim a conçu un fichier Excel « modèle » pour réaliser l'import ou l'export de données depuis ou vers ProSimPlus en utilisant PSPC.

Ce fichier permet à l'utilisateur de débiter avec l'utilisation de PSPC avec Excel.

Le fichier « modèle » est le suivant : « [PSPS_GS03_FR-ProSimPlusConsole.xlsm](#) »

Note : Il est possible d'obtenir plus d'aide sur l'utilisation de PSPC en se référant à l'aide disponible dans le répertoire d'installation de ProSimPlus : « PSPC Ligne de commande.pdf ».

Le fichier « modèle » utilise des macros VBA (Visual Basic for Applications) pour réaliser l'appel à PSPC.

1- Configurer le fichier Excel (données)

Le fichier Excel « modèle » est présenté ci-dessous. Il peut être divisé en 2 parties :

- 1- Les données envoyées à ProSimPlus
- 2- Les résultats récupérés à partir de ProSimPlus

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

Select the ProSimPlus file:



ProSimPlus file

Legend

XXX

User data

XXX

Results

Export Data from Excel to ProSimPlus

Tags number

Sets number

		Tags				
Sets	Set 1					
	Set 2					
	Set 3					
	Set 4					
	Set 5					
	...					
	...					
	...					
	...					
	...					
	...					
	...					

Données envoyées à ProSimPlus

Import Data from ProSimPlus to Excel

		Results				
Sets	Set 1					
	Set 2					
	Set 3					
	Set 4					
	Set 5					
	...					
	...					
	...					
	...					
	...					
	...					
	...					

Résultats récupérés à partir de ProSimPlus



1- Configurer le fichier Excel (données)

La 1^{ère} étape est la configuration des données envoyées à ProSimPlus via PSPC.

1. Définir le fichier ProSimPlus utilisé
2. Définir le nombre de données envoyées
3. Définir le nombre de jeux de données testés

Pour notre exemple, nous allons faire varier 2 paramètres (2 tags) sur le procédé étudié.

Nous ferons une étude de sensibilité sur 5 points de fonctionnement (5 sets).

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

Select the ProSimPlus file:



1

ProSimPlus file

C:\XXXXXXXXXXXX\PSPS_EX_FR-Exemple-Simple.pmp3

Export Data from Excel to ProSimPlus

Tags number

2

2


Sets number

5

3

		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		
	...				
	...				
	...				
	...				
	...				



L'icône  permet d'afficher une boîte de dialogue pour rechercher plus facilement le fichier ProSimPlus. Il est aussi possible de renseigner directement le chemin du fichier sans utiliser la boîte de dialogue.

1- Configurer le fichier Excel (données)

4. Il faut ensuite définir des « tags », c'est-à-dire des noms, aux données envoyées à ProSimPlus
5. Définir les valeurs pour les différents jeux de paramètres
6. Optionnel : Il est possible d'ajouter un descriptif ou une unité dans la cellule sous le « tag » du paramètre

Export Data from Excel to ProSimPlus

Tags number

2

Sets number

5

4

6

Tags

Temp	Pres			
°C	atm			
-60	100			
-50	110			
-40	120			
-30	130			
-20	140			

5

Sets

Set 1

Set 2

Set 3

Set 4

Set 5

Pour l'exemple étudié, les 2 paramètres modifiés dans ProSimPlus sont la température de sortie de l'échangeur « E101 » et la pression de l'alimentation « FEED ».

Les tags (noms des données) sont « Temp » pour la température de l'échangeur et « Pres » pour la pression de l'alimentation.

Nous testons 5 points de fonctionnement : la température varie de -60 à -20°C et la pression de 100 à 140 atm.

2- Configurer le « Windows Script » dans ProSimPlus

Le fichier Excel est configuré pour les données envoyées à ProSimPlus. Il faut maintenant récupérer ces données dans ProSimPlus et les affecter aux paramètres que nous souhaitons modifier.

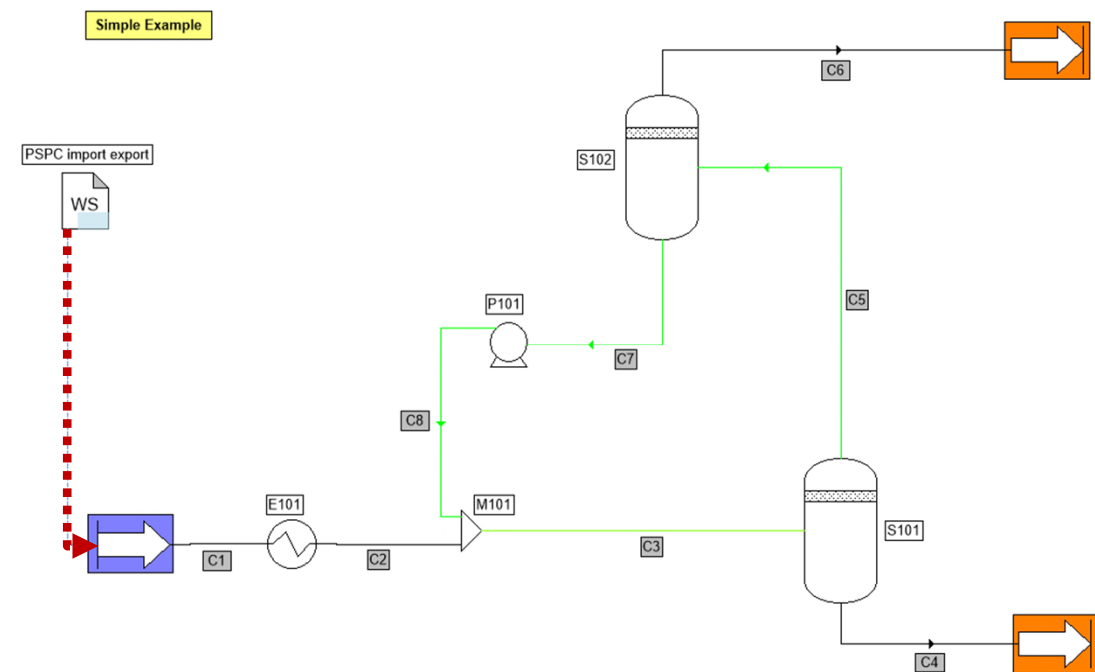
Pour ce faire, il faut ajouter un module utilisateur « Windows Script » qui servira pour l'import des données depuis Excel mais aussi pour l'export des données (résultats) vers Excel à convergence.



Ce module doit être calculé en premier dans la séquence de calcul.

C'est la seule façon de s'assurer que les données importées depuis Excel seront bien celles utilisées dans les opérations unitaires.

Dans l'exemple étudié, il est nécessaire d'imposer la séquence de calcul en ajoutant un courant d'information depuis le module « Windows Script » vers l'alimentation « FEED » du procédé.



2- Configurer le « Windows Script » dans ProSimPlus

9

L'ajout d'un courant d'information depuis le « Windows Script » vers l'alimentation « FEED » indique au logiciel que le module « Windows Script » doit être calculé avant l'alimentation.

Ce courant d'information est un courant d'information « fictif » dans le sens où il ne sert qu'à imposer la séquence de calcul (aucune information n'est transmise).

Courant information (SISTR)

Nom:

Desc:

Identification Paramètres Notes

Nature de l'information à émettre:

☒ Aucune information transmise

Le segment d'information à émettre sera déterminé automatiquement en fonction des paramètres du module "Alimentation du procédé"

Début: Fin:

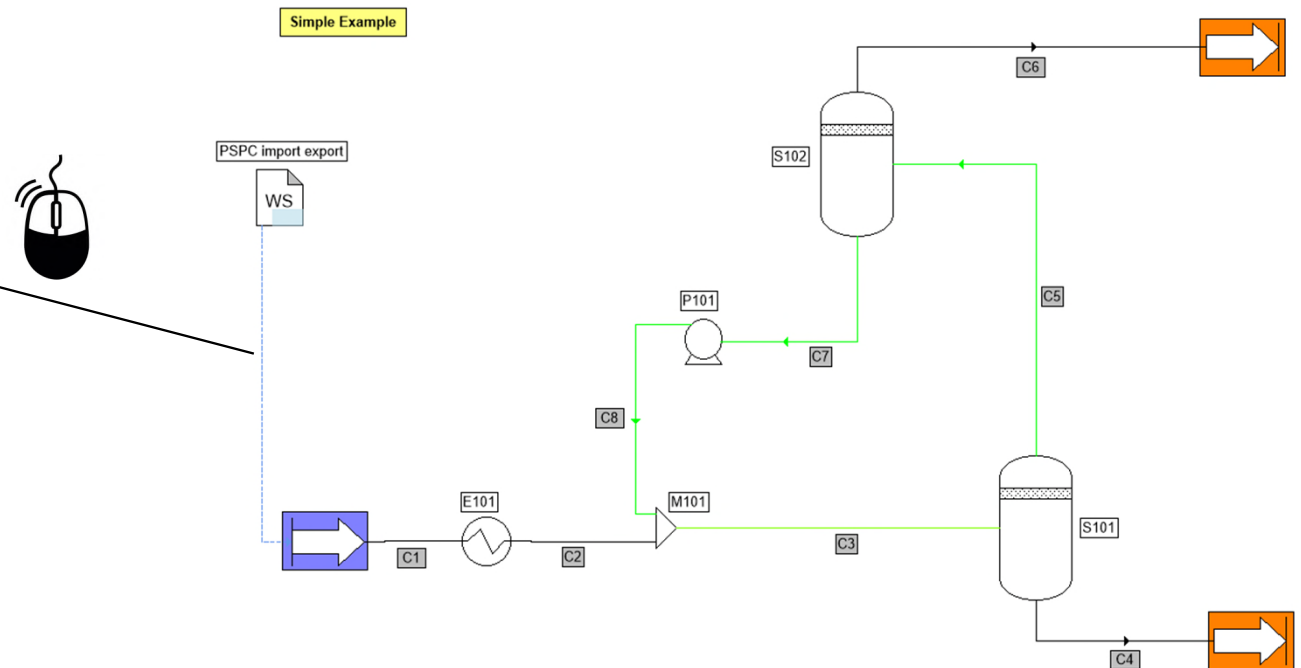
Nature de l'information à récupérer:

☒ Aucune information transmise

Le segment d'information à recevoir sera déterminé automatiquement en fonction des paramètres du module "Alimentation du procédé 1"

Début: Fin:

OK Annuler



2- Configurer le « Windows Script » dans ProSimPlus

Le script (code de calcul) du « Windows Script » est divisé en plusieurs fonctions :

1. La fonction « OnCalculation » : cette fonction sert à écrire le code de calcul du « Windows Script ».

Dans cet exemple, cette fonction est utilisée pour récupérer les valeurs importées par PSPC depuis Excel. Cette routine est appelée lors du calcul du « Windows Script » et permet d'affecter les données dans les différentes opérations unitaires.

2. La routine « OnPrintResults » : cette routine permet d'exporter les données (résultats) après convergence de la simulation.

2- Configurer le « Windows Script » dans ProSimPlus

Le code suivant est un code « modèle » qui peut être utilisé avec l'utilisation du fichier Excel « PSPS_GS03_FR-ProSimPlusConsole.xlsm »:

```
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 à modifier pour adaptation
(ajout ou suppression de paramètres)

2- Configurer le « Windows Script » dans ProSimPlus

Description de la fonction « OnCalculation » :

```
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
```

Récupération des valeurs des tags
définis dans Excel

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

Les valeurs récupérées dans les tags sont stockées dans 2 variables intermédiaires « TemperatureFromExcel » et « PressureFromExcel ».

Ces valeurs sont ensuite affectées aux paramètres opératoires désirés à savoir la température de sortie de l'échangeur « E101 » et la pression de l'alimentation « FEED ».

2- Configurer le « Windows Script » dans ProSimPlus

Description de la fonction « OnCalculation » :

En utilisant le script, il est possible d'accéder à tous les paramètres de la simulation afin d'affecter la valeur importée depuis Excel dans n'importe quel paramètre.

Dans l'onglet « Script » du module, vous pouvez accéder aux différents paramètres :

```
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")
```

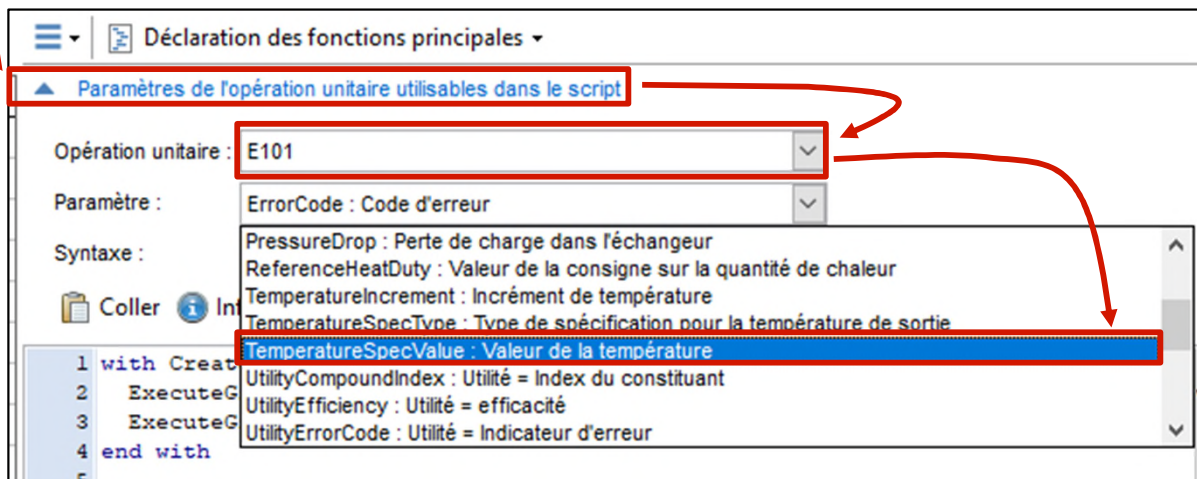
```
OnCalculation = true
```

```
End Function
```

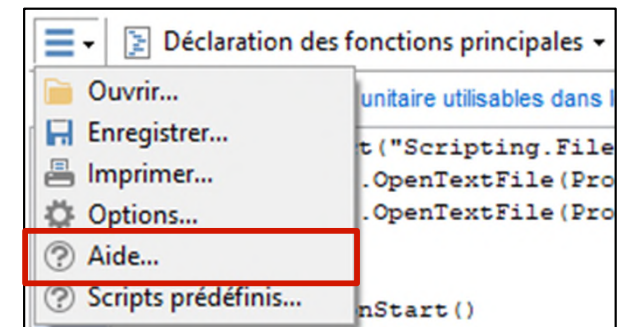


La fonction « ConvertToProSim » permet de convertir les valeurs dans les unités utilisées dans ProSimPlus

Temp	Pres
°C	atm



L'aide sur l'utilisation du script est également disponible



2- Configurer le « Windows Script » dans ProSimPlus

Description de la routine « OnPrintResults » :

Au moment d'imprimer les résultats, le module transmet les résultats à Excel.

Il est nécessaire de modifier la dimension du vecteur « Array » en fonction du nombre de résultats transmis à Excel.

Pour le cas étudié, nous décidons de récupérer le débit massique du courant « C5 », le débit molaire du courant « C6 » et la puissance électrique de la pompe « 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
```

Pour ajouter une donnée à exporter, il suffit de modifier la dimension du vecteur « Array » (ex : redim Array(3)) et d'ajouter le paramètre exporté (ex : Array(3) = Project.Streams("C8").MassFlowrateIn("kg/s"))

3- Configurer le fichier Excel (résultats)

Les résultats sont stockés dans un fichier texte (« .txt »). Ce fichier est enregistré dans le même dossier que le fichier ProSimPlus avec l'extension « -out.txt ». Dans notre cas, le fichier est « PSPS_EX_FR_Exemple-Simple-out.txt ».

Import Data from ProSimPlus to Excel

Ce fichier est lu par Excel et les résultats récupérés sont stockés dans la partie « Résultats » du fichier Excel pour chaque jeu de données « set » (point de fonctionnement)

		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- Configurer le fichier Excel (résultats)

Les colonnes du tableau des résultats sont affichées dans l'ordre imposé dans la routine « OnPrintResults » du « Windows Script » de ProSimPlus.

La colonne « Property n°1 »
affichera les valeurs de « Array (0) »

La colonne « Property n°2 »
affichera les valeurs de « Array (1) »

Pour le cas étudié :

1^{ère} colonne (Property n°1) : Débit
massique du courant « C5 »

2^{ème} colonne : Débit molaire du
courant « C6 »

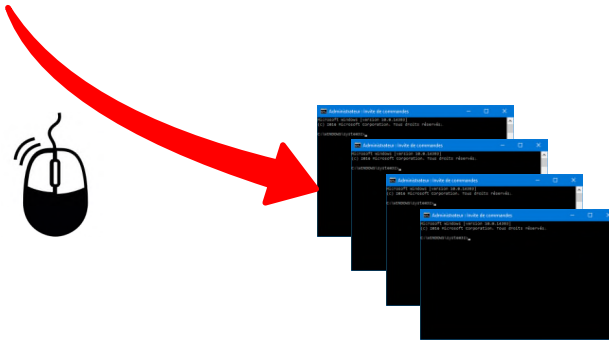
3^{ème} colonne : Puissance électrique
de la pompe « P101 »

Import Data from ProSimPlus to Excel

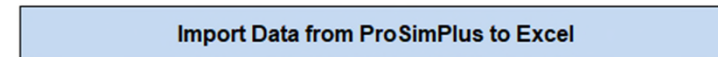
		Results				
		Property N°1	Property N°2	Property N°3	Property N°4	Property N°5
Sets	Set 1					
	Set 2					
	Set 3					
	Set 4					
	Set 5					
	...					
	...					
	...					
	...					
	...					

4- Utilisation de PSPC

Pour lancer la simulation avec PSPC, il suffit de cliquer sur le bouton « PSPC »



Selon le nombre de jeux de données, une ou plusieurs fenêtres en ligne de commande s'affichent à l'écran



A la fin du calcul, les données sont affichées dans le tableau des résultats

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	
Sets	---				

4- Utilisation de PSPC

Il est possible de changer les champs en vert afin de renseigner le type de propriété et son unité :

	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

Les paramètres stockés au niveau des modules (opérations unitaires) ou des courants dans ProSimPlus sont en unité « ProSim ».

Il est possible de convertir la valeur dans l'unité spécifiée par le rapport avec la fonction « ProSimToReport » dans le « Windows Script ». Exemple : Conversion de la puissance de la pompe :

```
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- Utilisation de PSPC

Conseil d'utilisation pour un autre cas d'étude :

Avant d'utiliser les modèles (fichier Excel et code « script » pour « Windows Script »), il est important que l'utilisateur teste les différents points de fonctionnement (cas de fonctionnement avec les valeurs minimales et maximales des différents paramètres modifiés) afin d'assurer la fiabilité et la convergence de la simulation avant même d'utiliser PSPC.

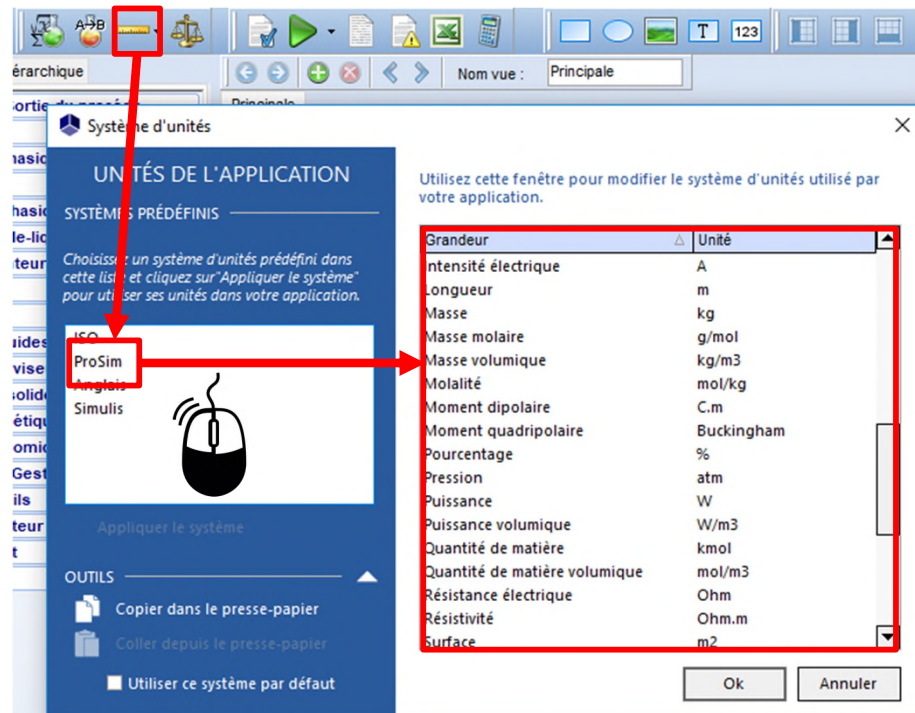
L'appel à ProSimPlus par PSPC a l'avantage de s'affranchir de l'interface de ProSimPlus.

Le principal inconvénient est l'absence d'information en cas de non-convergence (car aucune interface). Dans ce cas de figure, aucun résultat ne sera affiché et la raison de la non-convergence ne sera pas connue facilement (nécessité d'ouvrir l'interface ProSimPlus et de lancer la simulation pour le point de fonctionnement bloquant).

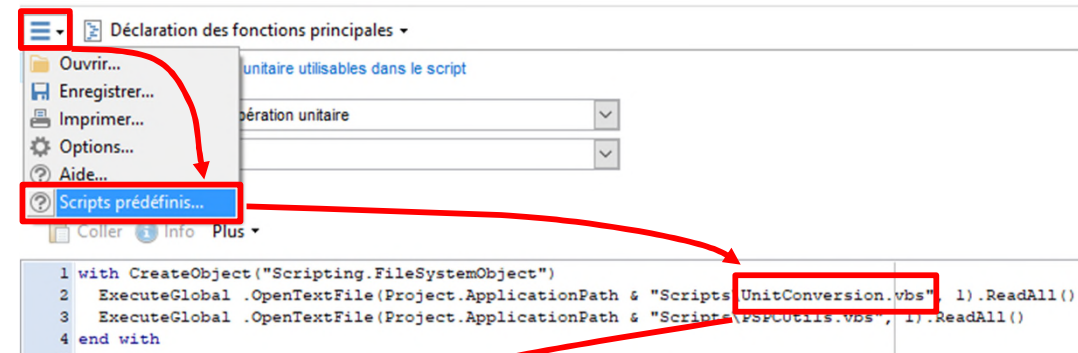
4- Utilisation de PSPC

Pour plus d'informations :

Sur les unités « ProSim » et les grandeurs



Sur les fonctions et scripts prédéfinis



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

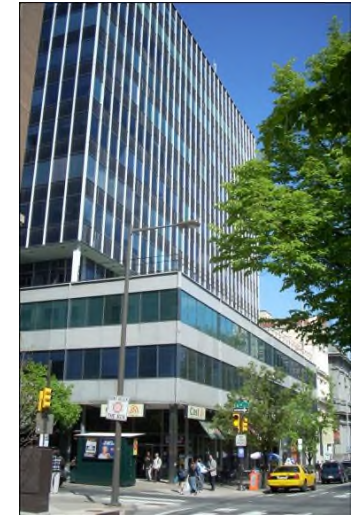
51, rue Ampère
Immeuble Stratège A
F-31670 Labège
France

☎: +33 (0) 5 62 88 24 30



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