

56. Sim Flowsheet Excel Wizards

General

Excel wizards are a quick way to set up the units according to models and also a way to store the models to be used in different process simulations. Of course you may create units manually (fill the Model-sheet using the same syntax as in MS Excel) but Excel wizards may be used to automate the creation of the most often used unit types.

HSC Sim Excel wizards are Excel files where each sheet is a wizard of its own. The content of the wizard is copied in the “Wizard” page of the unit (created if not existing) and wizards are applied by running the wizard. The upper part of the wizard is copied to the Model page and the lower part of the wizard makes appropriate changes to the unit using the command lines.

How to use the existing wizards?

Units can be prepared with wizards. Wizards are stored in Microsoft Excel files in the HSC Chemistry installation folder ...\\Sim\\Wizards\\... in which case wizards are called up from a unit by selecting it in the menu of Excel model of the unit.

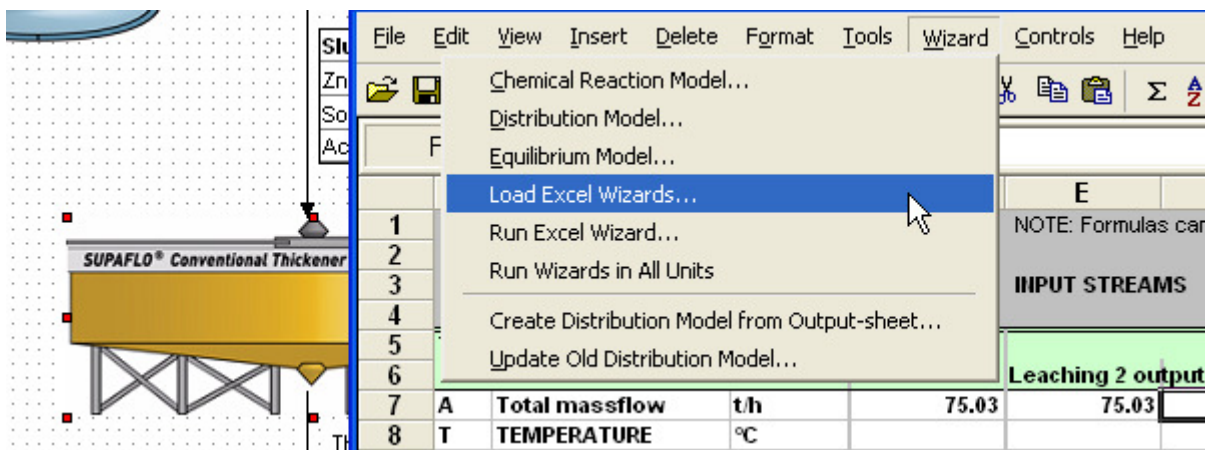


Figure 1. Selection of Excel wizards in a unit.

In next window HSC Sim show all units and the wizards that are already loaded to the units. If wizards are not loaded to the unit there is help text instead of wizard name. You need to double click the Wizard name cell, see Figure 2. You can load more than one wizard this way and then apply all wizards at the same time. To apply the wizards you need to select the wizard from the Apply column and the press Apply Selected button. Defining the wizard will automatically also select the wizard.

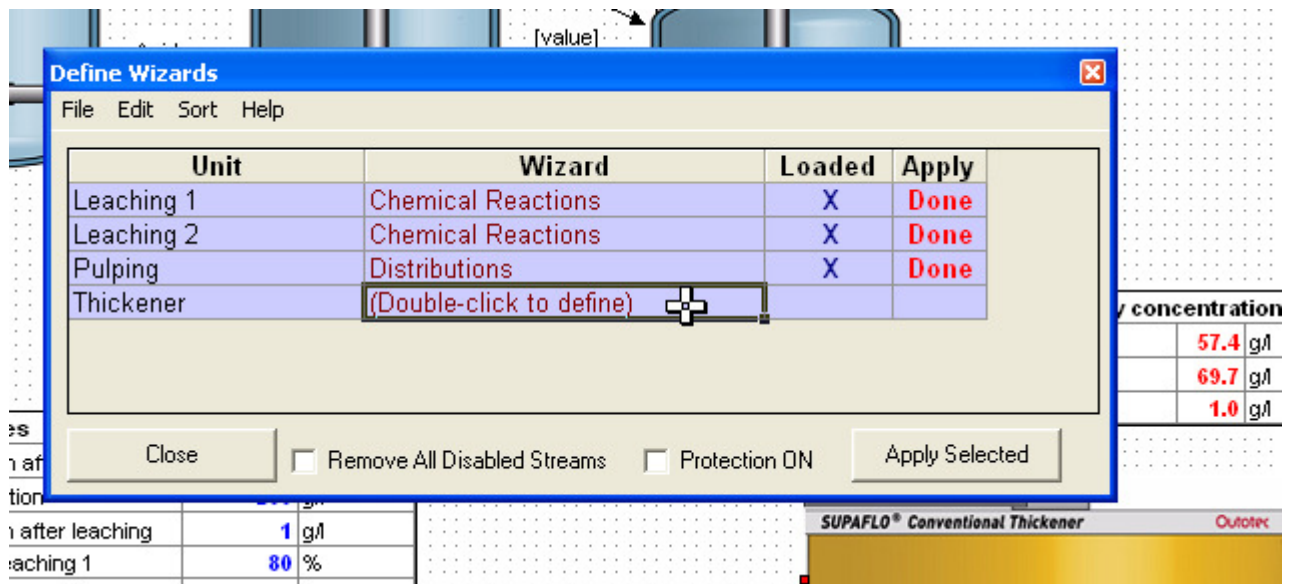


Figure 2. Defining the Wizards.

Double click loads the wizard navigator and shows all the folders in the Wizard folder on the left side, below that the files of the selected folder and then on the right side all the sheets of the selected file are shown.

Select the appropriate folder, file and sheet and click OK. The content of the wizard will be copied onto the wizard sheet of the unit. If the sheet is missing, HSC Sim will create one.

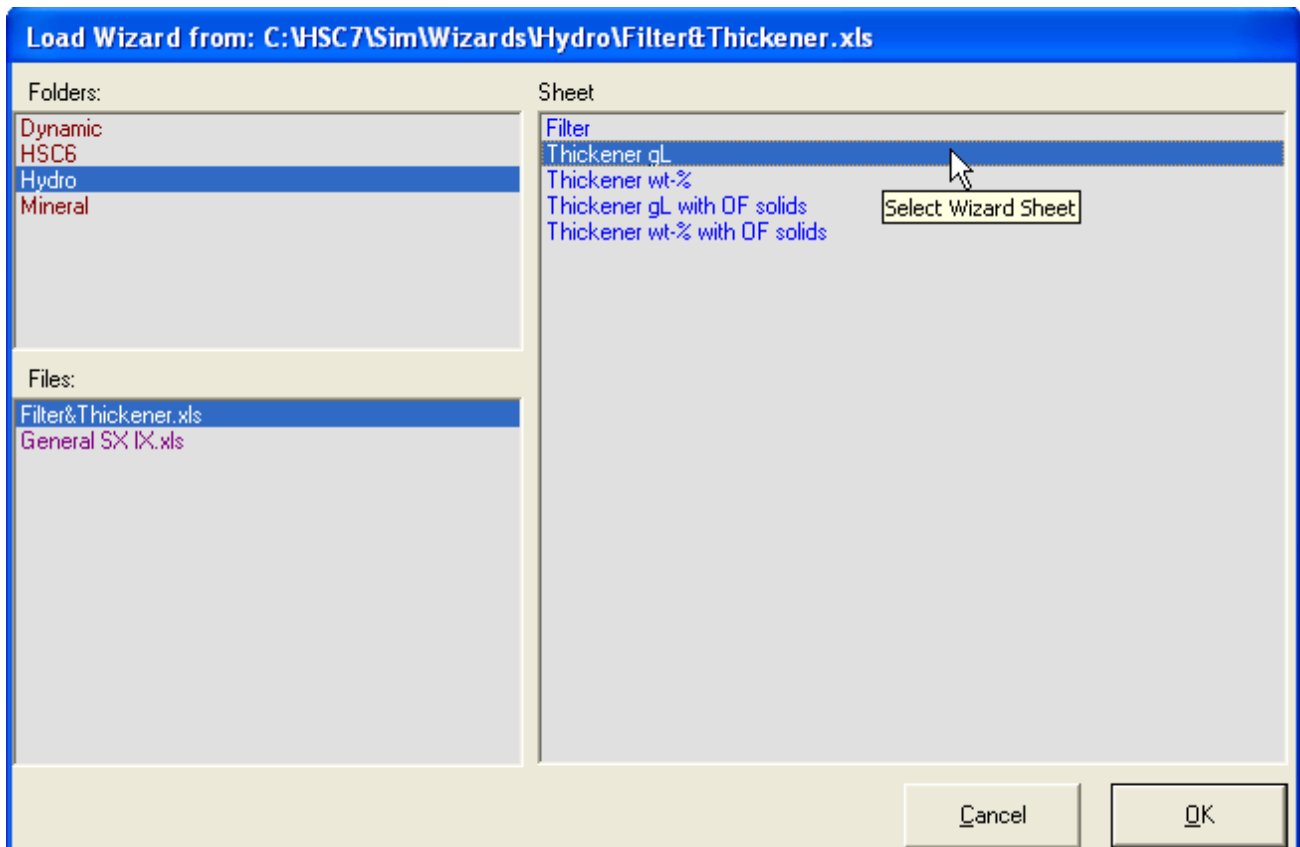


Figure 3. Selecting a wizard (Folders – Files – Sheets).

Now the wizard is loaded and previous window, Figure 2, will be shown. You can see the loaded wizard name in name column and the wizard is selected for apply. Now press Apply Selected button to run wizard.

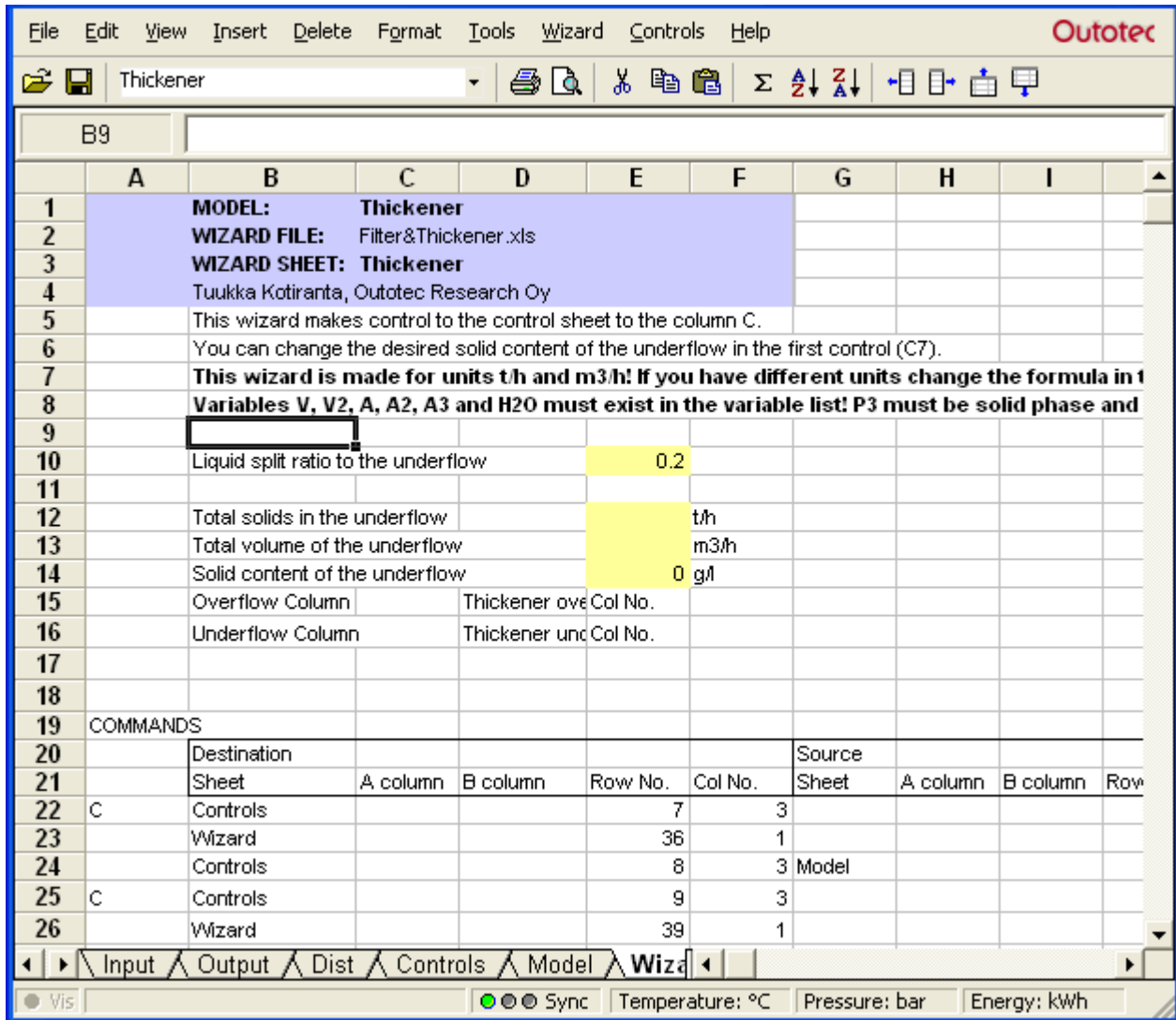


Figure 4. Wizard opened in a unit. Wizard creates a wizard –page and the content is copied onto it. The upper part is the model reference part and the lower part consists of command lines.

You can rerun the wizard at any time. The original command lines remain in the “Wizard” sheet of the unit. Cells with blue font are not overwritten in the Model sheet when the wizard is rerun (The blue font must be also in the Wizard sheet).

How to create a wizard?

You can create a new wizard with Microsoft Excel. The easiest way is probably to open an existing one and first copy the structure of the most appropriate wizard and then make changes.

It is important that **column A** is left empty up to the line where the command lines start, because an entry in column A will separate the wizard into an upper and lower part.

- The upper part of the wizard is copied as it is onto the Model page of the unit.
 - The exception is when the wizard is rerun. Then cells with blue font are not copied.
- The lower part consists of commands which are run to make changes in the unit

Command lines

The different parts of the wizard sheet are identified in the A column. The upper part is at least rows 1-10. Any entry in the first column below row 10 marks the beginning of the lower part, i.e. the commands. The text “Sheet” should exist in the cell of the B column and the rows following it are the command lines.

Columns B-F give the destination of the command, G-K give the source of the command and L-O define the command.

B	C	D	E	F	G	H	I	J	K	L	M	N
Destination					Source								
Sheet	Acolumn	Bcolumn	Row No.	Col No.	Sheet	Acolumn	Bcolumn	Row No.	Col No.	Formula	Function	Command	Prompt
Input	*			4						=SUM(\$E\$14:\$S\$14)			
Output	*			4						=SUM(\$E\$15:\$S\$15)			
Output	*			5						=Input!\$D\$16			

Table 1. Structure of the command lines. Column A is empty on command rows.

Column	Name	Definition
A	-	If column A contains letter "C" the row is skipped (comment row).
B	Destination Sheet	The name of the sheet where a new formula or value is written. (For example: Input or Output)
C	Destination A column	The value required in the A column to identify the row where the new formula or value is written. This is usually used with the Input and Output sheets. Here, you can use * for a wild card. (For example: P1, M.P.* or *)
D	Destination B column	The value required in the B column to identify the row where the new formula or value is written. Can be used independently or with the previous rule. Here you can use * for a wild card. (For example: H2SO4, Au or *)
E	Destination Row No.	The row number on the destination sheet where the new formula or value is written. This is used independently and usually for the Model or Control sheets. If you use this, please leave columns C and D empty. (For example: 4 or 10)
F	Destination Col. No.	The destination column number. You can use either the column number or any text. With a text the program asks which input or product column will be used when the wizard is executed. Please fill prompt column if you use text. (For example: 5 or Underflow). If the text is exact match with the stream name, that stream is selected without asking from the user.
G	Source Sheet	Source sheet name. (For example: Input or Output)
H	Source A column	The value required in the A column to identify the row the value is taken from. This is usually used with the Input and Output sheets. * is used for a wild card, i.e. for any characters following. (For example: M.P.* = any lines where an entry in the A column starts with M.P.; * = any lines where the A column is not empty)
I	Source B column	The value required in the B column to identify the row the value is taken from. Can be used independently or with the previous rule. (For example: H2SO4, Fe*)
J	Source Row No.	The row number on the source sheet the value is taken from. This is used independently and usually for the Model or Control sheets. If you use this, please leave columns H and I empty. * indicates that the row referred to is the same as where the command is placed, i.e. the same as the destination row. (For example: * or 5)
K	Source Col. No.	The destination column number. You can use either the column number or any text. With texts, the program asks which input or product column will be used when the wizard is executed. Please fill prompt column if you use text. (For example: 5 or Underflow)

Use preferentially the above given ways to refer to the cells. If desired you can refer to cells by Formula (column L, as given below) but since the value is copied you have to be careful as the formula changes when it is copied to a different location. You must use \$-character to prevent the change of the formula.

If the direct cell reference or formula cannot be made with the desired procedure there are a few other ways, i.e. functions and commands that can be used as described below. If you use a formula (L column) or a command (N column) the source area must be left empty. You can combine these two with the function column (M) but each command must be in a different row and the function must be filled to the latter command row. Please look at the wizards provided with HSC-Sim if you need an example.

L	Formula	In this you can write a formula. It is copied to the destination cell without any modification if you use \$-character in the formula. The equals character is not obligatory. (For example: SUM(\$E\$14:\$S\$14) or Model!\$E\$10). You can use this column to add text to target cell. Please note that then you need to use “ ” to define it as a text (For example: “g/L” or “Water feed”).
M	Function	To aggregate cell formulas. @ refers to the current cell content and # refers to Source. E.g. @*# modifies the current content of the cell so that it will be multiplied by the cell indicated in the Source. You can use most Excel commands, e.g. Column(@) will return the number of the column defined in the Source.
N	Command	With commands you can for example copy the selected cell down and the formula is changed according to Excel copying rules. If you use this, the source references, formula and function columns must be empty. The list of available commands is presented in Table 2.
O	Prompt	Prompt for the user when the source column is asked (column K).

Table 2. Description of available commands.

Commands	Parameter (in parenthesis)	Action	Example
CopyDown(n)	Number or cell address	Copies the destination cell down n times (if missing or zero then once)	CopyDown(nMinerals)
CopyRight(n)	Number or cell address	Copies the destination cell right n times (if missing or zero then once)	CopyRight(10)
HorizontalArray(n)	Number of rows	Creates based on destination cell by copying a horizontal array of n size, equals CopyRight(n-1)	HorizontalArray(nSizeClasses)
VerticalArray(n)	Number of columns	Creates based on destination cell by copying a vertical array of n size, equals CopyDown(n-1)	
TableRC(n;m)	Number of rows; Number of columns	Creates based on destination cell by copying a table of n rows and m columns	
NMinerals	-	Gives number of minerals	
NSizeClasses	-	Gives number of size classes	
nSizeFractions	-	Gives number of size classes	
FirstParticleRow	-	Gives row number where is first particle in variable list (M.P.1 in A column)	
NPhases	-	Gives number of phases	
nSpecies(n)	Number of the phase	Gives the number of species in the phase	nSpecies(2)
nOutputStreams	-	Gives number of output streams	
nInputStreams	-	Gives number of input streams	
DefineName(n)	Name of defined range	Defines a name for a range. Top left corner needs to be defined with Source columns (G-K) and bottom right corner with Destination columns (B-F). If only one cell is named, then only Destination column are filled.	DefineName(MineralGrades)
FormulaArray	-	Makes an array to area defined with Source and Destination columns in a same way as DefineName(n). The content of the array needs to be defined in Function column (M).	<i>Destination columns:</i> bottom right corner of array area. <i>Source columns:</i> top left corner of array area. <i>Function column:</i> array formula <i>Command column:</i> FormulaArray
IfExit	-	If source = Formula then exit wizard	