SIEMENS

Learn-/Training Document

Siemens Automation Cooperates with Education (SCE) | As of Version V9 SP1

PA Module P01-07

SIMATIC PCS 7 – Importing plant design data

siemens.com/sce

SIEMENS

Global Industry Partner of WorldSkills International

worldskills

Matching SCE Trainer Packages for this Learn-/Training Document

- **SIMATIC PCS 7 Software Package V9.0 (set of 3)**
  Order No.: 6ES7650-0XX58-0YS5
- **SIMATIC PCS 7 Software Package V9.0 (set of 6)**
  Order No.: 6ES7650-0XX58-2YS5
- **SIMATIC PCS 7 Software Upgrade Packages (set of 3)**
  Order No.: 6ES7650-0XX58-0YE5 (V8.x→ V9.0)
- **SIMIT Simulation Platform with Dongle V10**
  (contains SIMIT S & CTE, FLOWNET, CONTEC libraries) – 2500 simulation tags
  Order No.: 6DL8913-0AK00-0AS5
- **Upgrade SIMIT Simulation Platform V10**
  (contains SIMIT S & CTE, FLOWNET, CONTEC libraries) from V8.x/V9.x
  Order No.: 6DL8913-0AK00-0AS6
- **Demo Version SIMIT Simulation Platform V10**
  Download
- **SIMATIC PCS 7 AS RTX Box (PROFIBUS) only in combination with ET 200M for RTX –**
  Order No.: 6ES7654-0UE23-0XS1
- **ET 200M for RTX Box (PROFIBUS) only in combination with PCS 7 AS RTX Box –**
  Order No.: 6ES7153-2BA10-4AB1

Note that these trainer packages are replaced with successor packages when necessary.
An overview of the currently available SCE packages is available at: siemens.com/sce/tp

Continued training

For regional Siemens SCE continued training, get in touch with your regional SCE contact
siemens.com/sce/contact

Additional information regarding SCE

siemens.com/sce

Information regarding use

The SCE Learn-/Training Document for the integrated automation solution Totally Integrated Automation (TIA) was prepared for the program "Siemens Automation Cooperates with Education (SCE)" specifically for training purposes for public educational facilities and R&D institutions. Siemens does not guarantee the contents.

This document is to be used only for initial training on Siemens products/systems, which means it can be copied in whole or part and given to those being trained for use within the scope of their training. Circulation or copying this Learn-/Training Document and sharing its content is permitted within public training and advanced training facilities for training purposes.

Exceptions require written consent from the Siemens. Send all related requests to
scesupportfinder.i-ia@siemens.com.

Offenders will be held liable. All rights including translation are reserved, particularly if a patent is granted or a utility model or design is registered.

Use for industrial customer courses is explicitly not permitted. We do not consent to commercial use of the Learn-/Training Document.

We wish to thank the TU Dresden, particularly Prof. Dr.-Ing. Leon Urbas and the Michael Dziallas Engineering Corporation and all other involved persons for their support during the preparation of this Learn-/Training Document.

Table of contents

Importing plant design data

1 Goal

Students learn to identify recurring structures and to design templates. They know the difference between a process tag type and a model. They can create and implement both. This allows the students to implement many similar process tag types or units in ***PCS 7***. They become familiar with the process object view and can use it to display parameters system-wide and change them if needed.

2 Prerequisite

This chapter builds on chapter 'Control loop and other control functions'. To implement this chapter, you can use an existing project from the previous chapter or the archived project 'p01-06-exercise-r1905-en.zip' provided by SCE. The download of the project(s) is stored on the SCE Internet for the respective module.

The (optional) simulation for the SIMIT program can be retrieved from the file 'p01-04-plantsim-v10-r1905-en.simarc'. It can be run in Demo mode.

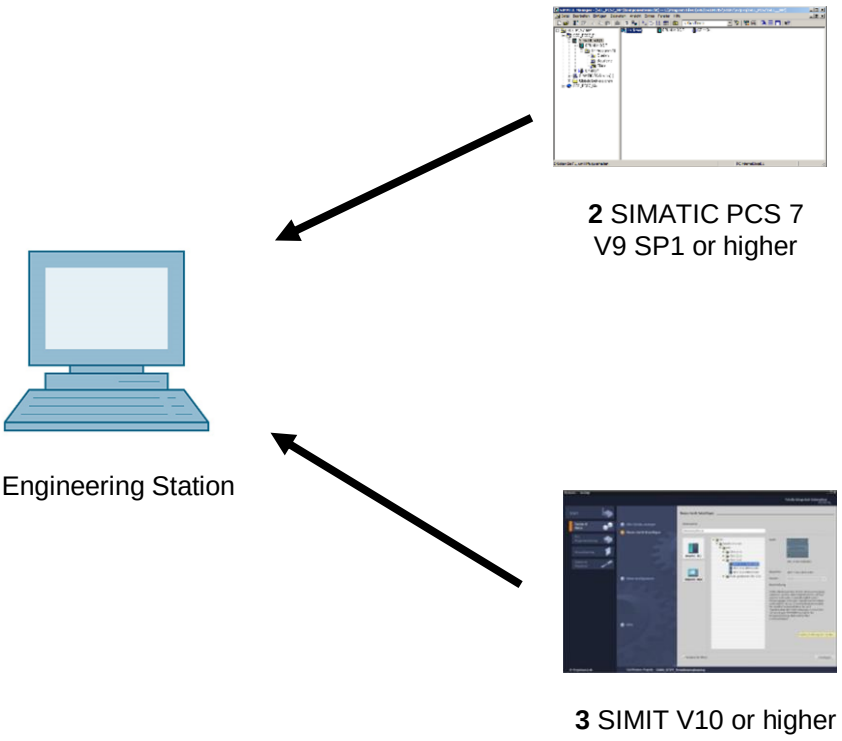
3 Required hardware and software

1 Engineering station: Requirements include hardware and operating system (for further information, see Readme on the PCS 7 installation DVD)

2 SIMATIC PCS 7 software V9 SP1 or higher

- Installed program packages (contained in SIMATIC PCS 7 Software Trainer Package):
  - *Engineering → PCS 7 Engineering*
  - *Engineering → BATCH Engineering*
  - *Runtime → Single Station → OS Single Station*
  - *Runtime → Single Station → BATCH Single Station*
  - *Options → SIMATIC Logon*
  - *Options → S7-PLCSIM V5.4 SP8*

3 Demo Version SIMIT Simulation Platform V10



4 Theory

4.1 Theory in brief

In a process plant, you will find recurring objects and structures that have identical behavior, are integrated identically in the control technology and are to be represented identically in the visualization.

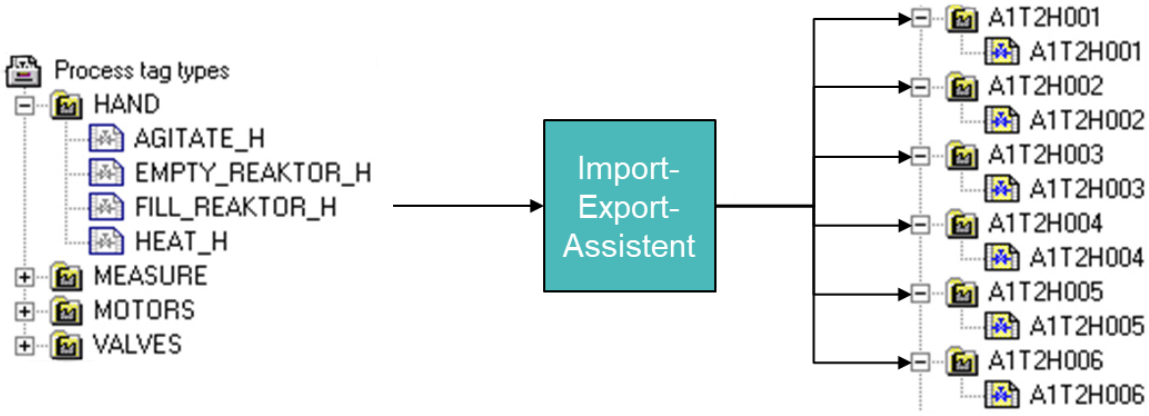


Figure 1: From the process tag type to replicas

Such an object can be stored in the project library as a process tag type. A process tag type is a single CFC. As shown in Figure 1, it is possible to use the Import/Export Assistant to create a large number of process tags as a copy of a process tag type in a single operation. This process is controlled by an import file. The process tags then have to be further manually adapted and interconnected according to the specific automation tasks.

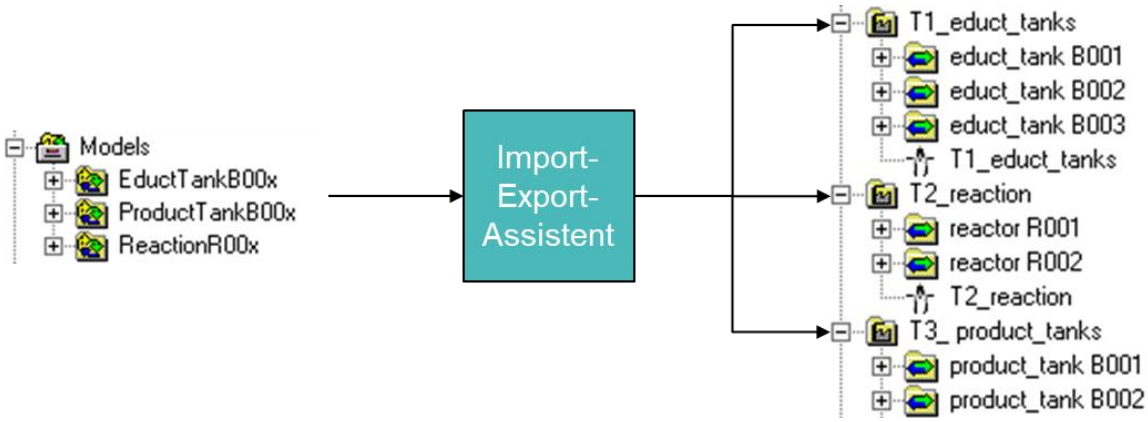


Figure 2: From the model to replicas

With ***models*** you define more complex functions than with process tag types (up to complete units). A model consists of hierarchy folders with CFC/SFCs, pictures, reports and additional documents. The entire structure can be stored in the project library as a reusable template. It is possible to use the Import/Export Assistant to create a large number of replicas based on an import file as a copy of a model in a single operation (see Figure 2). The replicas are then adapted to the specific requirements of the respective automation task.

The ***PCS 7*** libraries contain extensive ***templates***. If a template is to be used multiple times, it is copied from the ***PCS 7*** library to the project library, adapted as required and copied using the Import/Export Assistant based on an import file.

4.2 Project structuring

When designing an automation system with PCS 7, well-proven general design principles for complex systems can be used [1]. The three most important principles are:

– Principle of hierarchical structuring

– Principle of modularization

– Principle of reuse

The principle of hierarchical structuring was already used in the structuring of the plant in chapter 'Plant hierarchy'. Through structuring into subsystems that can be processed largely independent of each other, a design problem that initially seemed unmanageable is broken down into manageable and plannable sub-tasks.

According to the principle of modularization, a system to be designed is to be made up of components (here: blocks, CFCs, SFCs) that have the following characteristics:

– The scope is manageable and easy to understand

– Largely autonomous functions that can be separately checked

– As few references to other components as possible

– Defined interfaces to other components

This results in two rivaling complexity aspects when an automation solution is broken down into individual components:

– Low inner complexity of components: The more components, the smaller and more manageable the individual components.

– High exterior complexity of components: The more components, the greater the number of connections between the components.

Hierarchical structuring and modularization are inter-dependent. While hierarchical structuring is determined more by the process plant, modularization is dominated by the process control implementation. Due to the opposing complexity aspects mentioned above and the high dependency on the specific process and automation problem at hand, early coordination of both disciplines is advantageous.

Through the plant hierarchy, ***PCS 7*** supports the principle of hierarchical structuring. The principle of modularization and reuse is realized in ***PCS 7*** in the import of plant design data.

In larger projects or in the case of recurring similar projects, a large number of identical or at least very similar objects and structures must often be monitored. To save configuration time and expense, it is recommended that a targeted search for suitable, recurring objects and structures be included in the concept development and design phase of an automation project. Once such objects and structures are identified, the next step is to test and implement generic solutions. It is then possible to use these generic solutions for a large number of identical or similar objects and structures. The additional effort that the preparation of the generic solution (here also called a type or template) entails should lead to considerable time and cost savings over the life of the project due to the following factors:

– A type can be implemented multiple times, which means it has multiple replicas.
– By using a type in more than one replica, multiple tests are performed at the same time.
– If errors occur or changes become necessary, the user only has to adapt the generic solution and update all replicas.

Moreover, objects and structures that are available from earlier projects and libraries can be reused. Their advantage is that they have been tried and are largely free of errors. Through the reuse of proven components, the reliability of a new automation solution also generally increases.

4.3 Process tag type

The process tag type is used as a generic solution when a project contains many similar process tags [2].

First, a CFC is prepared that contains all internal blocks and their interconnections. All input and output parameters are defined clearly as parameters or signals. This CFC with all generally valid parameters is used to generate a process tag type. The process tag-specific parameters in which the replicas differ are specified in an "import file".

During the import, the Import/Export Assistant generates the replicas of the process tag type in the specified hierarchy folders. If the hierarchy does not yet exist, it is created as well. Each replica is an instance of the process tag type and has its properties.

In ***PCS 7***, the process tags (replicas) generated in this way can be specifically adapted by adding, for example, various interlocking mechanisms. Under certain preconditions, these are not overwritten even in the case of a new import.

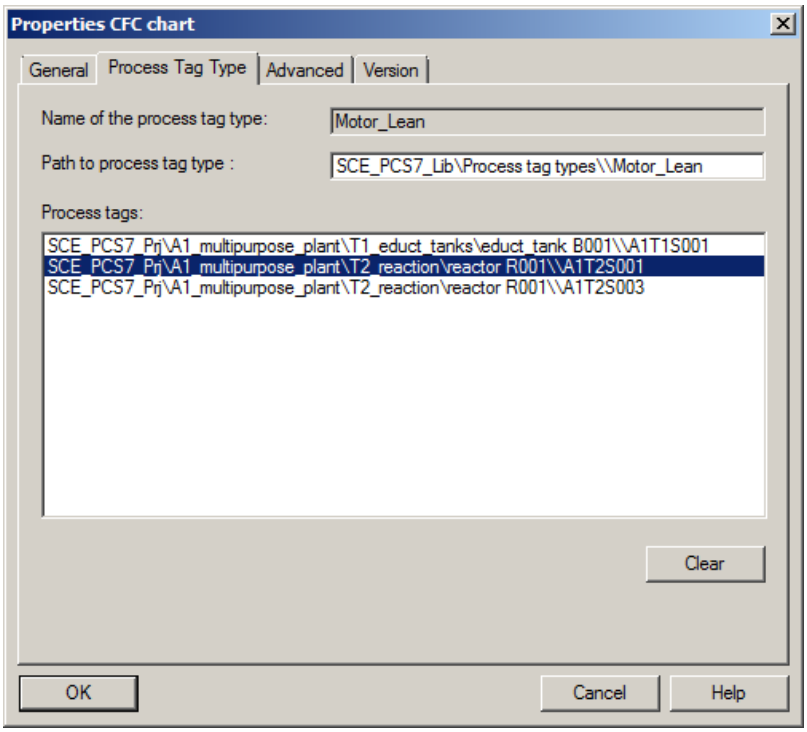


Figure 3: Replica A1T2H003 of FILL_REACTOR_H

The following must not be changed when process tags are generated:

– Specific adaptations to the block inputs/outputs for which parameters are assigned via the import file. When a new import takes place, these adaptations will be overwritten with the parameters defined in the import file.

– Changes to the block names

Subsequent changes can be made easily with process tag types by implementing the changes in the process tag type and the import file. The modified data is then transferred to all generated process tags by performing another import. Possible changes include:

– Addition of a parameter and assignment of this parameter via the import file.

– Deletion of all generated process tags of a process tag type (without manual deletion in the plant hierarchy)

– Addition of a block I/O and parameter assignment of the block I/O via the import file.

4.4 Model

The model is used as a generic solution when similar structures occur in the project.

As a rule, a plant is structured by breaking it down into smaller functional units whose interfaces, behavior and logic can be clearly described; for example, a tank with its instrumentation. Instead of implementing new functional units each time, there is the option to create a set of pre-assembled functional units (models).

So that a single version of a model can be used project wide, all models should be stored centrally in the master data library and adapted prior to generating replicas.

A model consists of hierarchy folders that contain the following elements:

– CFC/SFCs

– OS pictures

– OS reports

– Additional documents

Once a model has been configured and assigned an import file, replicas can be generated by means of an import. The following steps are executed automatically:

Step 1: The hierarchy path is read from the 'Hierarchy' column of the first row of data in the import file. A check is made to determine whether this path already exists. Additional actions depend on the result of the check:

– If the hierarchy folder exists and is already a replica of the model, the parameter settings from the import file are applied to the existing replica.

– If the hierarchy folder exists and is suitable as a replica of the model, it is made into a replica of the model with its CFC and assigned parameters based on the import file.

– If a hierarchy folder does not exist, it is created and a replica of the model is created and assigned parameters as appropriate.

Step 2: The following elements are inserted into the footer of the charts, if the columns are present:

– Function identifier (FID)

– Location identifier (LID)

– CFC name

– Chart comment

Step 3: Texts and values of the parameter descriptions and the interconnection descriptions (signals) are written to the corresponding block or chart I/Os of the replicas.

Notes:

– *An interconnection is deleted when the signal name (symbol or textual interconnection) consists of the code word "---" (three dashes).*

– *An interconnection remains unchanged, if no interconnection name (symbol or textual interconnection) is specified.*

Step 4: The data types of the I/Os for signals are determined and assigned to the interconnections.

Note:

– *The following applies to interconnections with shared addresses: If the 'Include signal in the symbol table' option is set, the names can be found in the symbol table of the resource of the model.*

For ***PCS 7*** it is recommended that this option not be used because these entries are made in ***HW Config*** when the hardware is configured.

Observe the following rules:

– The symbol name is present in the symbol table:

  The data type must be identical, and the symbol name must be unique. The data type parameters are assigned based on the block/chart I/O. The absolute address is overwritten, and the symbol comment is entered for the symbol (if present in the import file). Only the information that has changed will be overwritten, existing attributes are retained.

– The symbol name is not present in the symbol table:

  The interconnection is created, and the data type parameters are assigned based on the I/O. The absolute address and the symbol comment are entered for the symbol (if present in the import file).

Step 5: The message text is imported for each message.

Then, steps 1 to 5 are repeated for each row of the import file.

If a hierarchy folder containing several models was selected, the list contains the import files, each with the model. This list can be edited if required. Finally, the import operation, as described above, is carried out for all models in the list.

4.5 Parameters and signals

In order for process tag types and models to be successfully created, it is important to define all inputs and outputs of a CFC as a parameter or signal. Only I/Os that are defined as a parameter or signal can be included as a column in the import file and assigned parameters.

4.6 Process object view

In the process object view, all basic automation data project-wide can be displayed in a process control-oriented view. Project-wide means that, in a multiproject, the data of all the projects contained in it is collected.

The structure of the process object view is similar to that of the plant view:

– The left section of the window displays the plant hierarchy as a tree structure (hierarchy window). The possible operations offered here are identical. The hierarchy window also displays the CFCs, SFCs, pictures, reports and additional documents.

– The right section displays a table of the lower-level objects with their attributes (content window). The content window has the tabs shown in Table 1 and offers different views of the project data.

Tab	Use
General	Display of all lower-level process objects (process tags, CFCs, SFCs, OS pictures, OS reports or additional documents) with their general information for the plant unit currently selected in the tree view.
Blocks	Display of the block properties of the blocks of all lower-level CFCs for the plant unit currently selected in the tree view. In this context, SFC instances are also referred to as blocks.
Parameters	Display of the I/O points that were explicitly selected for editing in the process object view (S7_edit = para) for all the process tags and CFCs displayed in the "General" tab.
Signals	Display of the I/O points that were explicitly selected for editing in the process object view (S7_edit = signal) for all the process tags and CFCs displayed in the 'General' tab.
Messages	Display of the associated messages for all the process tags, CFCs and SFCs displayed in the 'General' tab.
Picture objects	Display of the picture interconnections present in WinCC (if required) for all the process tags and CFCs displayed in the 'General' tab.

Tab	Use
Archive tags	Display of the existing interconnected WinCC archive tags with their attributes for all the process tags, CFCs and SFCs displayed in the 'General' tab. Only those attributes that are relevant for PCS 7 (subset of all attributes defined in Tag Logging).
Hierarchy folder	Display of the hierarchy folders of the plant unit selected in the tree view (one line per hierarchy folder).
Equipment properties	Here, the equipment properties are displayed for the project selected in the tree view. These equipment properties are instances of equipment property types that have been configured in the shared declarations (one line for each equipment property). The attributes are applied to the instance when a type is changed.
Shared declarations	Here, you can edit the attributes of the types, enumerations, units of measure and equipment properties contained in the multiproject.

Table 1: Tabs of the process object view

4.7 References

[1] Lauber, R. and Göhner, P. (1999): Prozessautomatisierung 2., Springer Verlag

[2] PCS 7 online help for the plant hierarchy (PH). Siemens. (→ Open multiproject → Plant view → Select multiproject → Press F1 → The STEP 7 Help Viewer appears with "Help on plant hierarchy (PH)")

5 Task

PCS 7 is a software that provides users with many tools for efficient programming of large plants and copying of program sections.

In this task, charts and hierarchy structures will be created as library objects. There is then the option to use them multiple times. The Import/Export Assistant and the process object view are aids for this.

The chart of the 'A1T2X001' valve is to be used here as a process tag template. All other inflow valves for the reactors are to be created using this process tag.

For the model, you take educt tank B001 and generate all other educt tanks from it.

6 Planning

The level sensors in educt tank B001 are used in the same way in educt tanks B002 and B003. The same applies to the valves and pumps located between the educt tank and reactor.

A process tag type will be created based on valve A1T2X001 and then copied for all similar valves (A1T2X002 to A1T2X006).

The following symbols and parameters are relevant for this:

Block	I/O	Type
FbkOpen	PV_In	Signal
FbkClose	PV_In	Signal
Output	PV_Out	Signal
CMP_Interlock	In1	Parameter
Permit	In01	Parameter
Protect	In01	Parameter
Valve block	OpenLocal	Parameter
Valve block	CloseLocal	Parameter
Valve block	LocalLi	Parameter

In the second part, a complete structure will be created as a model based on educt tank B001 and then duplicated.

The following symbols and parameters are relevant for this:

CFC	Block	I/O	Type
A1T1L001	LSA+	PV_In	Signal
A1T1L001	LSA-	PV_In	Signal
A1T1S001	FbkRun	PV_In	Signal
A1T1S001	OutStart	PV_Out	Signal
A1T1S001	CMP_Interlock	In1	Parameter

A1T1S001	CMP_Interlock	In2	Parameter
A1T1S001	OR_Local	In1	Parameter
A1T1S001	OR_Local	In2	Parameter
A1T1S001	Pump block	LocalLi	Parameter
A1T1X004	FbkOpen	PV_In	Signal
A1T1X004	FbkClose	PV_In	Signal
A1T1X004	Output	PV_Out	Signal
A1T1X004	OR_Local	In1	Parameter
A1T1X004	OR_Local	In2	Parameter
A1T1X004	Valve block	LocalLi	Parameter

Figure 54 provides an overview of the new blocks to be created by importing plant design data.

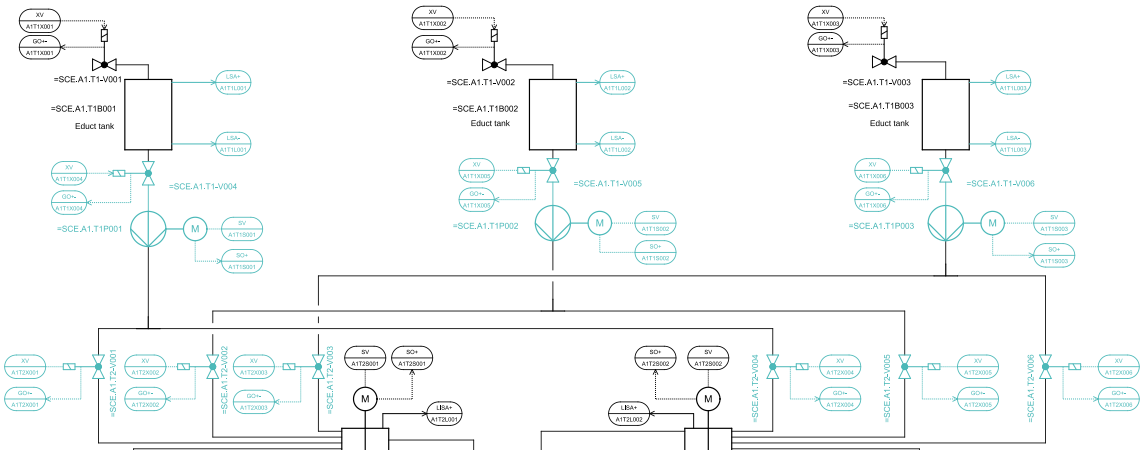


Figure 54: Portion of the P&ID flow chart to be programmed

7 Learning objective

In this chapter, students learn the following:

– Import of plant design data using the Import/Export Assistant

– Process object view

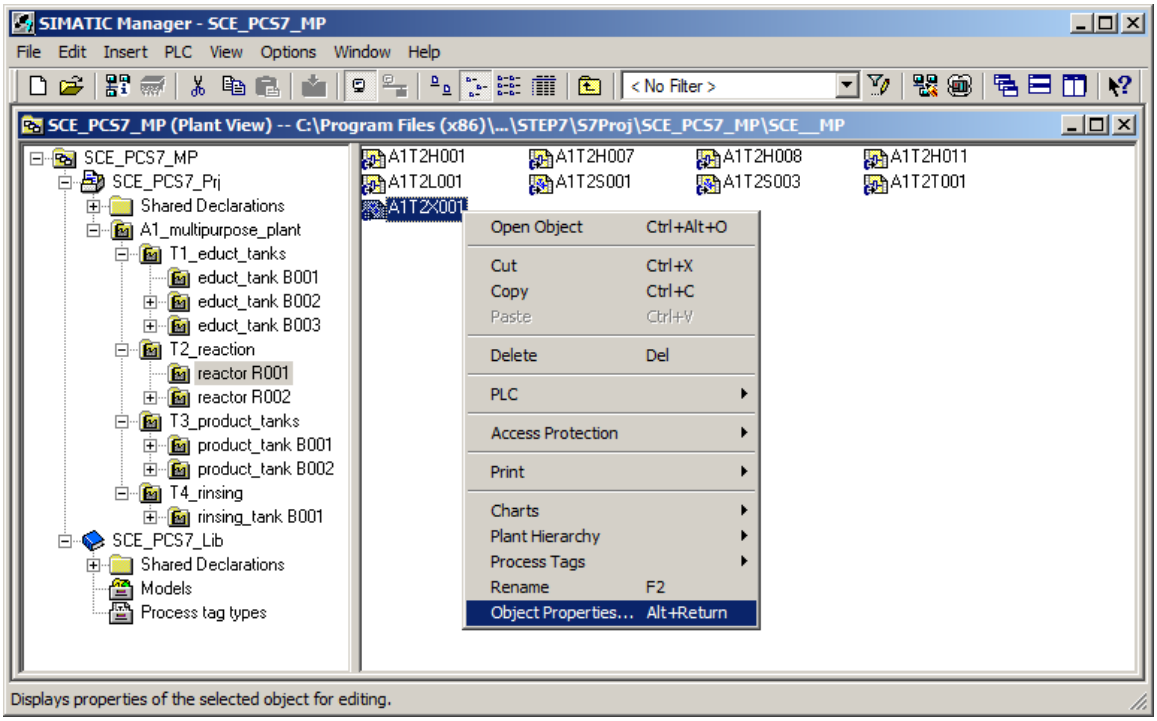
– Duplication of charts by creating process tags

– Duplication of folder structures by creating models

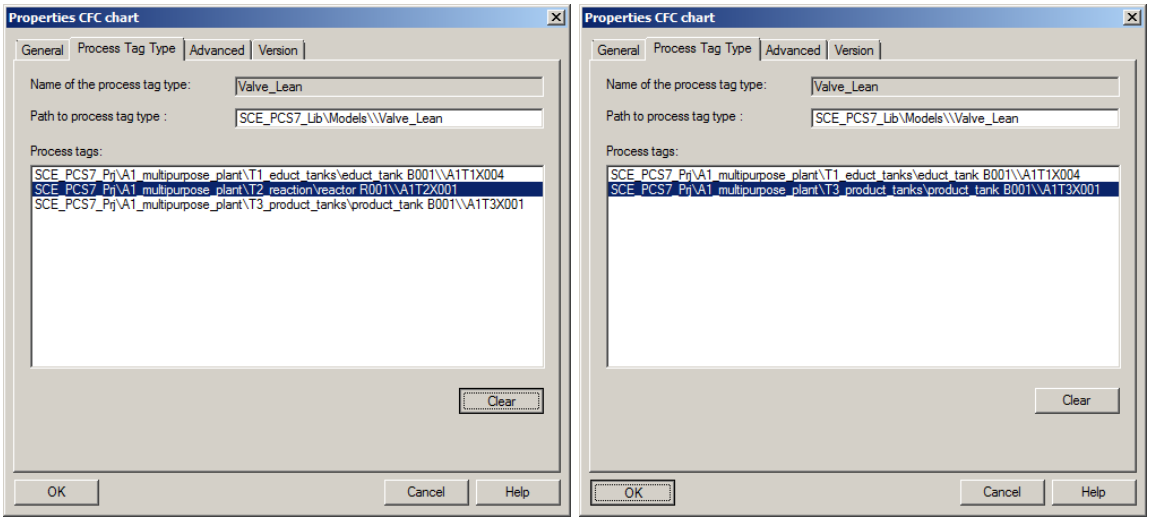
8 Structured step-by-step instructions

8.1 Creating a process tag type

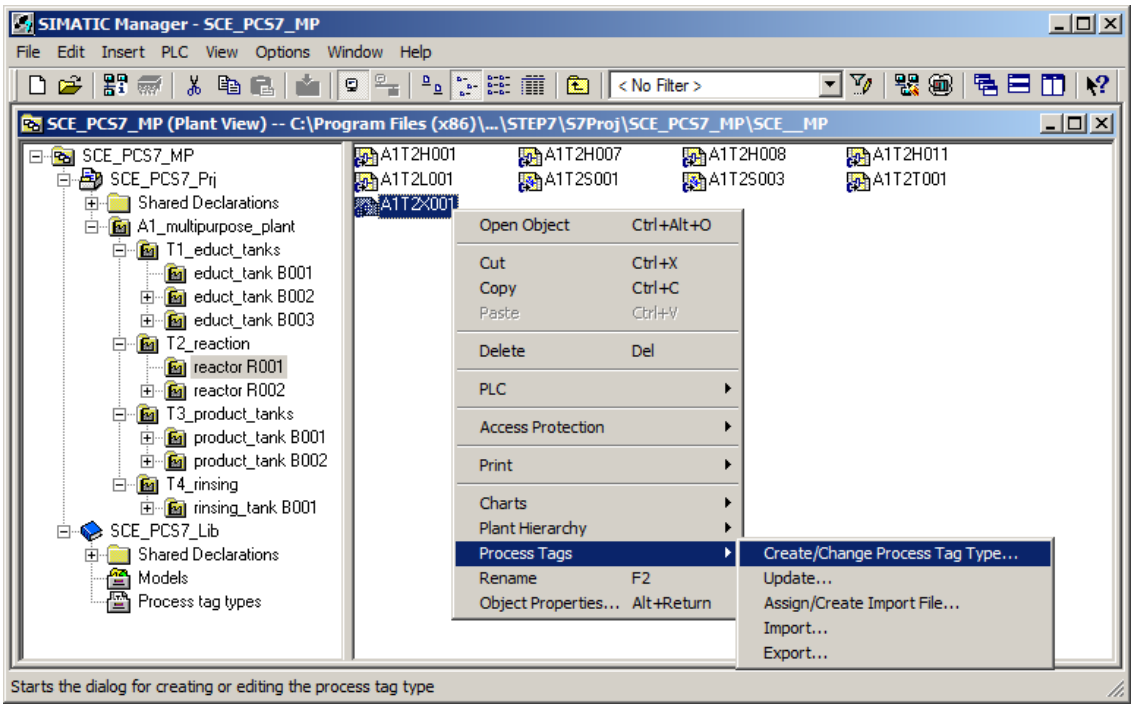
1. To duplicate a previously created and tested chart, a process tag is created from it. In this example, take the valve 'A1T2X001'. Because this chart already belongs to the 'Valve_Lean' process tag type, you must first clear the connection in the object properties. (→ A1T2X001 → Object Properties)



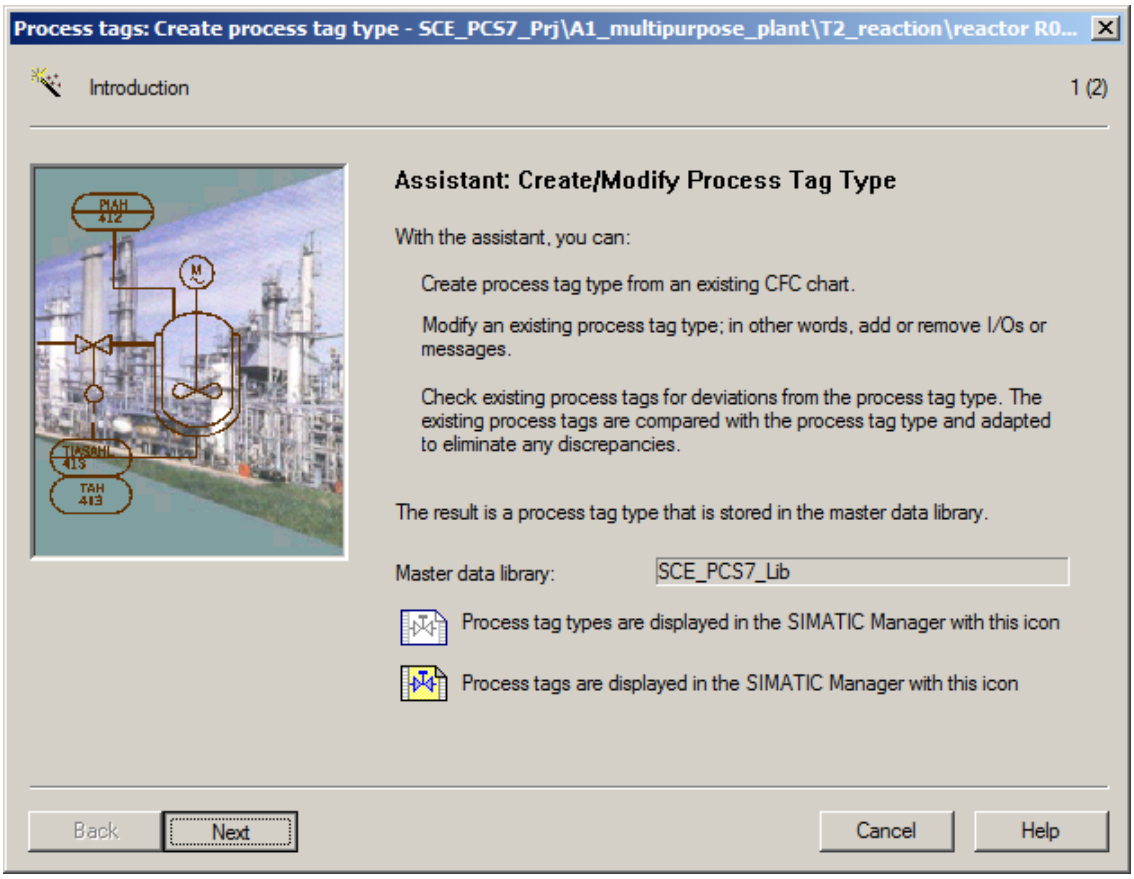
2. In the 'Process tag type' tab, select the row with the valve and then click 'Clear'. The valve is removed from the list (→ Process Tag Type → A1T2X001 → 'Clear'→ 'OK')



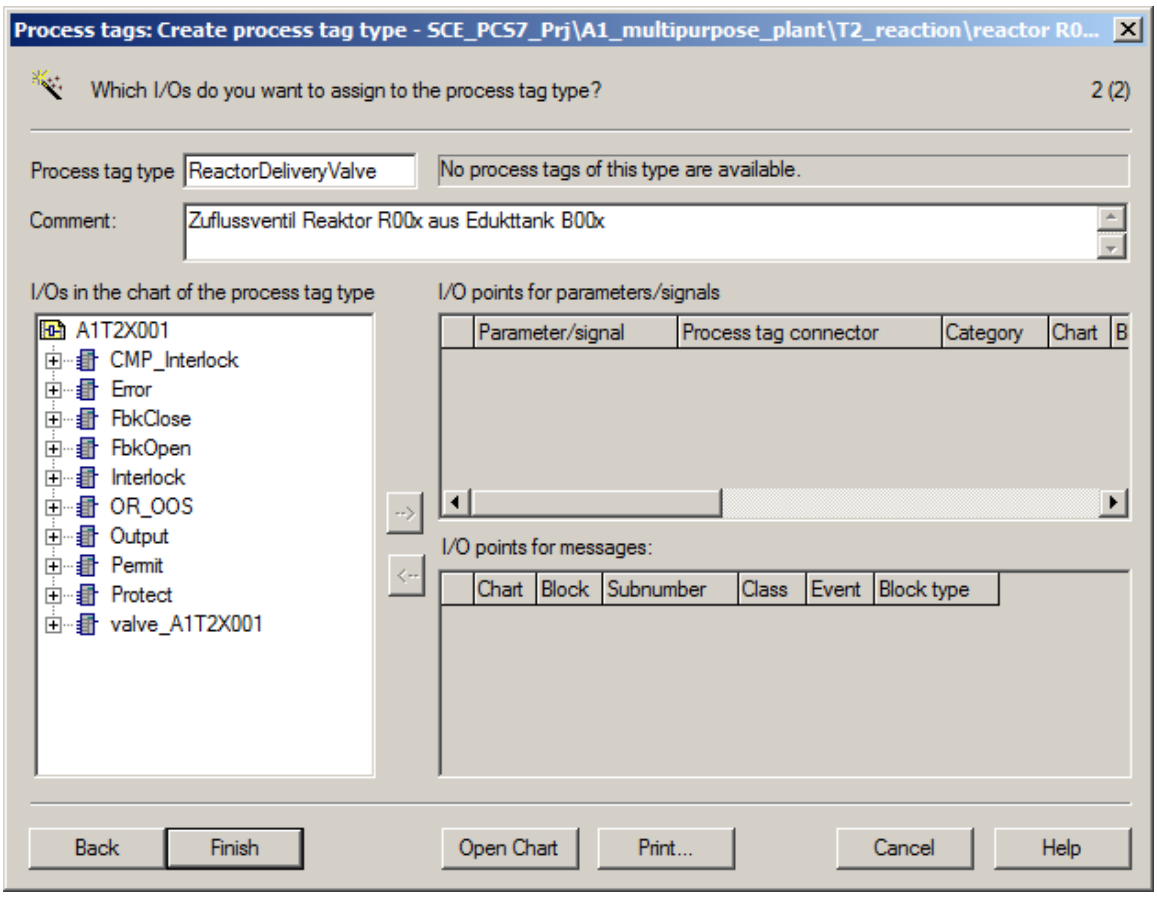
3. You can then create a process tag type from 'A1T2X001' by clicking on 'Process tags' in the shortcut menu and then on 'Create/Modify Process Tag Type…'. (→ A1T2X001 → Process Tags → Create/Modify Process Tag Type…)



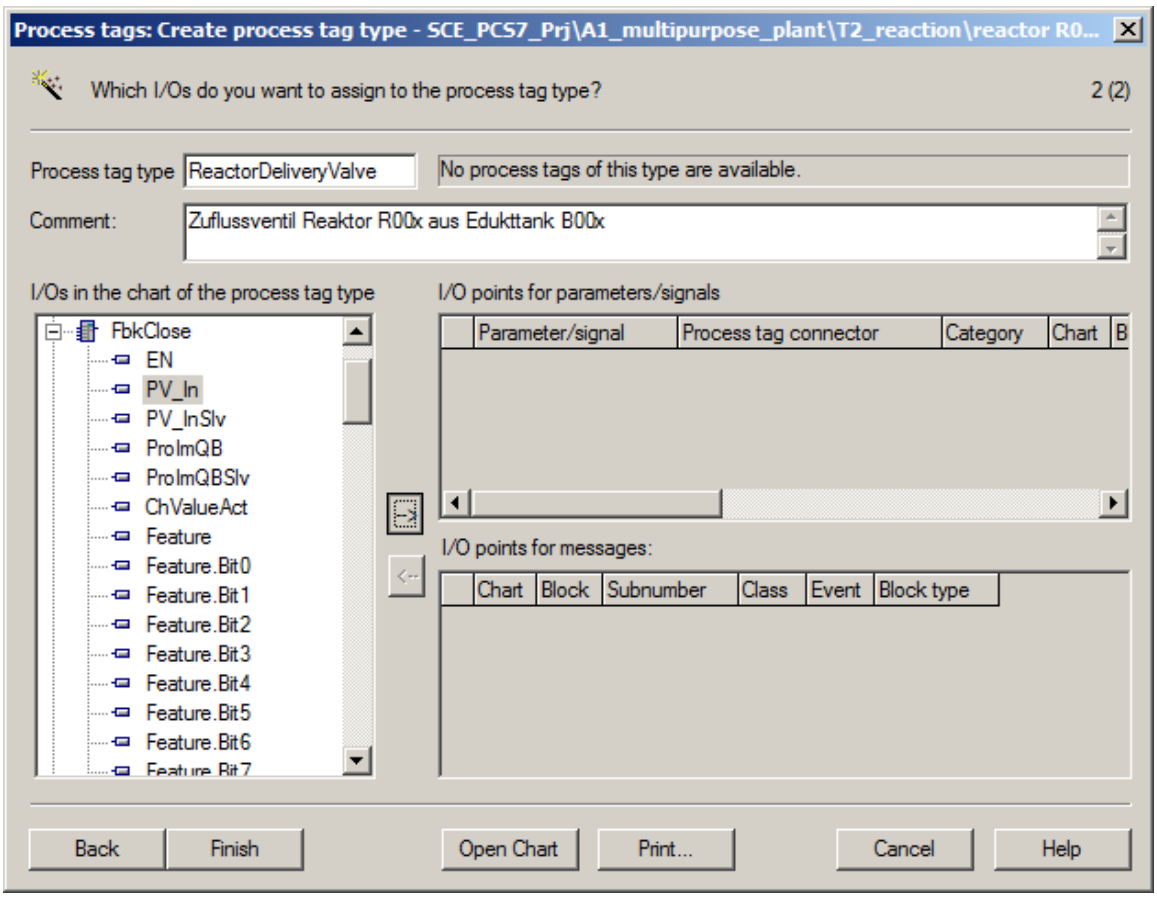
4. The Create/Modify Process Tag Type dialog opens. (→ Next)



5. First, the name of the process tag type is generalized to 'ReactorDeliveryValve' and the comment to 'Valve inlet reactor R00x from educt tank B00x'.



6. Next, the parameters and signals that are to be changed for the individual replicas of the process tag type must be selected on the left side of the window. (FbkClse_A1T2X001 → PV_In →  -)

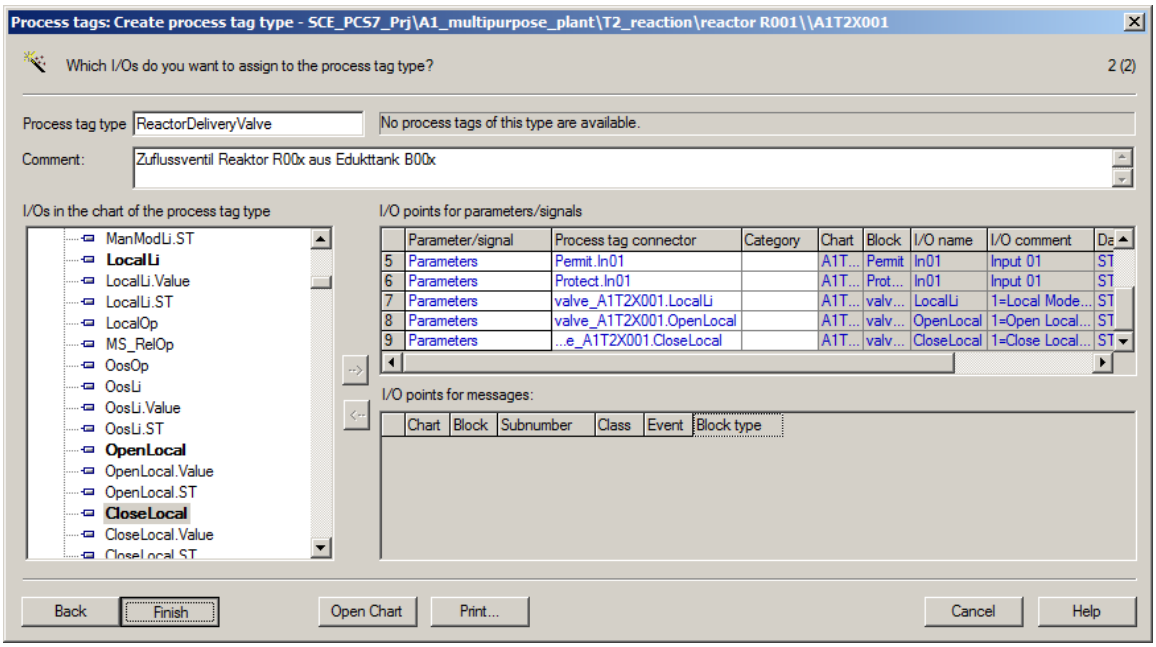


Note:

– *With the 'Open Chart' button, the associated CFC can be displayed to get a better overview. (→ Open Chart)*

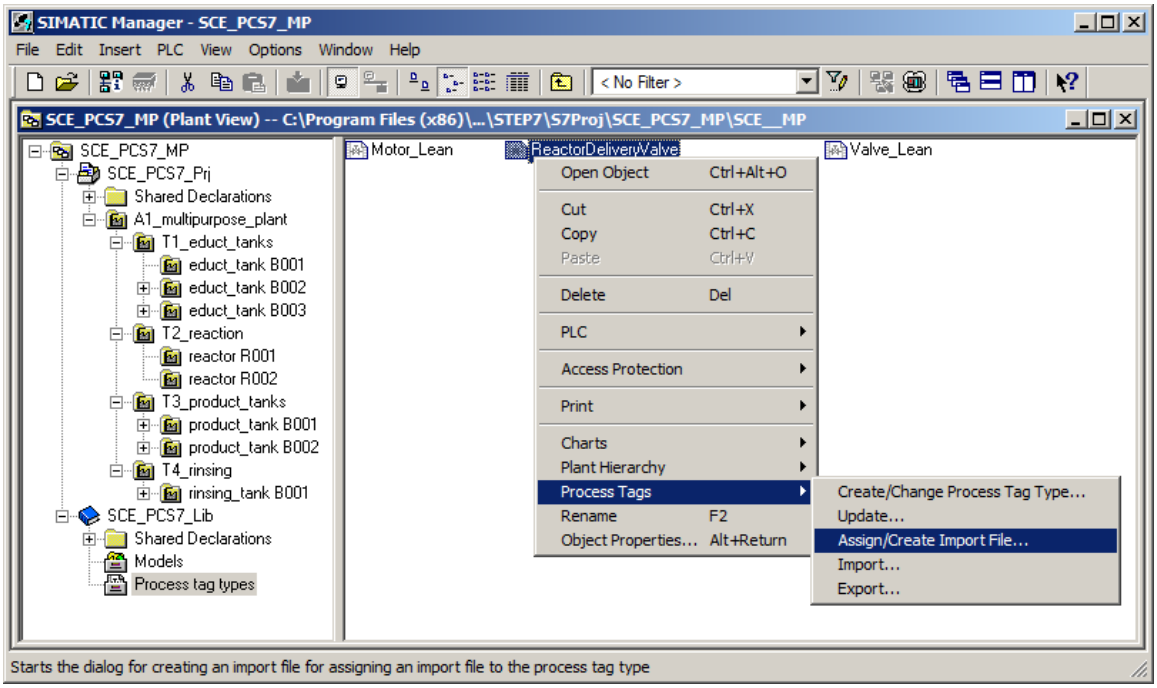
7. Now, add all signals and parameters that represent the input and output interconnections of the CFC. Signals are input and output signals and parameters are interconnections between charts. You must add the signals and parameters shown here for the inflow valves of the reactors. Then the process tag type can be finished. (→ Finish)

	Parameter/signal	**Process tag connector**	Category	Chart	Block	I/O name	I/O comment	Data type	I/O	Block type
1	Parameters	CMP_Interlock.In1		A1T2X001	CMP_Interlock	In1	Analogue Value 1	STRUCT	IN	CompAn02
2	Signal	FbkClose.PV_In		A1T2X001	FbkClose	PV_In	Input value	BOOL	IN	Pcs7DiIn
3	Signal	FbkOpen.PV_In		A1T2X001	FbkOpen	PV_In	Input value	BOOL	IN	Pcs7DiIn
4	Signal	Output.PV_Out		A1T2X001	Output	PV_Out	Output value	BOOL	OUT	Pcs7DiOu
5	Parameters	Permit.In01		A1T2X001	Permit	In01	Input 01	STRUCT	IN	Intlk02
6	Parameters	Protect.In01		A1T2X001	Protect	In01	Input 01	STRUCT	IN	Intlk02
7	Parameters	valve_A1T2X001.OpenLocal		A1T2X001	valve_A1T2X001	OpenLocal	1=Open Local:Field Open ...	STRUCT	IN	VlvL
8	Parameters	...e_A1T2X001.CloseLocal		A1T2X001	valve_A1T2X001	CloseLocal	1=Close Local: Field Close...	STRUCT	IN	VlvL
9	Parameters	valve_A1T2X001.LocalLi		A1T2X001	valve_A1T2X001	LocalLi	1=Local Mode: Local oper...	STRUCT	IN	VlvL

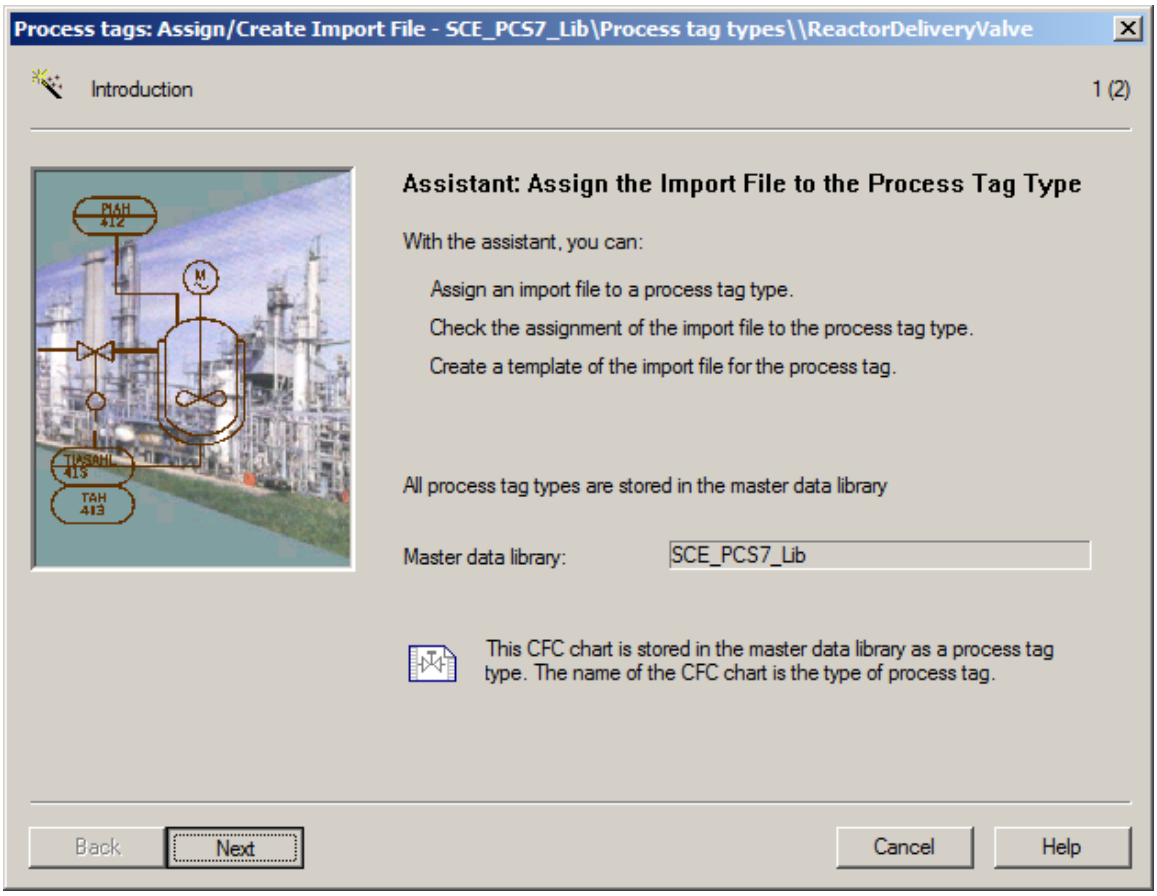


8.2 Creating an import file

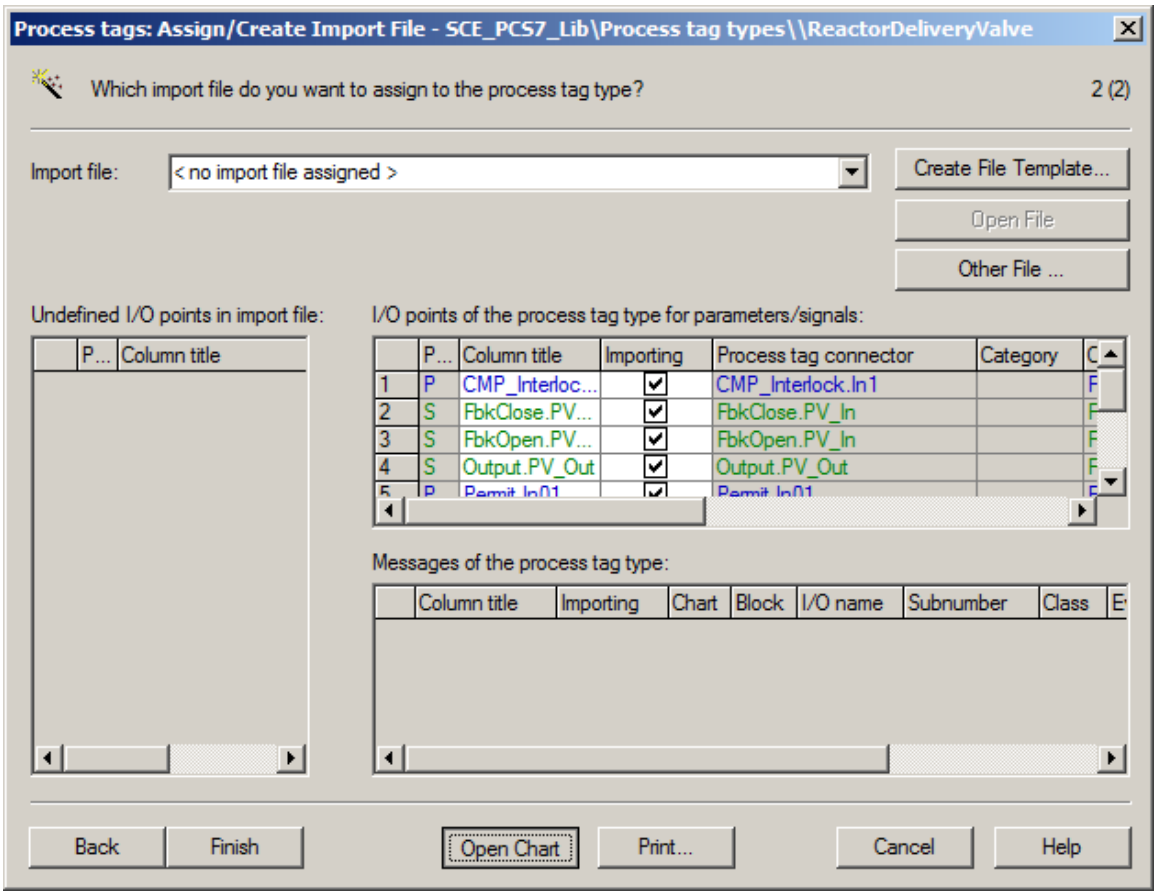
1. After the process tag type is finished, it is located in the plant view in the project library under 'Process tag types'. You must now create an import file for the process tag type just created. (→ Process tag types → ReactorDeliveryValve → Process Tags→ Assign/Create Import File)



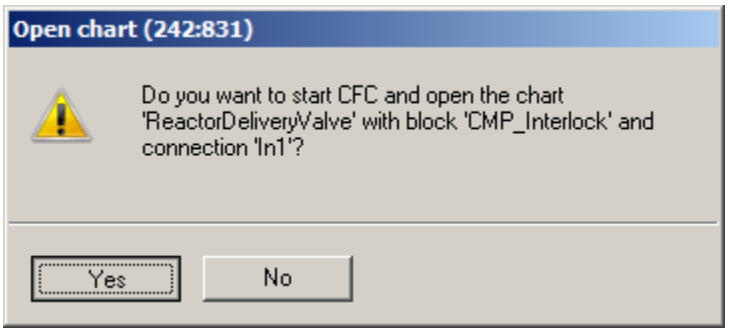
2. Confirm the first dialog with 'Next'. (→ Next)



3. First, open the chart. (→ Open Chart)



4. Confirm the following message. (→ Yes)



5. You can see that all cross-chart connections are created as textual interconnections and all input and output signals with their symbolic names. You can then close the chart again. (→ Close)

Note:

– *The textual interconnection A1H001\A1H001.PV_Out is structured as follows:*

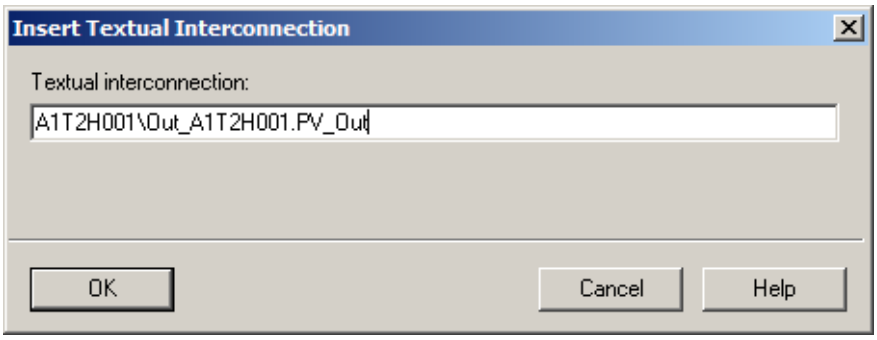
– *A1H001 Name of the CFC*

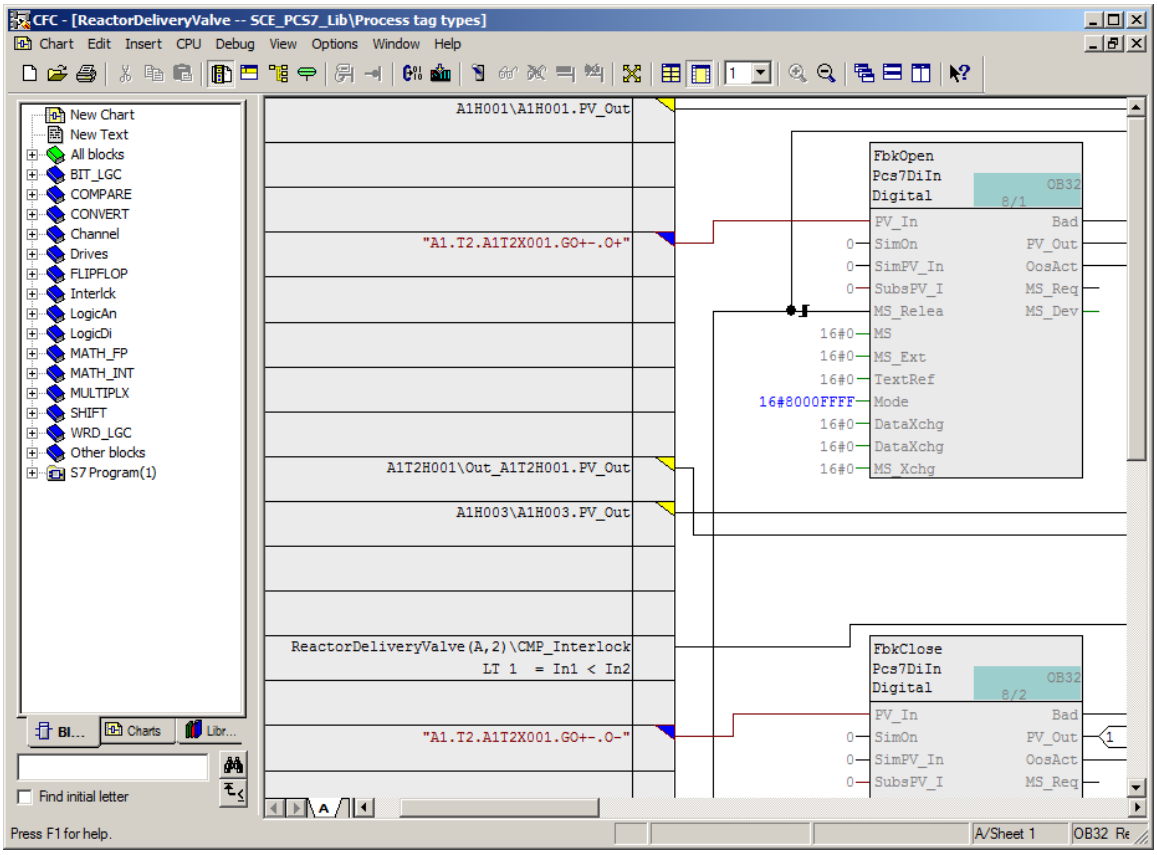
– \ *Separator*

– *A1H001 Name of the block in the CFC*

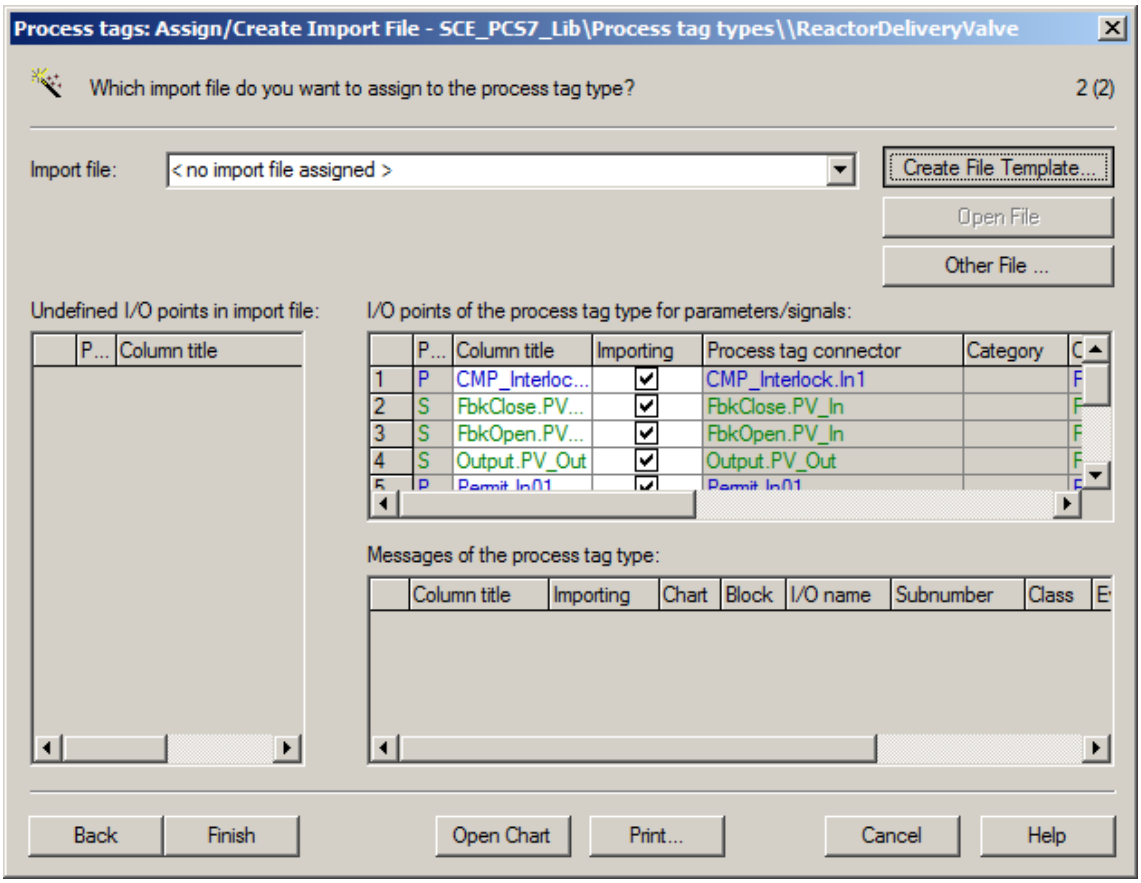
– . *Separator*

– *PV_Out I/O of the block that is to be connected*

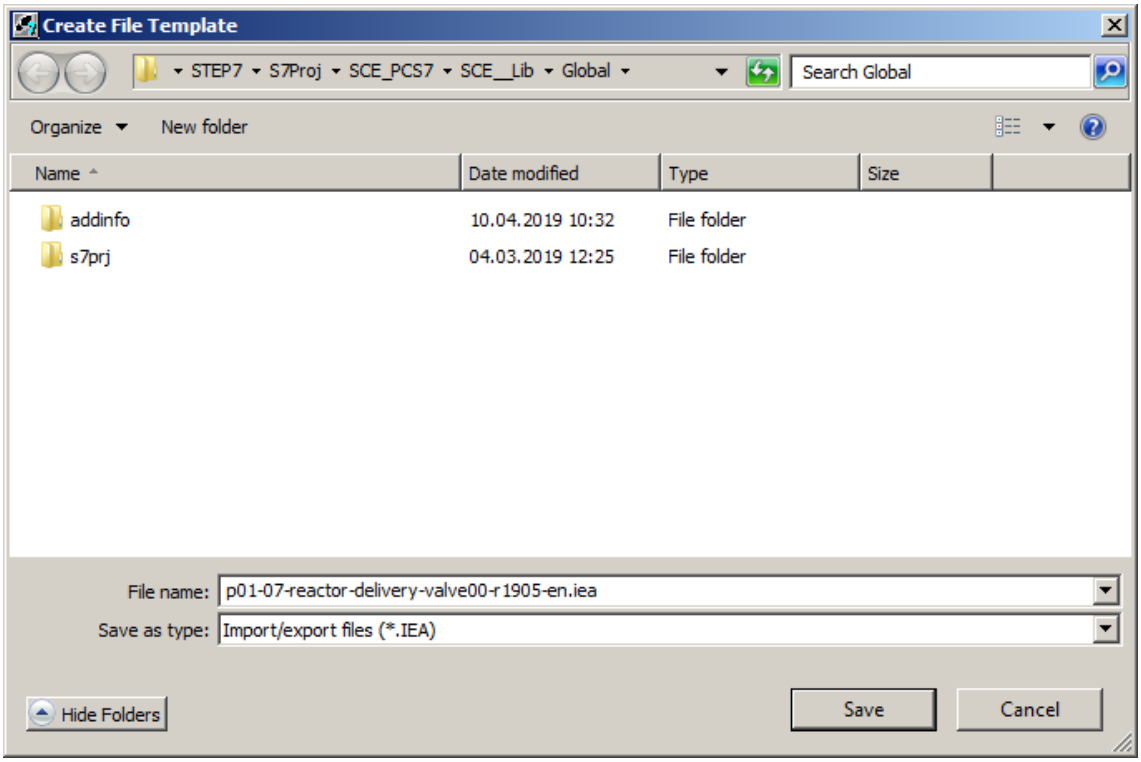




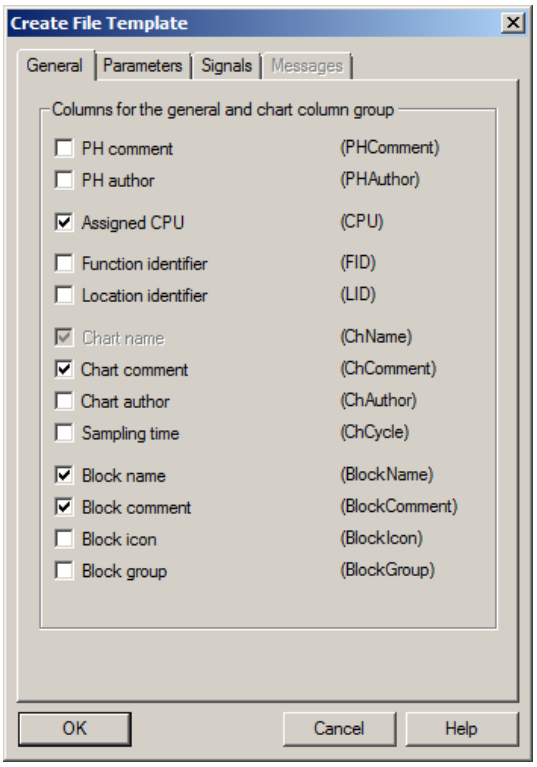
6. Next, create a new file template. (→ Create File Template…)



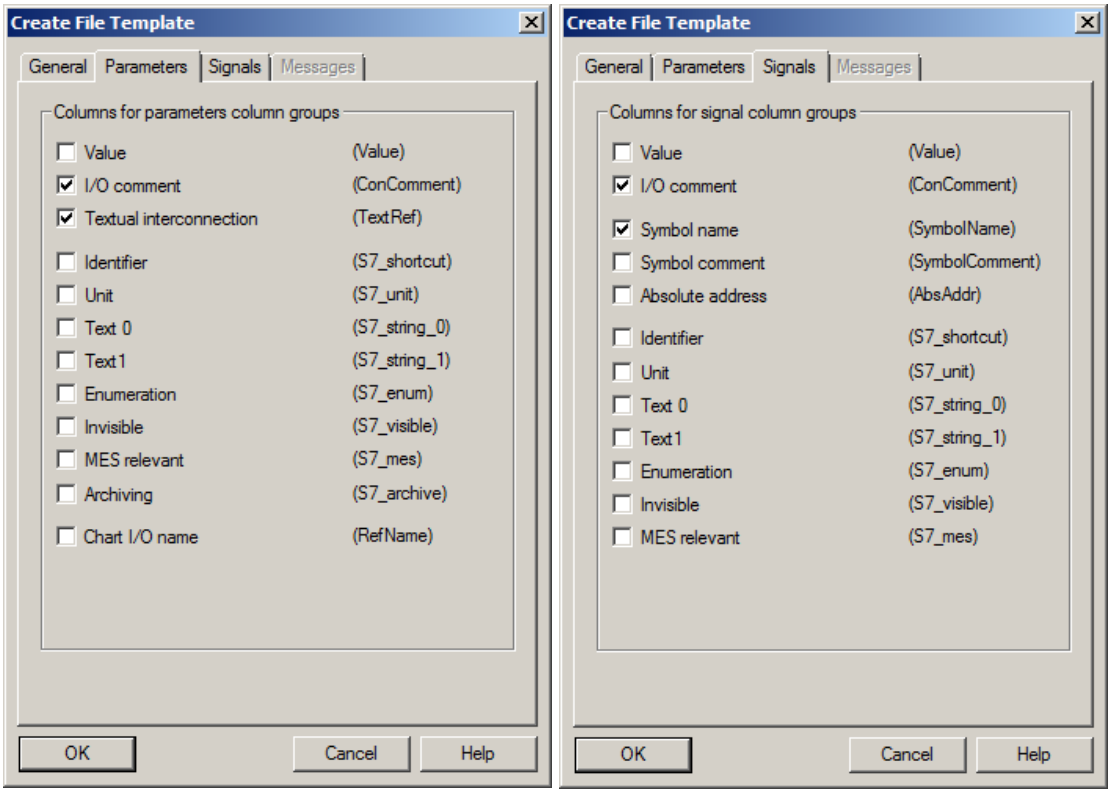
7. Assign the name p01-07-reactor-delivery-valve00-r1905-en.iea to the import file and select a memory location. (→ OK)



8. The next step is to select the general columns that are to be displayed in the import file. (→ General → Assigned CPU → Chart comment → Block name → Block comment)

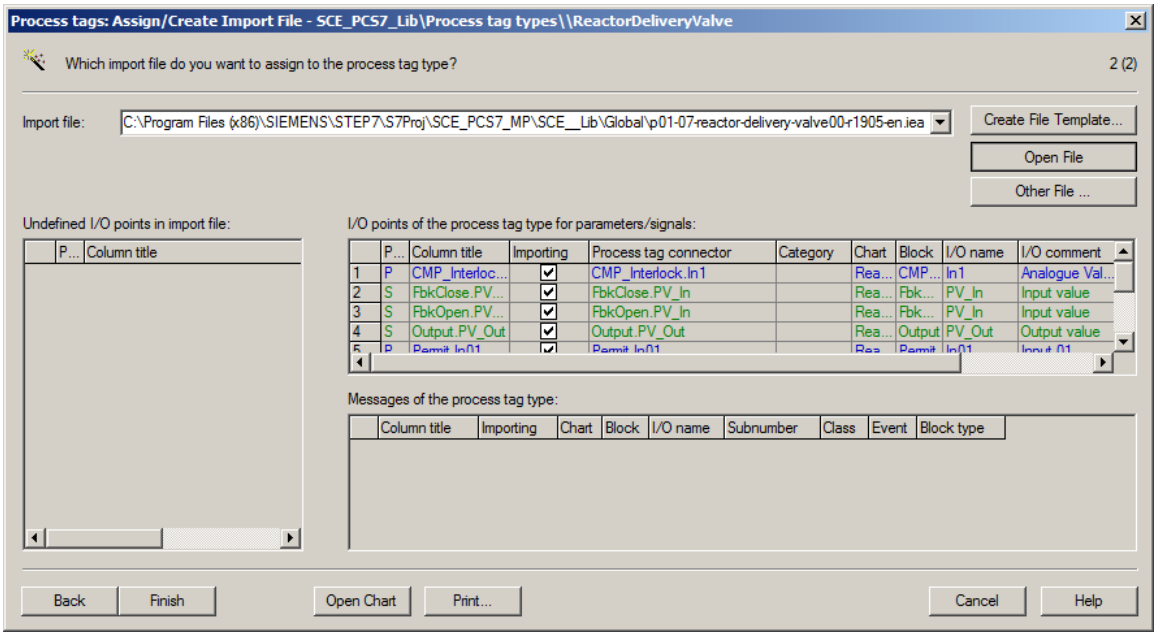


9. Then the columns to be displayed for the parameters and the signals in the import file are selected. (→ Parameters → I/O comment → Textual interconnection → Signals → I/O comment → Symbol name → OK)

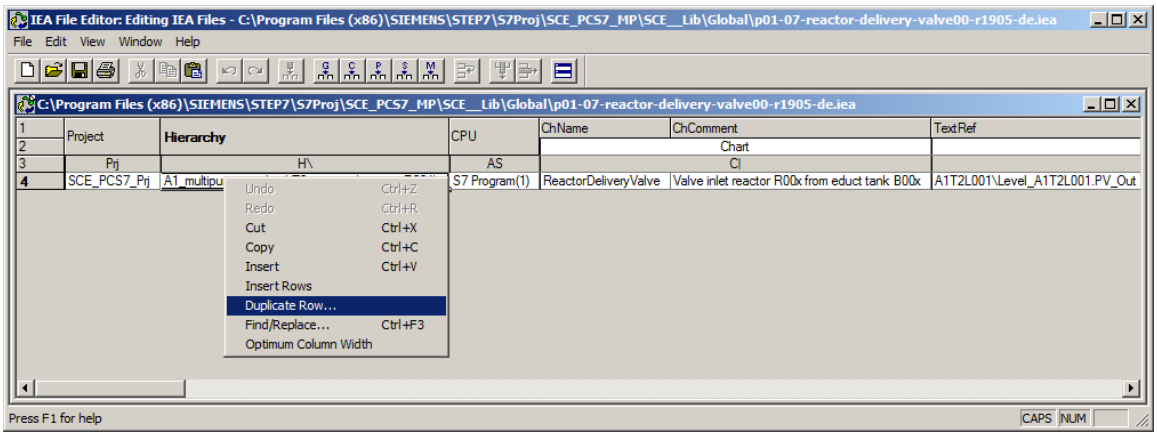


8.3 Editing the import file

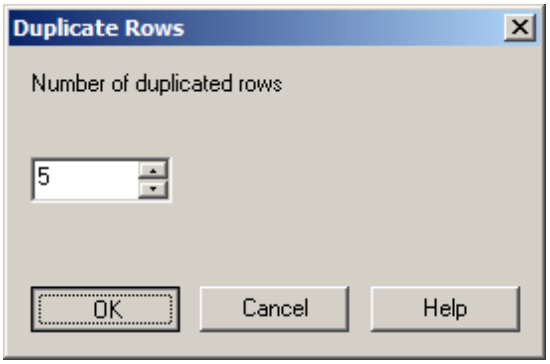
1. The import file created in this way is then opened. (→ Open File)

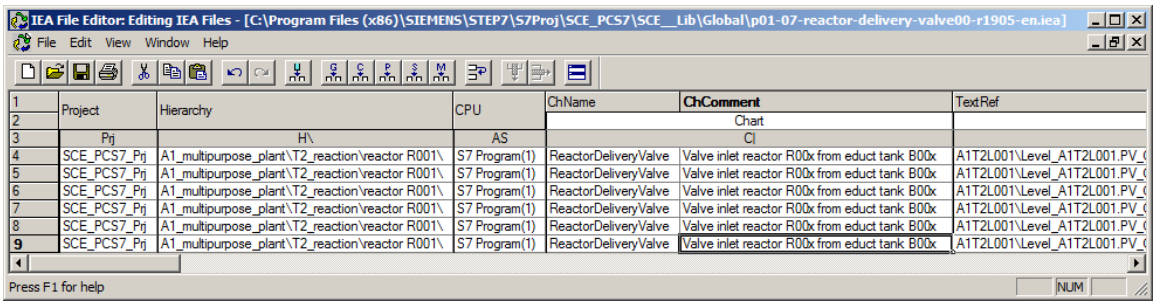


2. Now, duplicate the first row by right- clicking it and selecting menu command 'Duplicate row…'. (→ Duplicate Row…)

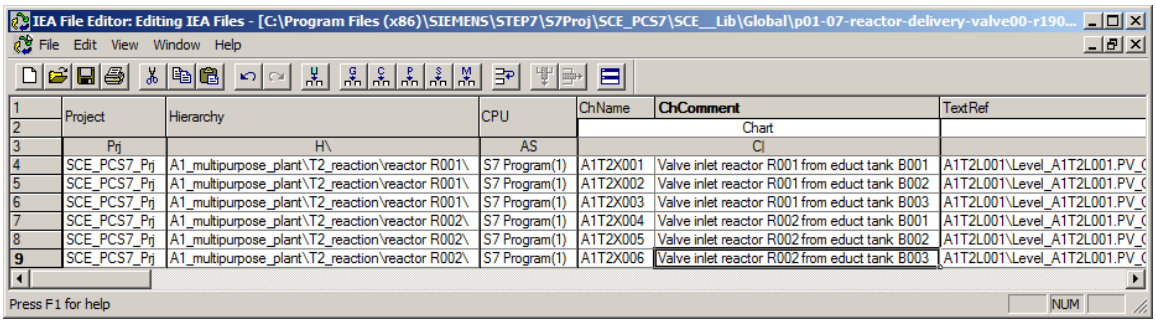


3. In the window that now opens, enter the number of rows. In this case, 5 rows are to be duplicated because the reactors have 6 inflow valves that are to be edited/created using this process tag type. (→ 5 → OK)

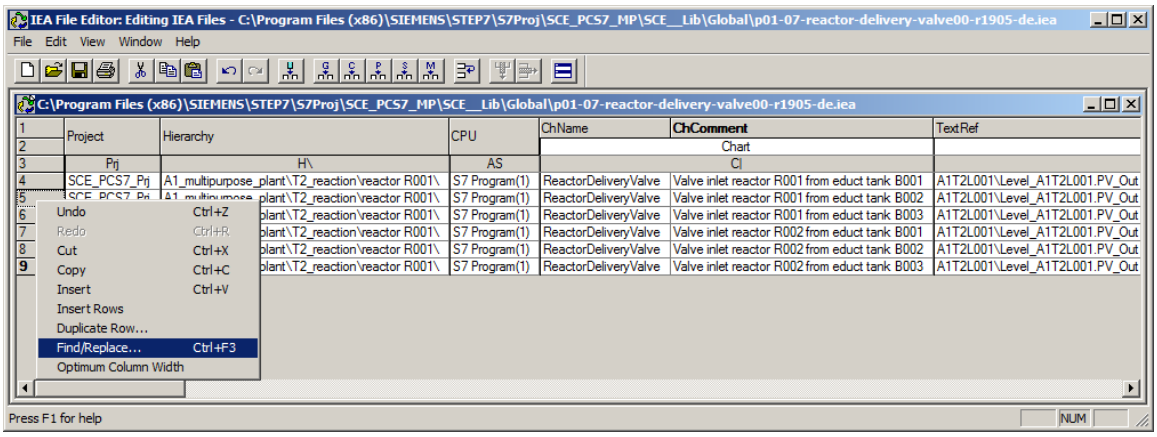


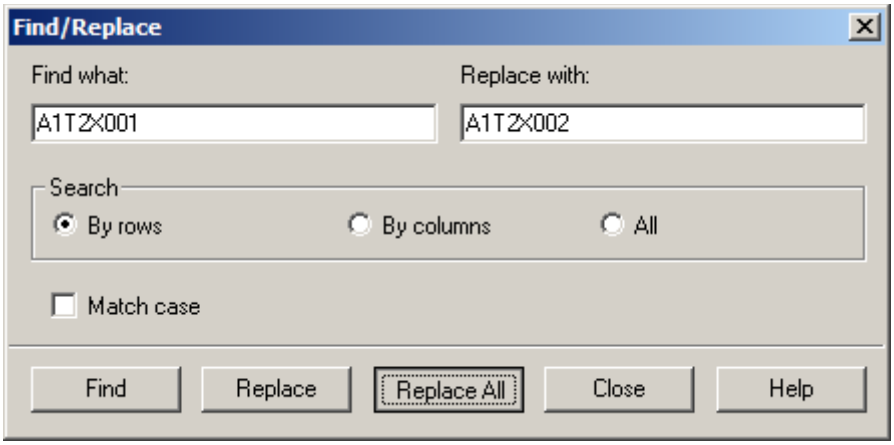


4. In the duplicated rows, enter the specific properties for each valve. Start with the hierarchy, ChName and ChComment.

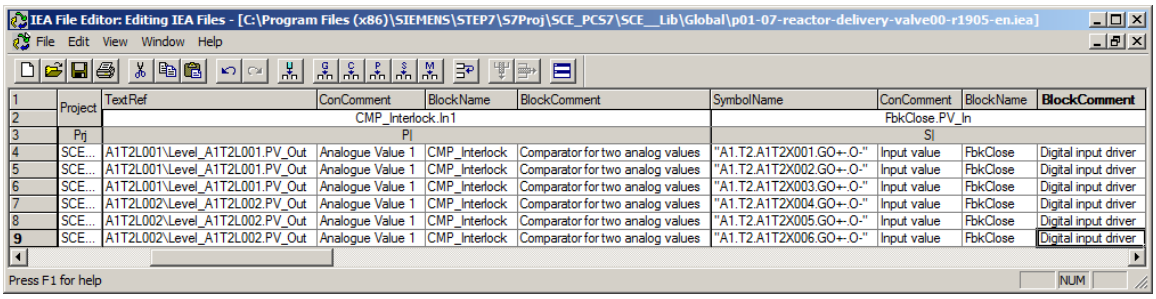


5. Next, you must set the correct parameters and signals for each row. This can be accelerated by finding and replacing by rows. In row 2, for example, you can replace 'A1T2X001' with 'A1T2X002'.

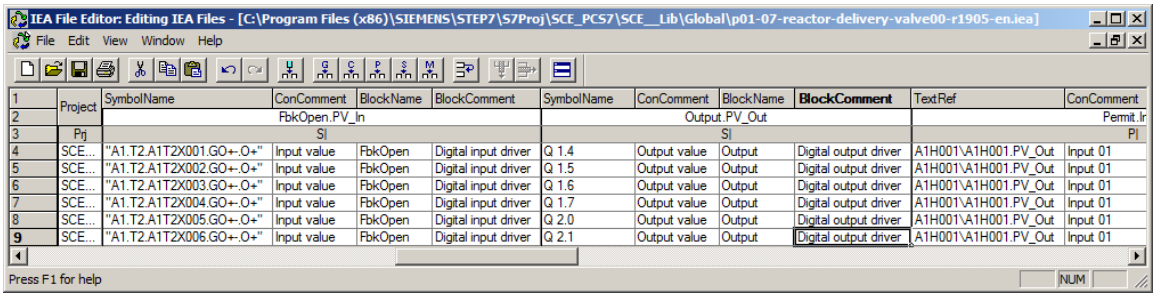




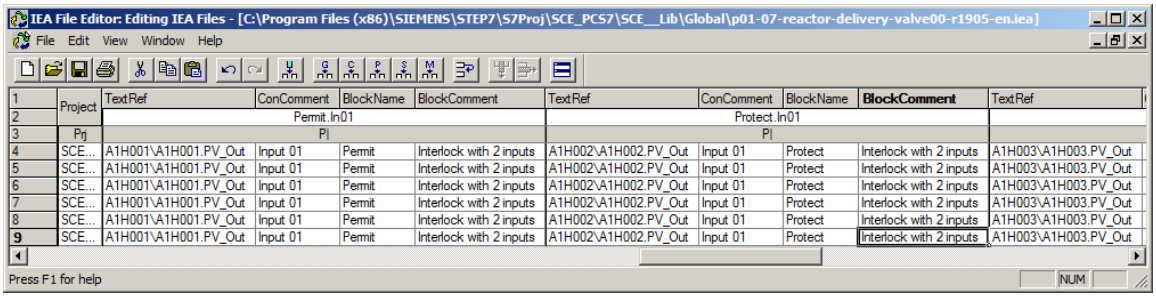
6. Now, edit the rows of the file as shown below. You should place the input signals (SymbolName column) inside quotation marks (") because they will otherwise not be found. You should enter the output signals (SymbolName column) as an absolute address or correct the CFCs afterwards.



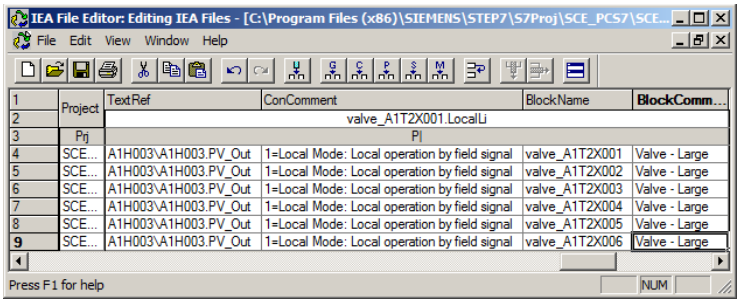
	Project	TextRef	ConComment	BlockName	BlockComment	SymbolName	ConComment	BlockName	BlockComment
2		CMP_Interlock.In1				FbkClose.PV_In			
3	Prj	PI				SI			
4	SCE...	A1T2L001\Level_A1T2L001.PV_Out	Analogue Value 1	CMP_Interlock	Comparator for two analog values	"A1.T2.A1T2X001.GO+-.O-"	Input value	FbkClose	Digital input driver
5	SCE...	A1T2L001\Level_A1T2L001.PV_Out	Analogue Value 1	CMP_Interlock	Comparator for two analog values	"A1.T2.A1T2X002.GO+-.O-"	Input value	FbkClose	Digital input driver
6	SCE...	A1T2L001\Level_A1T2L001.PV_Out	Analogue Value 1	CMP_Interlock	Comparator for two analog values	"A1.T2.A1T2X003.GO+-.O-"	Input value	FbkClose	Digital input driver
7	SCE...	A1T2L002\Level_A1T2L002.PV_Out	Analogue Value 1	CMP_Interlock	Comparator for two analog values	"A1.T2.A1T2X004.GO+-.O-"	Input value	FbkClose	Digital input driver
8	SCE...	A1T2L002\Level_A1T2L002.PV_Out	Analogue Value 1	CMP_Interlock	Comparator for two analog values	"A1.T2.A1T2X005.GO+-.O-"	Input value	FbkClose	Digital input driver
9	SCE...	A1T2L002\Level_A1T2L002.PV_Out	Analogue Value 1	CMP_Interlock	Comparator for two analog values	"A1.T2.A1T2X006.GO+-.O-"	Input value	FbkClose	Digital input driver



	Project	SymbolName	ConComment	BlockName	BlockComment	SymbolName	ConComment	BlockName	BlockComment	TextRef	ConComment
2		FbkOpen.PV_In				Output.PV_Out					Permit.In
3	Prj	SI				SI					PI
4	SCE...	"A1.T2.A1T2X001.GO+-.O+"	Input value	FbkOpen	Digital input driver	Q 1.4	Output value	Output	Digital output driver	A1H001\A1H001.PV_Out	Input 01
5	SCE...	"A1.T2.A1T2X002.GO+-.O+"	Input value	FbkOpen	Digital input driver	Q 1.5	Output value	Output	Digital output driver	A1H001\A1H001.PV_Out	Input 01
6	SCE...	"A1.T2.A1T2X003.GO+-.O+"	Input value	FbkOpen	Digital input driver	Q 1.6	Output value	Output	Digital output driver	A1H001\A1H001.PV_Out	Input 01
7	SCE...	"A1.T2.A1T2X004.GO+-.O+"	Input value	FbkOpen	Digital input driver	Q 1.7	Output value	Output	Digital output driver	A1H001\A1H001.PV_Out	Input 01
8	SCE...	"A1.T2.A1T2X005.GO+-.O+"	Input value	FbkOpen	Digital input driver	Q 2.0	Output value	Output	Digital output driver	A1H001\A1H001.PV_Out	Input 01
9	SCE...	"A1.T2.A1T2X006.GO+-.O+"	Input value	FbkOpen	Digital input driver	Q 2.1	Output value	Output	Digital output driver	A1H001\A1H001.PV_Out	Input 01

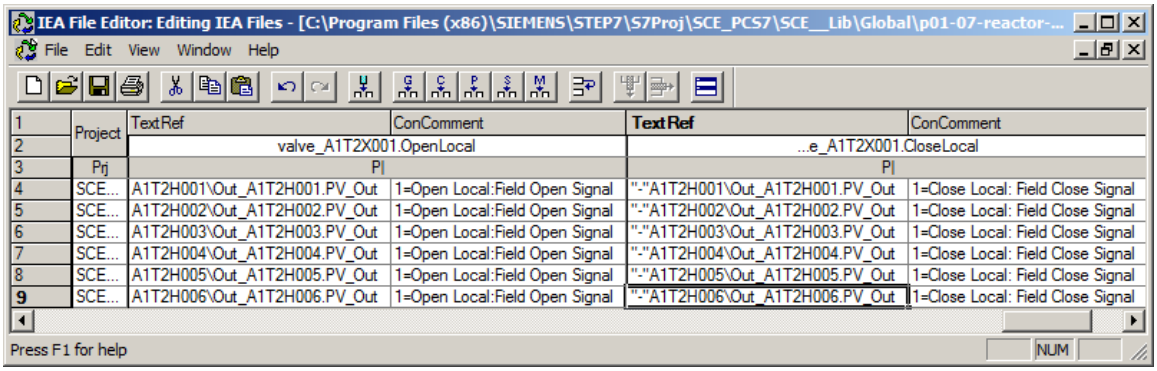


	Project	TextRef	ConComment	BlockName	BlockComment	TextRef	ConComment	BlockName	BlockComment	TextRef
2		Permit.In01				Protect.In01				
3	Prj	PI				PI				
4	SCE...	A1H001\A1H001.PV_Out	Input 01	Permit	Interlock with 2 inputs	A1H002\A1H002.PV_Out	Input 01	Protect	Interlock with 2 inputs	A1H003\A1H003.PV_Out
5	SCE...	A1H001\A1H001.PV_Out	Input 01	Permit	Interlock with 2 inputs	A1H002\A1H002.PV_Out	Input 01	Protect	Interlock with 2 inputs	A1H003\A1H003.PV_Out
6	SCE...	A1H001\A1H001.PV_Out	Input 01	Permit	Interlock with 2 inputs	A1H002\A1H002.PV_Out	Input 01	Protect	Interlock with 2 inputs	A1H003\A1H003.PV_Out
7	SCE...	A1H001\A1H001.PV_Out	Input 01	Permit	Interlock with 2 inputs	A1H002\A1H002.PV_Out	Input 01	Protect	Interlock with 2 inputs	A1H003\A1H003.PV_Out
8	SCE...	A1H001\A1H001.PV_Out	Input 01	Permit	Interlock with 2 inputs	A1H002\A1H002.PV_Out	Input 01	Protect	Interlock with 2 inputs	A1H003\A1H003.PV_Out
9	SCE...	A1H001\A1H001.PV_Out	Input 01	Permit	Interlock with 2 inputs	A1H002\A1H002.PV_Out	Input 01	Protect	Interlock with 2 inputs	A1H003\A1H003.PV_Out



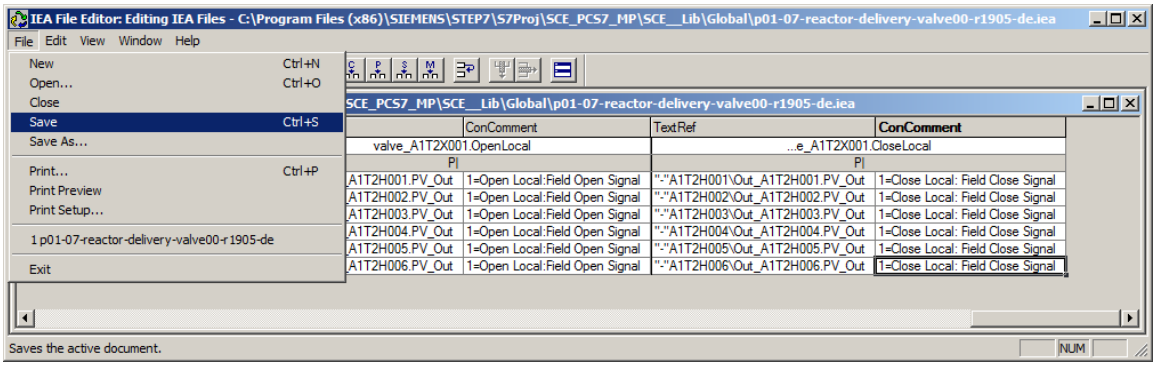
	Project	TextRef	ConComment	BlockName	BlockComm...
2		valve_A1T2X001.LocalLi			
3	Prj	PI			
4	SCE...	A1H003\A1H003.PV_Out	1=Local Mode: Local operation by field signal	valve_A1T2X001	Valve - Large
5	SCE...	A1H003\A1H003.PV_Out	1=Local Mode: Local operation by field signal	valve_A1T2X002	Valve - Large
6	SCE...	A1H003\A1H003.PV_Out	1=Local Mode: Local operation by field signal	valve_A1T2X003	Valve - Large
7	SCE...	A1H003\A1H003.PV_Out	1=Local Mode: Local operation by field signal	valve_A1T2X004	Valve - Large
8	SCE...	A1H003\A1H003.PV_Out	1=Local Mode: Local operation by field signal	valve_A1T2X005	Valve - Large
9	SCE...	A1H003\A1H003.PV_Out	1=Local Mode: Local operation by field signal	valve_A1T2X006	Valve - Large

7. Finally, change the parameter of the manual control as shown here. The character '-' in front of the textual interconnection means 'invert'. This character must be placed inside quotation marks (" ").



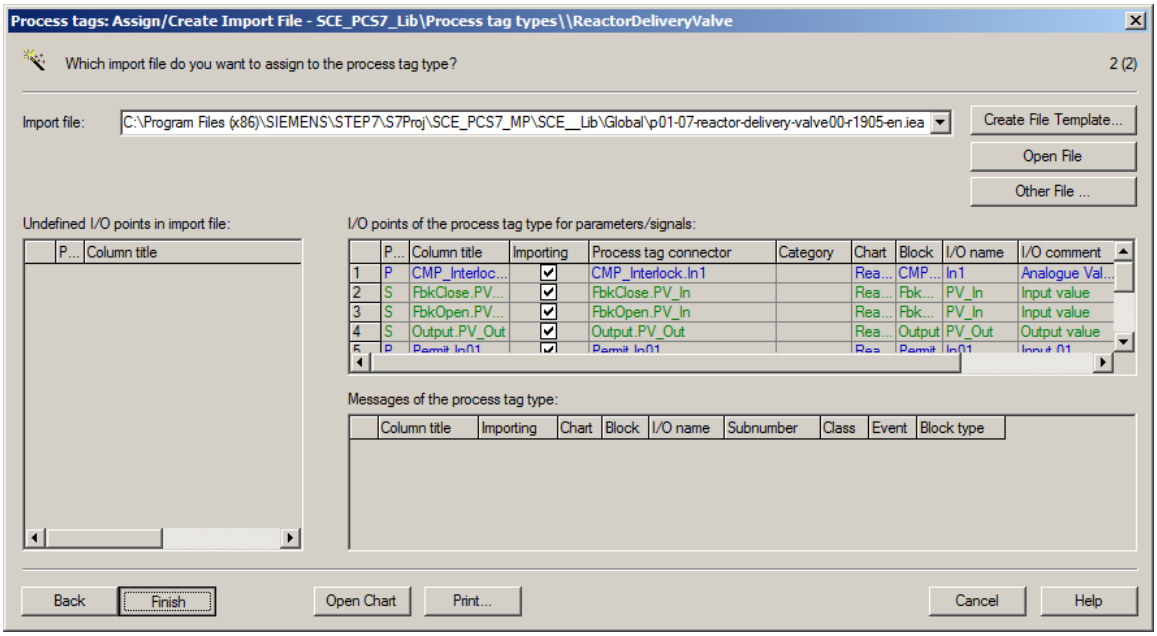
	Project	TextRef	ConComment	TextRef	ConComment
2		valve_A1T2X001.OpenLocal		...e_A1T2X001.CloseLocal	
3	Prj	PI		PI	
4	SCE...	A1T2H001\Out_A1T2H001.PV_Out	1=Open Local:Field Open Signal	"-"A1T2H001\Out_A1T2H001.PV_Out	1=Close Local: Field Close Signal
5	SCE...	A1T2H002\Out_A1T2H002.PV_Out	1=Open Local:Field Open Signal	"-"A1T2H002\Out_A1T2H002.PV_Out	1=Close Local: Field Close Signal
6	SCE...	A1T2H003\Out_A1T2H003.PV_Out	1=Open Local:Field Open Signal	"-"A1T2H003\Out_A1T2H003.PV_Out	1=Close Local: Field Close Signal
7	SCE...	A1T2H004\Out_A1T2H004.PV_Out	1=Open Local:Field Open Signal	"-"A1T2H004\Out_A1T2H004.PV_Out	1=Close Local: Field Close Signal
8	SCE...	A1T2H005\Out_A1T2H005.PV_Out	1=Open Local:Field Open Signal	"-"A1T2H005\Out_A1T2H005.PV_Out	1=Close Local: Field Close Signal
9	SCE...	A1T2H006\Out_A1T2H006.PV_Out	1=Open Local:Field Open Signal	"-"A1T2H006\Out_A1T2H006.PV_Out	1=Close Local: Field Close Signal

8. After all changes have been made, save the file. (→ File → Save → Close)



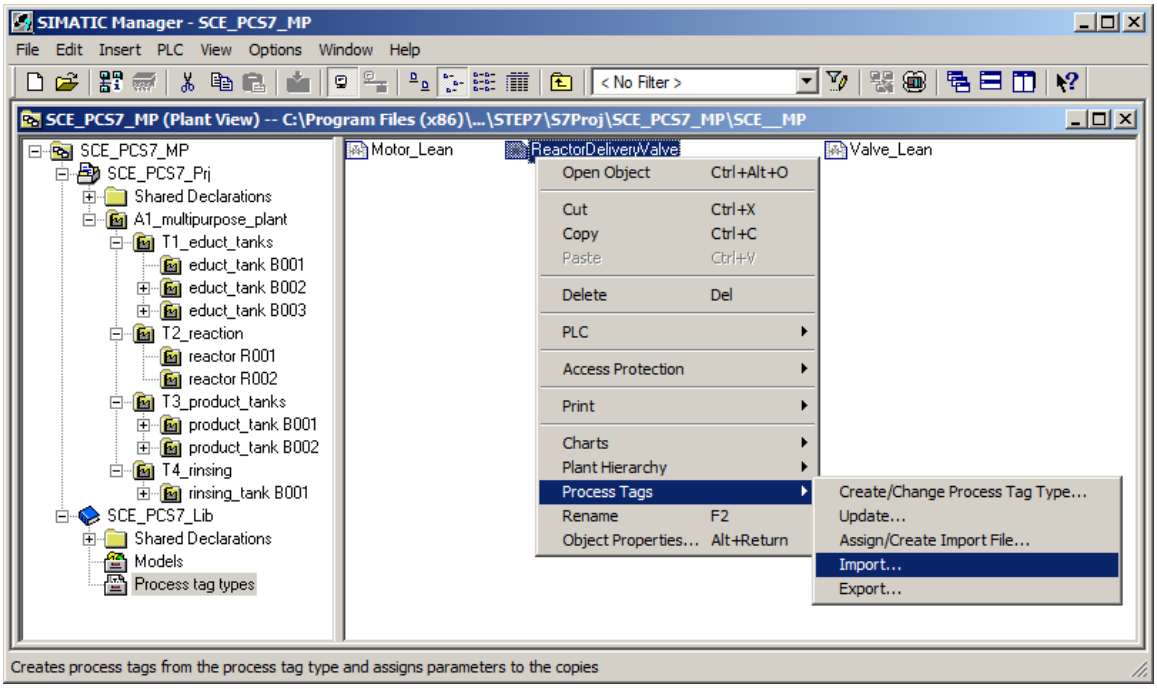
	ConComment	TextRef	ConComment
	valve_A1T2X001.OpenLocal		...e_A1T2X001.CloseLocal
	PI		PI
A1T2H001.PV_Out	1=Open Local:Field Open Signal	"-"A1T2H001\Out_A1T2H001.PV_Out	1=Close Local: Field Close Signal
A1T2H002.PV_Out	1=Open Local:Field Open Signal	"-"A1T2H002\Out_A1T2H002.PV_Out	1=Close Local: Field Close Signal
A1T2H003.PV_Out	1=Open Local:Field Open Signal	"-"A1T2H003\Out_A1T2H003.PV_Out	1=Close Local: Field Close Signal
A1T2H004.PV_Out	1=Open Local:Field Open Signal	"-"A1T2H004\Out_A1T2H004.PV_Out	1=Close Local: Field Close Signal
A1T2H005.PV_Out	1=Open Local:Field Open Signal	"-"A1T2H005\Out_A1T2H005.PV_Out	1=Close Local: Field Close Signal
A1T2H006.PV_Out	1=Open Local:Field Open Signal	"-"A1T2H006\Out_A1T2H006.PV_Out	1=Close Local: Field Close Signal

9. The creation and assignment of the import file will now be finished. (→ Finish)

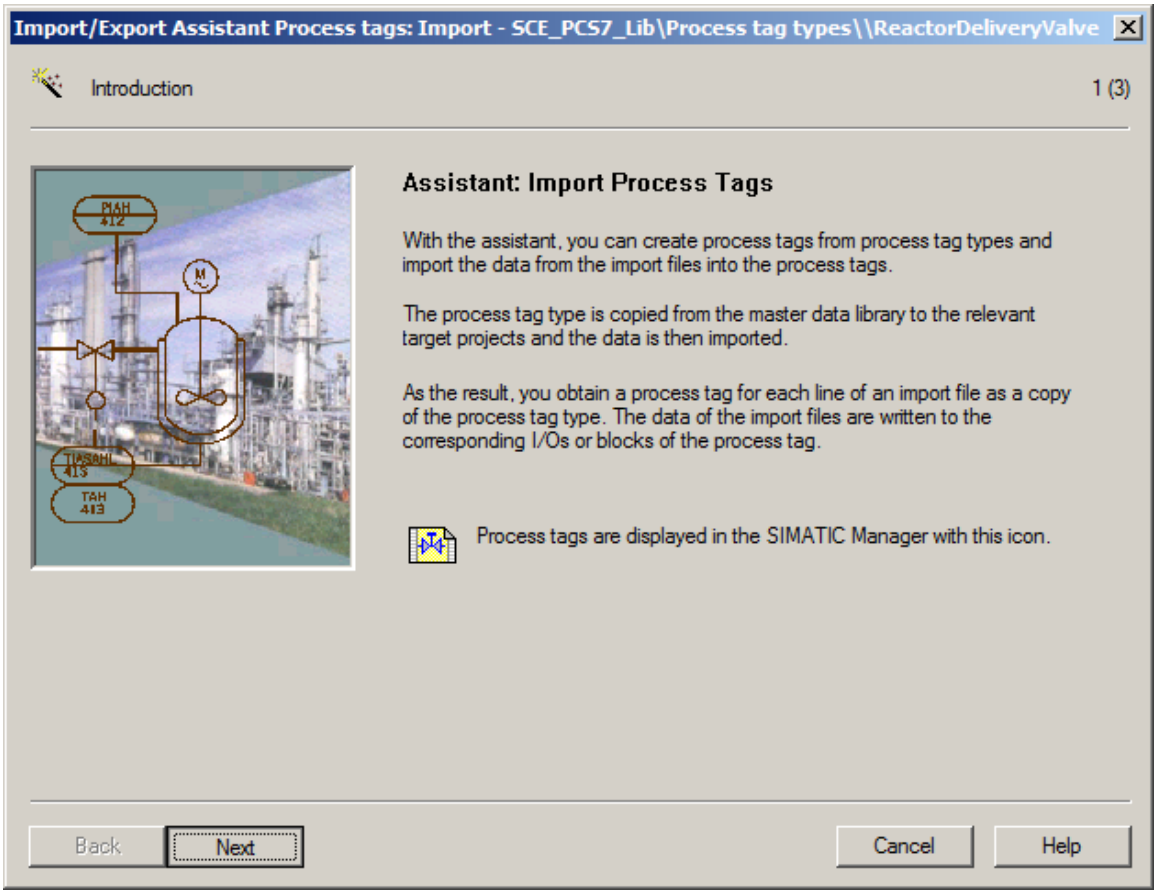


8.4 Importing process tags

1. The import of the created process tag type can now be started. (→ ReactorDeliveryValve → Process Tags → Import…)

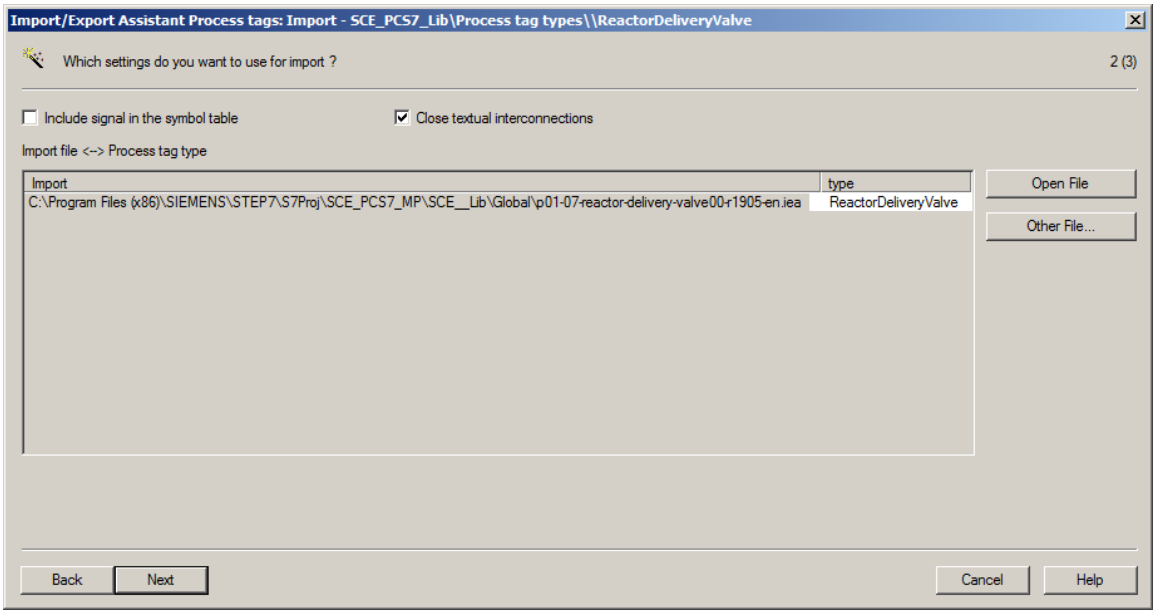


2. Confirm the first step of the dialog with 'Next'. (→ Next)



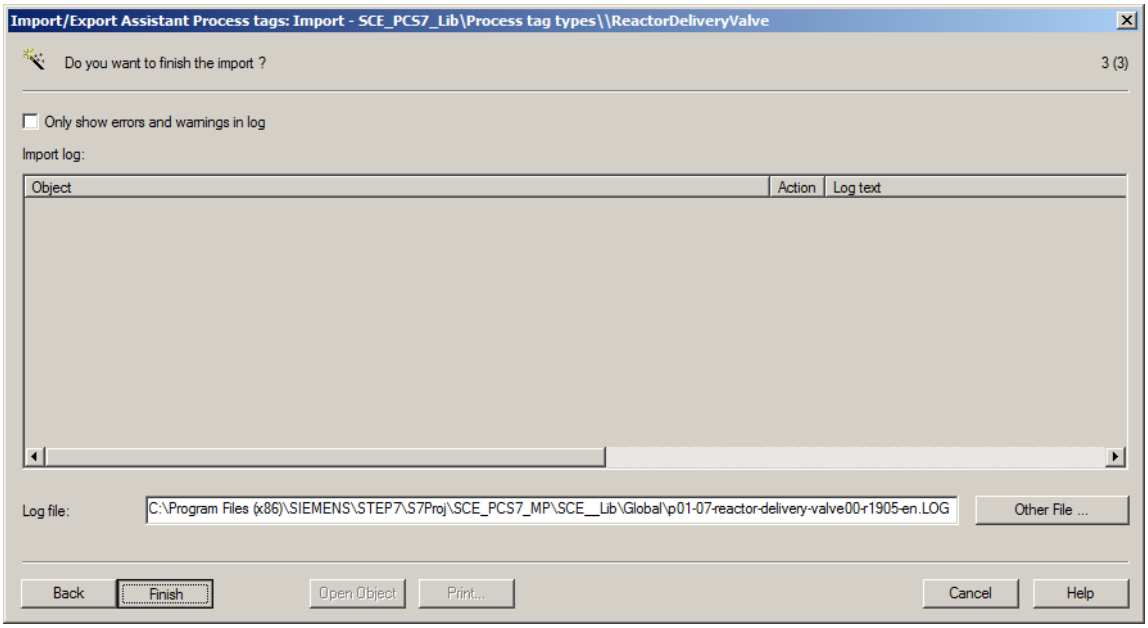
3. In the next dialog box, select the 'Close textual interconnections' option and then click 'Next'.

   (→ Close textual interconnections → Next)

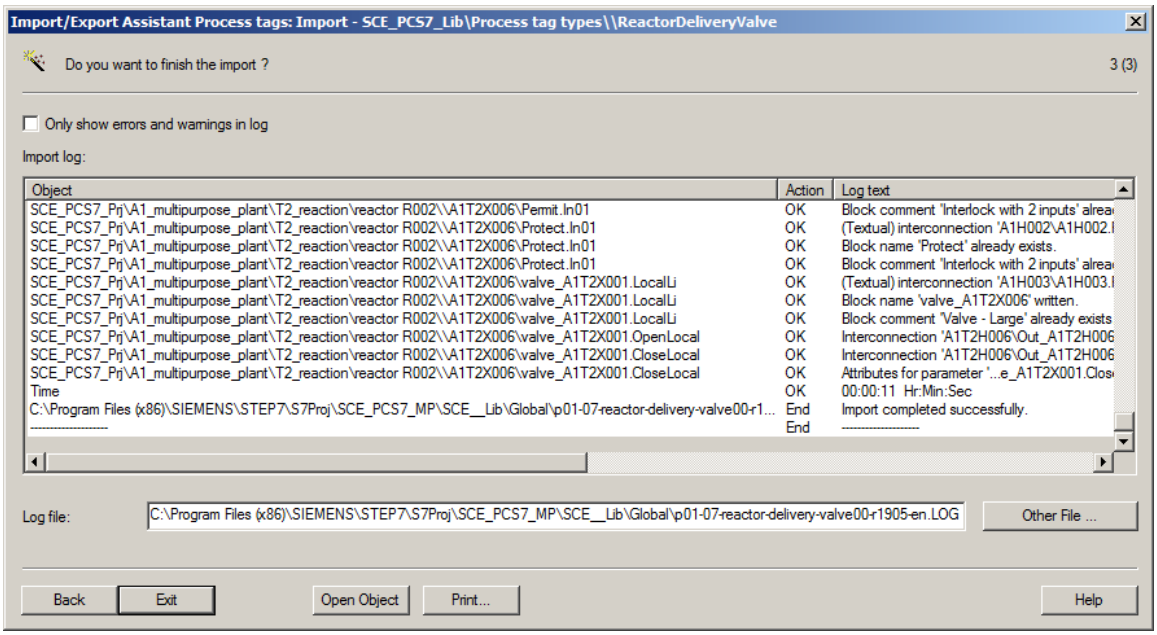


4. The import can now be started by selecting 'Finish'.

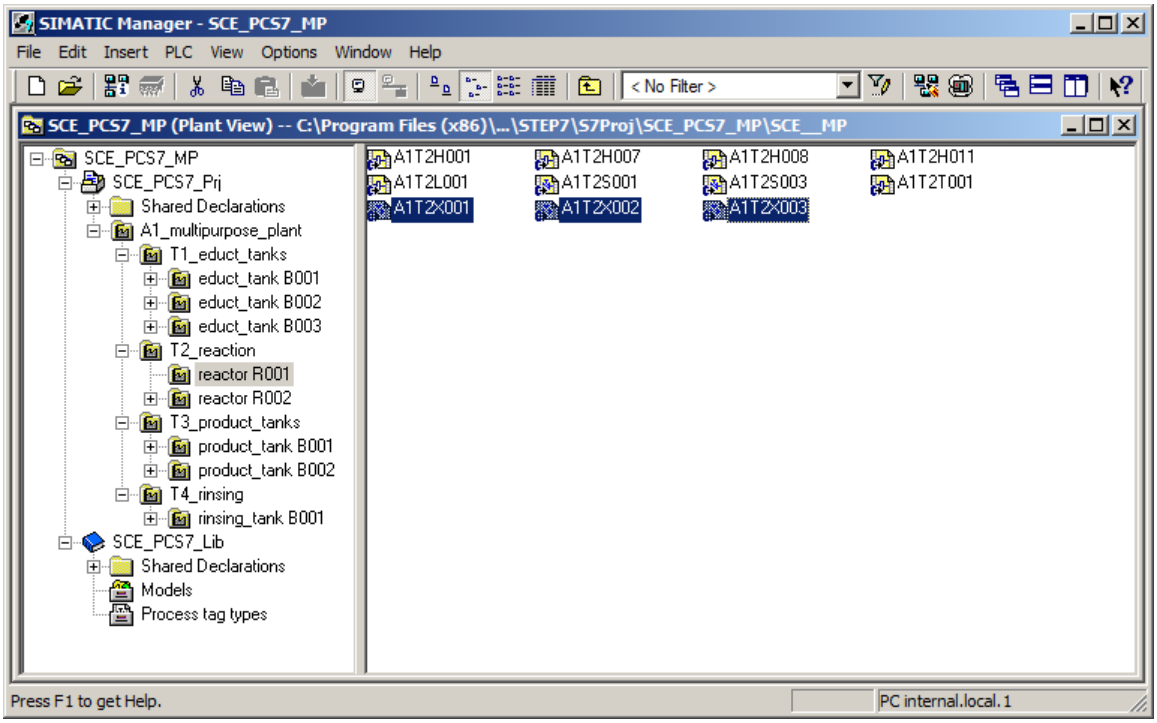
   (→ Finish)

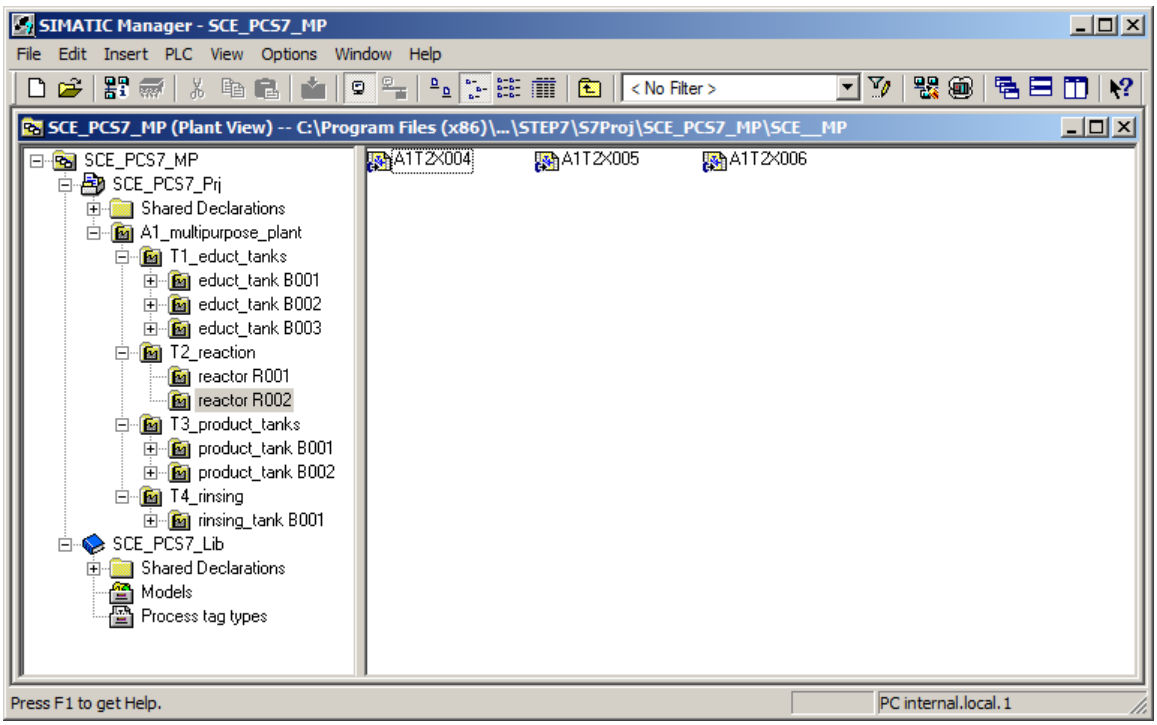


5. After conclusion of the import, the log is displayed. (→ Exit)



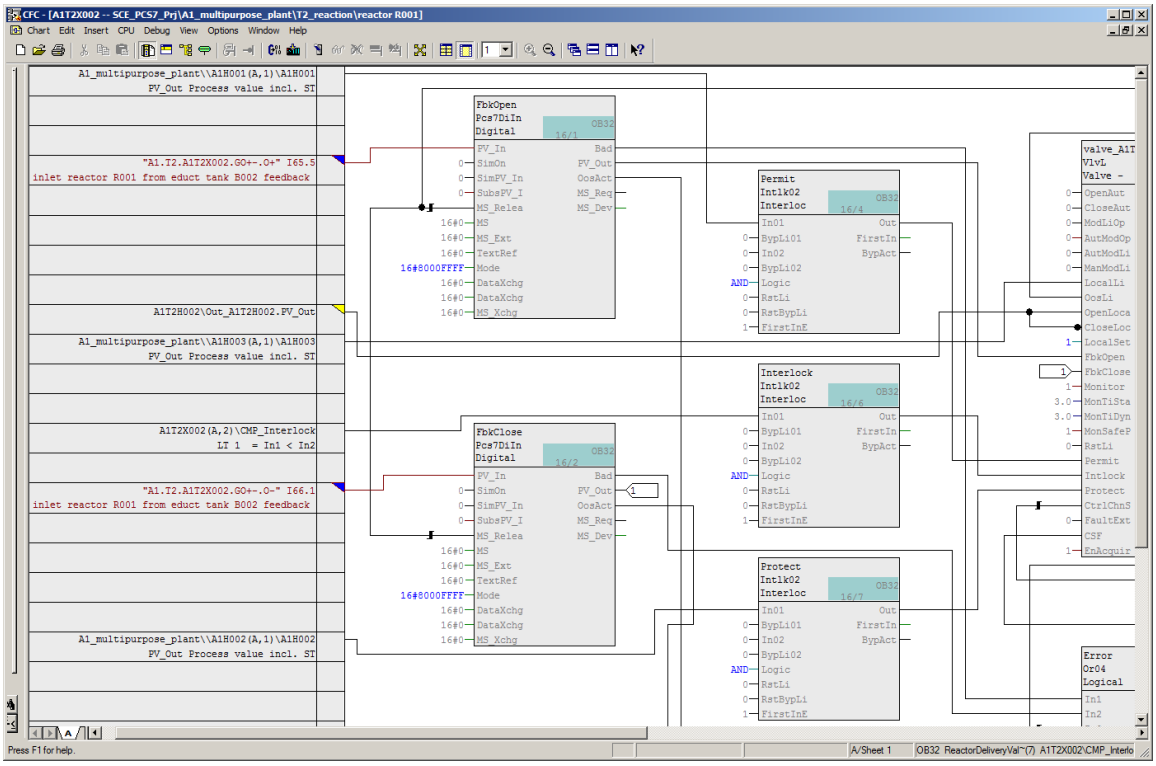
6. The newly imported CFCs are now located in the Reactor R001 hierarchy level. In this manner, a large number of charts can be created quickly and efficiently. The interesting aspect of this method is that the changes to the charts are not made individually but in the form of a table via the import file. Nevertheless, each individual chart can be viewed and changed with the CFC editor afterwards.

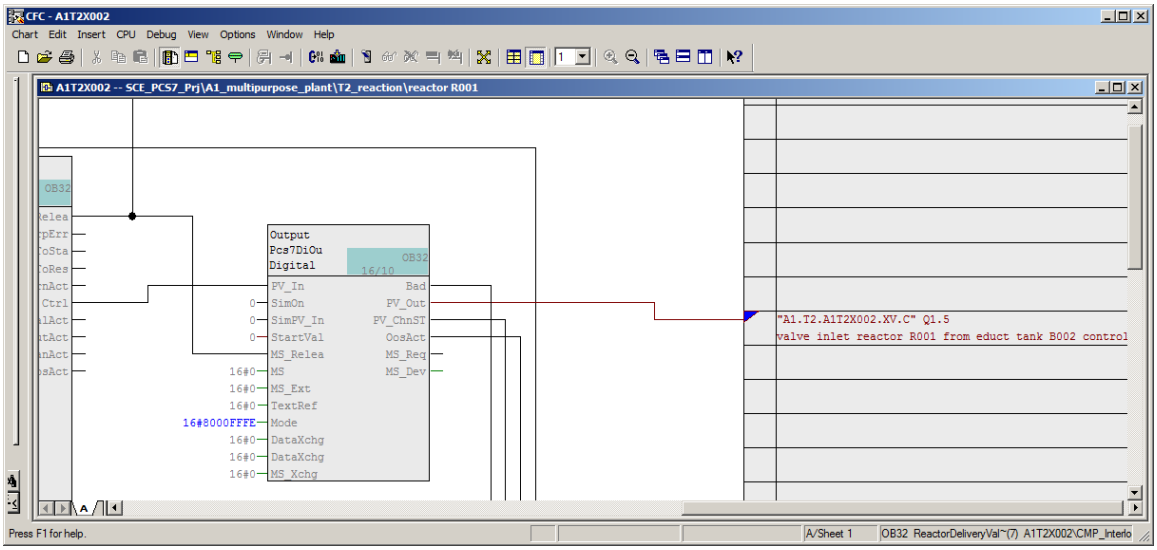




8.5 Check of the imported CFCs

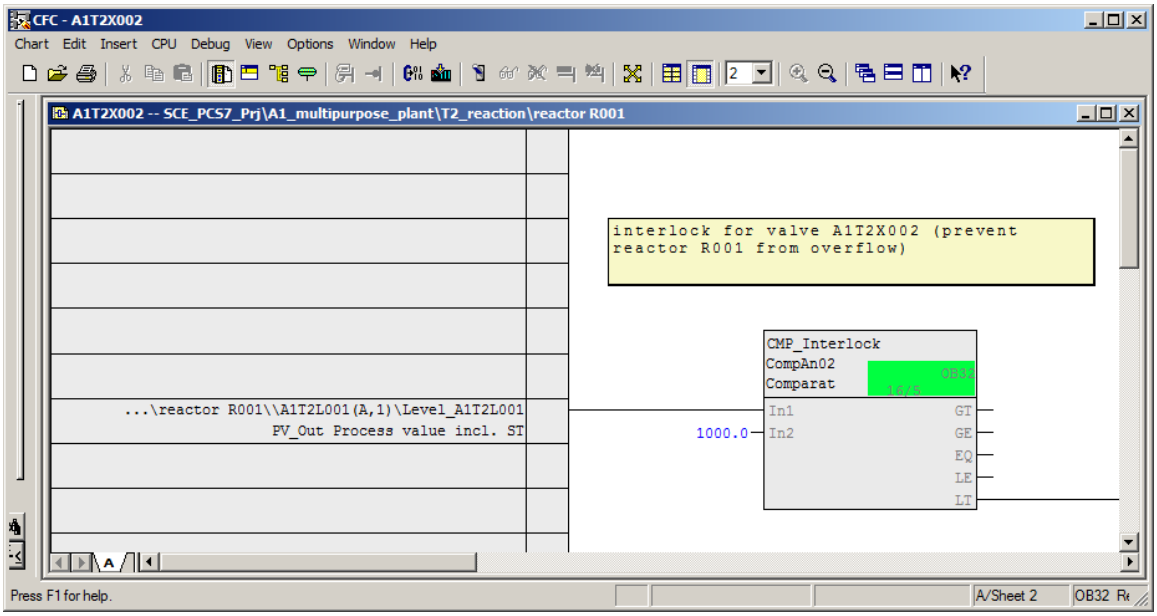
1. Open the newly created CFCs and check the input and output signals and the block names. Textual interconnections to existing CFCs should already be closed.



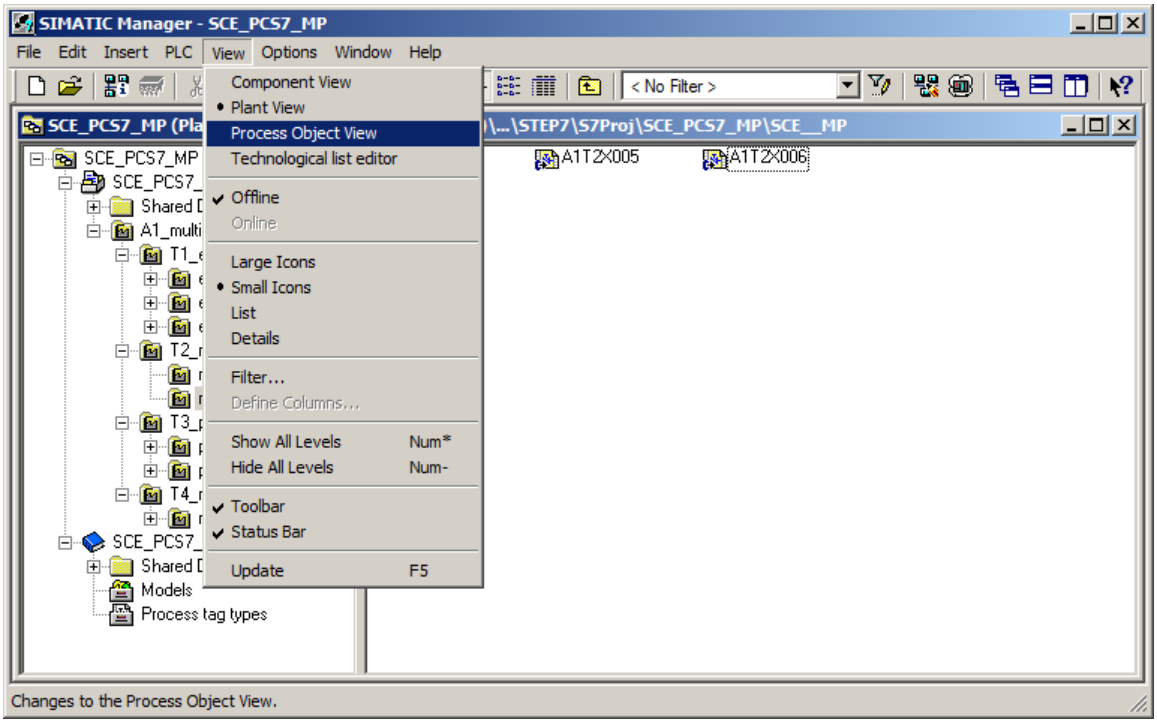


Note:

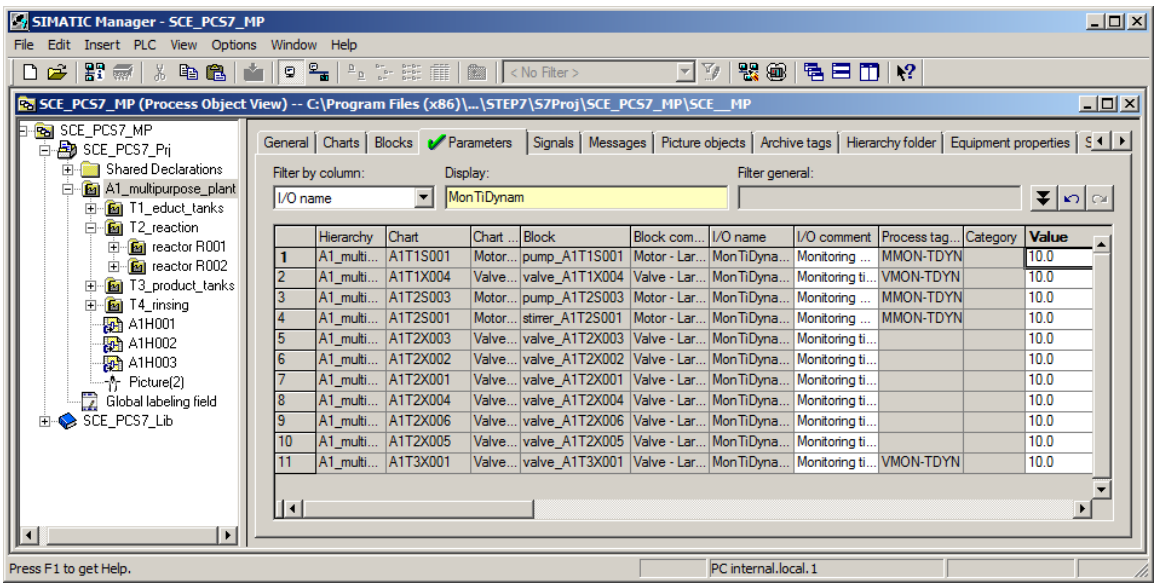
– *If you have placed the output signals inside quotation marks, the original output linked in the process tag type must still be deleted manually. Alternatively, the address linked at the output can also be deleted in the process tag type and then newly imported.*



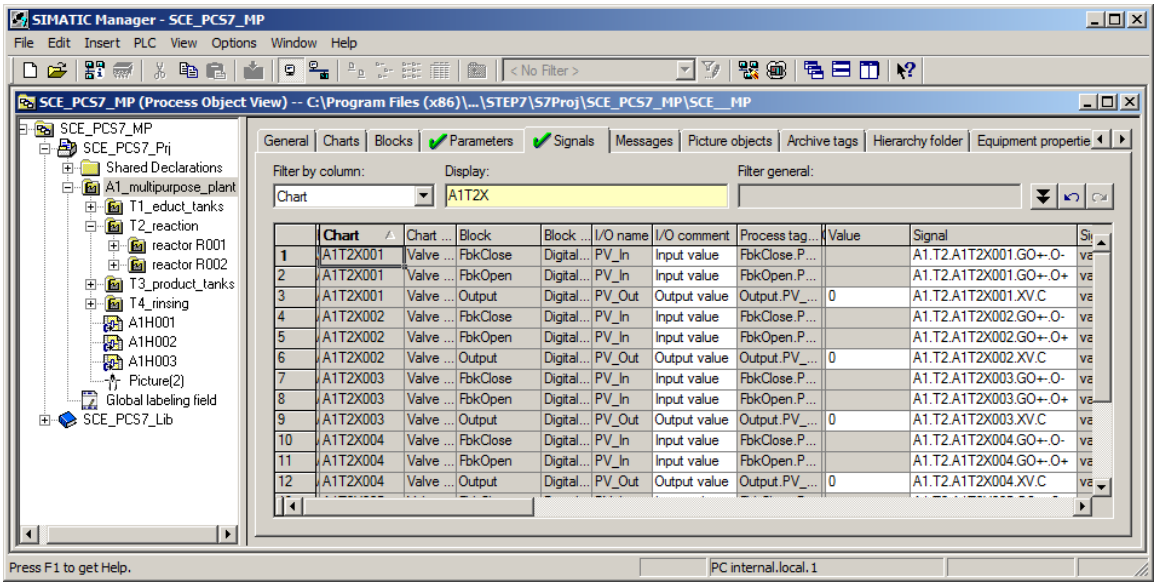
2. Another method for making changes in multiple existing charts without having to open them is the process object view. (→ View → Process Object View)



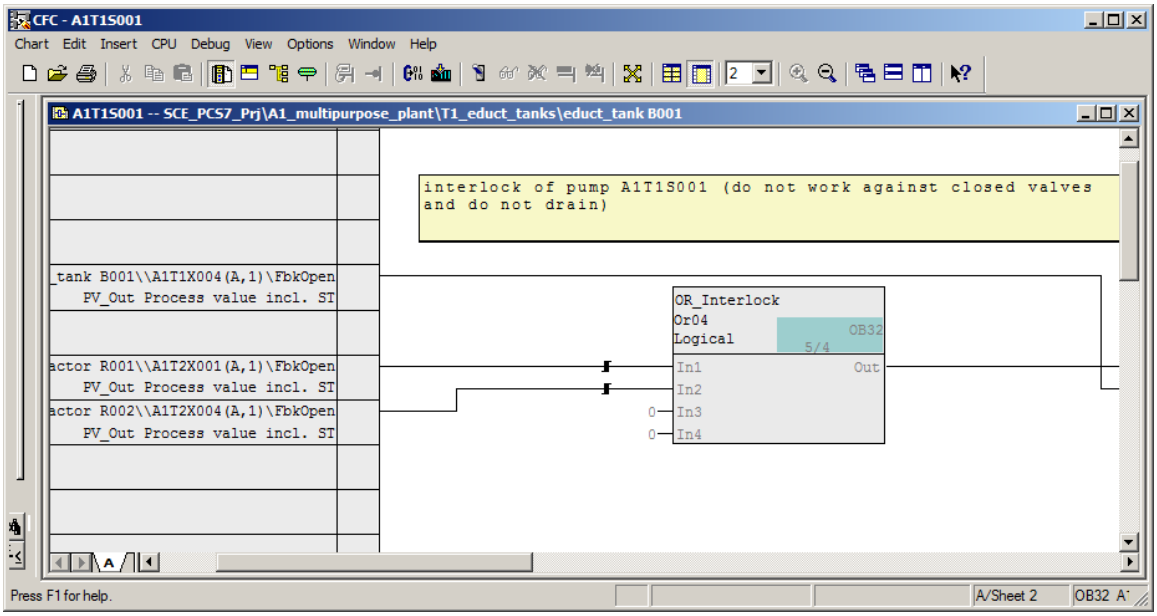
3. By setting a filter for the I/O 'MonTiDynamic' in the 'Parameters' tab, the user can change the value of a parameter for several CFCs, for example. Only the elements located below the hierarchy level selected in the left side of the window that correspond to the filter criteria are displayed. Change the value here for all displayed I/Os to '10.0'. (→ A1_multipurpose_plant → Parameters → Filter by column: I/O name → Display: MonTiDynam → Value → 10.0)



4. By utilizing the 'Parameters' or 'Signals' tabs, extensive changes can be made quickly in the CFCs. In this example, however, everything is to remain unchanged and you return to the plant view. (→ View → Plant View).

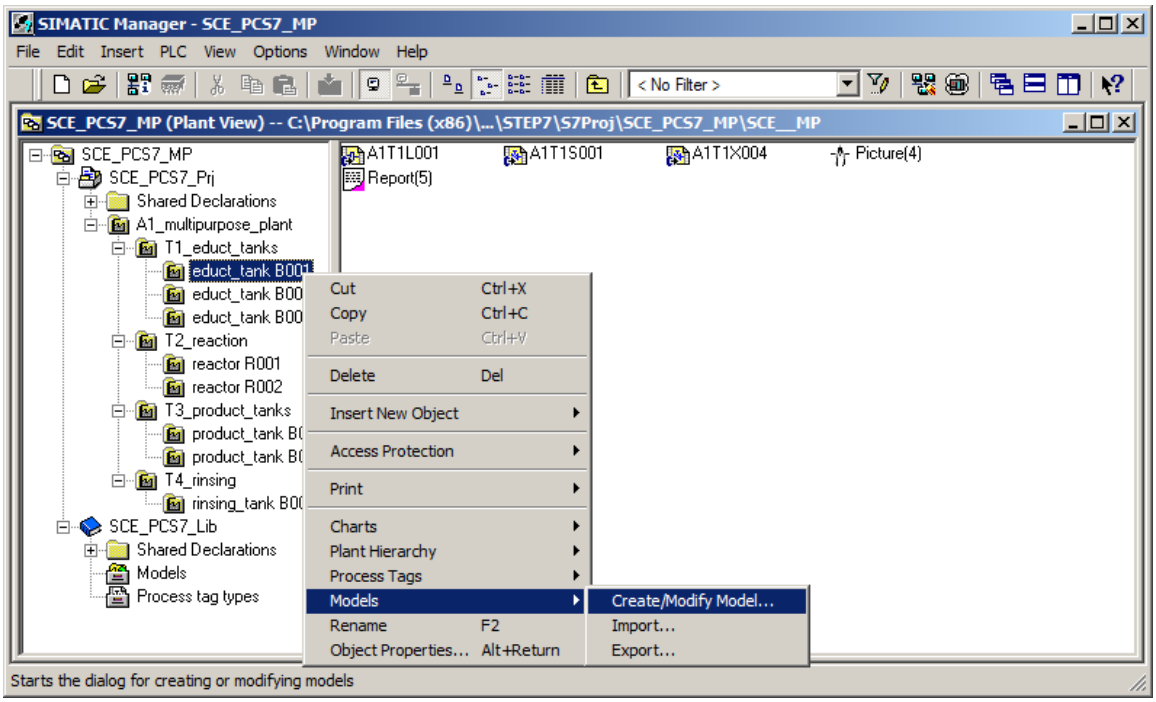


5. Before finally creating a model for the educt tank, if you have not already done so, complete the interlocking of the pump A1T1S001 with the valve A1T2X004 created from the process tag type as shown below.

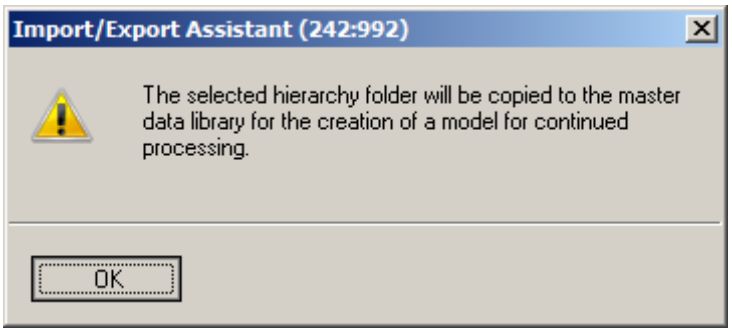


8.6 Creating a model

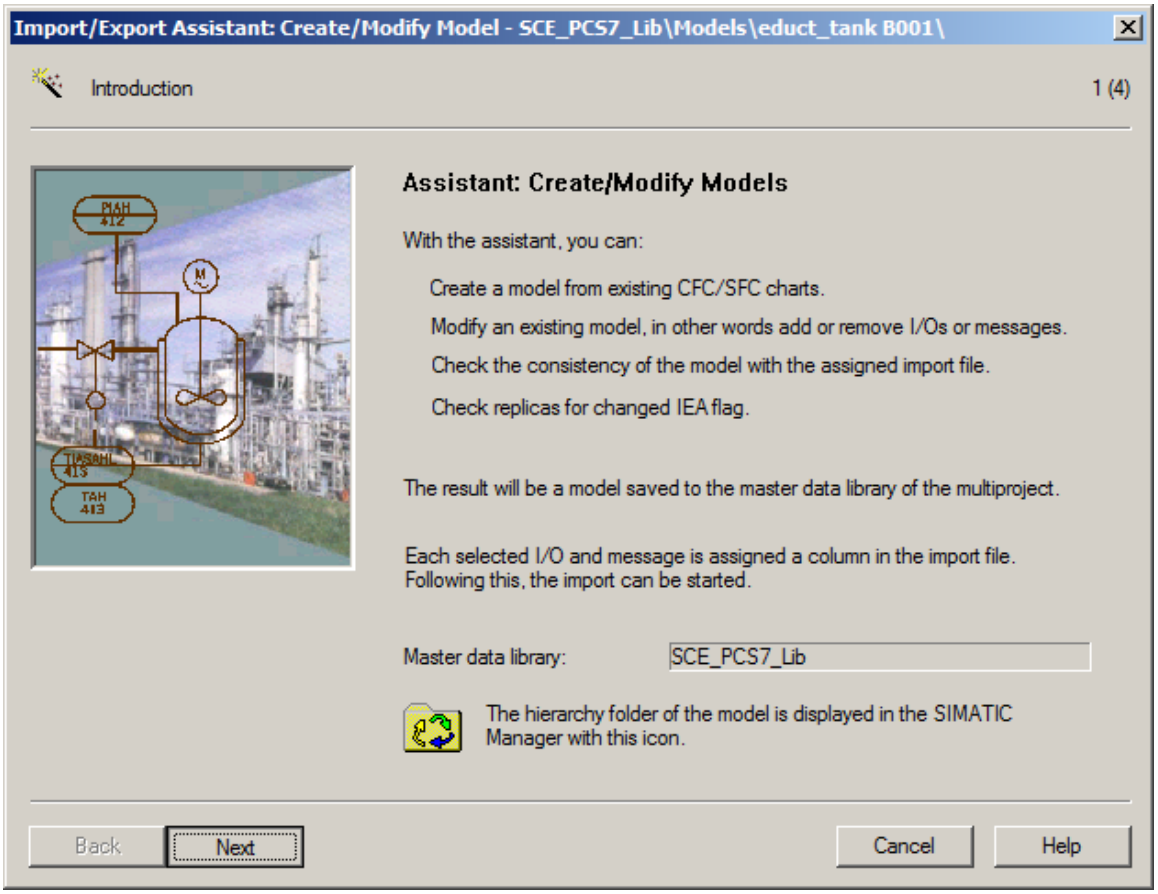
1. Educt tank B001 with all its CFCs will be used as a model. First, delete Picture(4) and then create a model. (→ Educt tank B001 → Models → Create/Modify Model…)



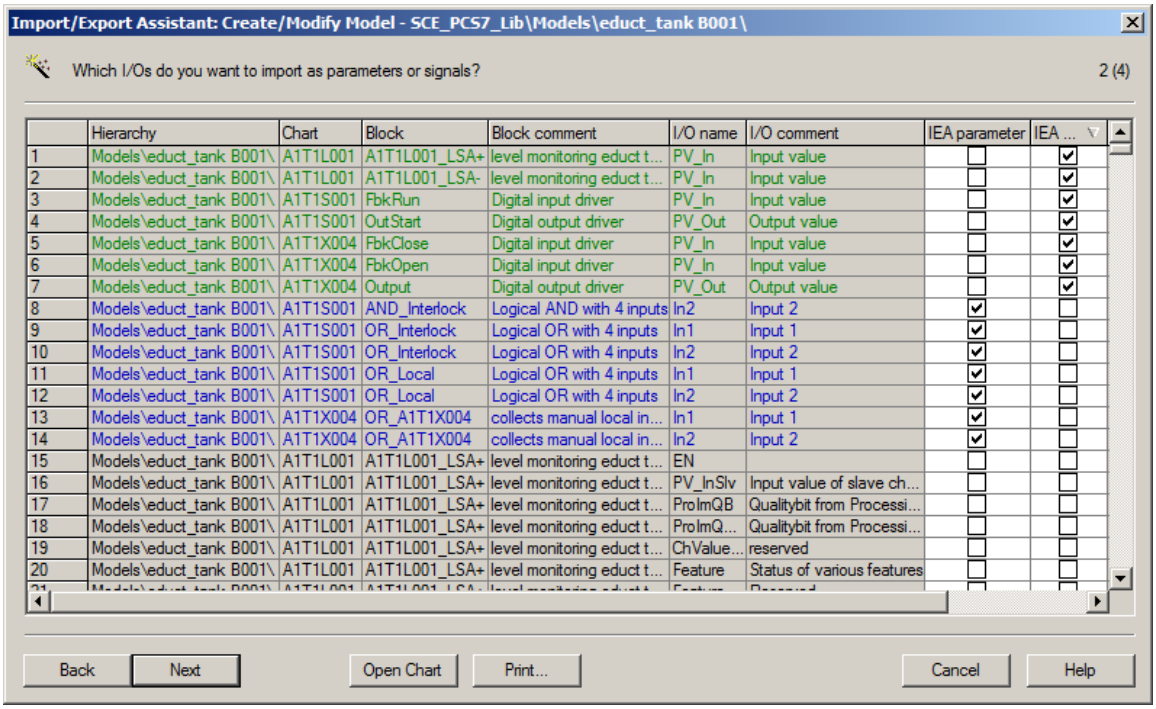
2. Confirm the message that follows with 'OK'. (→ OK)



3. Confirm the introductory screen of the dialog assistant with 'Next'. (→ Next)

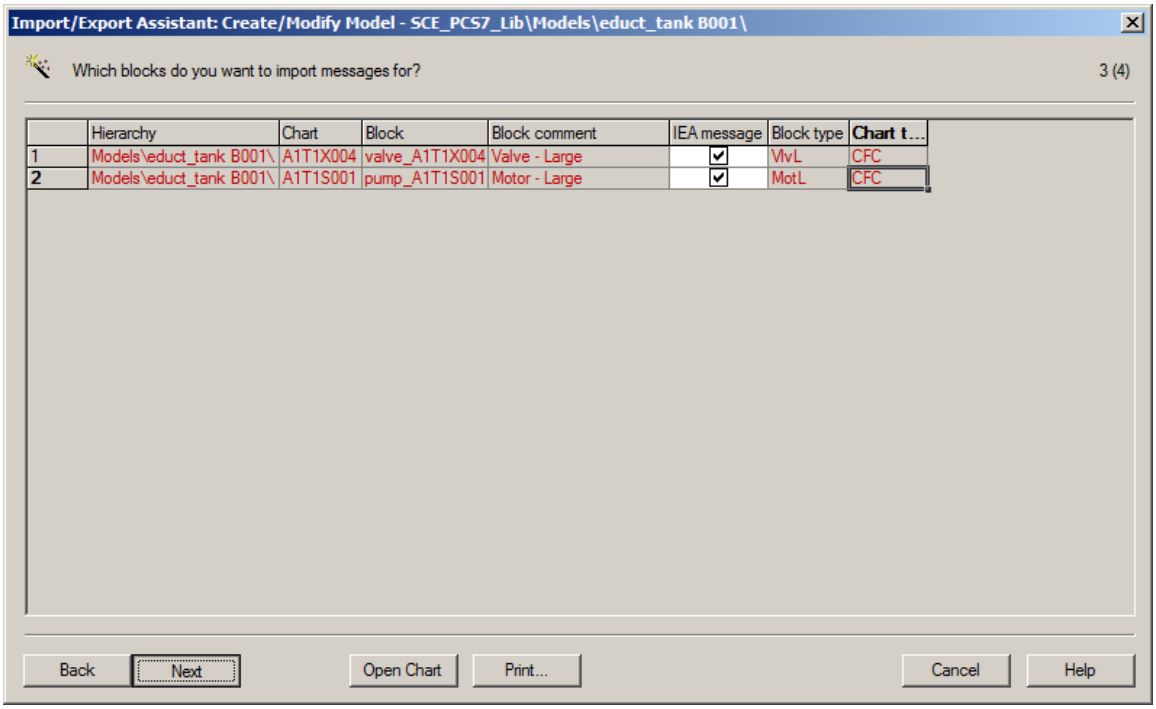


4. In the next step, specify which parameters (blue) and signals (green) will be displayed in the Import/Export Assistant. Select the parameters/signals shown below. (→ IEA parameter → IEA signals → Next)



	Hierarchy	Chart	Block	Block comment	I/O name	I/O comment	IEA parameter	IEA ...
1	Models\educt_tank B001\	A1T1L001	A1T1L001_LSA+	level monitoring educt t...	PV_In	Input value	☐	☑
2	Models\educt_tank B001\	A1T1L001	A1T1L001_LSA-	level monitoring educt t...	PV_In	Input value	☐	☑
3	Models\educt_tank B001\	A1T1S001	FbkRun	Digital input driver	PV_In	Input value	☐	☑
4	Models\educt_tank B001\	A1T1S001	OutStart	Digital output driver	PV_Out	Output value	☐	☑
5	Models\educt_tank B001\	A1T1X004	FbkClose	Digital input driver	PV_In	Input value	☐	☑
6	Models\educt_tank B001\	A1T1X004	FbkOpen	Digital input driver	PV_In	Input value	☐	☑
7	Models\educt_tank B001\	A1T1X004	Output	Digital output driver	PV_Out	Output value	☐	☑
8	Models\educt_tank B001\	A1T1S001	AND_Interlock	Logical AND with 4 inputs	In2	Input 2	☑	☐
9	Models\educt_tank B001\	A1T1S001	OR_Interlock	Logical OR with 4 inputs	In1	Input 1	☑	☐
10	Models\educt_tank B001\	A1T1S001	OR_Interlock	Logical OR with 4 inputs	In2	Input 2	☑	☐
11	Models\educt_tank B001\	A1T1S001	OR_Local	Logical OR with 4 inputs	In1	Input 1	☑	☐
12	Models\educt_tank B001\	A1T1S001	OR_Local	Logical OR with 4 inputs	In2	Input 2	☑	☐
13	Models\educt_tank B001\	A1T1X004	OR_A1T1X004	collects manual local in...	In1	Input 1	☑	☐
14	Models\educt_tank B001\	A1T1X004	OR_A1T1X004	collects manual local in...	In2	Input 2	☑	☐
15	Models\educt_tank B001\	A1T1L001	A1T1L001_LSA+	level monitoring educt t...	EN		☐	☐
16	Models\educt_tank B001\	A1T1L001	A1T1L001_LSA+	level monitoring educt t...	PV_InSlv	Input value of slave ch...	☐	☐
17	Models\educt_tank B001\	A1T1L001	A1T1L001_LSA+	level monitoring educt t...	ProImQB	Qualitybit from Processi...	☐	☐
18	Models\educt_tank B001\	A1T1L001	A1T1L001_LSA+	level monitoring educt t...	ProImQ...	Qualitybit from Processi...	☐	☐
19	Models\educt_tank B001\	A1T1L001	A1T1L001_LSA+	level monitoring educt t...	ChValue...	reserved	☐	☐
20	Models\educt_tank B001\	A1T1L001	A1T1L001_LSA+	level monitoring educt t...	Feature	Status of various features	☐	☐

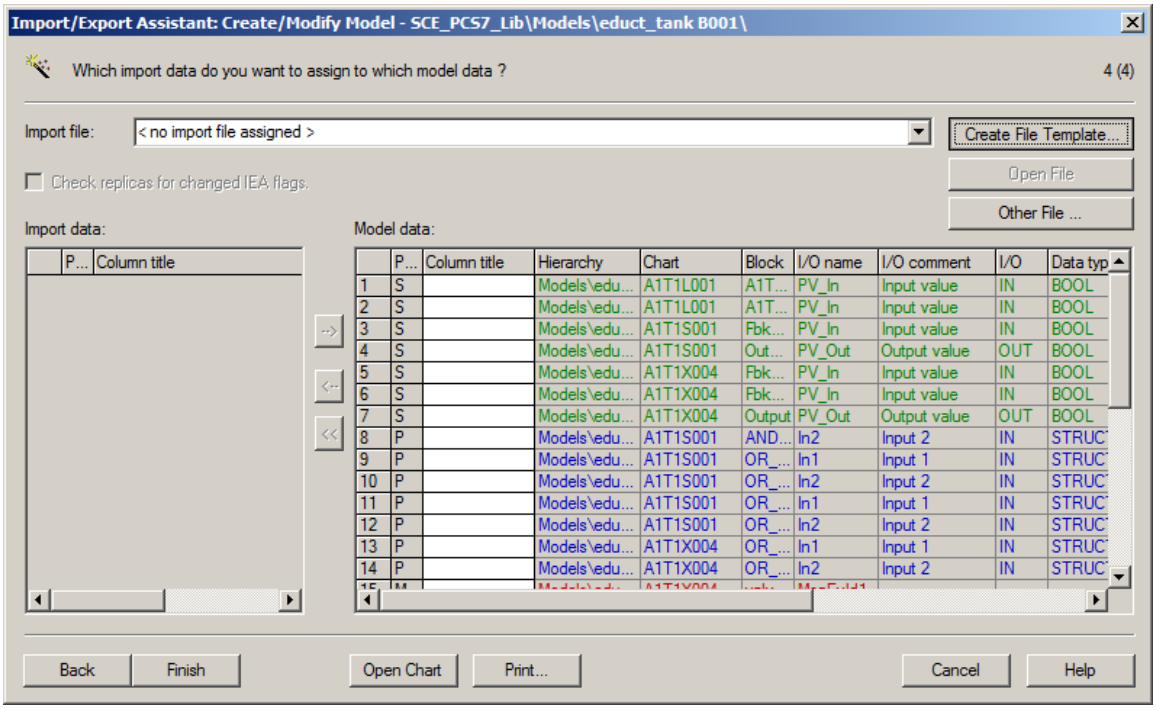
5. The messages that will be displayed in the Import/Export Assistant are then defined. (→ IEA message → Next)



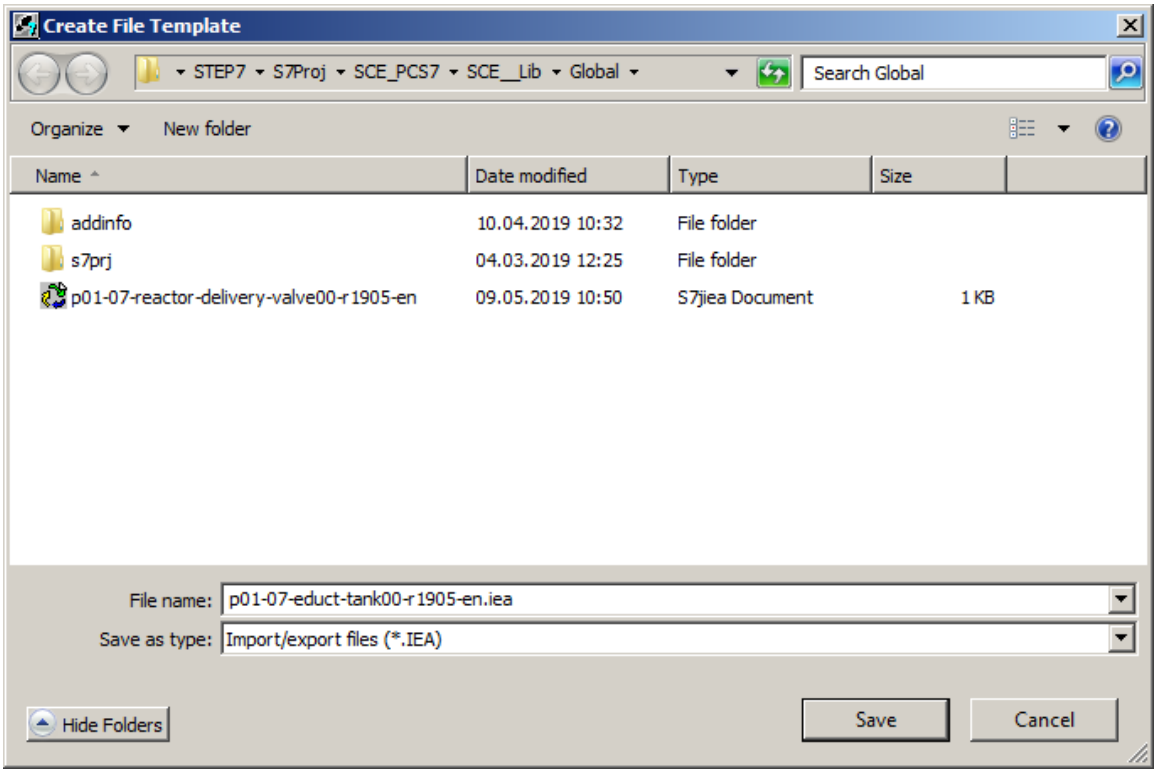
	Hierarchy	Chart	Block	Block comment	IEA message	Block type	Chart t...
1	Models\educt_tank B001\	A1T1X004	valve_A1T1X004	Valve - Large	☑	VlvL	CFC
2	Models\educt_tank B001\	A1T1S001	pump_A1T1S001	Motor - Large	☑	MotL	CFC

8.7 Creating an import file

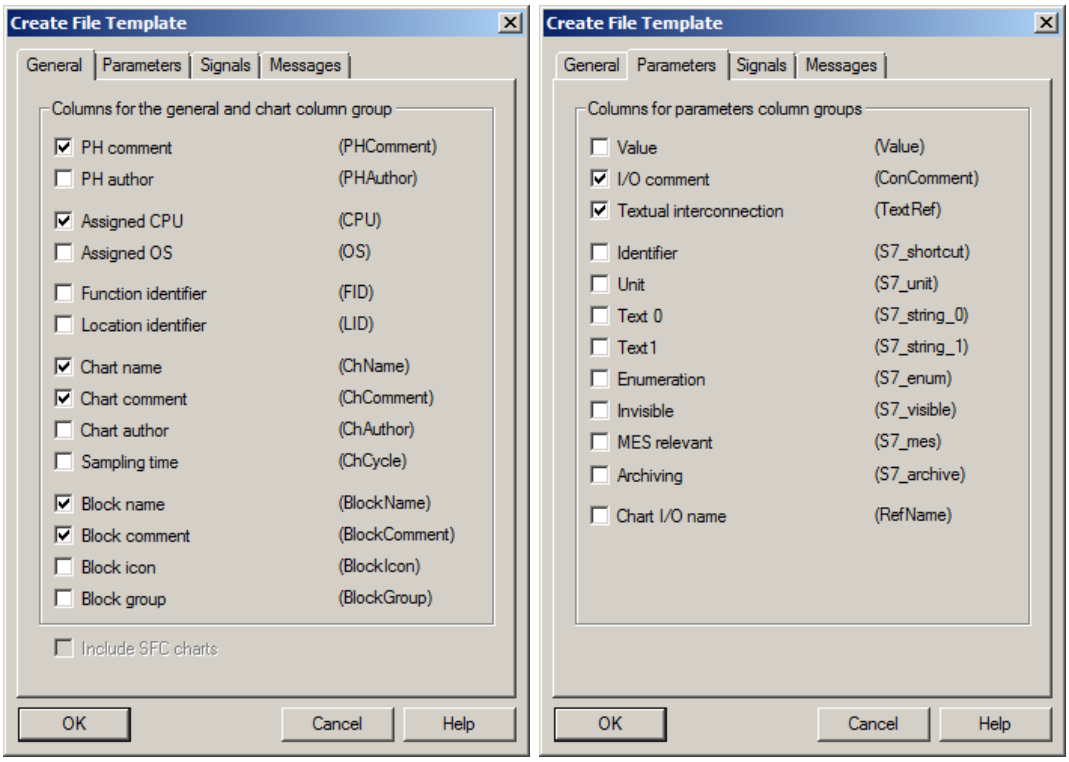
1. Now create a file template. (→ Create File Template…)



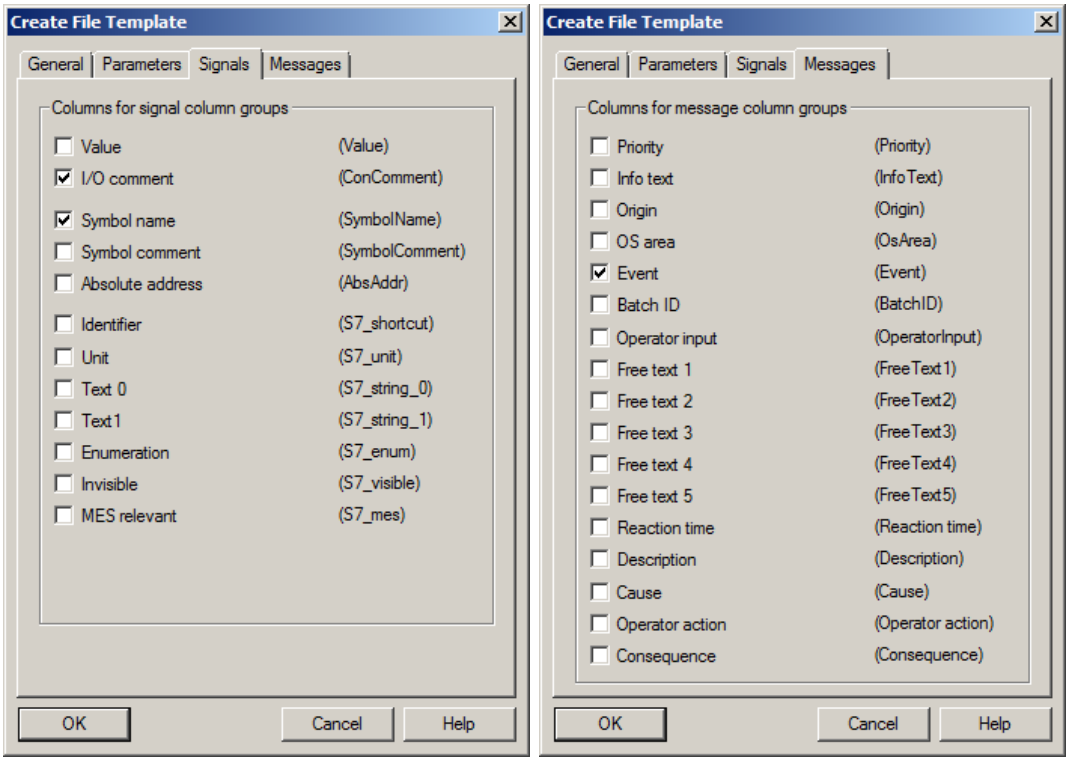
2. Name the file template 'p01-07-educt-tank00-r1905-en.iea'. (→ Save)



3. The next step is to select the columns that will be displayed generally in the import file and for the parameters. (→ Tab: General → PH comment → Assigned CPU → Chart name → Chart comment → Block name → Block comment → Tab: Parameters → I/O comment → Textual interconnection)

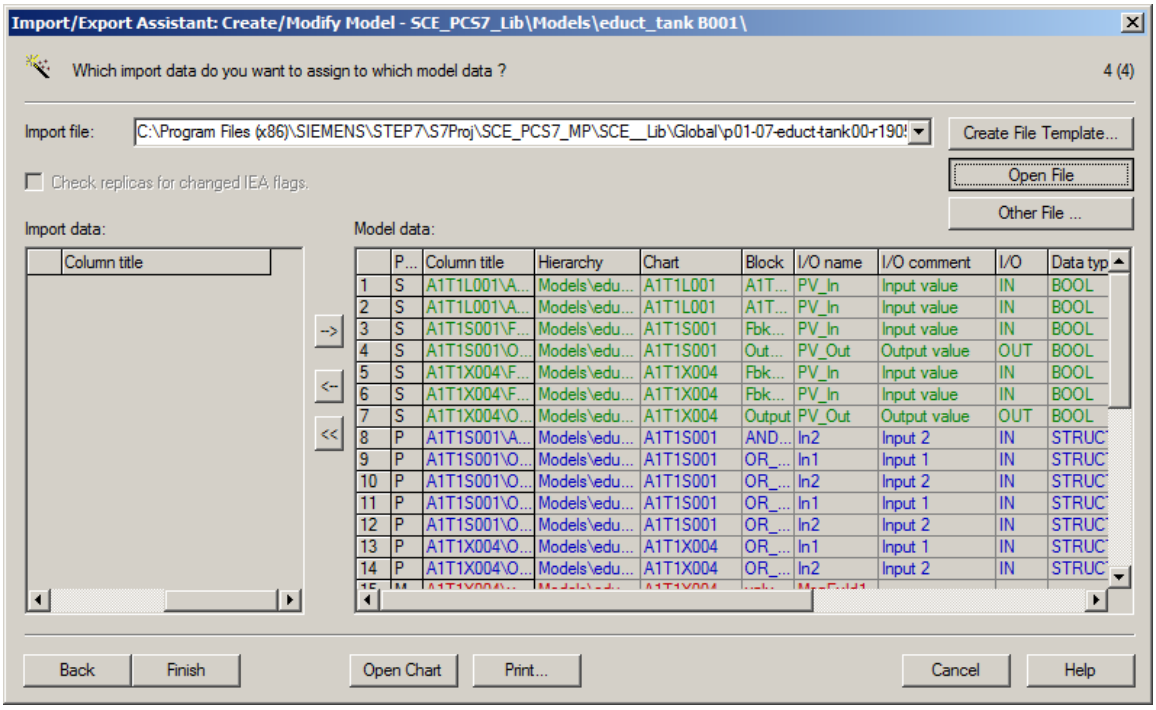


4. The columns that will be displayed for the signals and the messages in the import file are selected here. (→ Tab: Signals → I/O comment → Symbol name → Tab: Messages → Event → OK)



8.8 Editing the import file

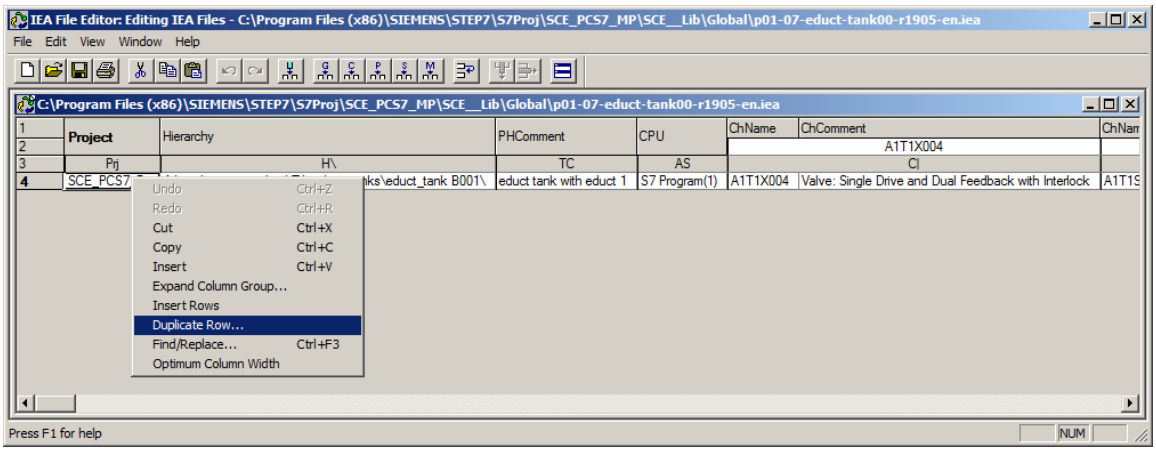
1. Now open the created file. (→ Open File)



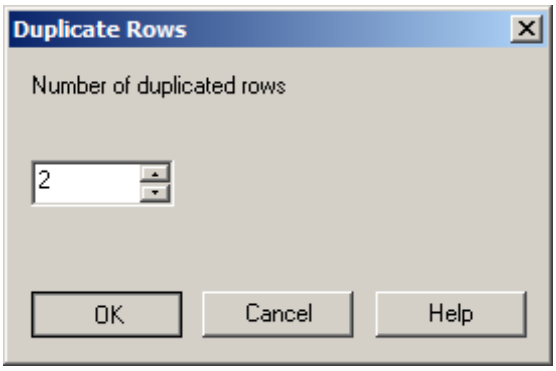
Note:

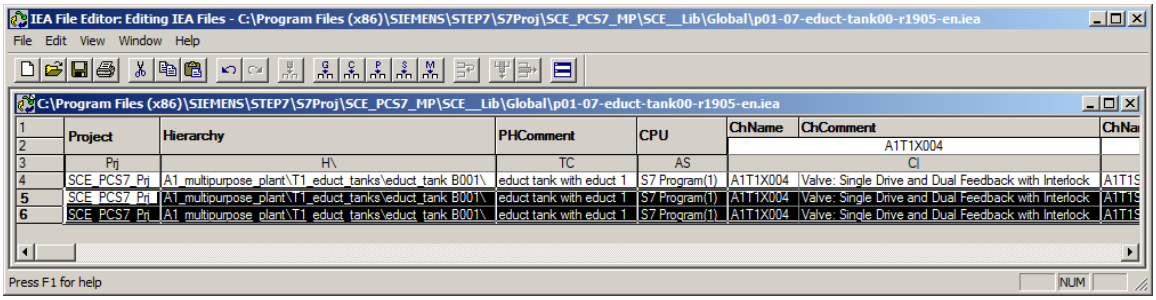
– *As an alternative, the supplied import file 'p01-07-educt-tank00-r1905-en.iea' can be used. To do so, instead of 'Open File', select the 'Other File' button and select the file 'p01-07-educt-tank00-r1905-en.iea'. With that file, the steps below can be skipped. Continue now with step 51.*

2. The first row is duplicated again as many times as models are needed. (→ Duplicate Row)

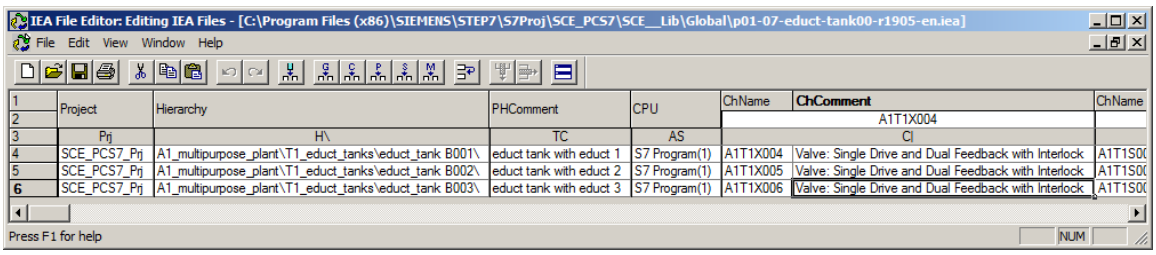


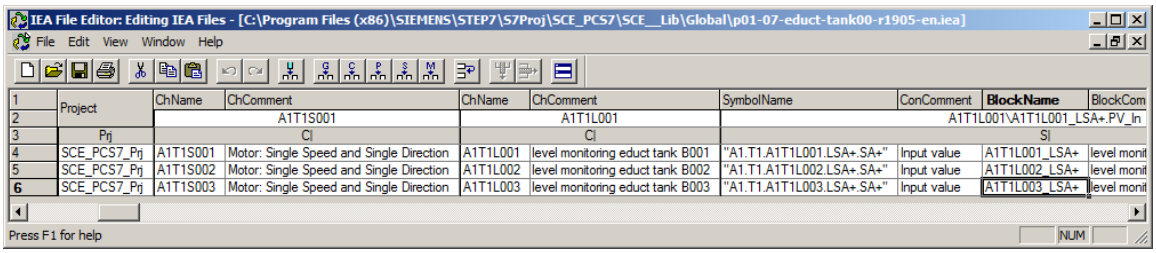
3. Set the number of duplicated rows to '2' and confirm with 'OK'. (→ 2 → OK)

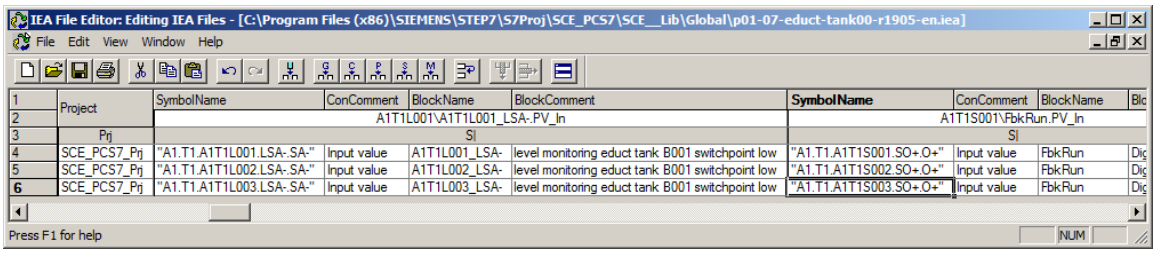




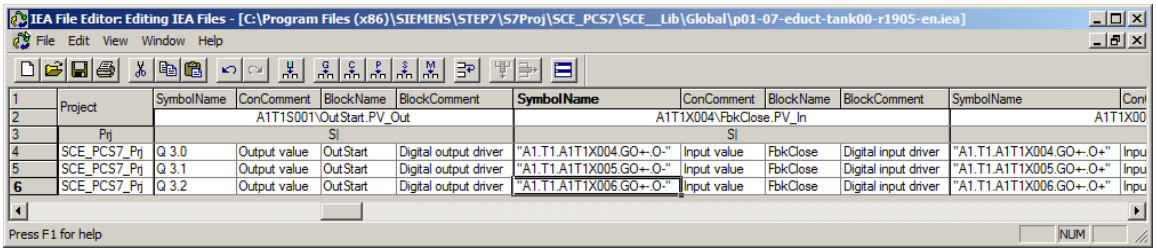
4. Next, change the general information in the Hierarchy and PHComment columns. Then change the ChName and ChComment of the CFCs. For the signals and parameters, you must adapt the SymbolName (inside quotation marks for input signals and as an absolute address for output signals), the BlockName or BlockComment and TextRef.



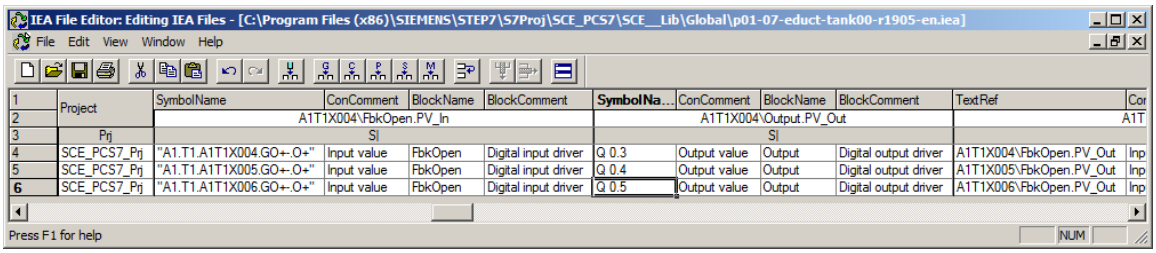




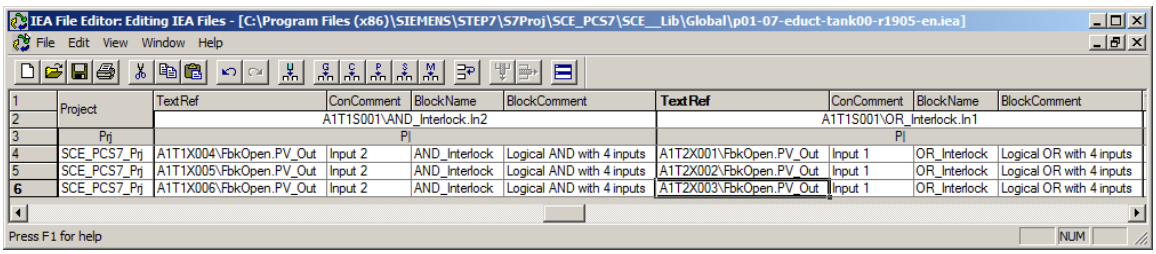
	Project	SymbolName	ConComment	BlockName	BlockComment	SymbolName	ConComment	BlockName	Blc
2		A1T1L001\A1T1L001_LSA-.PV_In				A1T1S001\FbkRun.PV_In			
3	Prj	SI				SI			
4	SCE_PCS7_Prj	"A1.T1.A1T1L001.LSA-.SA-"	Input value	A1T1L001_LSA-	level monitoring educt tank B001 switchpoint low	"A1.T1.A1T1S001.SO+.O+"	Input value	FbkRun	Dig
5	SCE_PCS7_Prj	"A1.T1.A1T1L002.LSA-.SA-"	Input value	A1T1L002_LSA-	level monitoring educt tank B001 switchpoint low	"A1.T1.A1T1S002.SO+.O+"	Input value	FbkRun	Dig
6	SCE_PCS7_Prj	"A1.T1.A1T1L003.LSA-.SA-"	Input value	A1T1L003_LSA-	level monitoring educt tank B001 switchpoint low	"A1.T1.A1T1S003.SO+.O+"	Input value	FbkRun	Dig



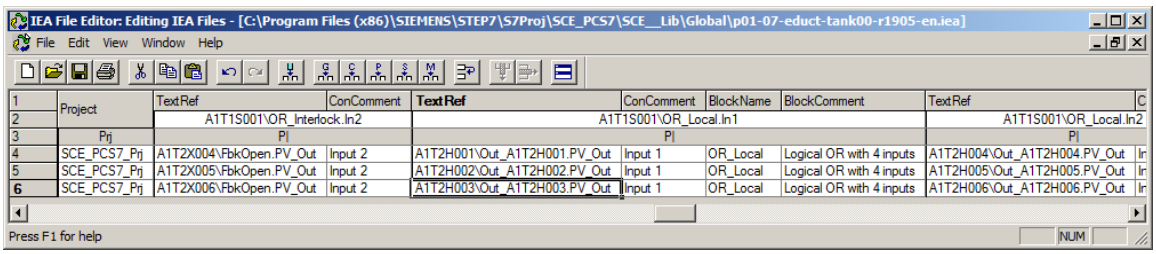
	Project	SymbolName	ConComment	BlockName	BlockComment	SymbolName	ConComment	BlockName	BlockComment	SymbolName	Con
2		A1T1S001\OutStart.PV_Out				A1T1X004\FbkClose.PV_In				A1T1X00	
3	Prj	SI				SI					
4	SCE_PCS7_Prj	Q 3.0	Output value	OutStart	Digital output driver	"A1.T1.A1T1X004.GO+.O-"	Input value	FbkClose	Digital input driver	"A1.T1.A1T1X004.GO+.O+"	Inpu
5	SCE_PCS7_Prj	Q 3.1	Output value	OutStart	Digital output driver	"A1.T1.A1T1X005.GO+.O-"	Input value	FbkClose	Digital input driver	"A1.T1.A1T1X005.GO+.O+"	Inpu
6	SCE_PCS7_Prj	Q 3.2	Output value	OutStart	Digital output driver	"A1.T1.A1T1X006.GO+.O-"	Input value	FbkClose	Digital input driver	"A1.T1.A1T1X006.GO+.O+"	Inpu



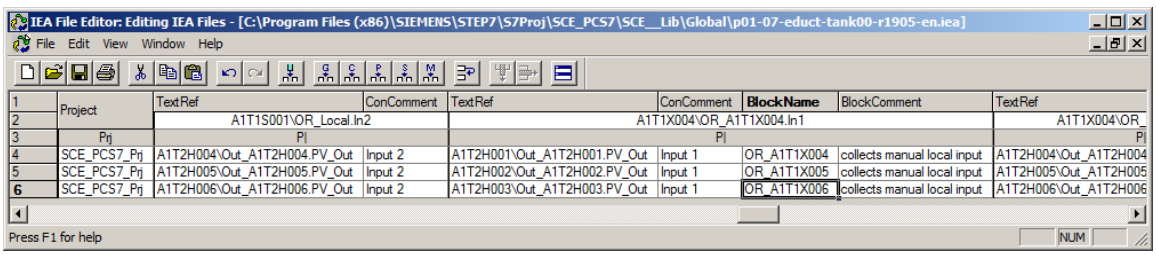
	Project	SymbolName	ConComment	BlockName	BlockComment	SymbolNa...	ConComment	BlockName	BlockComment	TextRef	Cor
2		A1T1X004\FbkOpen.PV_In				A1T1X004\Output.PV_Out					A1T
3	Prj	SI				SI					
4	SCE_PCS7_Prj	"A1.T1.A1T1X004.GO+.O+"	Input value	FbkOpen	Digital input driver	Q 0.3	Output value	Output	Digital output driver	A1T1X004\FbkOpen.PV_Out	Inp
5	SCE_PCS7_Prj	"A1.T1.A1T1X005.GO+.O+"	Input value	FbkOpen	Digital input driver	Q 0.4	Output value	Output	Digital output driver	A1T1X005\FbkOpen.PV_Out	Inp
6	SCE_PCS7_Prj	"A1.T1.A1T1X006.GO+.O+"	Input value	FbkOpen	Digital input driver	Q 0.5	Output value	Output	Digital output driver	A1T1X006\FbkOpen.PV_Out	Inp



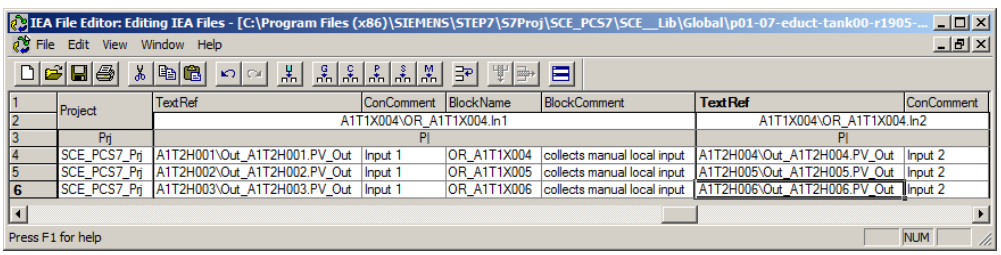
	Project	TextRef	ConComment	BlockName	BlockComment	TextRef	ConComment	BlockName	BlockComment
2		A1T1S001\AND_Interlock.In2				A1T1S001\OR_Interlock.In1			
3	Prj	PI				PI			
4	SCE_PCS7_Prj	A1T1X004\FbkOpen.PV_Out	Input 2	AND_Interlock	Logical AND with 4 inputs	A1T2X001\FbkOpen.PV_Out	Input 1	OR_Interlock	Logical OR with 4 inputs
5	SCE_PCS7_Prj	A1T1X005\FbkOpen.PV_Out	Input 2	AND_Interlock	Logical AND with 4 inputs	A1T2X002\FbkOpen.PV_Out	Input 1	OR_Interlock	Logical OR with 4 inputs
6	SCE_PCS7_Prj	A1T1X006\FbkOpen.PV_Out	Input 2	AND_Interlock	Logical AND with 4 inputs	A1T2X003\FbkOpen.PV_Out	Input 1	OR_Interlock	Logical OR with 4 inputs



	Project	TextRef	ConComment	TextRef	ConComment	BlockName	BlockComment	TextRef	C
2		A1T1S001\OR_Interlock.In2		A1T1S001\OR_Local.In1				A1T1S001\OR_Local.In2	
3	Prj	PI		PI				PI	
4	SCE_PCS7_Prj	A1T2X004\FbkOpen.PV_Out	Input 2	A1T2H001\Out_A1T2H001.PV_Out	Input 1	OR_Local	Logical OR with 4 inputs	A1T2H004\Out_A1T2H004.PV_Out	In
5	SCE_PCS7_Prj	A1T2X005\FbkOpen.PV_Out	Input 2	A1T2H002\Out_A1T2H002.PV_Out	Input 1	OR_Local	Logical OR with 4 inputs	A1T2H005\Out_A1T2H005.PV_Out	In
6	SCE_PCS7_Prj	A1T2X006\FbkOpen.PV_Out	Input 2	A1T2H003\Out_A1T2H003.PV_Out	Input 1	OR_Local	Logical OR with 4 inputs	A1T2H006\Out_A1T2H006.PV_Out	In

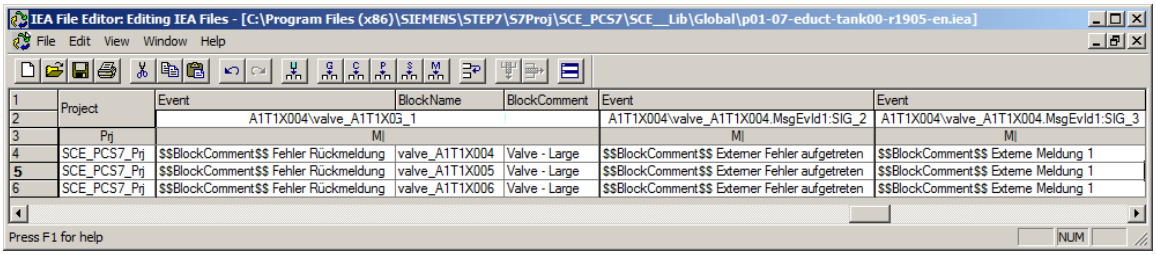


	Project	TextRef	ConComment	TextRef	ConComment	BlockName	BlockComment	TextRef
2		A1T1S001\OR_Local.In2		A1T1X004\OR_A1T1X004.In1				A1T1X004\OR_
3	Prj	PI		PI				PI
4	SCE_PCS7_Prj	A1T2H004\Out_A1T2H004.PV_Out	Input 2	A1T2H001\Out_A1T2H001.PV_Out	Input 1	OR_A1T1X004	collects manual local input	A1T2H004\Out_A1T2H004
5	SCE_PCS7_Prj	A1T2H005\Out_A1T2H005.PV_Out	Input 2	A1T2H002\Out_A1T2H002.PV_Out	Input 1	OR_A1T1X005	collects manual local input	A1T2H005\Out_A1T2H005
6	SCE_PCS7_Prj	A1T2H006\Out_A1T2H006.PV_Out	Input 2	A1T2H003\Out_A1T2H003.PV_Out	Input 1	OR_A1T1X006	collects manual local input	A1T2H006\Out_A1T2H006



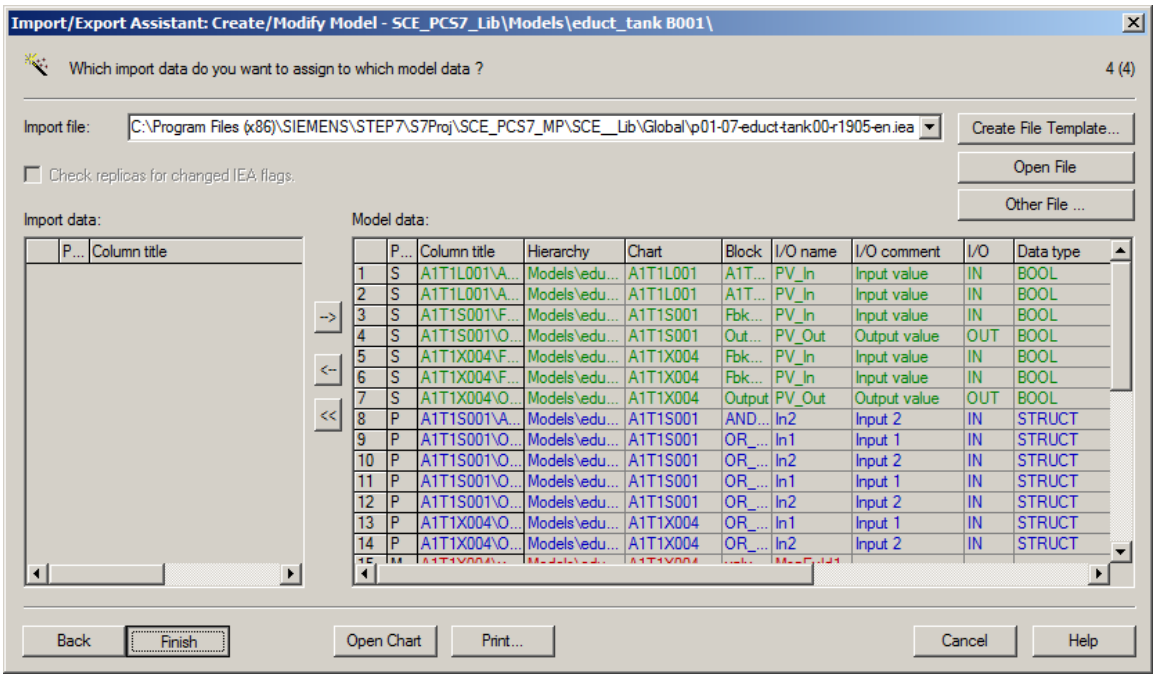
	Project	TextRef	ConComment	BlockName	BlockComment	TextRef	ConComment
2		A1T1X004\OR_A1T1X004.In1				A1T1X004\OR_A1T1X004.In2	
3	Prj	PI				PI	
4	SCE_PCS7_Prj	A1T2H001\Out_A1T2H001.PV_Out	Input 1	OR_A1T1X004	collects manual local input	A1T2H004\Out_A1T2H004.PV_Out	Input 2
5	SCE_PCS7_Prj	A1T2H002\Out_A1T2H002.PV_Out	Input 1	OR_A1T1X005	collects manual local input	A1T2H005\Out_A1T2H005.PV_Out	Input 2
6	SCE_PCS7_Prj	A1T2H003\Out_A1T2H003.PV_Out	Input 1	OR_A1T1X006	collects manual local input	A1T2H006\Out_A1T2H006.PV_Out	Input 2

5. Finally, leave the messages unchanged. Save the file and close the editor. (→ Save →)



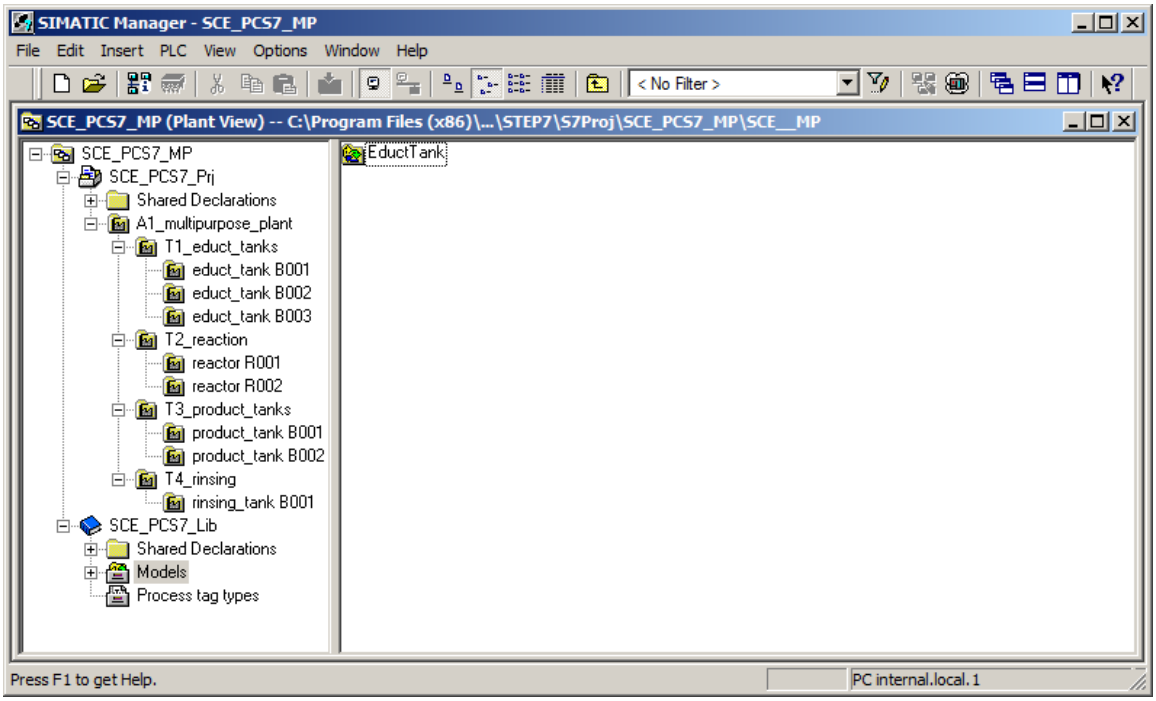
	Project	Event	BlockName	BlockComment	Event	Event
2		A1T1X004\valve_A1T1X03_1			A1T1X004\valve_A1T1X004.MsgEvId1:SIG_2	A1T1X004\valve_A1T1X004.MsgEvId1:SIG_3
3	Prj	MI			MI	MI
4	SCE_PCS7_Prj	$$BlockComment$$ Fehler Rückmeldung	valve_A1T1X004	Valve - Large	$$BlockComment$$ Externer Fehler aufgetreten	$$BlockComment$$ Externe Meldung 1
5	SCE_PCS7_Prj	$$BlockComment$$ Fehler Rückmeldung	valve_A1T1X005	Valve - Large	$$BlockComment$$ Externer Fehler aufgetreten	$$BlockComment$$ Externe Meldung 1
6	SCE_PCS7_Prj	$$BlockComment$$ Fehler Rückmeldung	valve_A1T1X006	Valve - Large	$$BlockComment$$ Externer Fehler aufgetreten	$$BlockComment$$ Externe Meldung 1

6. The Assistant is exited with 'Finish'. (→ Finish)



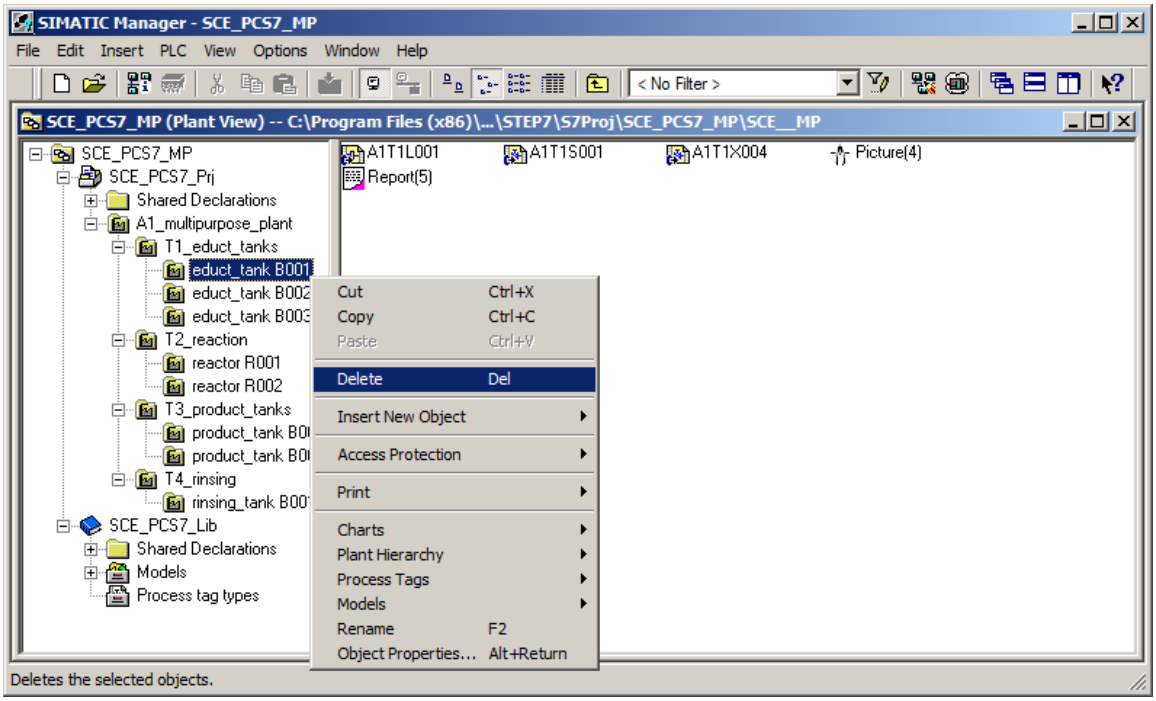
	P...	Column title	Hierarchy	Chart	Block	I/O name	I/O comment	I/O	Data type
1	S	A1T1L001\A...	Models\edu...	A1T1L001	A1T...	PV_In	Input value	IN	BOOL
2	S	A1T1L001\A...	Models\edu...	A1T1L001	A1T...	PV_In	Input value	IN	BOOL
3	S	A1T1S001\F...	Models\edu...	A1T1S001	Fbk...	PV_In	Input value	IN	BOOL
4	S	A1T1S001\O...	Models\edu...	A1T1S001	Out...	PV_Out	Output value	OUT	BOOL
5	S	A1T1X004\F...	Models\edu...	A1T1X004	Fbk...	PV_In	Input value	IN	BOOL
6	S	A1T1X004\F...	Models\edu...	A1T1X004	Fbk...	PV_In	Input value	IN	BOOL
7	S	A1T1X004\O...	Models\edu...	A1T1X004	Output	PV_Out	Output value	OUT	BOOL
8	P	A1T1S001\A...	Models\edu...	A1T1S001	AND...	In2	Input 2	IN	STRUCT
9	P	A1T1S001\O...	Models\edu...	A1T1S001	OR_...	In1	Input 1	IN	STRUCT
10	P	A1T1S001\O...	Models\edu...	A1T1S001	OR_...	In2	Input 2	IN	STRUCT
11	P	A1T1S001\O...	Models\edu...	A1T1S001	OR_...	In1	Input 1	IN	STRUCT
12	P	A1T1S001\O...	Models\edu...	A1T1S001	OR_...	In2	Input 2	IN	STRUCT
13	P	A1T1X004\O...	Models\edu...	A1T1X004	OR_...	In1	Input 1	IN	STRUCT
14	P	A1T1X004\O...	Models\edu...	A1T1X004	OR_...	In2	Input 2	IN	STRUCT

7. The newly created model is located in the project library in the Models folder. Here, the created model is renamed to 'EductTank'.

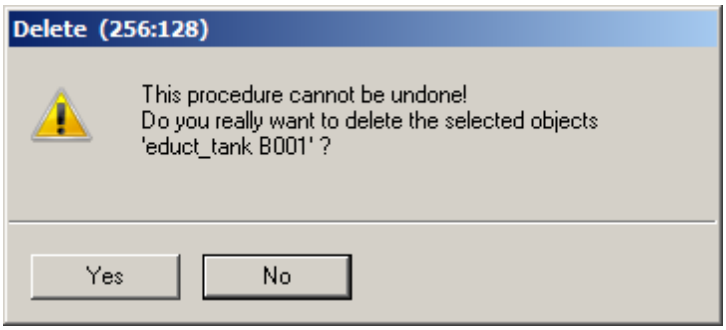


8.9 Importing a model

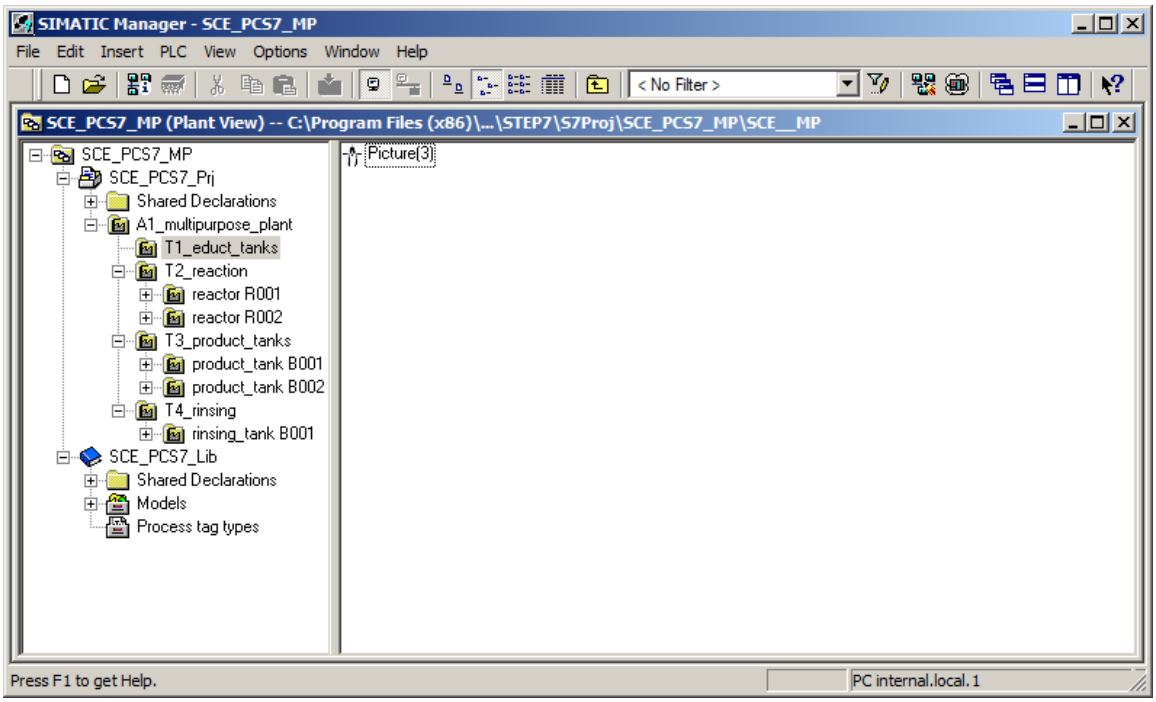
1. Before starting the import, you must delete the hierarchy folders B001 to B003 including the CFCs they contain. (→ educt_tank B00x → Delete)



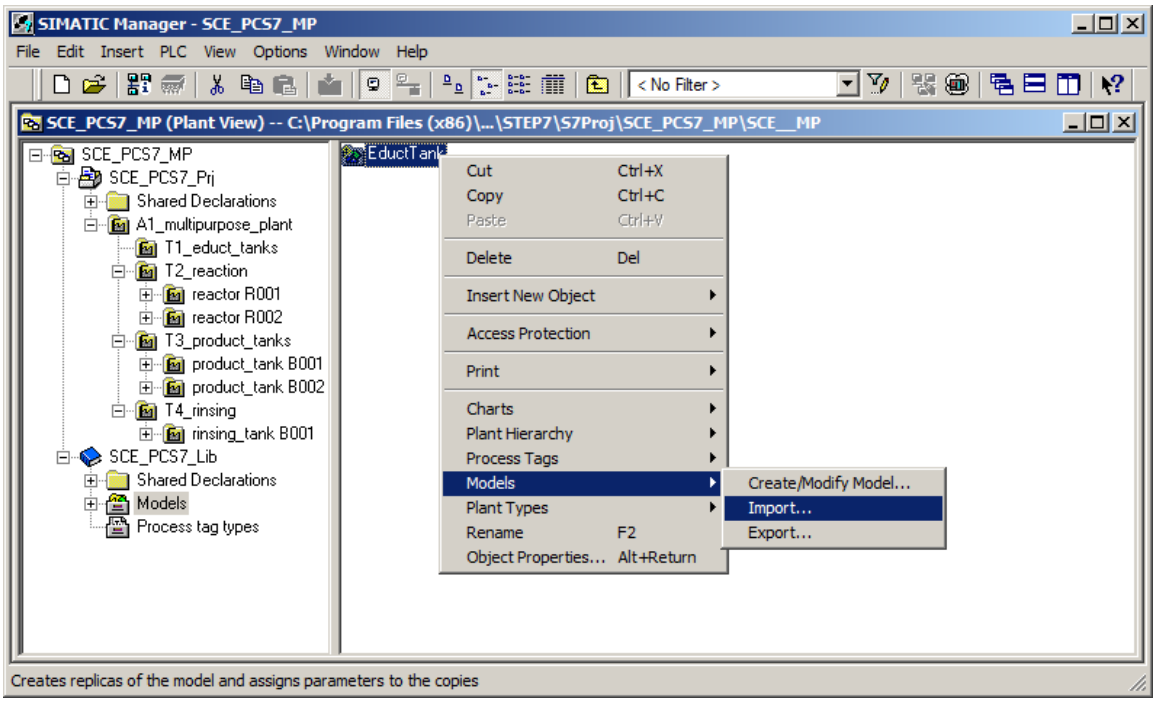
2. Confirm the warning with 'Yes'. (→ Yes)



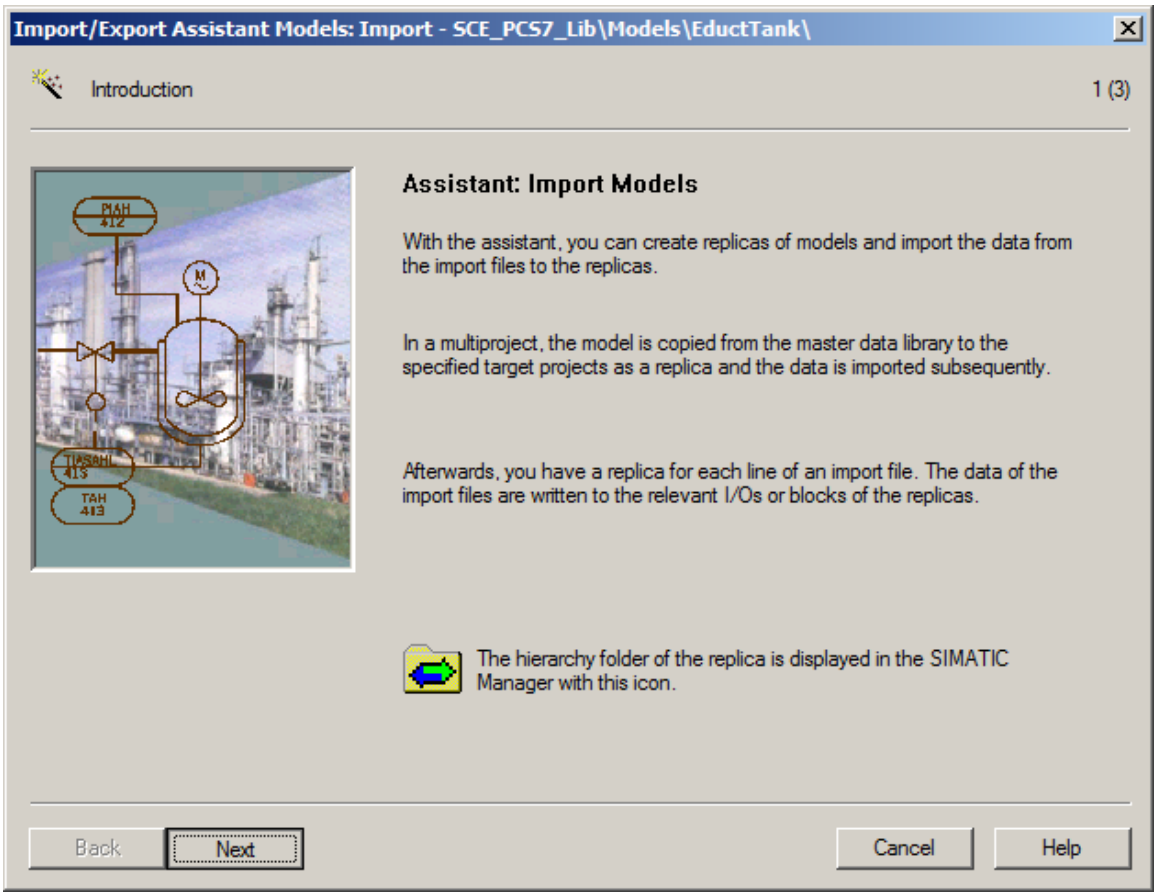
3. After the deletion, the plant hierarchy looks like this.



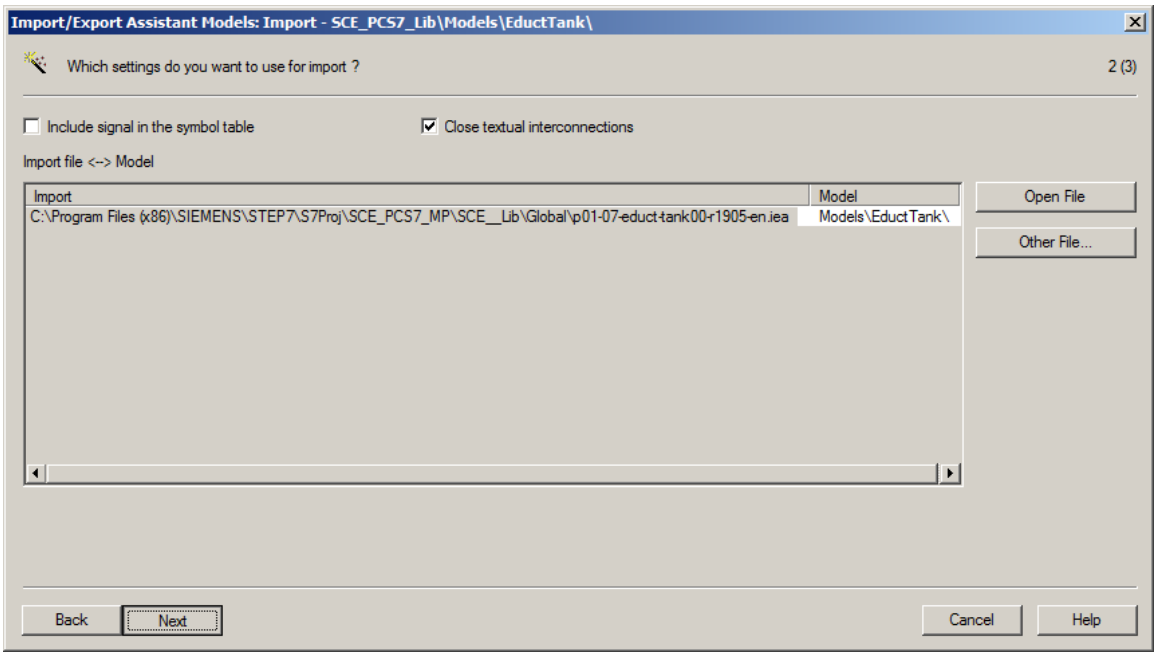
4. You can then start the import of the model. (→ EductTank → Models→ Import…)



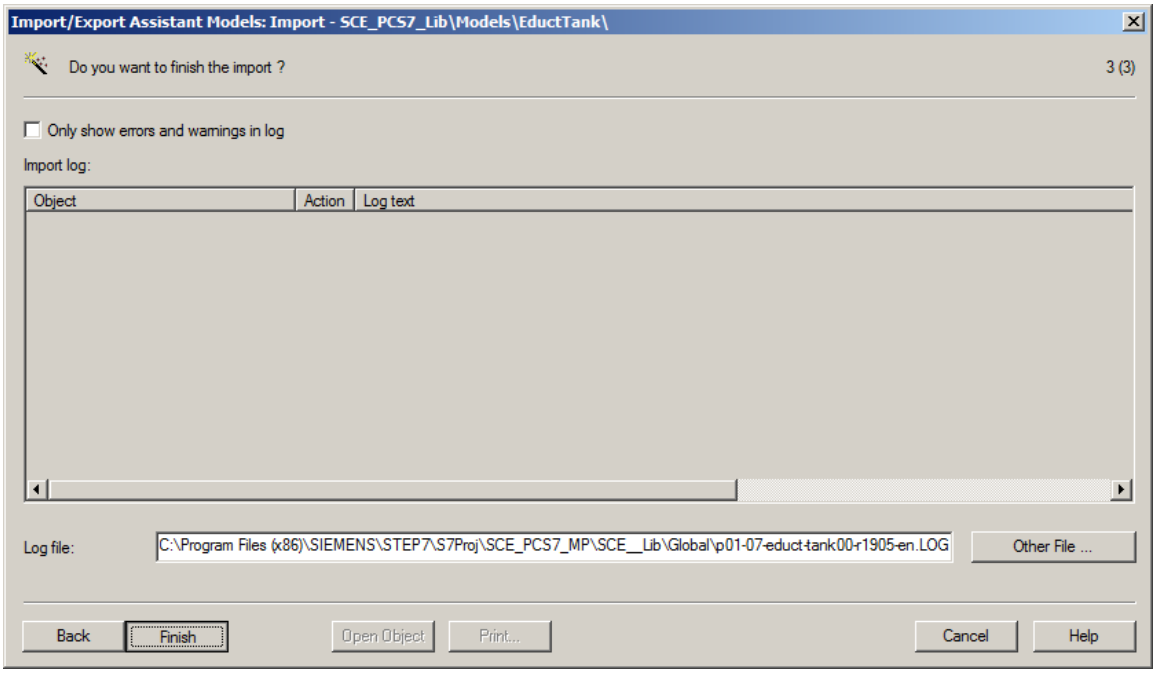
5. Confirm the start screen of the Import/Export Assistant with 'Next'. (→ Next)



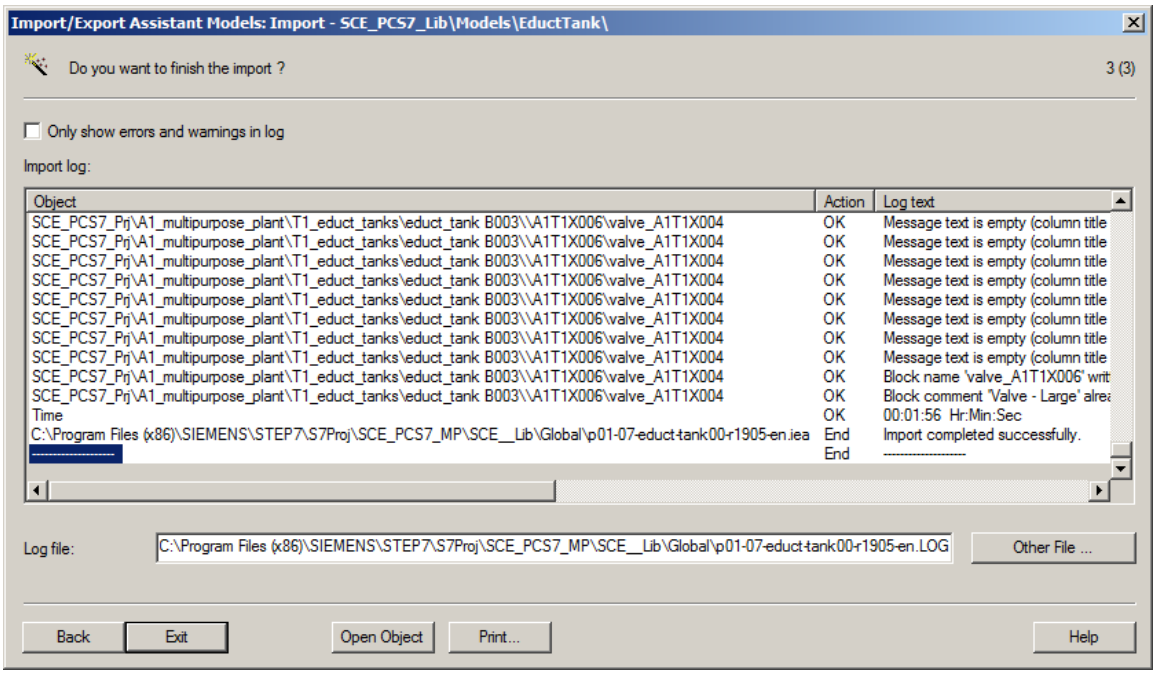
6. Select the 'Close textual interconnections' check box and click 'Next'. (→ Close textual interconnections → Next)



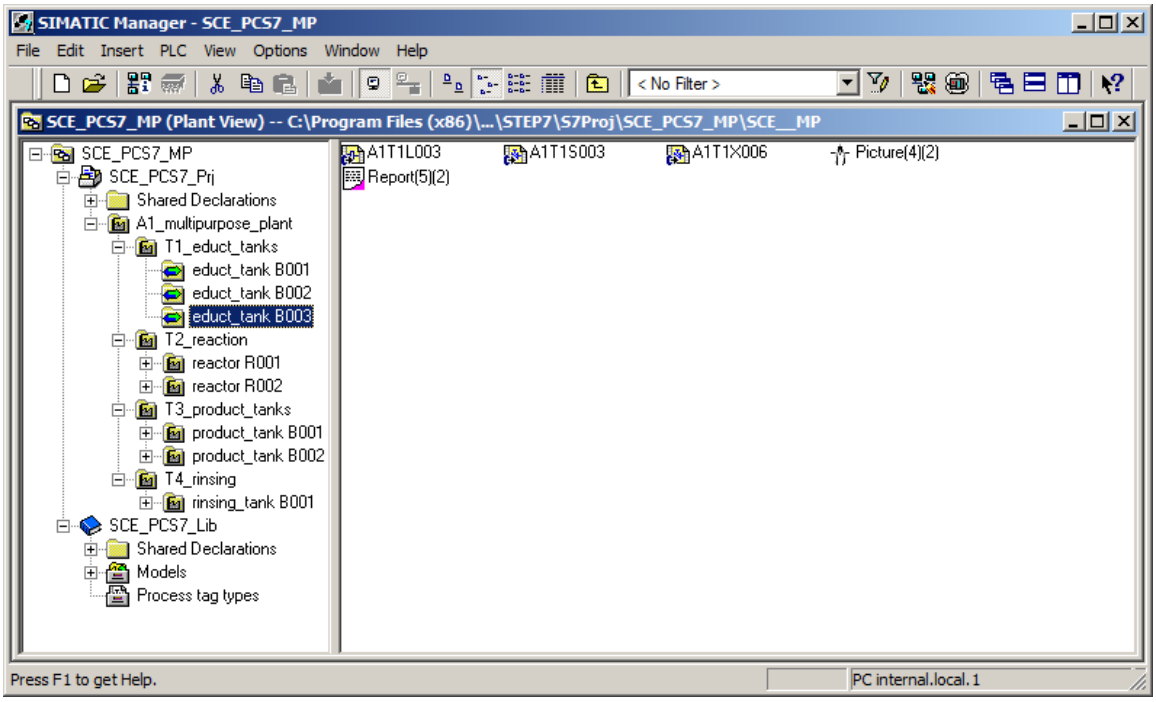
7. The assistant is now finished and the import is started. (→ Finish)



8. The import log is created again and the result is displayed. (→ Exit)



9. The imported models are now present in the plant hierarchy.



10. Check to see if the textual interconnections with the existing CFCs are closed.

Input:	Textual interconnection:	Inverted
MotL.Pumpe_A1T1S001.LocalLi	A1H003\A1H003.PV_Out	No
Intlk02.Permit.In01	A1H001\A1H001.PV_Out	No
Intlk02.Protect.In01	A1H002\A1H002.PV_Out	No
Or04.Or_Interlock.In1	A1T2X001\FbkOpen.PV_Out	No
Or04.Or_Interlock.In2	A1T2X004\FbkOpen.PV_Out	No
Or04.Or_Local.In1	A1T2H001\Out_A1T2H001.PV_Out	No
Or04.Or_Local.In2	A1T2H004\Out_A1T2H004.PV_Out	No

Table 2: Textual interconnections in chart 'A1T1S001'

Input:	Textual interconnection:	Inverted
VlvL.Ventil_A1T1X004.LocalLi	A1H003\A1H003.PV_Out	No
Intlk02.Permit.In01	A1H001\A1H001.PV_Out	No
Intlk02.Protect.In01	A1H002\A1H002.PV_Out	No
Or04.Or_Local.In1	A1T2H001\Out_A1T2H001.PV_Out	No
Or04.Or_Local.In2	A1T2H004\Out_A1T2H004.PV_Out	No

Table 3: Textual interconnections in chart 'A1T1X004'

Input:	Textual interconnection:	Inverted
Or08.Or_A1T2H001.In7	A1T1L001\A1T1L001_LSA-.PV_Out	Yes

Table 4: Textual interconnections in chart 'A1T2H001'

8.10 Checklist – step-by-step instruction

The following checklist helps students to independently check whether all steps of the step-by-step instruction have been carefully completed and enables them to successfully complete the module on their own.

No.	Description	Checked
1	Process tag type ReactorDeliveryValve created	
2	Process tag type ReactorDeliveryValve successfully imported	
3	Imported valve CFCs successfully tested (optional)	
4	Interlocking of pump A1T1S001 complete (no textual interconnections)	
5	EductTank model created	
6	EductTank model successfully imported	
7	Textual interconnections in imported models are closed	
8	Imported models successfully tested (optional)	
9	Project successfully archived	

Table 5: Checklist for step-by-step instructions

9 Exercises

In the exercises, you apply what you learned in the theory section and in the step-by-step instructions. The existing multiproject from the step-by-step instructions (p01-07-project-r1905-en.zip) is to be used and expanded for this. The download of the project is stored as zip file "Projects" on the SCE Internet for the respective module.

The tasks in this exercise will add all the objects not implemented so far to the plant. It is up to you where you want to utilize the tools for importing plant design data. Effective utilization of the import of plant design data not only depends on the plant structure but also on the mapping of this structure in the plant hierarchy. With some practice, you will deepen your knowledge regarding useful plant designations and the structure of the plant hierarchy.

Note:

– *The generated import files are stored in zip file 'p01-07-files-r1905-en.zip'. However, these files may not work if there are discrepancies in the created process tag type or model.*

9.1 Tasks

1. Complete the following CFCs in Reactor R001:

– A1T2H002 and A1T2H003

– A1T2H013 and A1T2H015

– A1T2X007.

2. Check open textual interconnections between the manual controls in the reactor and other CFCs in Reactor R001. To do so, you can also use the 'Close textual interconnections' function under Options in the CFC Editor, because it will show you the interconnections that could not yet be closed. Double-click or use the 'Go to' button to select an interconnection that is still open and correct it manually.

Note:

– *Not all open textual interconnections can be closed here. Most important are the connections within Reactor R001.*

3. Now, create a model of Reactor R001. Delete the Reactor R002 folder and import the model. Reactor R001 is automatically skipped because the folder already exists. If you delete the folder, it will also be generated from the model.

4. Next, create a model of Product Tank B001. Delete at least the product_tank B002 folder and import the model.

5. Now create the missing CFCs for the rinsing tank:

– A1T4L001

– A1T4S001

– A1T4X001, A1T4X002, A1T4X003 and A1T4X004.

6. Interconnect the manual control for rinsing in such a way that the rinsing water flows from the rinsing tank into the reactor and then directly back into the rinsing tank.

7. Check whether textual interconnections are still open and close them if necessary.

8. Finally, check all CFCs for correct designations and correct connections. For the former, it is best to utilize the process object view. Always select one CFC in the left window while you check the name of the blocks in the 'Blocks' tab in the right window. To look for errors, however, you should use the simulation.

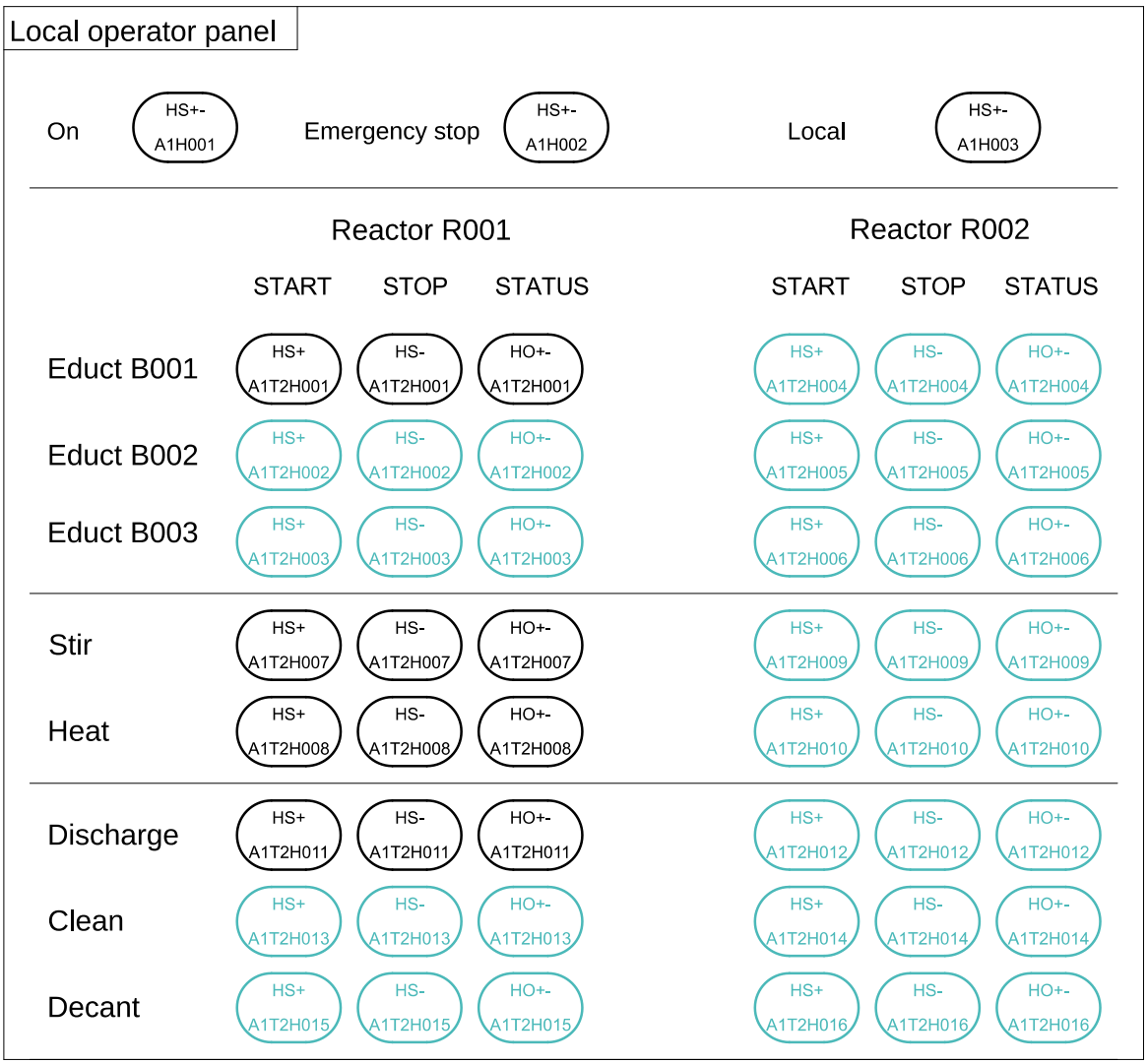


Figure 5: Excerpt from the local operator station

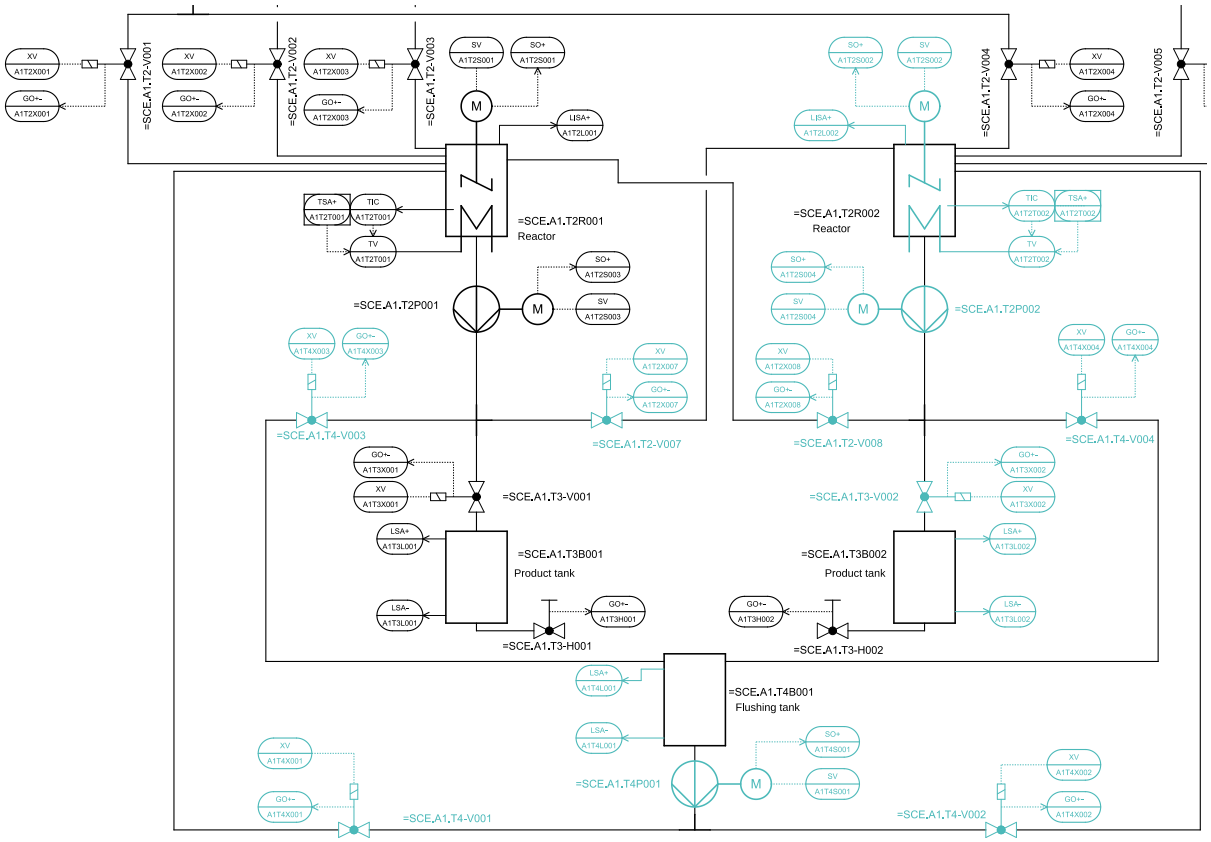


Figure 6: Excerpt from P&ID flowchart

9.2 Checklist – exercise

The following checklist helps students to independently check whether all steps of the exercise have been carefully completed and enables them to successfully complete the module on their own.

No.	Description	Checked
1	CFCs for A1T2H002, A1T2H003, A1T2H013, A1T2H015, A1T2X007 in 'reactor R001' complete	
2	Open textual interconnections in CFCs in 'reactor R001' checked	
3	CFCs in 'reactor R002' complete	
4	CFCs in 'product_tank B002' complete	
5	CFCs for A1T4L001, A1T4S001, A1T4X001, A1T4X002, A1T4X003, A1T4X004 in 'rinsing_tank B001‘ complete	
6	Manual control for rinsing correct	
7	Open textual interconnections all closed	
8	Block names in all CFCs correct	
9	Blocks successfully tested (optional)	
10	Project successfully archived	

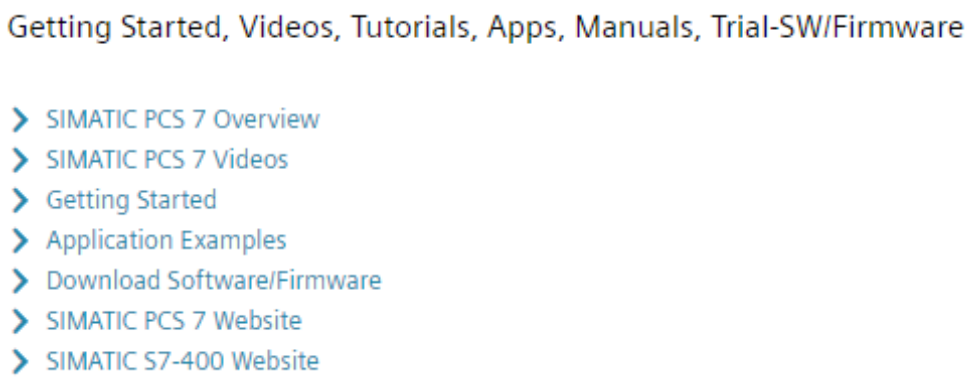
Table 6: Checklist for exercises

10 Additional information

More information for further practice and consolidation is available as orientation, for example: Getting Started, videos, tutorials, apps, manuals, programming guidelines and trial software/ firmware, under the following link:

siemens.com/sce/pcs7

Preview "Additional information"



Further Information

Siemens Automation Cooperates with Education
siemens.com/sce

Siemens SIMATIC PCS 7
siemens.com/pcs7

SCE Learn-/Training Documents
siemens.com/sce/documents

SCE Trainer Packages
siemens.com/sce/tp

SCE Contact Partners
siemens.com/sce/contact

Digital Enterprise
siemens.com/digital-enterprise

Industrie 4.0
siemens.com/future-of-manufacturing

Totally Integrated Automation (TIA)
siemens.com/tia

TIA Portal
siemens.com/tia-portal

SIMATIC Controller
siemens.com/controller

SIMATIC Technical Documentation
siemens.com/simatic-docu

Industry Online Support
support.industry.siemens.com

Product catalogue and online ordering system Industry Mall
mall.industry.siemens.com

Siemens
Digital Industries, FA
P.O. Box 4848
90026 Nuremberg
Germany