IMPORTING PLANT DESIGN DATA

OBJECTIVE

After working through this instruction module, students know how to identify recurring structures, and how to design templates. They know the difference between a process tag type and a model. They are able to generate and implement both. This allows the students to implement many similar process tags or units in ***PCS7***. They get to know the process object view and are able to use it in order to represent parameters plant-wide, and to change them if needed.

THEORY IN SHORT

In process engineering plants there are always recurring objects and structures that behave in the same way, are interfaced in process control in the same way, and are to be visualized in the same way.

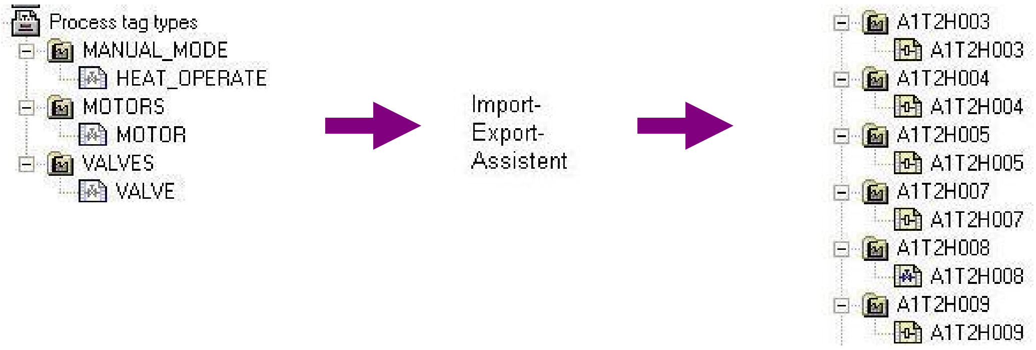


Figure 1: From process tag type to replicas

Such objects can be stored in the project's own library as ***process tag types***. A process tag type is a single CF chart. As shown in Figure 1, one process tag type is able to generate numerous process tags as a duplicate in a single process by using the import-export assistant. This process is controlled in an import file. It is then possible to manually adapt and wire the process tags according to their additional specific automation tasks.

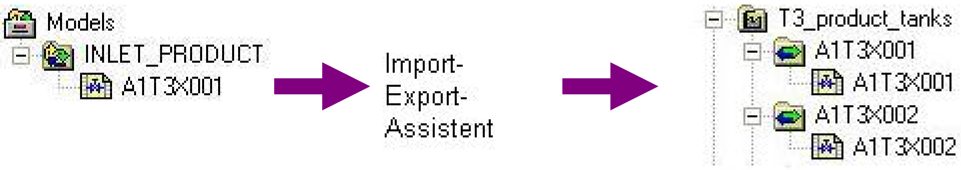


Figure 2: From model to replica

With ***models*** we define more complex functions than with process tag types (up to complete units). A model consists of hierarchy folders containing CFCs and SFCs, displays, reports, and supplementary documents. The entire structure can be stored in the project's own library as a reusable template. By using the import-export assistant it is possible to generate -based on an import file- from a model numerous replicas in a single process as a duplicate (refer to Figure 2). Then, the replicas are adapted to the respective automation task.

In the ***PCS7*** libraries, extensive templates are already stored as process tag types. If a template is to be used multiple times, the template is copied from the ***PCS7*** library to the project's library, adapted if needed and copied by means of the import-export assistant based on an import file.

THEORY

When designing an automation system with ***PCS7***, we can resort to general design principles for complex systems; principles that have proven themselves multiple times [1]. The most important ones are:

– The principle of hierarchical arrangement

– The principle of modularization

– The principle of re-use.

The *principle of hierarchical arrangements* was already used for structuring the plant in the chapter 'Plant Hierarchy' By subdividing units that can be processed largely independent of each other, first a huge design task that seems highly involved is broken down into sub-problems that can be grasped and planned.

The *principle of modularization* means that the system to be designed is to consist of constituent parts (here, blocks, CFCs, SFCs) that are to have the following properties:

– Its scope is to be clearly laid out and easy to follow

– Largely autonomous functions that can be checked

– As few relationships to other constituent parts as possible

– Defined interfaces to other constituent parts

This results in two rivaling complexity aspects when breaking down an automation solution into individual parts:

– Low complexity of the constituent parts: the more parts, the smaller the individual parts are, and easier to follow.

– High external complexity of the constituent parts: the more parts, the higher the number of connections between them.

Hierarchical arrangement and modularization are dependent on one another. While hierarchical arrangements are determined more by the process engineering plant, modularization is dominated by the process control implementation. Based on the countercurrent complexity aspects mentioned above, and the great dependence on the concrete process and automation engineering task, early coordination of both implementations is of advantage.

Through the plant hierarchy, ***PCS7*** supports the principle of hierarchical arrangements. The principle of modularization and re-use is implemented in ***PCS7*** with duplication processing.

In larger projects or recurring similar projects, often numerous identical or at least very similar objects and structures can be observed. To save time and outlay for configuration tasks, it is advisable for that reason to plan the specific search for suitable, recurring objects and structures during the concept determination and design phase of an automation project. After such objects and structures are identified, first, generic solutions are implemented and tested that can then be used for numerous identical or similar objects and structures. Based on the factors below, the initial additional outlay required to prepare the generic solution (here also called type or template) should lead to appreciable time and cost savings over the duration of the project.

– A type can be implemented multiple times; that is, it has several replicas

– By using a type in several replicas, several tests are performed at he same time

– If errors should occur or changes should be necessary, only the generic solution has to be adapted and all replicas have to be updated.

In addition, existing objects and structures from earlier projects and libraries can be reused. They have the advantage of being already tried and are largely faultless. By re-using proven parts, generally the reliability of a new automation solution also increases.

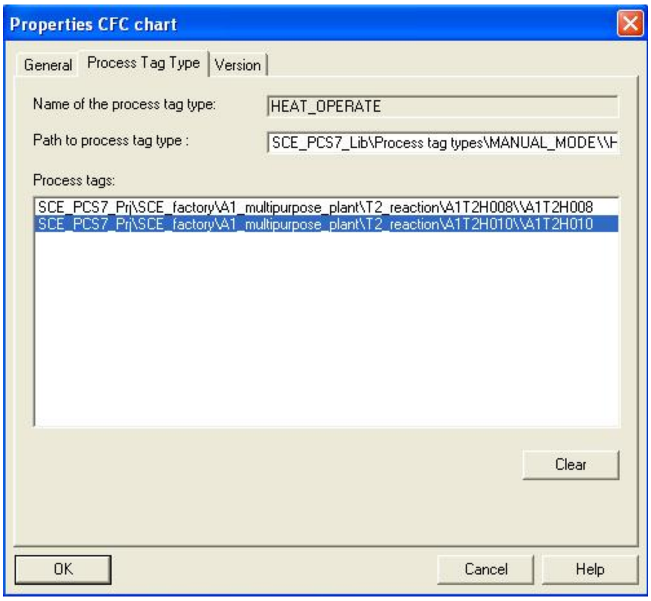
PROCESS TAG TYPE

Process tag types are used as a generic solution if a project contains many process tags of the same kind [2].

First, a CF chart is set up that contains all internal blocks and their connections. All input and output parameters are clearly defined as parameters or signals. From this CF chart now containing all generally valid parameters, a process tag type is generated. In an import file, all process tag specific parameters where the replicas differ are then specified.

During the import process, the import-export assistant generates the replicas of the process tag type in the specified hierarchy folders. If the hierarchy should not be available as yet, it is set up also. Each replica is an instance of the process tag type and has its properties.

In ***PCS7***, the process tags (replicas) generated in this way can also be adapted specifically by adding different locking mechanisms, for example. Under certain conditions, they are not overwritten if there is another import.



Name of the process tag type: HEAT_OPERATE

Path to process tag type : SCE_PCS7_Lib\Process tag types\MANUAL_MODE\\H

Process tags:

SCE_PCS7_Prj\SCE_factory\A1_multipurpose_plant\T2_reaction\A1T2H008\\A1T2H008
SCE_PCS7_Prj\SCE_factory\A1_multipurpose_plant\T2_reaction\A1T2H010\\A1T2H010

Figure 3: Replica A1T2H003 of process tag type FILL_REAKTOR_H

The following must not be changed at the process tags that were generated:

- Specific adaptations at the block connections that are parameterized by means of the import file. These adaptations are overwritten during a new import process with the parameters that are specified in the import file.

With process tag types, subsequent changes can be carried out simply by making the changes at the process tag type and in the import file. Then, the modified data is transmitted to all generated process tags with a new import process. The following changes are conceivable:

- Adding another parameter and assigning this parameter by means of the import file
- Deleting all generated process tags of a process tag type (without manual deletion in the plant hierarchy)
- Adding another block connection and parameterizing the block connection by means of the import file.

MODEL

The model is used as a generic solution if the project has structures that are alike.

As a rule, a plant is structured by dividing it into smaller functional units whose interfaces, performance and logic can be clearly described; for example a tank with its instrumentation. Instead of implementing these functional units anew each time, an inventory of preassembled function units (models) can be set up.

So that a model is used in only one version 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 with the following elements:

– CFCs/SFCs

– OS displays

– OS reports

– Supplementary documents

After a model was configured and an import file was assigned to it, replicas can be generated by means of the import process. The following steps are carried out automatically:

Step 1: The hierarchy path from the column 'Hierarchy' of the first data line of the import file is read. A check is performed whether this path already exists. Additional actions depend on the results of the check:

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

– If the hierarchy folder exists and if it is suitable as a replica of the model, it is -together with its CFC- made into a replica of the model, and parameterized according to the import file.

– If the hierarchy folder does not exist, it is set up, a replica of the model is generated and parameterized correspondingly.

Step 2: The following elements are inserted in the tile block of the charts if the columns are available:

– Function identifier (FKZ)

– Location designation (OKZ)

– CFC name

– Chart comment

Step 3: Texts and values of the parameter descriptions and the wiring descriptions (signals) are written to the corresponding block and chart connections of the replicas.

Note: A connection is cleared if the signal name (symbol or textual connection) consists of the code word ’---’ (three dashes).

A connection remains unchanged if no connection name (symbol or textual connection) is specified.

Step 4: The data types of the connections for signals are ascertained and assigned to the connections.

Note: The following applies to connections with global operands: If the option ’Enter signal also in symbol table’ is set, the names are searched for in the symbol table of the model’s resource.

It is not advisable to use this option for ***PCS7***, since these entries are made in ***HWConfig*** when the hardware is configured.

Please note the following rules:

- The symbol name exists in the symbol table:

  The data type has to be the same, the symbol name must exist only once. The data type is parameterized according to the block/chart connection. The absolute address is overwritten and for the symbol, the symbol comment is entered (if it exists in the import file). Only what has changed is overwritten; existing attributes are retained.

- The symbol name does not yet exist in the symbol table:

  The connection is set up and the data type is parameterized according to the connection. The absolute address and the symbol comment for the symbol are entered (if present in the import file).

Step 5: For each message, the message text is imported.

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

If a hierarchy folder was highlighted that contains several models, the import files appear with the model respectively in the list. It can be edited if needed. Then, as described above, all models in the list are imported.

PARAMETERS AND SIGNALS

For process tag types and models to be generated successfully, it is important to define all inputs and outputs of a CFC as parameter or as signal. Only connections that are defined as parameter or signal can be included and parameterized as a column in the import file.

PROCESS OBJECT VIEW

With the process object view, all basic automation data is represented project-wide in a process engineering oriented view. Project-wide means that the data of all included projects is recorded in a multi-project.

The process object view has a similar structure as the plant view:

- In the left half of the window, the plant hierarchy is represented as a tree structure (hierarchy window). There, identical operating options are offered. In addition, the CFCs, SFCs, pictures, reports and supplementary documents are displayed.

– In the right half, a table of the lower level objects with their attributes is displayed (content window). The content window includes the tabs shown in Table 1 and thus provides different views to the project data.

Table 1: Tabs oft he process object view

Tab	Usage
General	This tab displays all the lower-level ES objects (process tags, CFCs, SFCs, pictures, reports, or additional documents) and their general information for the plant unit currently selected in the left window.
Blocks	This tab displays every block property of all CFC blocks in the selected object of the hierarchy window. In this context, SFC instances are also referred to as blocks.
Parameters	This tab displays the I/O points that you can configure or interconnect with other I/O points in the process object view, for all the process tags and CFCs displayed in the "General" tab (S7_edit = 'para').
Signals	This tab displays the I/O points that you can interconnect with a signal in the process object view, for all the process tags and CFCs displayed in the "General" tab (S7_edit = 'signal').
Messages	This tab displays the corresponding messages for all the process tags and CFCs displayed in the "General" tab.
Picture objects	This tab displays any picture interconnections which may exist in WinCC for all the process tags and CFCs displayed in the "General" tab.
Archive tags	This tab displays any archive interconnections which may exist in WinCC for all the process tags and CFCs displayed in the "General" tab.
Hierarchy folder	This tab displays the hierarchy folders of the plant hierarchy contained in the selected object of the hierarchy window.
Equipment properties	This tab displays the equipment properties contained in the selected project. These equipment properties are instances created by the equipment property types configured in the shared declarations.
Shared Declarations	This tab shows the attributes of the enumeration, units and equipment-property types in the project.

LITERATURE

[1] Lauber, R. and Göhner, P. (1999): Process Automation 2. Springer Publishers

[2] Help for PCS 7. Siemens

STEP BY STEP INSTRUCTIONS

TASK

PCS7 is a software that provides the user with many aids to effectively program large plants and to duplicate program parts.

In this task, charts and hierarchy structures are generated as library objects. This makes it possible to use them multiple times. As aids, the import-export assistant and the project object view are used.

Here, we are using chart ’A1T2H008’ -for manually operating the heater in reactor R001- as the process tag template. With the aid of this process tag, we are generating chart ’A1T2H010’ for manually operating the heater in reactor R002.

As our model, we are using folder ’A1T3X001’ for the inlet valve Product Tank B001 as the template.

From this, folder ’A1T3X002’ is generated for the inlet valve Product Tank B002.

OBJECTIVE

In this chapter, the student will learn the following:

– Duplication processing by using the import-export assistant

– Duplication processing in the project object view

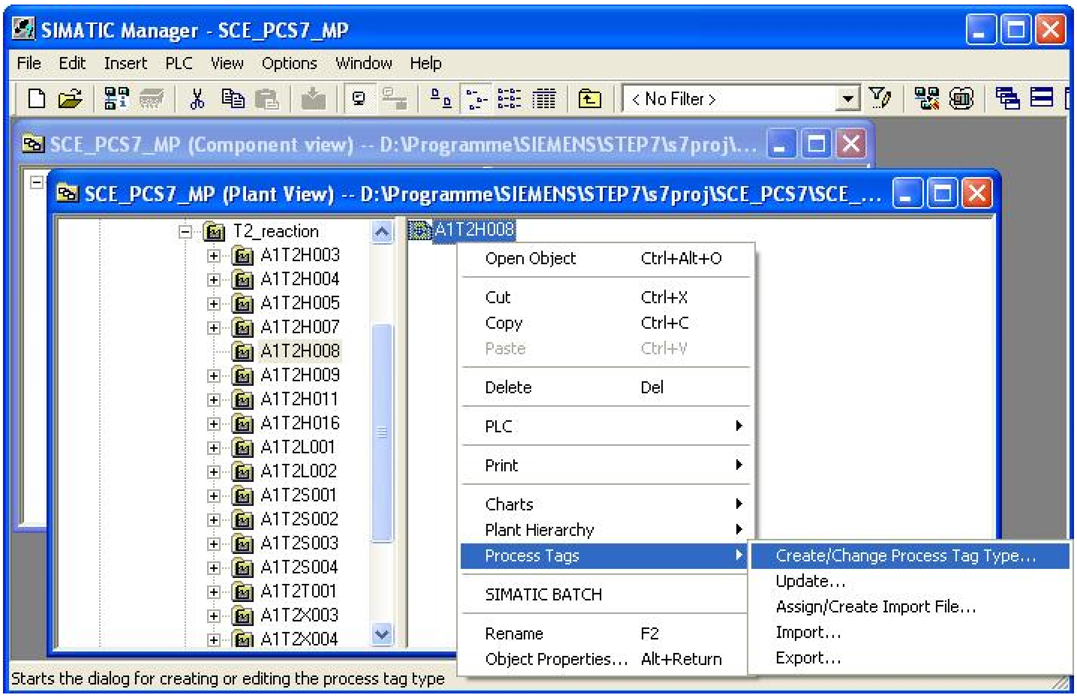
– Duplicating charts by generating process tags

– Duplicating folder structures by generating models

PROGRAMMING

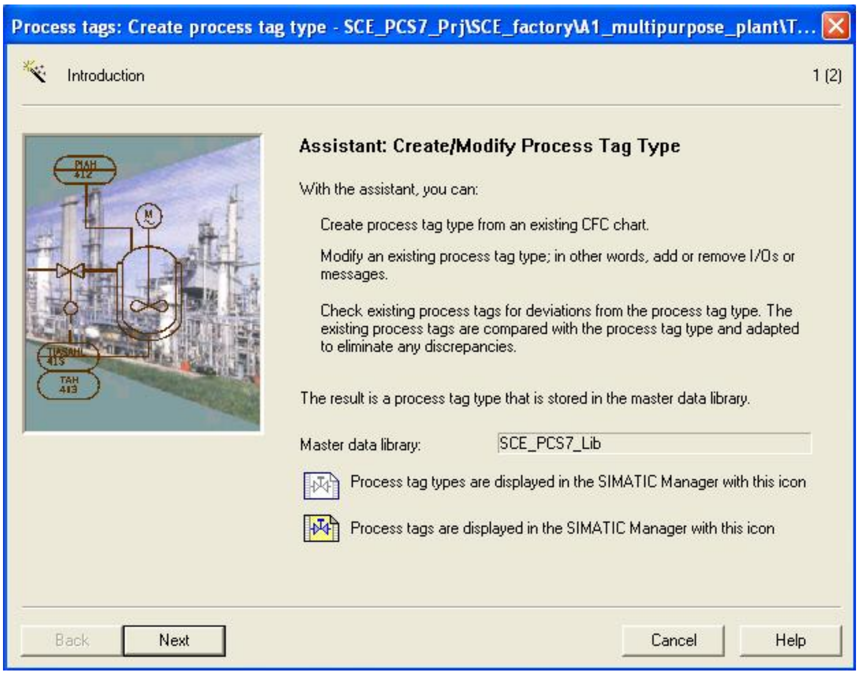
1. To duplicate a chart that was already generated and tested, we are generating from it a so-called process tag. In this example, we are using chart ’A1T2H008’ for manually operating the heater.

   (→ A1T2H008 → Process tags → Create/Change Process Tag Type)



Starts the dialog for creating or editing the process tag type

2. Then, information for the assistant is displayed. (→ Next)



Process tags: Create process tag type - SCE_PCS7_Prj\SCE_factory\A1_multipurpose_plant\T...

Introduction 1 (2)

Assistant: Create/Modify Process Tag Type

With the assistant, you can:

Create process tag type from an existing CFC chart.

Modify an existing process tag type; in other words, add or remove I/Os or messages.

Check existing process tags for deviations from the process tag type. The existing process tags are compared with the process tag type and adapted to eliminate any discrepancies.

The result is a process tag type that is stored in the master data library.

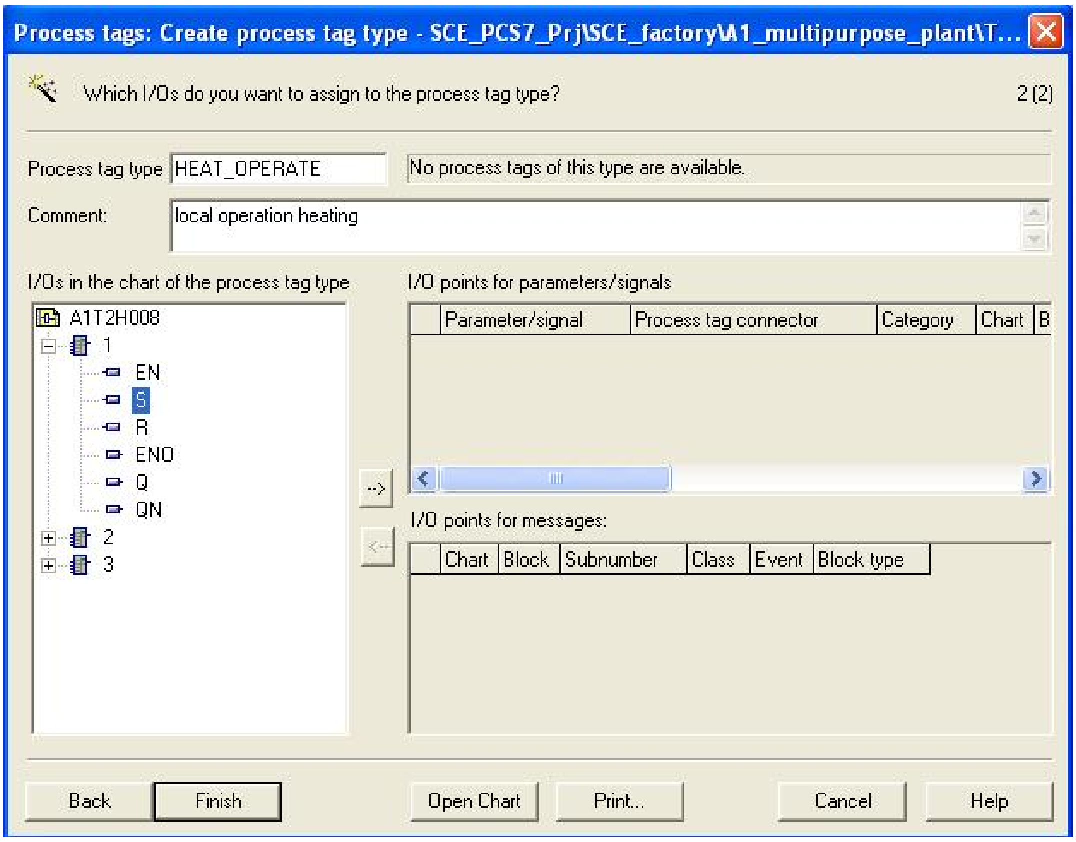
Master data library: SCE_PCS7_Lib

Process tag types are displayed in the SIMATIC Manager with this icon

Process tags are displayed in the SIMATIC Manager with this icon

3. In the following dialog, the name of the process tag type is specified, and a comment is entered. Then, with a double click on the desired block connections we specify which of them are later available as connection points during the import.

(→ HEAT_OPERATE → local operation heating → A1T2H008 → 1 → S)



Process tags: Create process tag type - SCE_PCS7_Prj\SCE_factory\A1_multipurpose_plant\T...

Which I/Os do you want to assign to the process tag type? 2 (2)

Process tag type: HEAT_OPERATE

No process tags of this type are available.

Comment: local operation heating

I/Os in the chart of the process tag type

```
A1T2H008
 1
   EN
   S
   R
   ENO
   Q
   QN
 2
 3
```

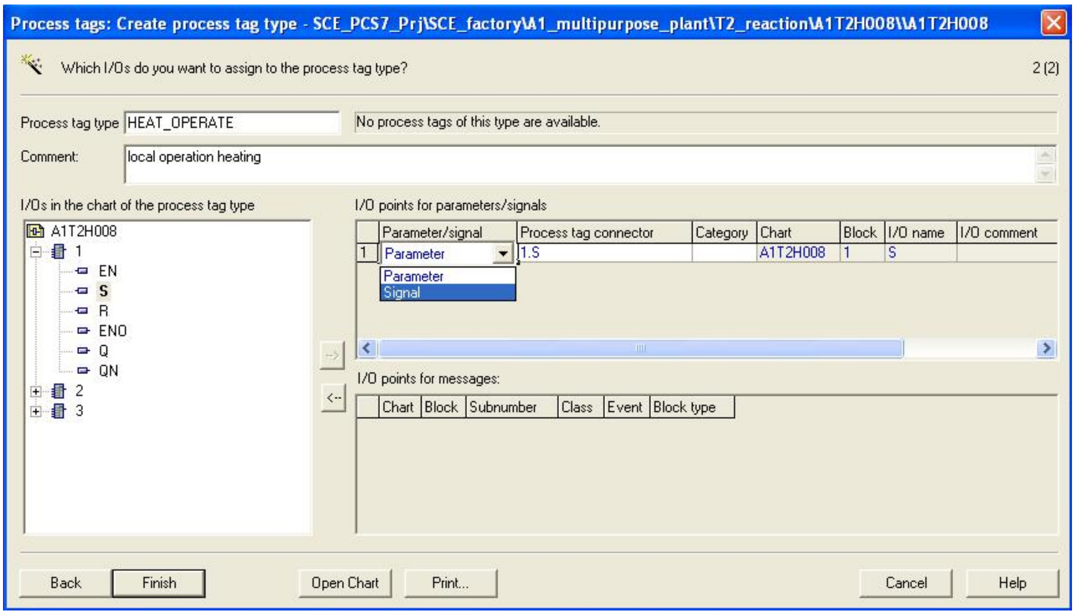
I/O points for parameters/signals

Parameter/signal	Process tag connector	Category	Chart	B

I/O points for messages:

Chart	Block	Subnumber	Class	Event	Block type

4. For each connection we have to specify whether it is available as parameter or as a signal connection. (→ Signal)



Process tags: Create process tag type - SCE_PCS7_Prj\SCE_factory\A1_multipurpose_plant\T2_reaction\A1T2H008\\A1T2H008

Which I/Os do you want to assign to the process tag type? 2 (2)

Process tag type: HEAT_OPERATE

No process tags of this type are available.

Comment: local operation heating

I/Os in the chart of the process tag type

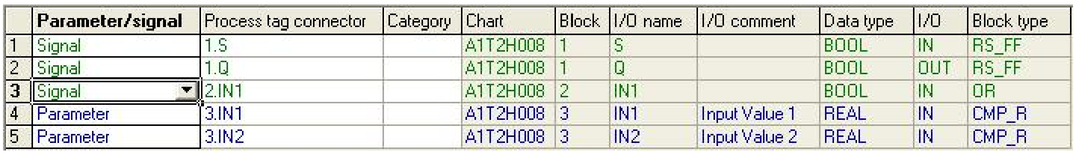
I/O points for parameters/signals

	Parameter/signal	Process tag connector	Category	Chart	Block	I/O name	I/O comment
1	Parameter	1.S		A1T2H008	1	S	

I/O points for messages:

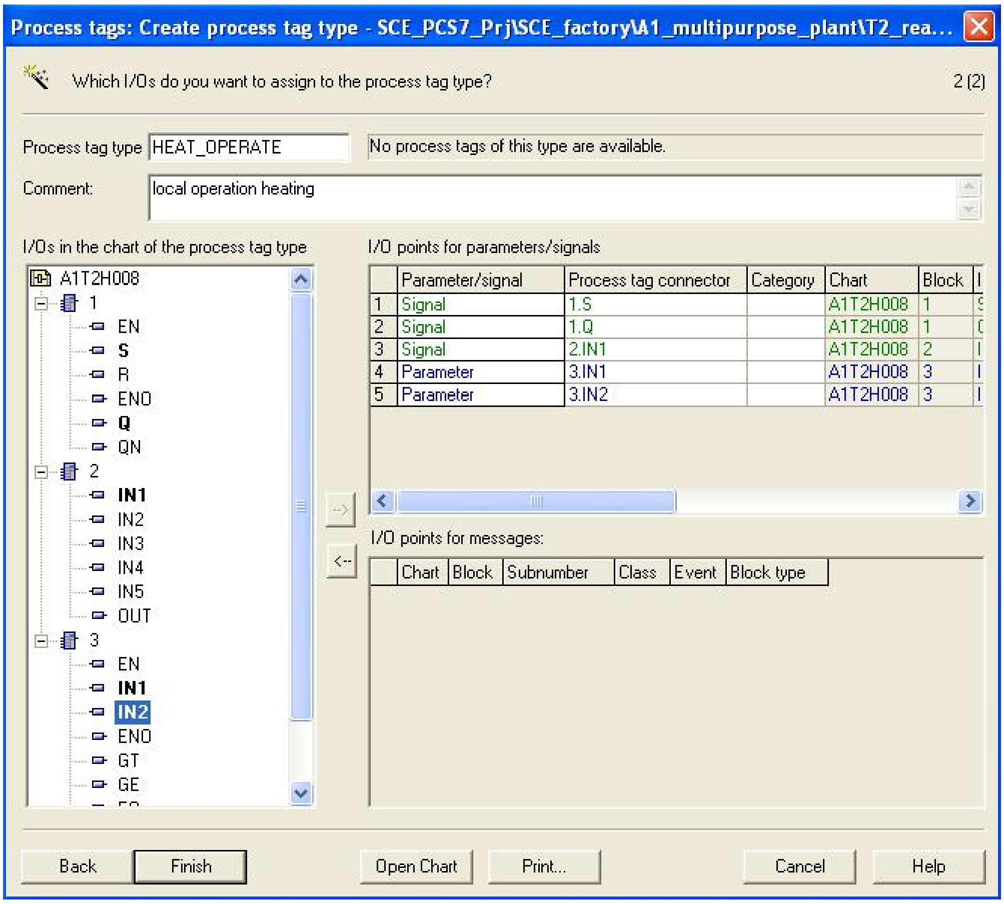
	Chart	Block	Subnumber	Class	Event	Block type

5. Our example has -as shown here- three signals ’1.S’, ’1.Q’ and ’2.IN1’ and two parameters ’7.IN1’ und ’7.IN2’.



	Parameter/signal	Process tag connector	Category	Chart	Block	I/O name	I/O comment	Data type	I/O	Block type
1	Signal	1.S		A1T2H008	1	S		BOOL	IN	RS_FF
2	Signal	1.Q		A1T2H008	1	Q		BOOL	OUT	RS_FF
3	Signal	2.IN1		A1T2H008	2	IN1		BOOL	IN	OR
4	Parameter	3.IN1		A1T2H008	3	IN1	Input Value 1	REAL	IN	CMP_R
5	Parameter	3.IN2		A1T2H008	3	IN2	Input Value 2	REAL	IN	CMP_R

6. The process tag 'HEAT_OPERATE' is now completed. (→ Finish)



Process tags: Create process tag type - SCE_PCS7_Prj\SCE_factory\A1_multipurpose_plant\T2_rea...

Which I/Os do you want to assign to the process tag type? 2 (2)

Process tag type: HEAT_OPERATE

No process tags of this type are available.

Comment: local operation heating

I/Os in the chart of the process tag type

I/O points for parameters/signals

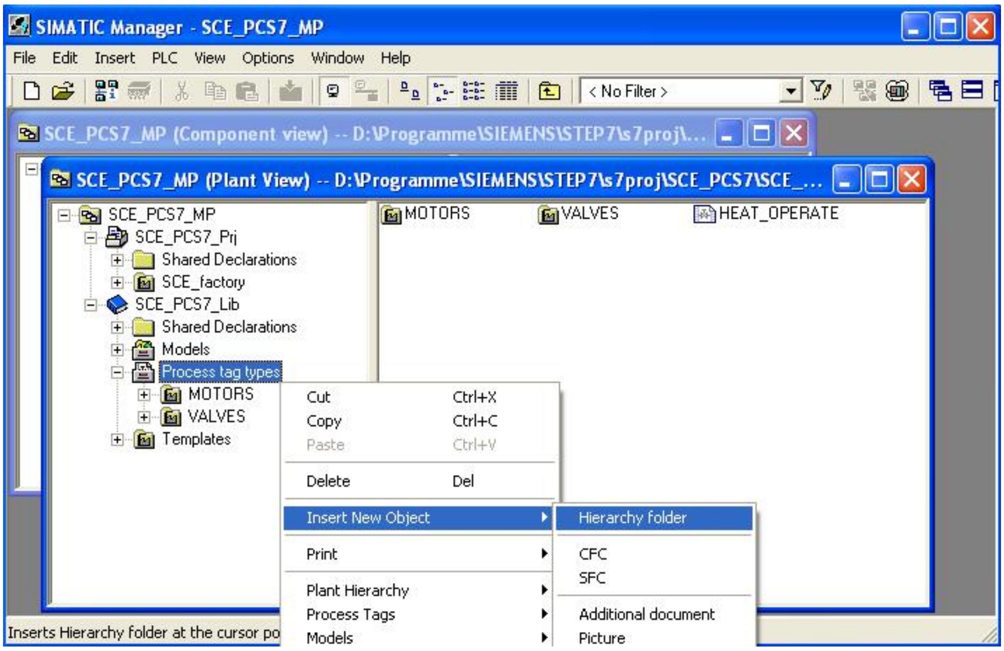
	Parameter/signal	Process tag connector	Category	Chart	Block	I
1	Signal	1.S		A1T2H008	1	S
2	Signal	1.Q		A1T2H008	1	Q
3	Signal	2.IN1		A1T2H008	2	I
4	Parameter	3.IN1		A1T2H008	3	I
5	Parameter	3.IN2		A1T2H008	3	I

I/O points for messages:

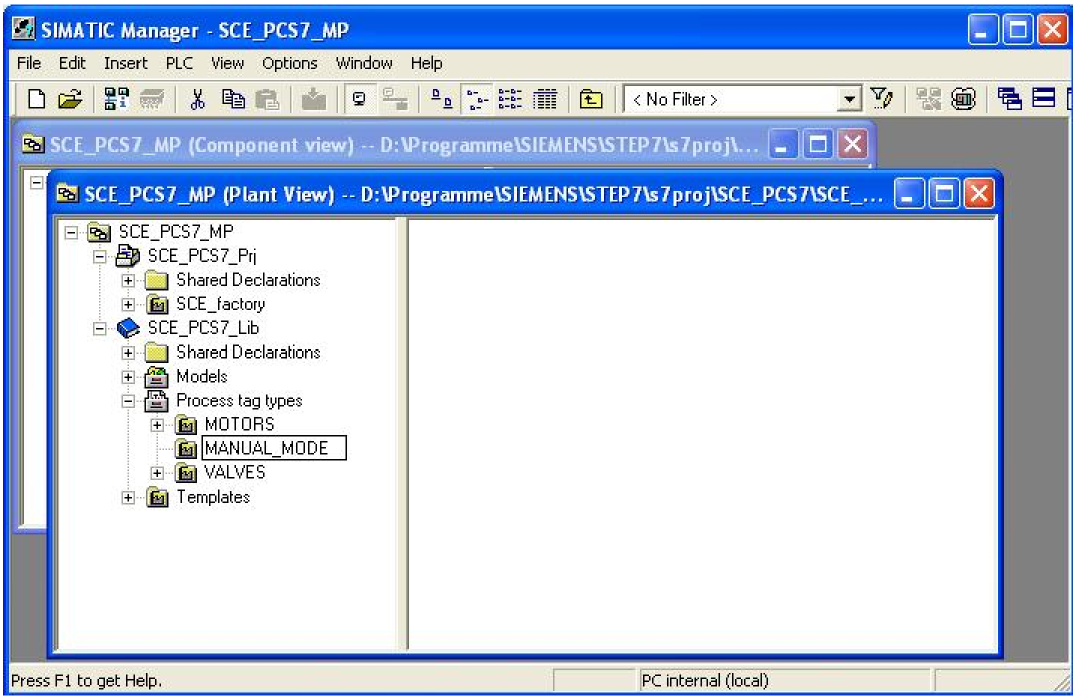
Chart	Block	Subnumber	Class	Event	Block type

7. For our process tag, a hierarchy folder is inserted in the plant view of the ***SIMATIC Manager***.

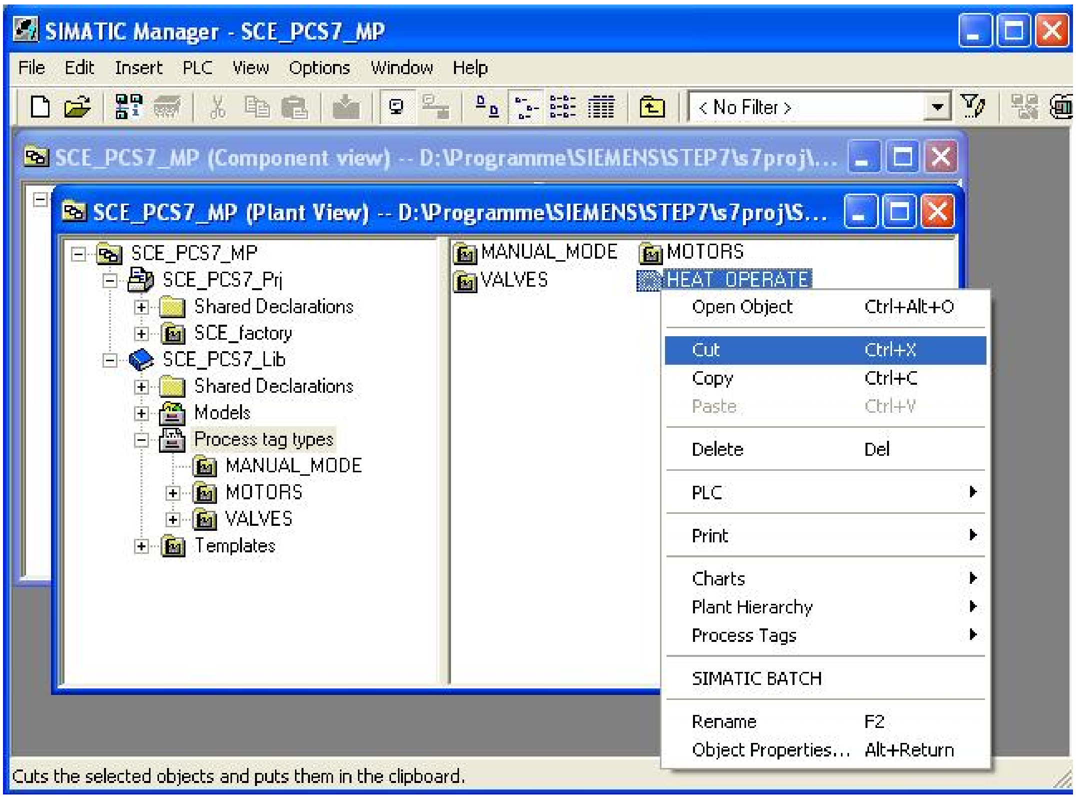
(→ Process tag types → Insert New Object → Hierarchy folder)



8. We now rename this folder ’MANUAL_MODE’. (→ MANUAL_MODE)

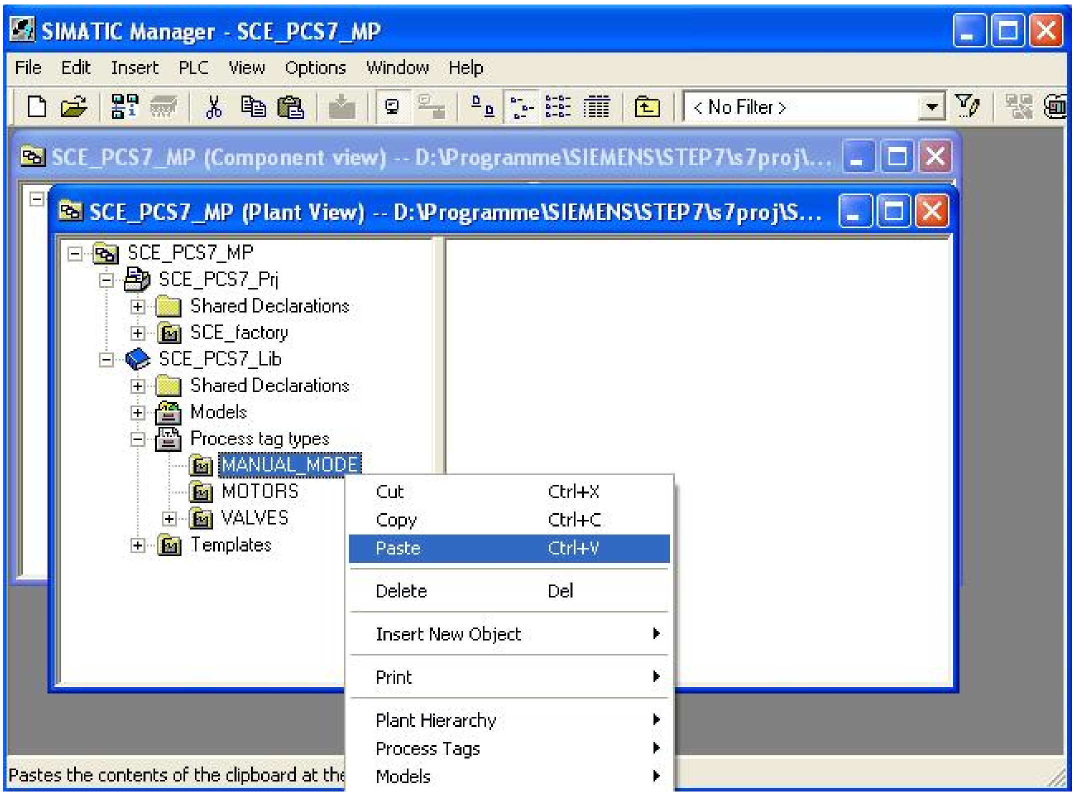

SIMATIC Manager - SCE_PCS7_MP
File Edit Insert PLC View Options Window Help
< No Filter >
SCE_PCS7_MP (Component view) -- D:\Programme\SIEMENS\STEP7\s7proj\...
SCE_PCS7_MP (Plant View) -- D:\Programme\SIEMENS\STEP7\s7proj\SCE_PCS7\SCE_...
SCE_PCS7_MP
SCE_PCS7_Prj
Shared Declarations
SCE_factory
SCE_PCS7_Lib
Shared Declarations
Models
Process tag types
MOTORS
MANUAL_MODE
VALVES
Templates
Press F1 to get Help.
PC internal (local)

9. Then, we cut the process tag ’HEAT_OPERATE’.
(→ HEAT_OPERATE → Cut)



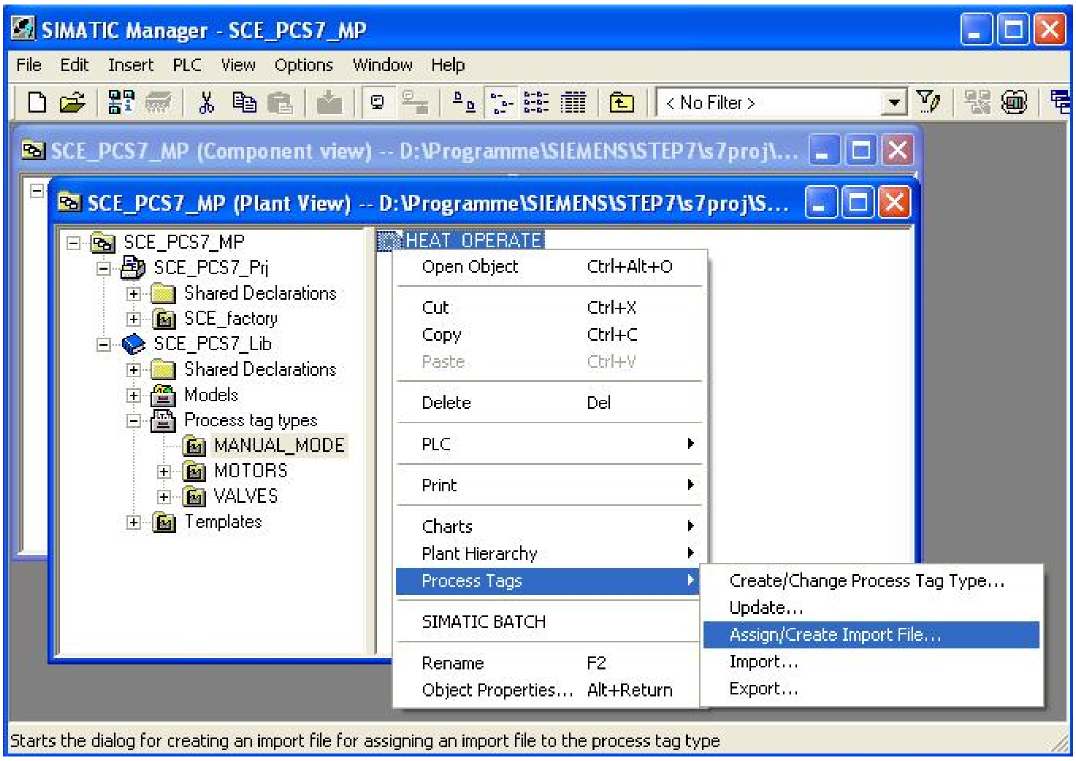
10. Next we paste it in the hierarchy folder ’MANUAL_MODE’.

    (→ MANUAL_MODE → Paste)

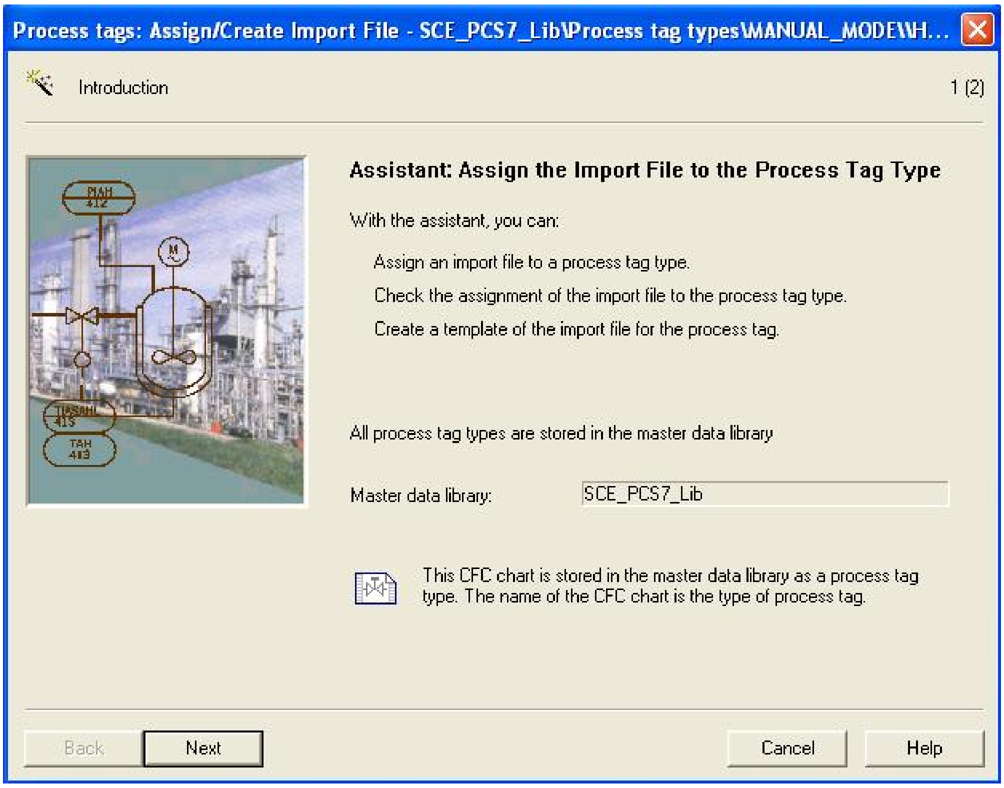

SIMATIC Manager - SCE_PCS7_MP
File Edit Insert PLC View Options Window Help
< No Filter >
SCE_PCS7_MP (Component view) -- D:\Programme\SIEMENS\STEP7\s7proj\...
SCE_PCS7_MP (Plant View) -- D:\Programme\SIEMENS\STEP7\s7proj\S...
SCE_PCS7_MP
SCE_PCS7_Prj
Shared Declarations
SCE_factory
SCE_PCS7_Lib
Shared Declarations
Models
Process tag types
MANUAL_MODE
MOTORS
VALVES
Templates
Cut Ctrl+X
Copy Ctrl+C
Paste Ctrl+V
Delete Del
Insert New Object
Print
Plant Hierarchy
Process Tags
Models
Pastes the contents of the clipboard at the

11. To generate numerous CFCs of the process tag type ’HEAT_OPERATE’, it is assigned an import file.

    (→ HEAT_OPERATE → Process Tags → Assign/Create Import File)


SIMATIC Manager - SCE_PCS7_MP
File Edit Insert PLC View Options Window Help
< No Filter >
SCE_PCS7_MP (Component view) -- D:\Programme\SIEMENS\STEP7\s7proj\...
SCE_PCS7_MP (Plant View) -- D:\Programme\SIEMENS\STEP7\s7proj\S...
SCE_PCS7_MP
SCE_PCS7_Prj
Shared Declarations
SCE_factory
SCE_PCS7_Lib
Shared Declarations
Models
Process tag types
MANUAL_MODE
MOTORS
VALVES
Templates
HEAT_OPERATE
Open Object Ctrl+Alt+O
Cut Ctrl+X
Copy Ctrl+C
Paste Ctrl+V
Delete Del
PLC
Print
Charts
Plant Hierarchy
Process Tags
SIMATIC BATCH
Rename F2
Object Properties... Alt+Return
Create/Change Process Tag Type...
Update...
Assign/Create Import File...
Import...
Export...
Starts the dialog for creating an import file for assigning an import file to the process tag type

12. Then information for the assistant is displayed. (→ Next)



Process tags: Assign/Create Import File - SCE_PCS7_Lib\Process tag types\MANUAL_MODE\H...

Introduction 1 (2)

Assistant: Assign the Import File to the Process Tag Type

With the assistant, you can:

Assign an import file to a process tag type.

Check the assignment of the import file to the process tag type.

Create a template of the import file for the process tag.

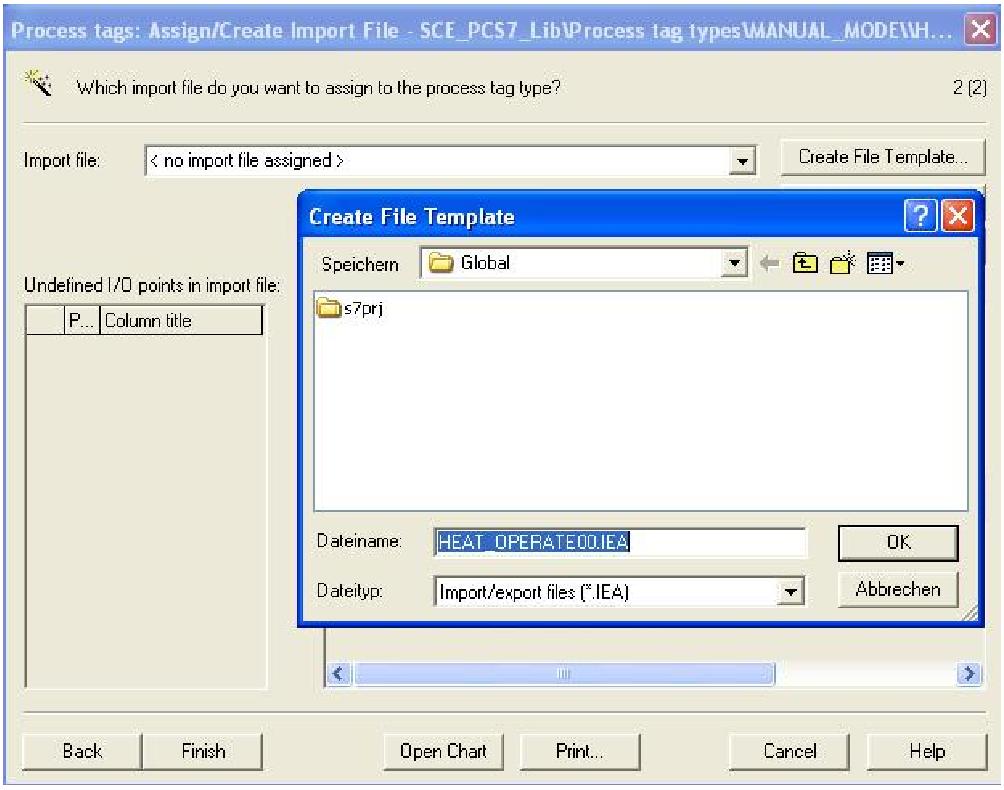
All process tag types are stored in the master data library

Master data library: SCE_PCS7_Lib

This CFC chart is stored in the master data library as a process tag type. The name of the CFC chart is the type of process tag.

13. First, we generate a file template.

(→ Create File Template → HEAT_OPERATE.IEA → OK)



Process tags: Assign/Create Import File - SCE_PCS7_Lib\Process tag types\MANUAL_MODE\H...

Which import file do you want to assign to the process tag type? 2 (2)

Import file: < no import file assigned >

Undefined I/O points in import file:

P...	Column title

Create File Template

Speichern: Global

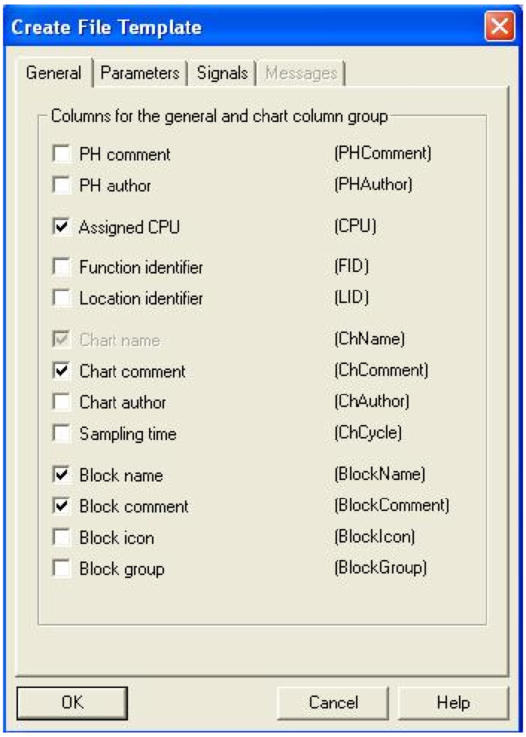
s7prj

Dateiname: HEAT_OPERATE00.IEA

Dateityp: Import/export files (*.IEA)

14. In the following dialog we select the general columns that are displayed in the import file.

    (→ General → Assigned CPU → Chart comment → Block name → Block comment)



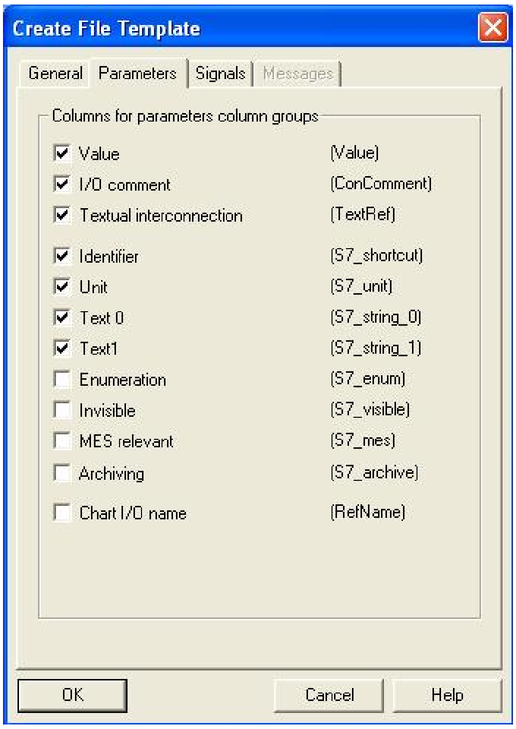
Create File Template

Columns for the general and chart column group

☐ PH comment (PHComment)
☐ PH author (PHAuthor)
☑ Assigned CPU (CPU)
☐ Function identifier (FID)
☐ Location identifier (LID)
☑ Chart name (ChName)
☑ Chart comment (ChComment)
☐ Chart author (ChAuthor)
☐ Sampling time (ChCycle)
☑ Block name (BlockName)
☑ Block comment (BlockComment)
☐ Block icon (BlockIcon)
☐ Block group (BlockGroup)

15. Here, we select which columns are displayed for the parameters in the import file.

    (→ Parameters → Value→ I/O comment → Textual interconnections → Identifier → Unit → Text 0 → Text 1)

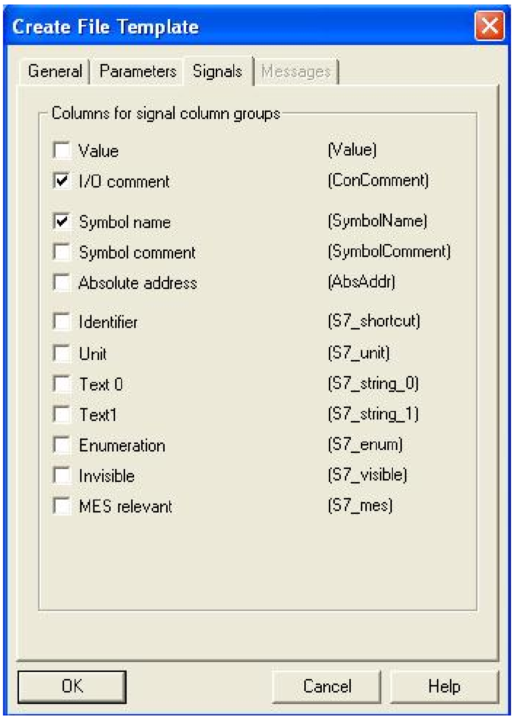


Create File Template

Columns for parameters column groups

☑ Value (Value)
☑ I/O comment (ConComment)
☑ Textual interconnection (TextRef)
☑ Identifier (S7_shortcut)
☑ Unit (S7_unit)
☑ Text 0 (S7_string_0)
☑ Text1 (S7_string_1)
☐ Enumeration (S7_enum)
☐ Invisible (S7_visible)
☐ MES relevant (S7_mes)
☐ Archiving (S7_archive)
☐ Chart I/O name (RefName)

16. Here, we select the columns that are displayed for the signals in the import file. (→ Signals → I/O comment → Symbol name → OK)

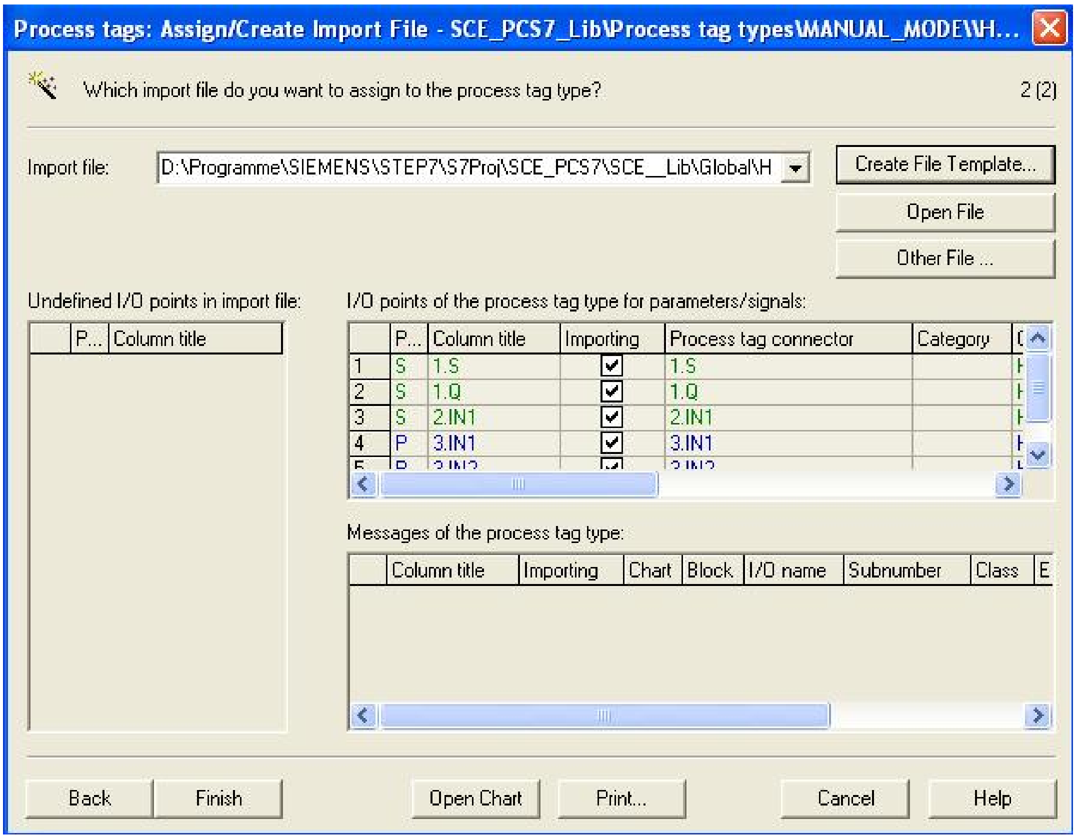


Create File Template

Columns for signal column groups

☐ Value	(Value)
☑ I/O comment	(ConComment)
☑ Symbol name	(SymbolName)
☐ Symbol comment	(SymbolComment)
☐ Absolute address	(AbsAddr)
☐ Identifier	(S7_shortcut)
☐ Unit	(S7_unit)
☐ Text 0	(S7_string_0)
☐ Text1	(S7_string_1)
☐ Enumeration	(S7_enum)
☐ Invisible	(S7_visible)
☐ MES relevant	(S7_mes)

17. We then open the import file that we created. (→ Open File)



Process tags: Assign/Create Import File - SCE_PCS7_Lib\Process tag types\MANUAL_MODE\H...

Which import file do you want to assign to the process tag type? 2 (2)

Import file: D:\Programme\SIEMENS\STEP7\S7Proj\SCE_PCS7\SCE__Lib\Global\H

Undefined I/O points in import file:

	P...	Column title

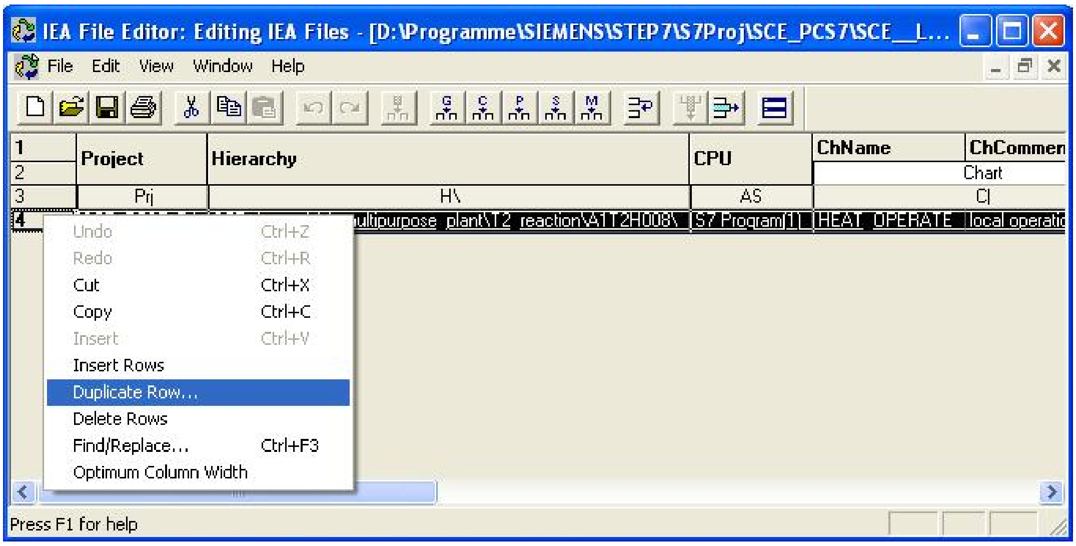
I/O points of the process tag type for parameters/signals:

	P...	Column title	Importing	Process tag connector	Category	C
1	S	1.S	☑	1.S		H
2	S	1.Q	☑	1.Q		H
3	S	2.IN1	☑	2.IN1		H
4	P	3.IN1	☑	3.IN1		H

Messages of the process tag type:

	Column title	Importing	Chart	Block	I/O name	Subnumber	Class	E

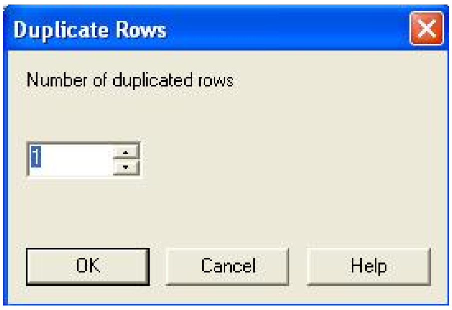
18. The row of chart ’A1T2H008’ is duplicated once to set up chart ’A1T2H010’. (→ Duplicate Row → 1 → OK)



IEA File Editor: Editing IEA Files - [D:\Programme\SIEMENS\STEP7\S7Proj\SCE_PCS7\SCE__L...

	Project	Hierarchy	CPU	ChName	ChComment
2				Chart	
3	Prj	H\	AS	CI	
4		...ultipurpose_plant\T2_reaction\A1T2H008\	S7 Program(1)	HEAT_OPERATE	local operatio

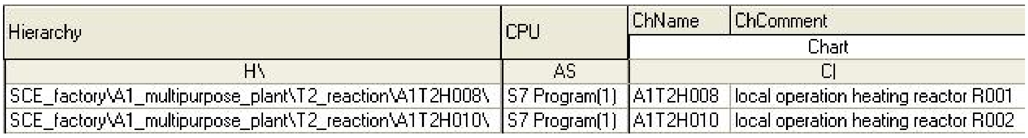
Press F1 for help



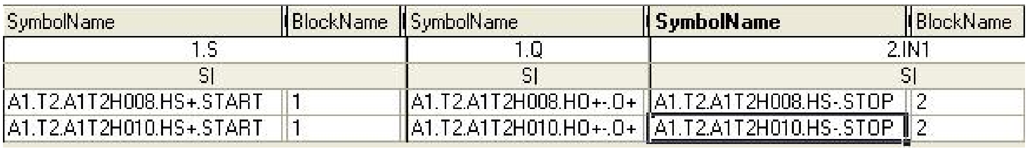
Duplicate Rows

Number of duplicated rows: 1

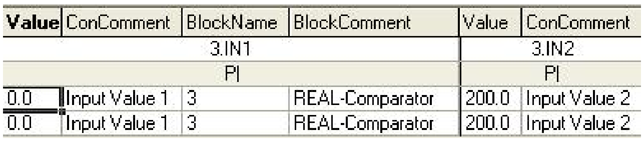
19. We then change the entries for charts ’A1T2H008’ and ’A1T2H010’ in the columns of the import files, as shown here.



Hierarchy	CPU	ChName	ChComment
		Chart	
H\	AS	CI	
SCE_factory\A1_multipurpose_plant\T2_reaction\A1T2H008\	S7 Program(1)	A1T2H008	local operation heating reactor R001
SCE_factory\A1_multipurpose_plant\T2_reaction\A1T2H010\	S7 Program(1)	A1T2H010	local operation heating reactor R002

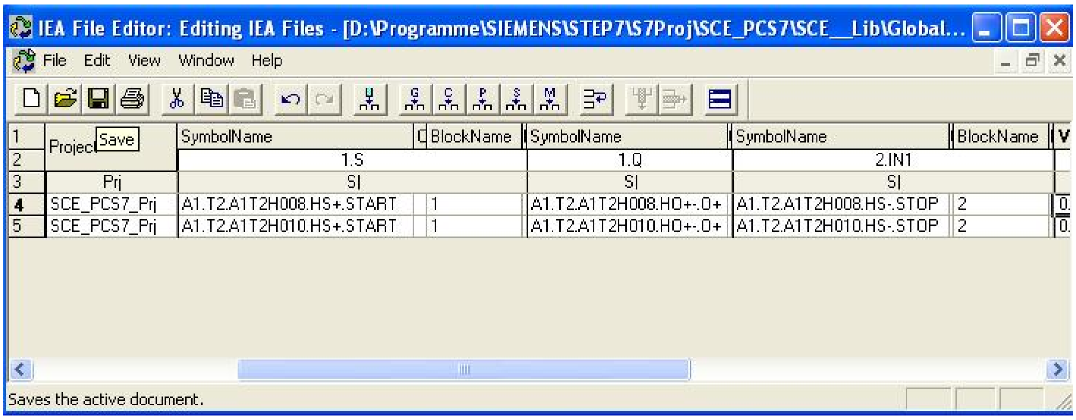


SymbolName	BlockName	SymbolName	SymbolName	BlockName
1.S		1.Q	2.IN1	
SI		SI	SI	
A1.T2.A1T2H008.HS+.START	1	A1.T2.A1T2H008.HO+-.O+	A1.T2.A1T2H008.HS-.STOP	2
A1.T2.A1T2H010.HS+.START	1	A1.T2.A1T2H010.HO+-.O+	A1.T2.A1T2H010.HS-.STOP	2



Value	ConComment	BlockName	BlockComment	Value	ConComment
3.IN1				3.IN2	
PI				PI	
0.0	Input Value 1	3	REAL-Comparator	200.0	Input Value 2
0.0	Input Value 1	3	REAL-Comparator	200.0	Input Value 2

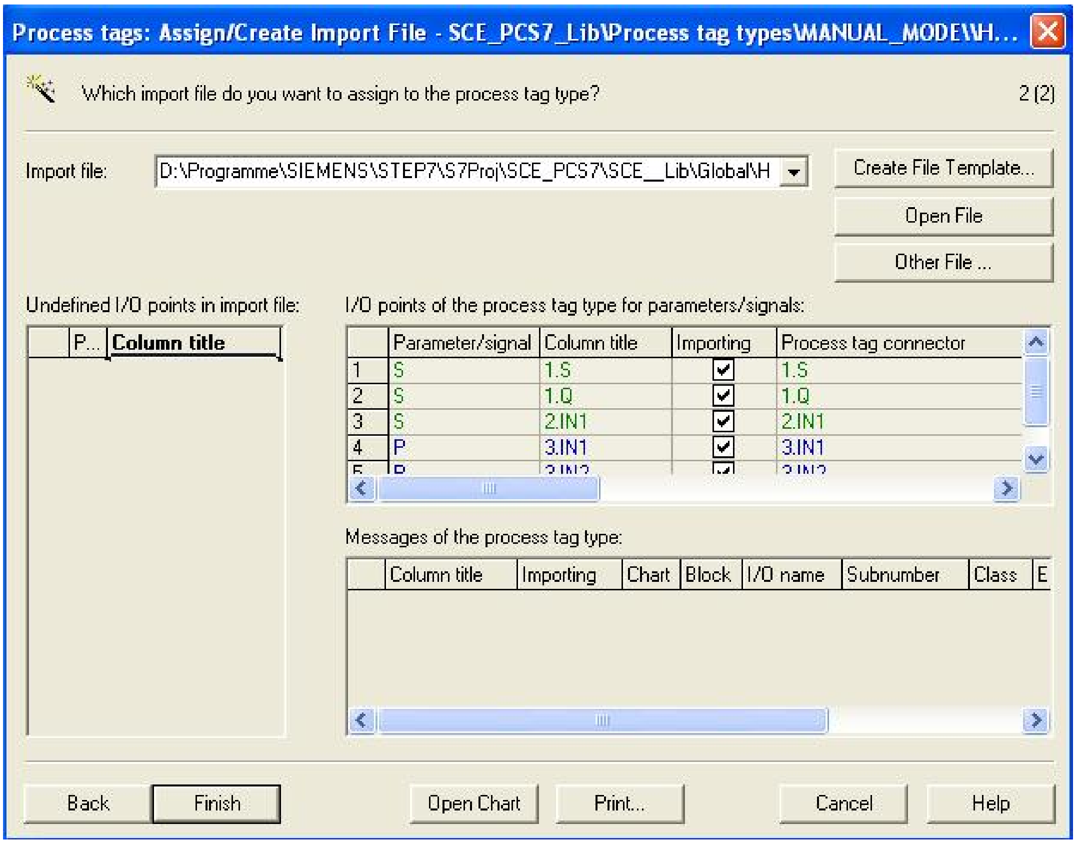
20. We now save the import file and close the window of the IEA file editor. (→ 💾 → ☒)


IEA File Editor: Editing IEA Files - [D:\Programme\SIEMENS\STEP7\S7Proj\SCE_PCS7\SCE__Lib\Global...
File Edit View Window Help
Save

1	Projec	SymbolName	C	BlockName	SymbolName	SymbolName	BlockName	V
2		1.S			1.Q	2.IN1		
3	Prj	SI			SI	SI		
4	SCE_PCS7_Prj	A1.T2.A1T2H008.HS+.START		1	A1.T2.A1T2H008.HO+-.0+	A1.T2.A1T2H008.HS-.STOP	2	0.
5	SCE_PCS7_Prj	A1.T2.A1T2H010.HS+.START		1	A1.T2.A1T2H010.HO+-.0+	A1.T2.A1T2H010.HS-.STOP	2	0.

Saves the active document.

21. Process tag 'HEAT_OPERATE' can then be completed. (→ Finish)


Process tags: Assign/Create Import File - SCE_PCS7_Lib\Process tag types\MANUAL_MODE\H...
Create File Template...
Open File
Other File ...
Back
Finish
Open Chart
Print...
Cancel
Help

Which import file do you want to assign to the process tag type? 2 (2)

Import file: D:\Programme\SIEMENS\STEP7\S7Proj\SCE_PCS7\SCE__Lib\Global\H

Undefined I/O points in import file:

	P...	Column title

I/O points of the process tag type for parameters/signals:

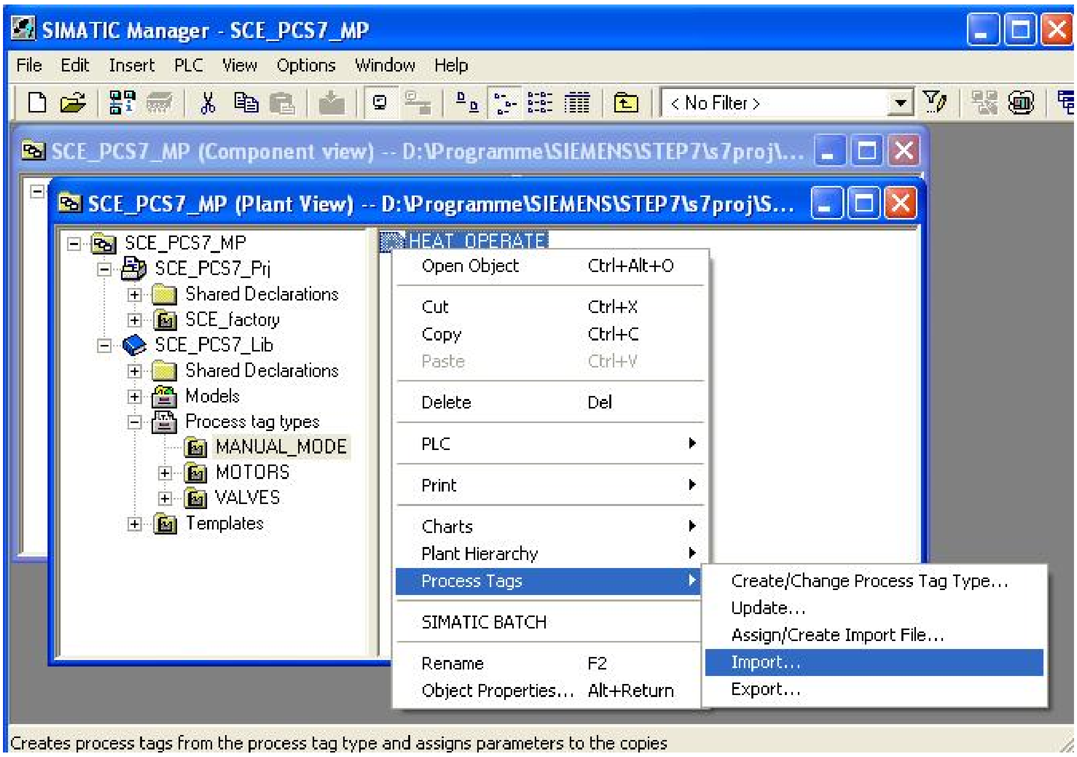
	Parameter/signal	Column title	Importing	Process tag connector
1	S	1.S	☑	1.S
2	S	1.Q	☑	1.Q
3	S	2.IN1	☑	2.IN1
4	P	3.IN1	☑	3.IN1
5	P	3.IN2	☑	3.IN2

Messages of the process tag type:

	Column title	Importing	Chart	Block	I/O name	Subnumber	Class	E

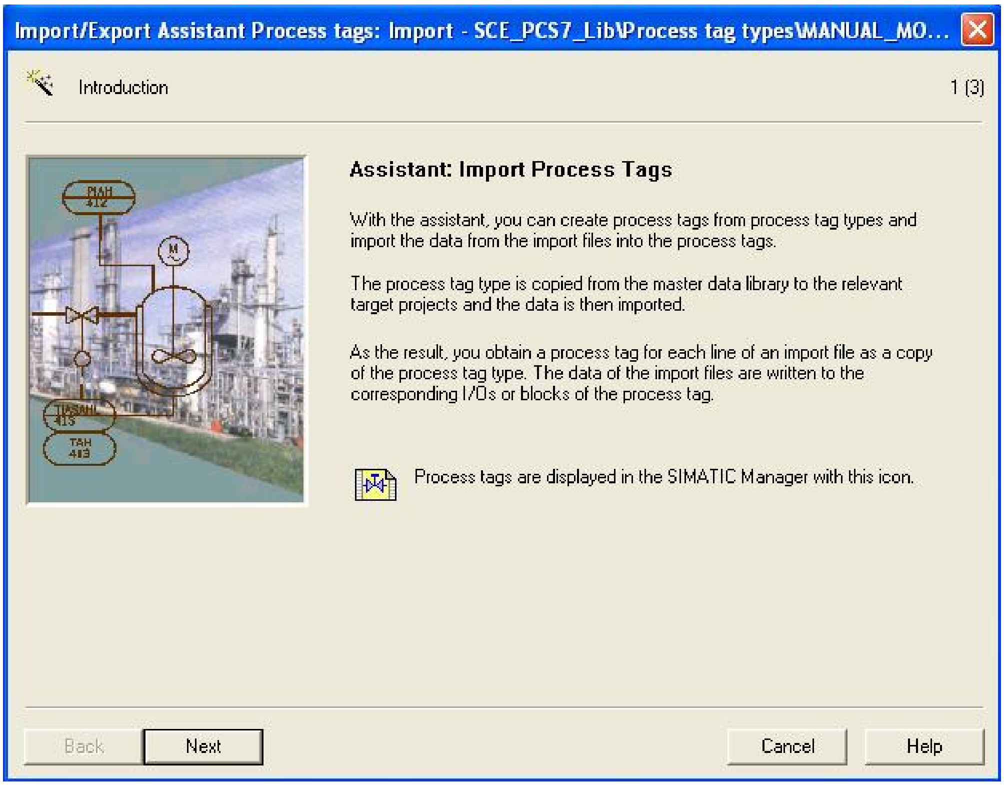
22. We now start the import with the import file that we set up.

(→ HEAT_OPERATE → Process Tags → Import)



Creates process tags from the process tag type and assigns parameters to the copies

23. Then, the information is displayed for the assistant. (→ Next)


Import/Export Assistant Process tags: Import - SCE_PCS7_Lib\Process tag types\MANUAL_MO...
Introduction
1 (3)
PIAH 412
M
TIASAHL 413
TAH 413
Back
Next
Cancel
Help

Assistant: Import Process Tags

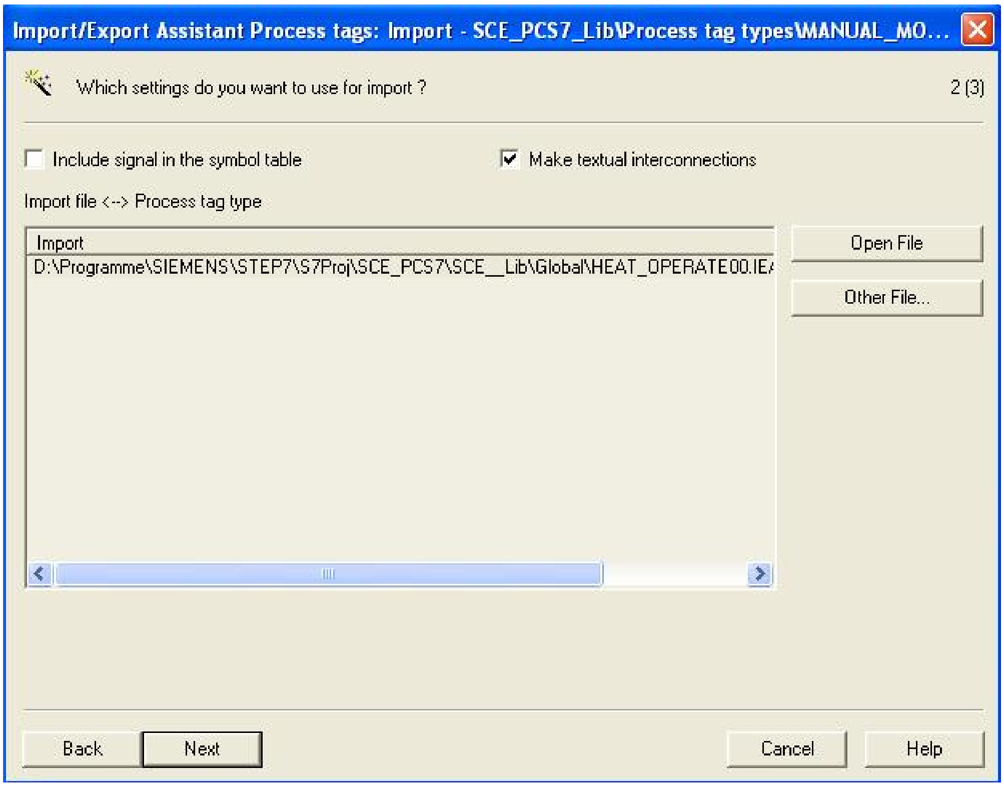
With the assistant, you can create process tags from process tag types and import the data from the import files into the process tags.

The process tag type is copied from the master data library to the relevant target projects and the data is then imported.

As the result, you obtain a process tag for each line of an import file as a copy of the process tag type. The data of the import files are written to the corresponding I/Os or blocks of the process tag.

Process tags are displayed in the SIMATIC Manager with this icon.

24. We now select the import file we previously set up and the option ’Make textual interconnections’. (→ Make textual interconnections → Next)



Import/Export Assistant Process tags: Import - SCE_PCS7_Lib\Process tag types\MANUAL_MO...

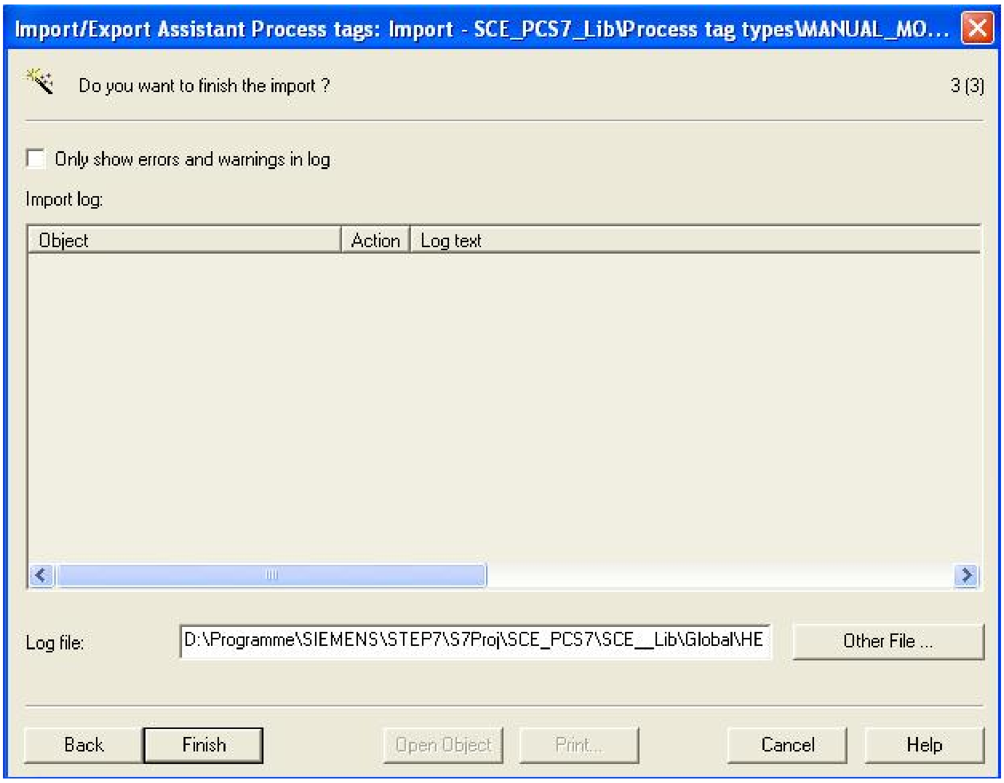
Which settings do you want to use for import ? 2 (3)

□ Include signal in the symbol table ☑ Make textual interconnections

Import file <--> Process tag type

Import
D:\Programme\SIEMENS\STEP7\S7Proj\SCE_PCS7\SCE__Lib\Global\HEAT_OPERATE00.IE/

25. We then start the import. (→ Finish)



Import/Export Assistant Process tags: Import - SCE_PCS7_Lib\Process tag types\MANUAL_MO...

Do you want to finish the import ? 3 (3)

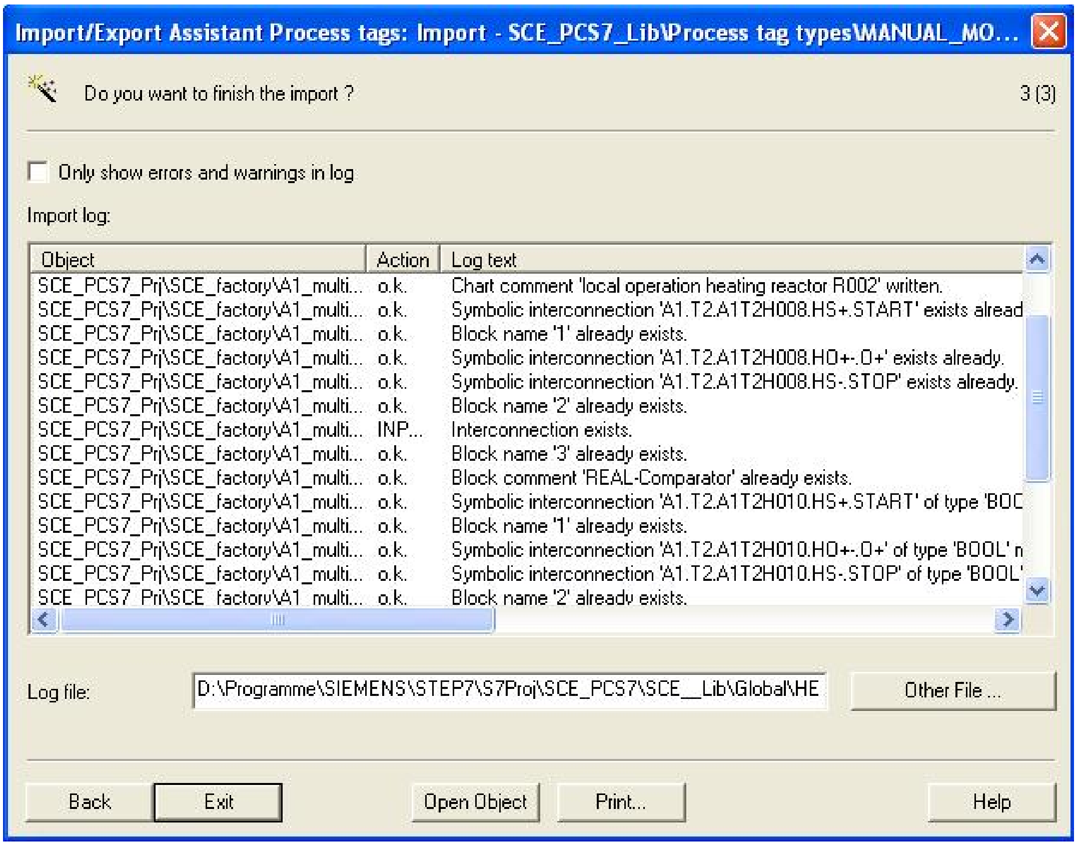
□ Only show errors and warnings in log

Import log:

Object	Action	Log text

Log file: D:\Programme\SIEMENS\STEP7\S7Proj\SCE_PCS7\SCE__Lib\Global\HE

26. An import log is then displayed. (→ Exit)



Import/Export Assistant Process tags: Import - SCE_PCS7_Lib\Process tag types\MANUAL_MO...

Do you want to finish the import ? 3 (3)

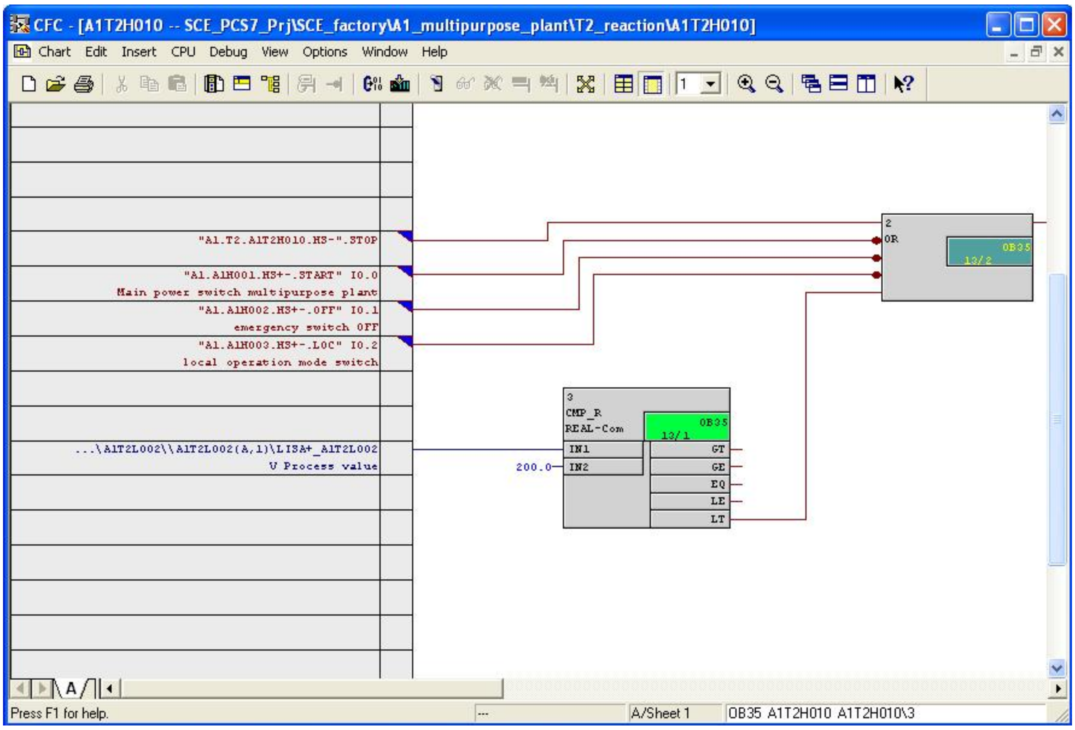
□ Only show errors and warnings in log

Import log:

Object	Action	Log text
SCE_PCS7_Prj\SCE_factory\A1_multi...	o.k.	Chart comment 'local operation heating reactor R002' written.
SCE_PCS7_Prj\SCE_factory\A1_multi...	o.k.	Symbolic interconnection 'A1.T2.A1T2H008.HS+.START' exists alread
SCE_PCS7_Prj\SCE_factory\A1_multi...	o.k.	Block name '1' already exists.
SCE_PCS7_Prj\SCE_factory\A1_multi...	o.k.	Symbolic interconnection 'A1.T2.A1T2H008.HO+-.O+' exists already.
SCE_PCS7_Prj\SCE_factory\A1_multi...	o.k.	Symbolic interconnection 'A1.T2.A1T2H008.HS-.STOP' exists already.
SCE_PCS7_Prj\SCE_factory\A1_multi...	o.k.	Block name '2' already exists.
SCE_PCS7_Prj\SCE_factory\A1_multi...	INP...	Interconnection exists.
SCE_PCS7_Prj\SCE_factory\A1_multi...	o.k.	Block name '3' already exists.
SCE_PCS7_Prj\SCE_factory\A1_multi...	o.k.	Block comment 'REAL-Comparator' already exists.
SCE_PCS7_Prj\SCE_factory\A1_multi...	o.k.	Symbolic interconnection 'A1.T2.A1T2H010.HS+.START' of type 'BOC
SCE_PCS7_Prj\SCE_factory\A1_multi...	o.k.	Block name '1' already exists.
SCE_PCS7_Prj\SCE_factory\A1_multi...	o.k.	Symbolic interconnection 'A1.T2.A1T2H010.HO+-.O+' of type 'BOOL' n
SCE_PCS7_Prj\SCE_factory\A1_multi...	o.k.	Symbolic interconnection 'A1.T2.A1T2H010.HS-.STOP' of type 'BOOL'
SCE_PCS7_Prj\SCE_factory\A1_multi...	o.k.	Block name '2' already exists.

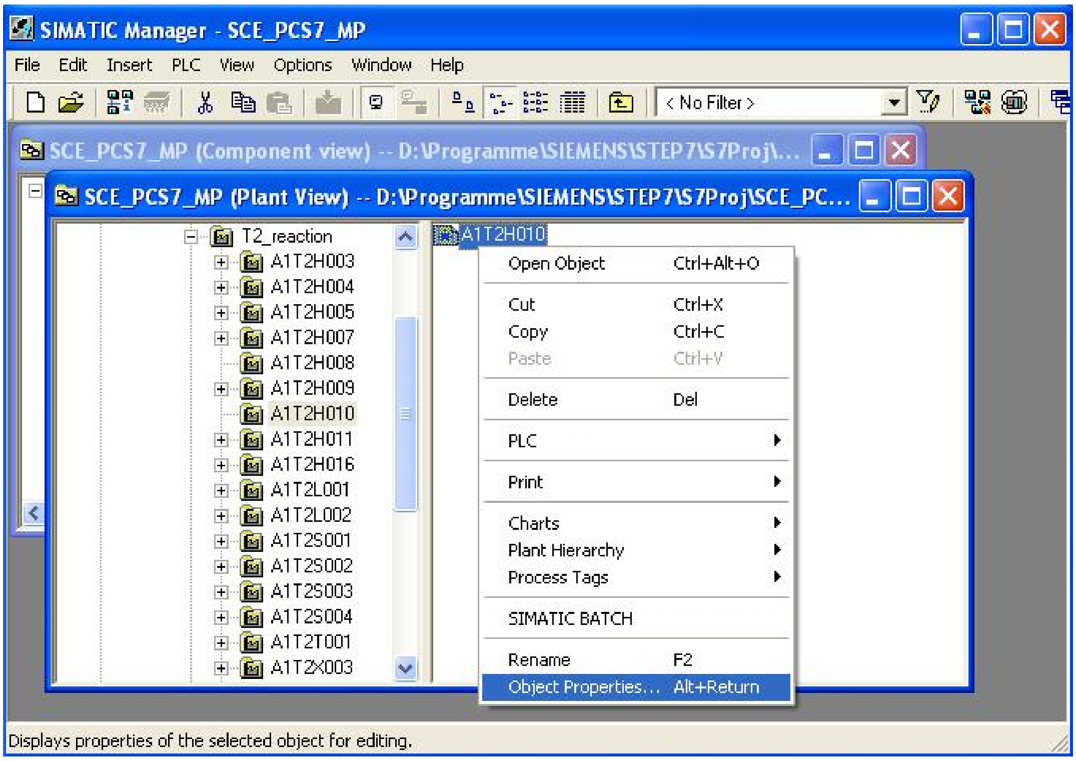
Log file: D:\Programme\SIEMENS\STEP7\S7Proj\SCE_PCS7\SCE__Lib\Global\HE

27. In this manner, we can quickly and effectively set up numerous charts. The interesting aspect regarding this method is that the changes in the charts are not performed individually, but by means of the import file in table form. Nevertheless, each individual chart can be monitored and changed afterwards of course with the CFC editor. (→ A1T2H010)


CFC - [A1T2H010 -- SCE_PCS7_Prj\SCE_factory\A1_multipurpose_plant\T2_reaction\A1T2H010]
Chart Edit Insert CPU Debug View Options Window Help
"A1.T2.A1T2H010.HS-".STOP
"A1.A1H001.HS+-.START" I0.0
Main power switch multipurpose plant
"A1.A1H002.HS+-.OFF" I0.1
emergency switch OFF
"A1.A1H003.HS+-.LOC" I0.2
local operation mode switch
...\A1T2L002\\A1T2L002(A,1)\LISA+_A1T2L002
V Process value
2
OR
OB35
13/2
3
CMP_R
REAL-Com
OB35
13/1
IN1
IN2
200.0
GT
GE
EQ
LE
LT
A
Press F1 for help.
A/Sheet 1
OB35 A1T2H010 A1T2H010\3

28. If we want to clear the assignment of a CFC to a process tag type, its object properties have to be selected.

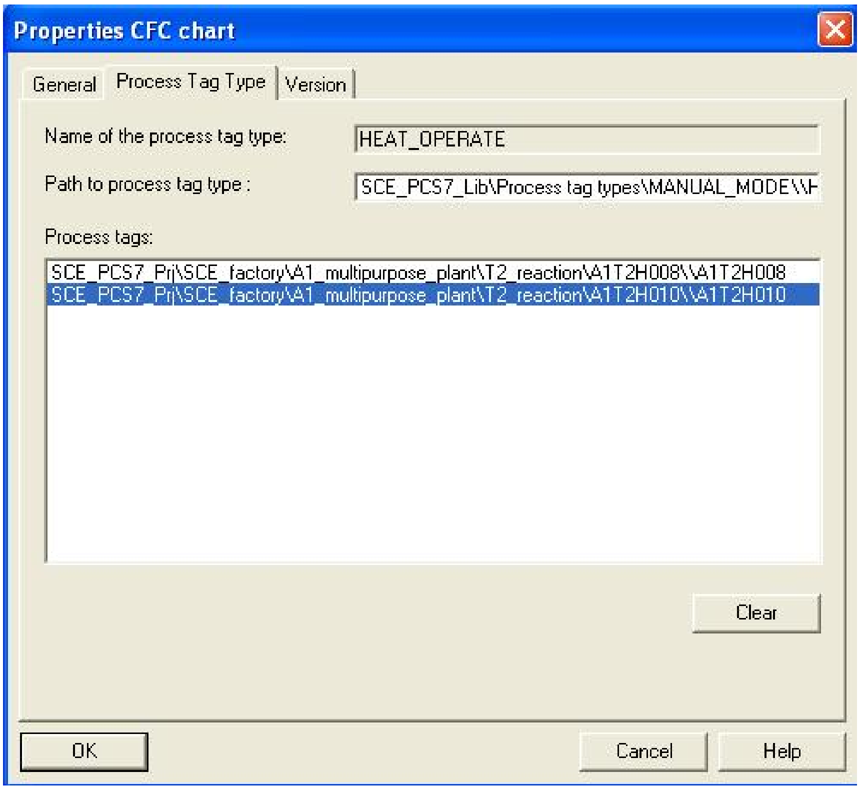
(→ A1T2H010 → Object Properties)



Displays properties of the selected object for editing.

29. Then we select the view Process Tag Type and clear the assignment there.

(→ Process Tag Type → Clear → OK)



Name of the process tag type: HEAT_OPERATE

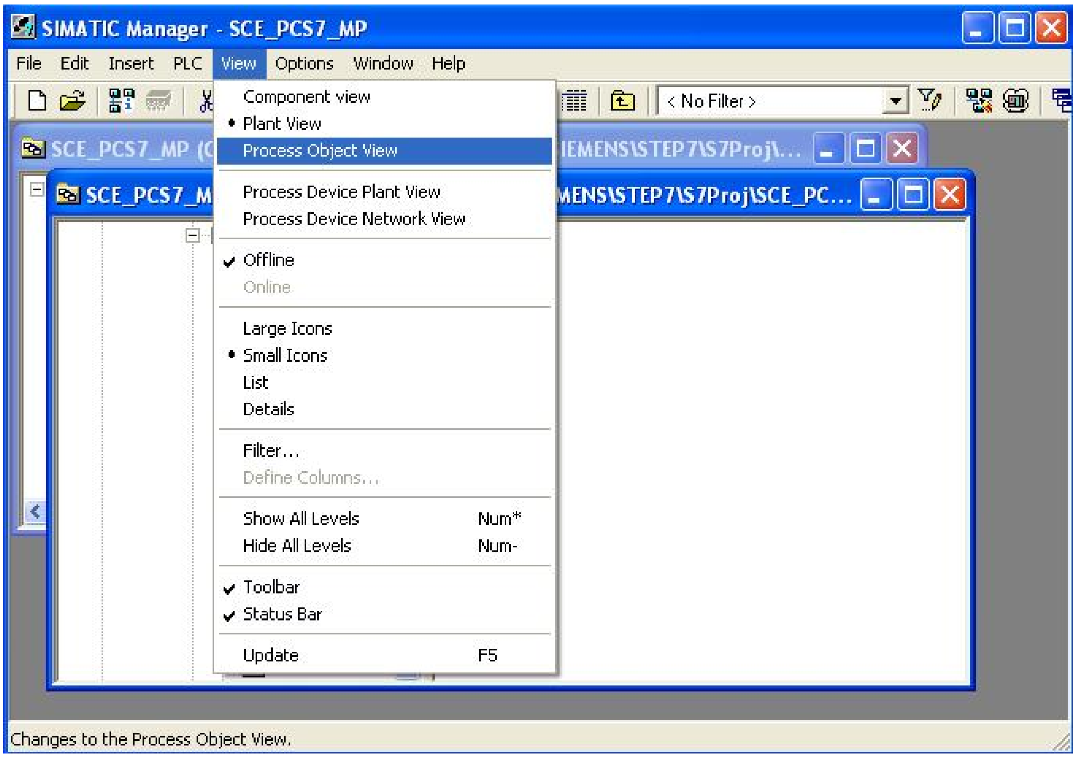
Path to process tag type : SCE_PCS7_Lib\Process tag types\MANUAL_MODE\\H

Process tags:

SCE_PCS7_Prj\SCE_factory\A1_multipurpose_plant\T2_reaction\A1T2H008\\A1T2H008
SCE_PCS7_Prj\SCE_factory\A1_multipurpose_plant\T2_reaction\A1T2H010\\A1T2H010

30. Another method to make changes in several already established charts without opening them is using the process object view.

    (→ View→ Process Object View)



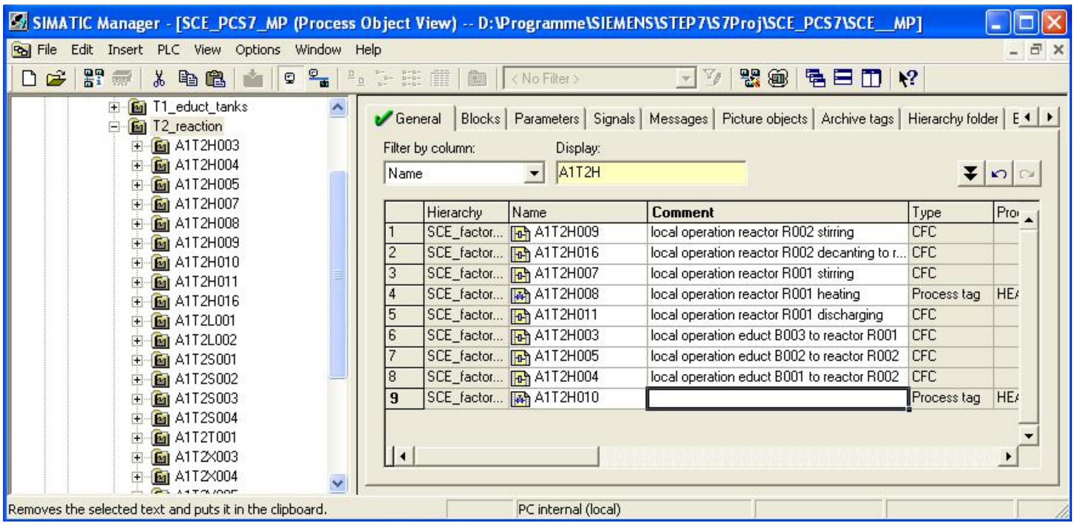
Changes to the Process Object View.

⚠

Note: Below, three examples are shown for using the process object view. Here, additional entries, texts, parameters and assignments can be changed also of course.

31. Since in large plants numerous folders exist, it is important to use the process object view filter advantageously. In addition, it is important to know that always only objects are displayed below the selected hierarchy folder.

(→ T2_reaction → Hierarchy folder → Filter by column: Name → Display: A1T2H)



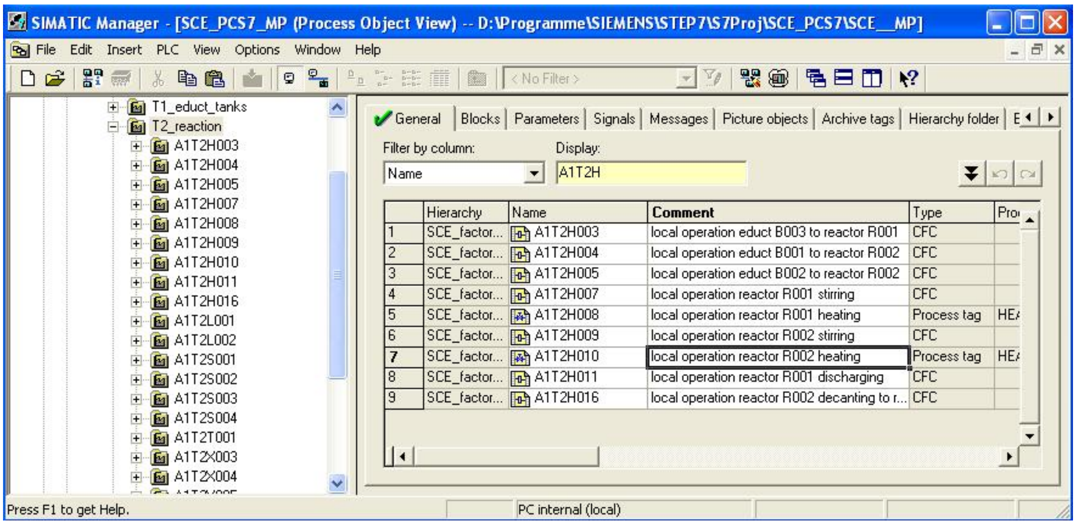
Filter by column: Name

Display: A1T2H

	Hierarchy	Name	Comment	Type	Pro
1	SCE_factor...	A1T2H009	local operation reactor R002 stirring	CFC	
2	SCE_factor...	A1T2H016	local operation reactor R002 decanting to r...	CFC	
3	SCE_factor...	A1T2H007	local operation reactor R001 stirring	CFC	
4	SCE_factor...	A1T2H008	local operation reactor R001 heating	Process tag	HE/
5	SCE_factor...	A1T2H011	local operation reactor R001 discharging	CFC	
6	SCE_factor...	A1T2H003	local operation educt B003 to reactor R001	CFC	
7	SCE_factor...	A1T2H005	local operation educt B002 to reactor R002	CFC	
8	SCE_factor...	A1T2H004	local operation educt B001 to reactor R002	CFC	
9	SCE_factor...	A1T2H010		Process tag	HE/

32. In this example, we enter the comment for chart ’A1T2H010’. We can also do this, of course, by first copying the text of ’A1T2H008’ and then change it.

(→ A1T2H010 → local operation reactor R002 heating)



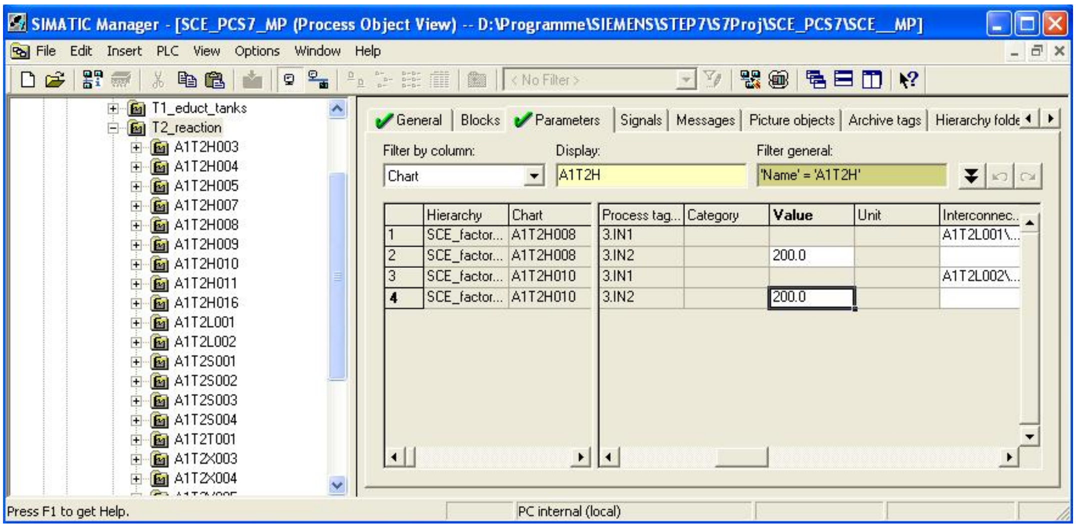
Filter by column: Name

Display: A1T2H

	Hierarchy	Name	Comment	Type	Pro
1	SCE_factor...	A1T2H003	local operation educt B003 to reactor R001	CFC	
2	SCE_factor...	A1T2H004	local operation educt B001 to reactor R002	CFC	
3	SCE_factor...	A1T2H005	local operation educt B002 to reactor R002	CFC	
4	SCE_factor...	A1T2H007	local operation reactor R001 stirring	CFC	
5	SCE_factor...	A1T2H008	local operation reactor R001 heating	Process tag	HE/
6	SCE_factor...	A1T2H009	local operation reactor R002 stirring	CFC	
7	SCE_factor...	A1T2H010	local operation reactor R002 heating	Process tag	HE/
8	SCE_factor...	A1T2H011	local operation reactor R001 discharging	CFC	
9	SCE_factor...	A1T2H016	local operation reactor R002 decanting to r...	CFC	

33. Here, we see how, with the process object view, parameters can be changed in our charts ’A1T2H008’ and ’A1T2H010’.

    (→ T2_reaction → Parameters → Filter by column: Chart → Display: A1T2H)



SIMATIC Manager - [SCE_PCS7_MP (Process Object View) -- D:\Programme\SIEMENS\STEP7\S7Proj\SCE_PCS7\SCE__MP]

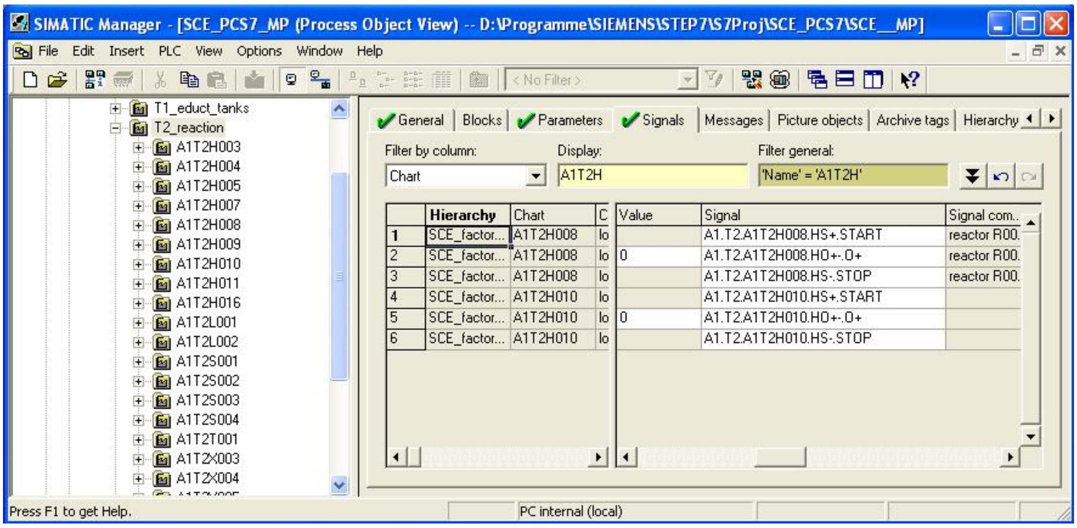
Filter by column: Chart
Display: A1T2H
Filter general: 'Name' = 'A1T2H'

	Hierarchy	Chart	Process tag...	Category	Value	Unit	Interconnec...
1	SCE_factor...	A1T2H008	3.IN1				A1T2L001\...
2	SCE_factor...	A1T2H008	3.IN2		200.0		
3	SCE_factor...	A1T2H010	3.IN1				A1T2L002\...
4	SCE_factor...	A1T2H010	3.IN2		200.0		

Press F1 to get Help. PC internal (local)

34. Here we see how, with the process object view, signal assignments can be changed in our charts ’A1T2H008’ and’A1T2H010’.

    (→ T2_reaction → Signals → Filter by column: Chart → A1T2H)



SIMATIC Manager - [SCE_PCS7_MP (Process Object View) -- D:\Programme\SIEMENS\STEP7\S7Proj\SCE_PCS7\SCE__MP]

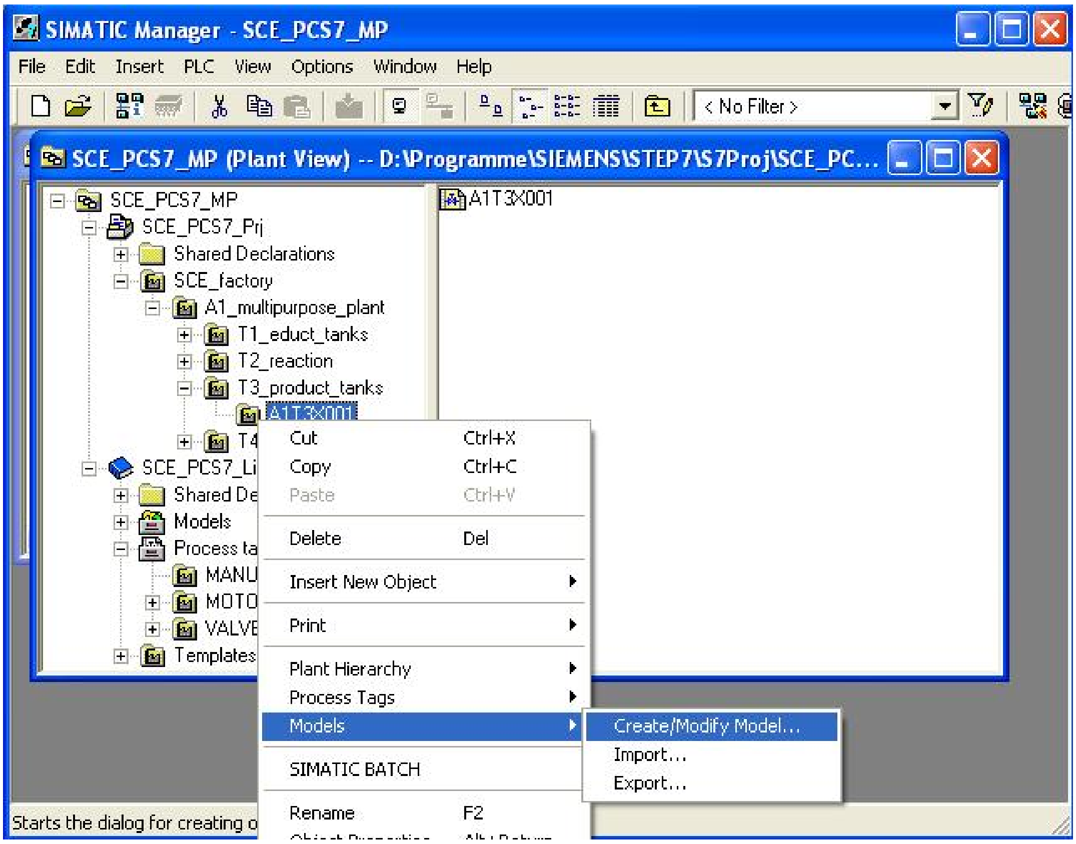
Filter by column: Chart
Display: A1T2H
Filter general: 'Name' = 'A1T2H'

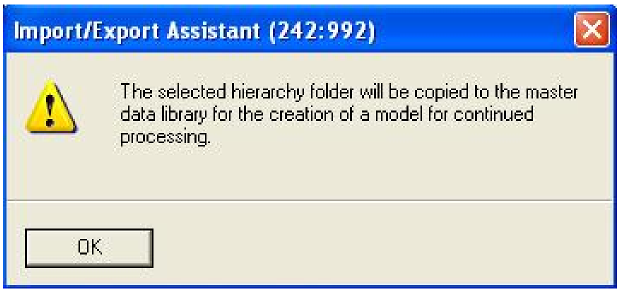
	Hierarchy	Chart	C	Value	Signal	Signal com...
1	SCE_factor...	A1T2H008	lo		A1.T2.A1T2H008.HS+.START	reactor R00.
2	SCE_factor...	A1T2H008	lo	0	A1.T2.A1T2H008.HO+-.0+	reactor R00.
3	SCE_factor...	A1T2H008	lo		A1.T2.A1T2H008.HS-.STOP	reactor R00.
4	SCE_factor...	A1T2H010	lo		A1.T2.A1T2H010.HS+.START	
5	SCE_factor...	A1T2H010	lo	0	A1.T2.A1T2H010.HO+-.0+	
6	SCE_factor...	A1T2H010	lo		A1.T2.A1T2H010.HS-.STOP	

Press F1 to get Help. PC internal (local)

35. To duplicate folder structures that were already created and tested, a model is generated from them. In the current example, we are taking the folder 'A1T3X001' for the inlet valve Product Tank B001.

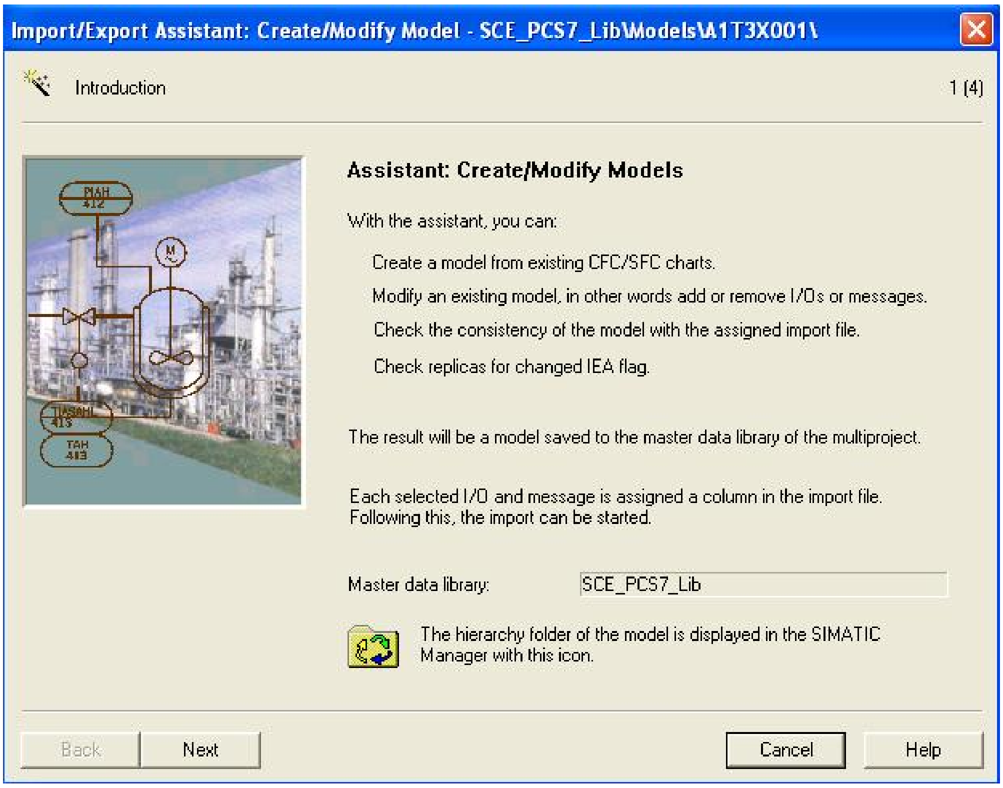
    (→ A1T3X001 → Models → Create/Modify Model → OK)




Import/Export Assistant (242:992)

The selected hierarchy folder will be copied to the master data library for the creation of a model for continued processing.

36. Then, information for the assistant is displayed. (→ Next)



Import/Export Assistant: Create/Modify Model - SCE_PCS7_Lib\Models\A1T3X001\

Introduction 1 (4)

Assistant: Create/Modify Models

With the assistant, you can:

Create a model from existing CFC/SFC charts.

Modify an existing model, in other words add or remove I/Os or messages.

Check the consistency of the model with the assigned import file.

Check replicas for changed IEA flag.

The result will be a model saved to the master data library of the multiproject.

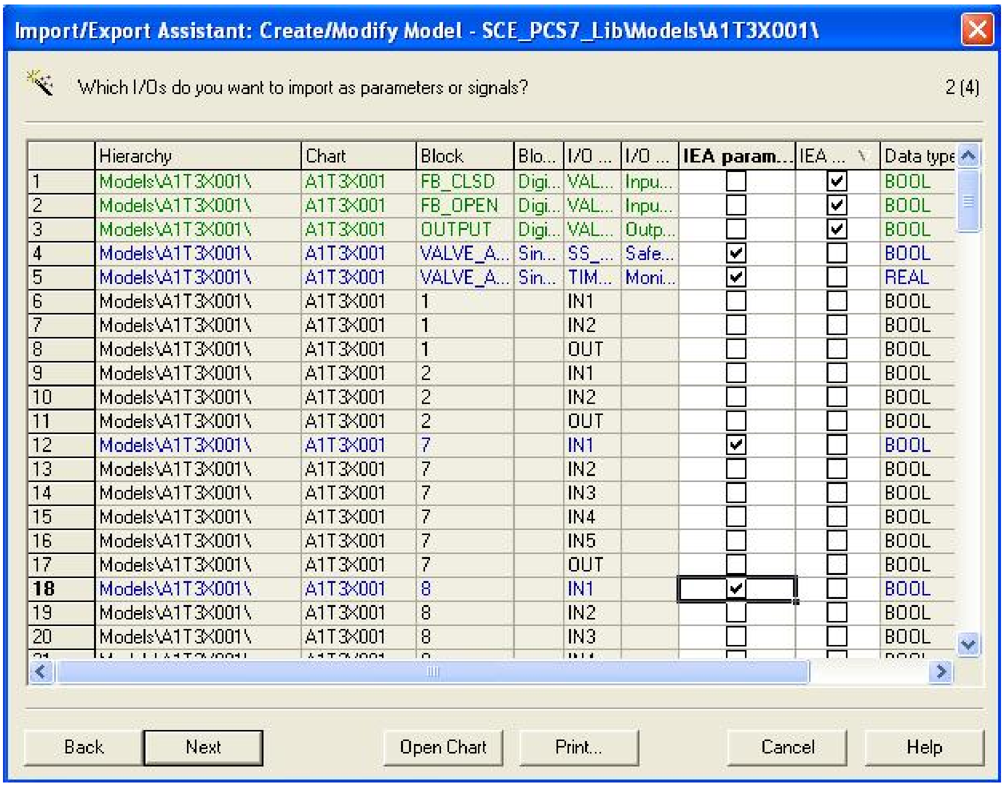
Each selected I/O and message is assigned a column in the import file.
Following this, the import can be started.

Master data library: SCE_PCS7_Lib

The hierarchy folder of the model is displayed in the SIMATIC Manager with this icon.

37. Net, we specify which parameters (blue) and signals (green) are displayed in the Export Assistant.

(→ IEA param. → IEA signals → Next)

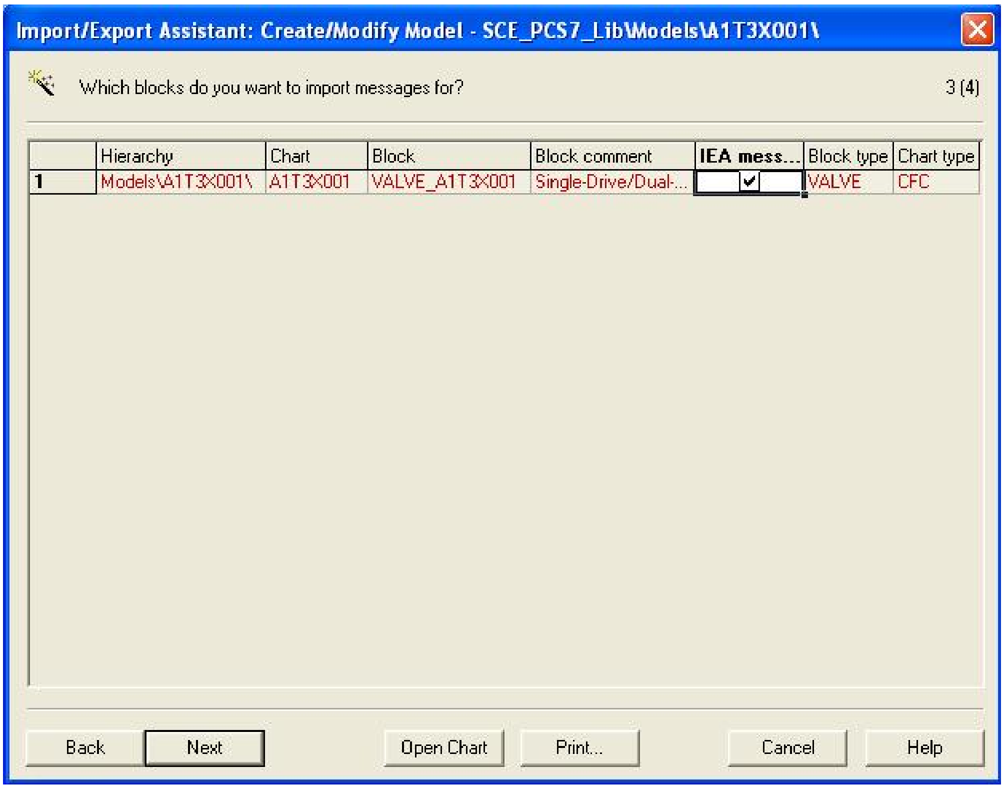


Import/Export Assistant: Create/Modify Model - SCE_PCS7_Lib\Models\A1T3X001\

Which I/Os do you want to import as parameters or signals? 2 (4)

	Hierarchy	Chart	Block	Blo...	I/O ...	I/O ...	IEA param...	IEA ...	Data type
1	Models\A1T3X001\	A1T3X001	FB_CLSD	Digi...	VAL...	Inpu...	□	☑	BOOL
2	Models\A1T3X001\	A1T3X001	FB_OPEN	Digi...	VAL...	Inpu...	□	☑	BOOL
3	Models\A1T3X001\	A1T3X001	OUTPUT	Digi...	VAL...	Outp...	□	☑	BOOL
4	Models\A1T3X001\	A1T3X001	VALVE_A...	Sin...	SS_...	Safe...	☑	□	BOOL
5	Models\A1T3X001\	A1T3X001	VALVE_A...	Sin...	TIM...	Moni...	☑	□	REAL
6	Models\A1T3X001\	A1T3X001	1		IN1		□	□	BOOL
7	Models\A1T3X001\	A1T3X001	1		IN2		□	□	BOOL
8	Models\A1T3X001\	A1T3X001	1		OUT		□	□	BOOL
9	Models\A1T3X001\	A1T3X001	2		IN1		□	□	BOOL
10	Models\A1T3X001\	A1T3X001	2		IN2		□	□	BOOL
11	Models\A1T3X001\	A1T3X001	2		OUT		□	□	BOOL
12	Models\A1T3X001\	A1T3X001	7		IN1		☑	□	BOOL
13	Models\A1T3X001\	A1T3X001	7		IN2		□	□	BOOL
14	Models\A1T3X001\	A1T3X001	7		IN3		□	□	BOOL
15	Models\A1T3X001\	A1T3X001	7		IN4		□	□	BOOL
16	Models\A1T3X001\	A1T3X001	7		IN5		□	□	BOOL
17	Models\A1T3X001\	A1T3X001	7		OUT		□	□	BOOL
18	Models\A1T3X001\	A1T3X001	8		IN1		☑	□	BOOL
19	Models\A1T3X001\	A1T3X001	8		IN2		□	□	BOOL
20	Models\A1T3X001\	A1T3X001	8		IN3		□	□	BOOL

38. Next, we specify the messages that are displayed in the import-export assistant. (→ IEA message → Next)



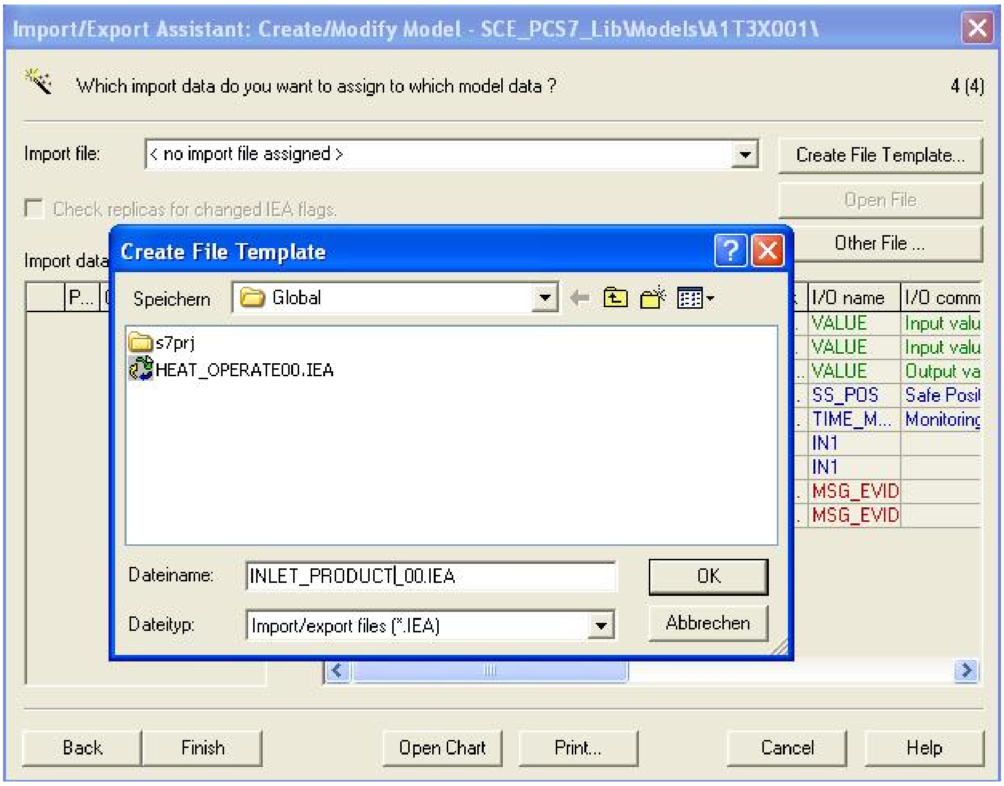
Import/Export Assistant: Create/Modify Model - SCE_PCS7_Lib\Models\A1T3X001\

Which blocks do you want to import messages for? 3 (4)

	Hierarchy	Chart	Block	Block comment	IEA mess...	Block type	Chart type
1	Models\A1T3X001\	A1T3X001	VALVE_A1T3X001	Single-Drive/Dual-...	☑	VALVE	CFC

39. We then generate a file template.

(→ Create File Template → INLET_PRODUCT.IEA → OK)



Import/Export Assistant: Create/Modify Model - SCE_PCS7_Lib\Models\A1T3X001\

Which import data do you want to assign to which model data ? 4 (4)

Import file: < no import file assigned >

□ Check replicas for changed IEA flags.

Import data

I/O name	I/O comm
VALUE	Input valu
VALUE	Input valu
VALUE	Output va
SS_POS	Safe Posi
TIME_M...	Monitoring
IN1	
IN1	
MSG_EVID	
MSG_EVID	

Create File Template

Speichern: Global

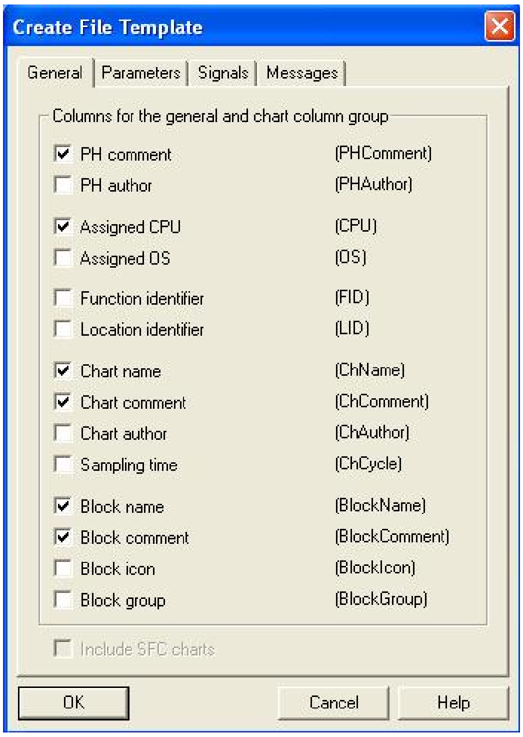
s7prj
HEAT_OPERATE00.IEA

Dateiname: INLET_PRODUCT_00.IEA

Dateityp: Import/export files (*.IEA)

40. In the following dialog we can select which general columns are displayed in the import file.

    (→ General → PH comment → Assigned CPU → Chart name → Chart comment → Block name → Block comment)



Create File Template

Columns for the general and chart column group

☑ PH comment (PHComment)
☐ PH author (PHAuthor)

☑ Assigned CPU (CPU)
☐ Assigned OS (OS)

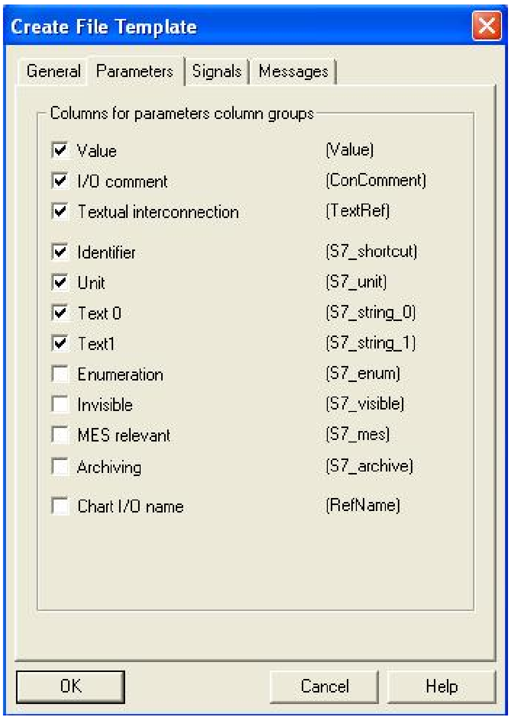
☐ Function identifier (FID)
☐ Location identifier (LID)

☑ Chart name (ChName)
☑ Chart comment (ChComment)
☐ Chart author (ChAuthor)
☐ Sampling time (ChCycle)

☑ Block name (BlockName)
☑ Block comment (BlockComment)
☐ Block icon (BlockIcon)
☐ Block group (BlockGroup)

☐ Include SFC charts

41. Here we select which columns are displayed for the parameters in the import file. (→ Parameters → Value → I/O comment → Textual interconnections → Identifier → Unit→ Text 0 → Text 1)



Create File Template

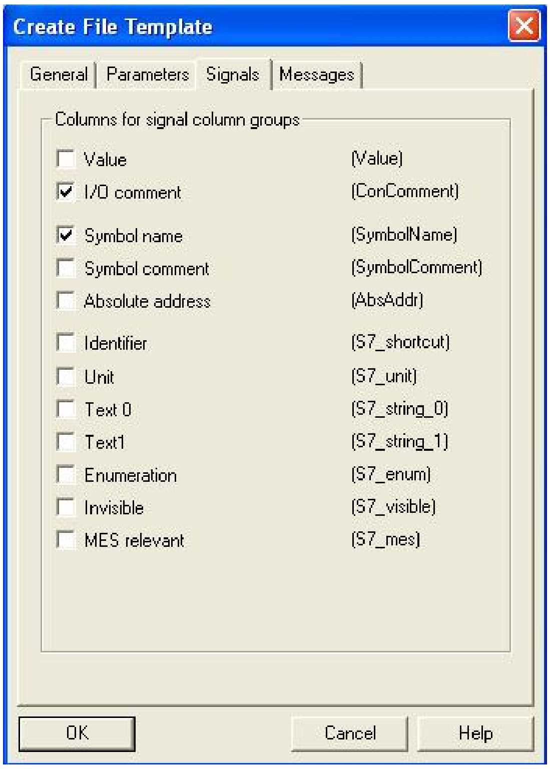
Columns for parameters column groups

☑ Value (Value)
☑ I/O comment (ConComment)
☑ Textual interconnection (TextRef)

☑ Identifier (S7_shortcut)
☑ Unit (S7_unit)
☑ Text 0 (S7_string_0)
☑ Text1 (S7_string_1)
☐ Enumeration (S7_enum)
☐ Invisible (S7_visible)
☐ MES relevant (S7_mes)
☐ Archiving (S7_archive)

☐ Chart I/O name (RefName)

42. Here we select which columns for the signals are displayed in the import file. (→ Signals → I/O comment → Symbol name)

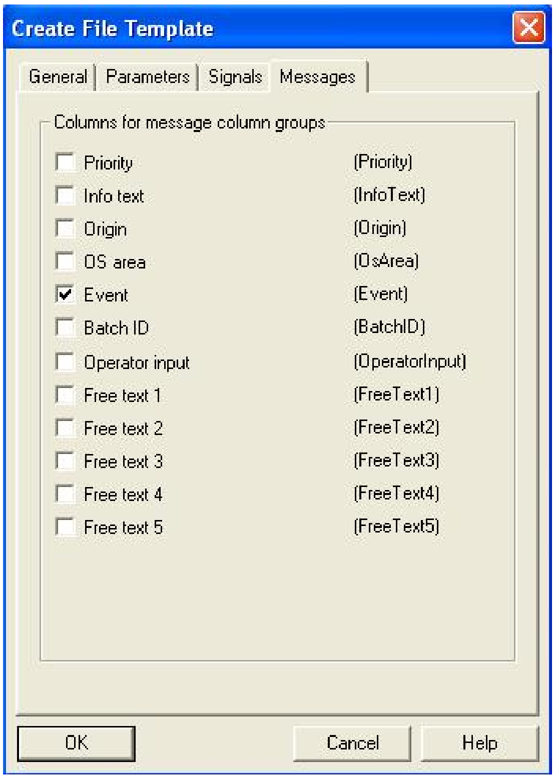


Create File Template

Columns for signal column groups

☐ Value	(Value)
☑ I/O comment	(ConComment)
☑ Symbol name	(SymbolName)
☐ Symbol comment	(SymbolComment)
☐ Absolute address	(AbsAddr)
☐ Identifier	(S7_shortcut)
☐ Unit	(S7_unit)
☐ Text 0	(S7_string_0)
☐ Text1	(S7_string_1)
☐ Enumeration	(S7_enum)
☐ Invisible	(S7_visible)
☐ MES relevant	(S7_mes)

43. Here we select which columns for the messages are displayed in the import file. (→ Messages → Event → OK)

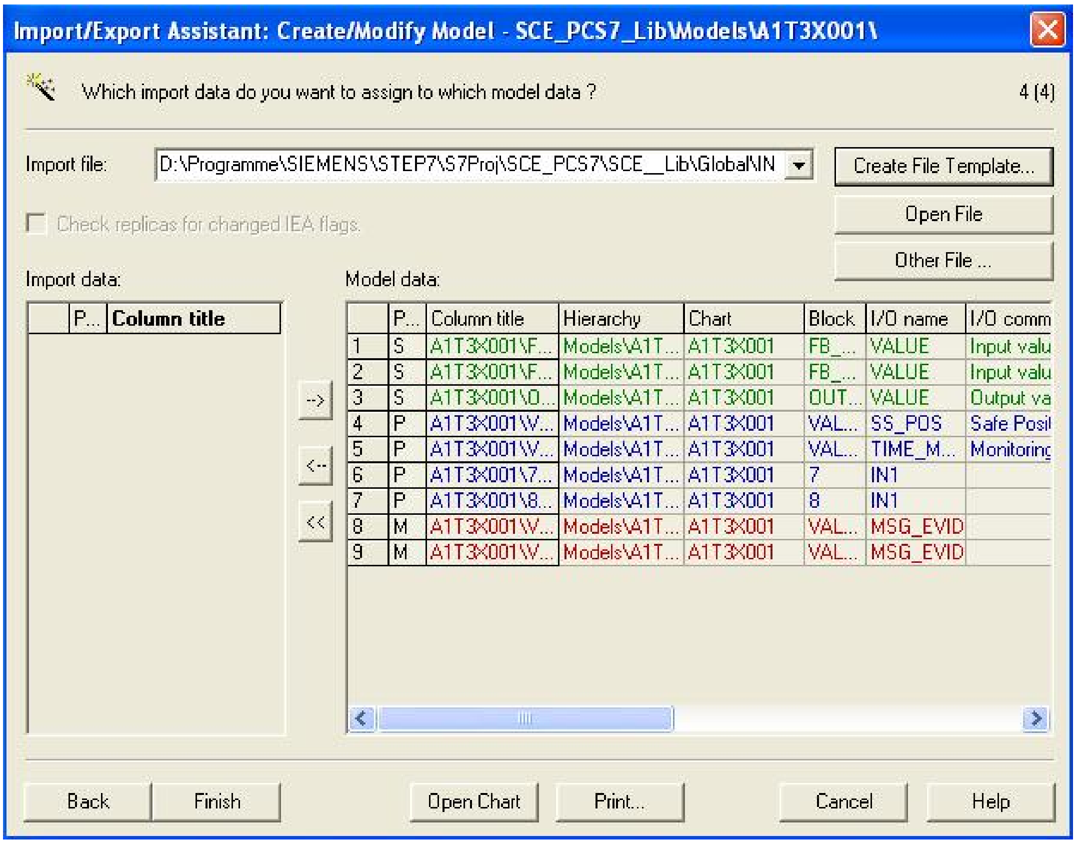


Create File Template

Columns for message column groups

☐ Priority	(Priority)
☐ Info text	(InfoText)
☐ Origin	(Origin)
☐ OS area	(OsArea)
☑ Event	(Event)
☐ Batch ID	(BatchID)
☐ Operator input	(OperatorInput)
☐ Free text 1	(FreeText1)
☐ Free text 2	(FreeText2)
☐ Free text 3	(FreeText3)
☐ Free text 4	(FreeText4)
☐ Free text 5	(FreeText5)

44. We now open the import file that we generated. (→ Open file)



Import/Export Assistant: Create/Modify Model - SCE_PCS7_Lib\Models\A1T3X001\

Which import data do you want to assign to which model data ? 4 (4)

Import file: D:\Programme\SIEMENS\STEP7\S7Proj\SCE_PCS7\SCE__Lib\Global\IN

□ Check replicas for changed IEA flags.

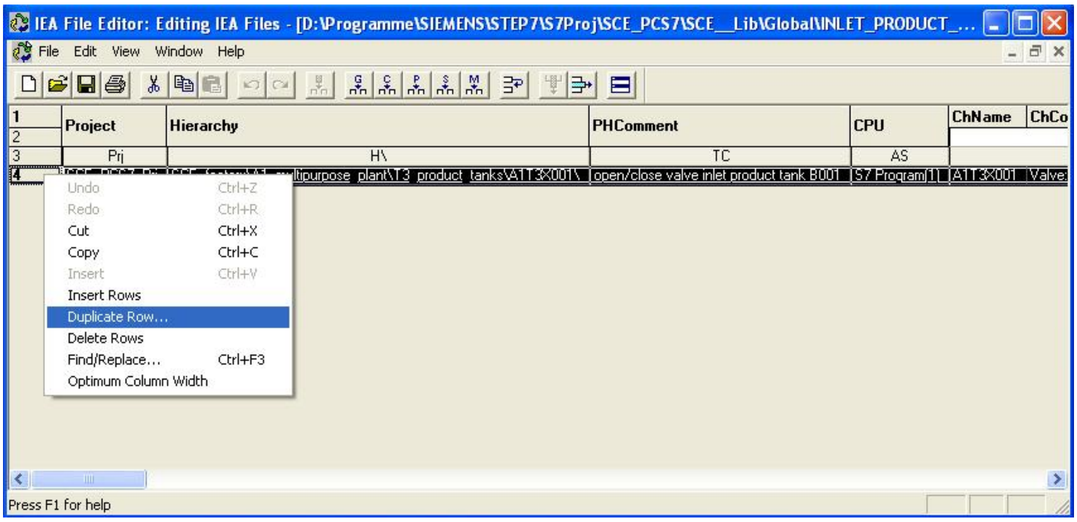
Import data:

	P...	Column title

Model data:

	P...	Column title	Hierarchy	Chart	Block	I/O name	I/O comm
1	S	A1T3X001\F...	Models\A1T...	A1T3X001	FB_...	VALUE	Input valu
2	S	A1T3X001\F...	Models\A1T...	A1T3X001	FB_...	VALUE	Input valu
3	S	A1T3X001\O...	Models\A1T...	A1T3X001	OUT...	VALUE	Output va
4	P	A1T3X001\V...	Models\A1T...	A1T3X001	VAL...	SS_POS	Safe Posit
5	P	A1T3X001\V...	Models\A1T...	A1T3X001	VAL...	TIME_M...	Monitoring
6	P	A1T3X001\7...	Models\A1T...	A1T3X001	7	IN1	
7	P	A1T3X001\8...	Models\A1T...	A1T3X001	8	IN1	
8	M	A1T3X001\V...	Models\A1T...	A1T3X001	VAL...	MSG_EVID	
9	M	A1T3X001\V...	Models\A1T...	A1T3X001	VAL...	MSG_EVID	

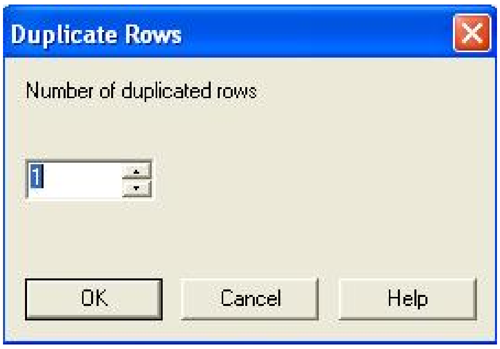
45. We then duplicate the row from hierarchy folder ’A1T3X001’ once to set up hierarchy folder ’A1T3X002’. (→ Duplicate Row → 1 → OK)



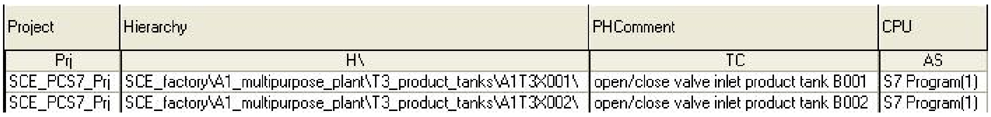
IEA File Editor: Editing IEA Files - [D:\Programme\SIEMENS\STEP7\S7Proj\SCE_PCS7\SCE__Lib\Global\INLET_PRODUCT_...

	Project	Hierarchy	PHComment	CPU	ChName	ChCo
1 2						
3	Prj	H\	TC	AS		
4		...ltipurpose_plant\T3_product_tanks\A1T3X001\	open/close valve inlet product tank B001	S7 Program(1)	A1T3X001	Valve:

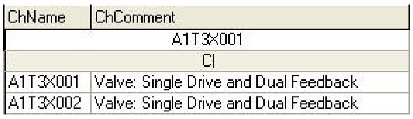
Press F1 for help



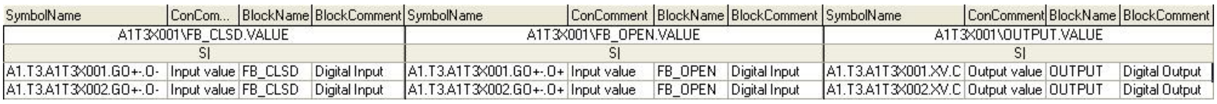
46. As is shown here, we can then change the entries for hierarchy folders ’A1T3X001’ and ’A1T3X002’ in the columns of the import file. .



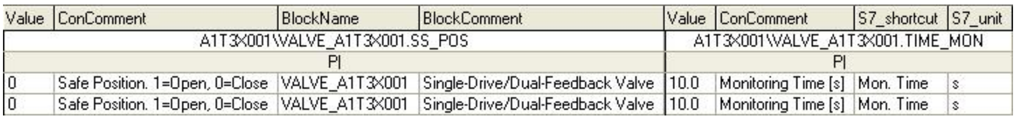
Project	Hierarchy	PHComment	CPU
Prj	H\	TC	AS
SCE_PCS7_Prj	SCE_factory\A1_multipurpose_plant\T3_product_tanks\A1T3X001\	open/close valve inlet product tank B001	S7 Program(1)
SCE_PCS7_Prj	SCE_factory\A1_multipurpose_plant\T3_product_tanks\A1T3X002\	open/close valve inlet product tank B002	S7 Program(1)



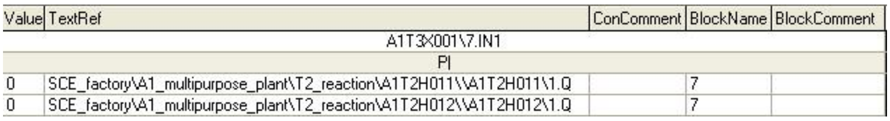
ChName	ChComment
A1T3X001	
CI	
A1T3X001	Valve: Single Drive and Dual Feedback
A1T3X002	Valve: Single Drive and Dual Feedback



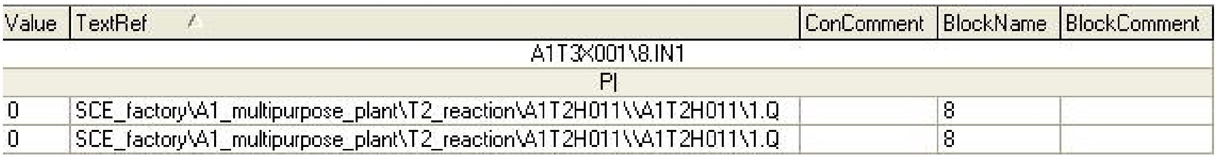
SymbolName	ConCom...	BlockName	BlockComment	SymbolName	ConComment	BlockName	BlockComment	SymbolName	ConComment	BlockName	BlockComment
A1T3X001\FB_CLSD.VALUE				A1T3X001\FB_OPEN.VALUE				A1T3X001\OUTPUT.VALUE			
SI				SI				SI			
A1.T3.A1T3X001.GO+-.O-	Input value	FB_CLSD	Digital Input	A1.T3.A1T3X001.GO+-.O+	Input value	FB_OPEN	Digital Input	A1.T3.A1T3X001.XV.C	Output value	OUTPUT	Digital Output
A1.T3.A1T3X002.GO+-.O-	Input value	FB_CLSD	Digital Input	A1.T3.A1T3X002.GO+-.O+	Input value	FB_OPEN	Digital Input	A1.T3.A1T3X002.XV.C	Output value	OUTPUT	Digital Output



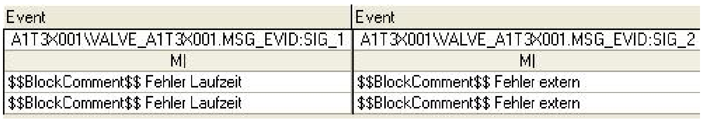
Value	ConComment	BlockName	BlockComment	Value	ConComment	S7_shortcut	S7_unit
A1T3X001\VALVE_A1T3X001.SS_POS				A1T3X001\VALVE_A1T3X001.TIME_MON			
PI				PI			
0	Safe Position. 1=Open, 0=Close	VALVE_A1T3X001	Single-Drive/Dual-Feedback Valve	10.0	Monitoring Time [s]	Mon. Time	s
0	Safe Position. 1=Open, 0=Close	VALVE_A1T3X001	Single-Drive/Dual-Feedback Valve	10.0	Monitoring Time [s]	Mon. Time	s



Value	TextRef	ConComment	BlockName	BlockComment
A1T3X001\7.IN1				
PI				
0	SCE_factory\A1_multipurpose_plant\T2_reaction\A1T2H011\\A1T2H011\1.Q		7	
0	SCE_factory\A1_multipurpose_plant\T2_reaction\A1T2H012\\A1T2H012\1.Q		7	

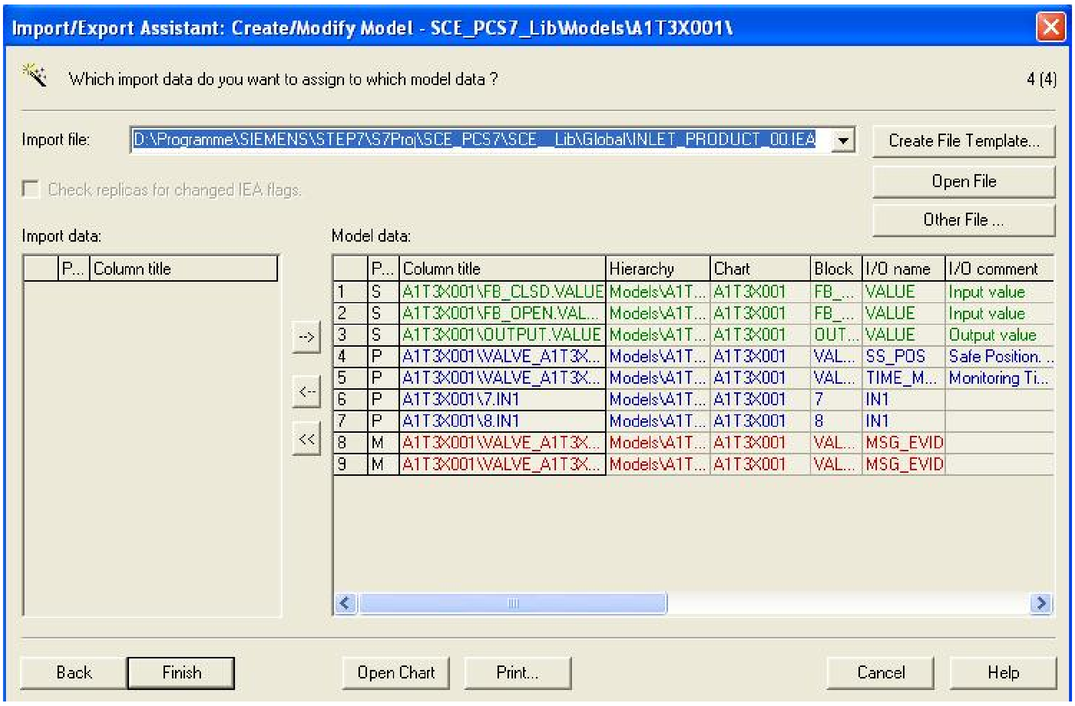


Value	TextRef	ConComment	BlockName	BlockComment
A1T3X001\8.IN1				
PI				
0	SCE_factory\A1_multipurpose_plant\T2_reaction\A1T2H011\\A1T2H011\1.Q		8	
0	SCE_factory\A1_multipurpose_plant\T2_reaction\A1T2H011\\A1T2H011\1.Q		8	



Event	Event
A1T3X001\VALVE_A1T3X001.MSG_EVID:SIG_1	A1T3X001\VALVE_A1T3X001.MSG_EVID:SIG_2
MI	MI
$$BlockComment$$ Fehler Laufzeit	$$BlockComment$$ Fehler extern
$$BlockComment$$ Fehler Laufzeit	$$BlockComment$$ Fehler extern

47. The model can then be completed. (→ Finish)



Import/Export Assistant: Create/Modify Model - SCE_PCS7_Lib\Models\A1T3X001\

Which import data do you want to assign to which model data ? 4 (4)

Import file: D:\Programme\SIEMENS\STEP7\S7Proj\SCE_PCS7\SCE__Lib\Global\INLET_PRODUCT_00.IEA

□ Check replicas for changed IEA flags.

Import data:

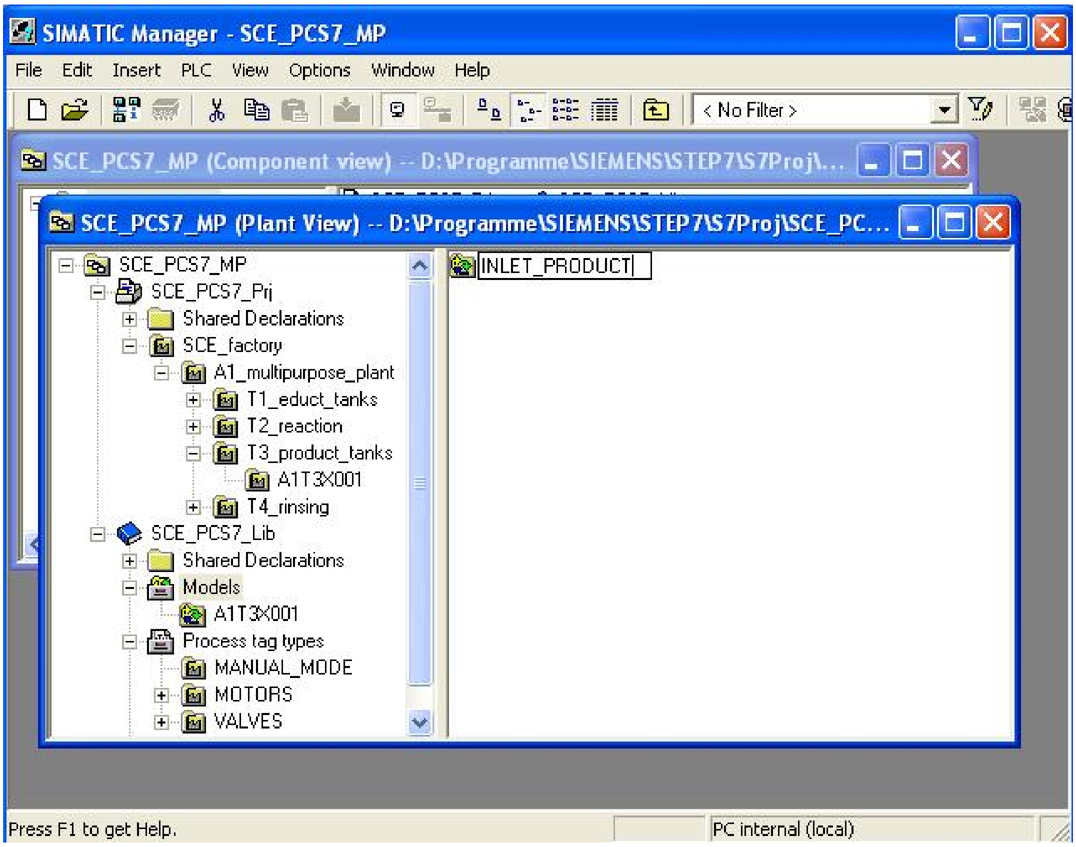
	P...	Column title

Model data:

	P...	Column title	Hierarchy	Chart	Block	I/O name	I/O comment
1	S	A1T3X001\FB_CLSD.VALUE	Models\A1T...	A1T3X001	FB_...	VALUE	Input value
2	S	A1T3X001\FB_OPEN.VAL...	Models\A1T...	A1T3X001	FB_...	VALUE	Input value
3	S	A1T3X001\OUTPUT.VALUE	Models\A1T...	A1T3X001	OUT...	VALUE	Output value
4	P	A1T3X001\VALVE_A1T3X...	Models\A1T...	A1T3X001	VAL...	SS_POS	Safe Position. ..
5	P	A1T3X001\VALVE_A1T3X...	Models\A1T...	A1T3X001	VAL...	TIME_M...	Monitoring Ti...
6	P	A1T3X001\7.IN1	Models\A1T...	A1T3X001	7	IN1	
7	P	A1T3X001\8.IN1	Models\A1T...	A1T3X001	8	IN1	
8	M	A1T3X001\VALVE_A1T3X...	Models\A1T...	A1T3X001	VAL...	MSG_EVID	
9	M	A1T3X001\VALVE_A1T3X...	Models\A1T...	A1T3X001	VAL...	MSG_EVID	

48. We still have to change the name of the model to 'INLET_PRODUCT'.

(→ INLET_PRODUCT)



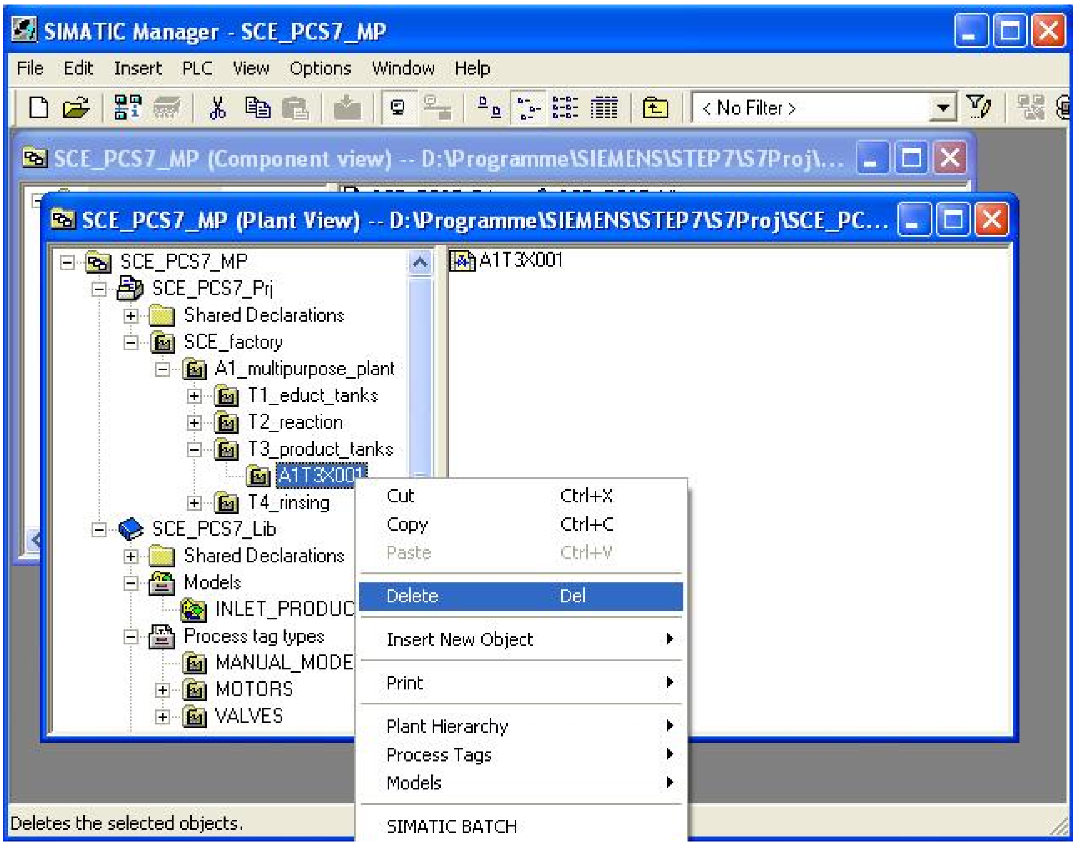
SIMATIC Manager - SCE_PCS7_MP

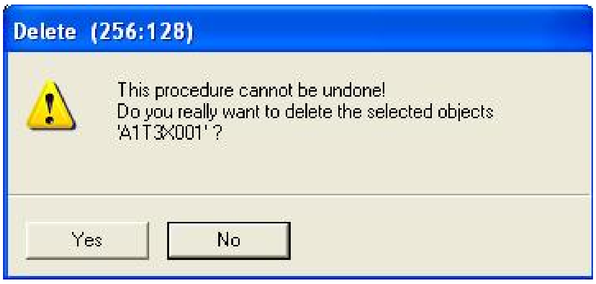
SCE_PCS7_MP (Component view) -- D:\Programme\SIEMENS\STEP7\S7Proj\...

SCE_PCS7_MP (Plant View) -- D:\Programme\SIEMENS\STEP7\S7Proj\SCE_PC...

INLET_PRODUCT

49. Before we can start importing, we have to delete the original hierarchy folder ’A1T3X001’. (→ A1T3X001 → Delete → Yes)

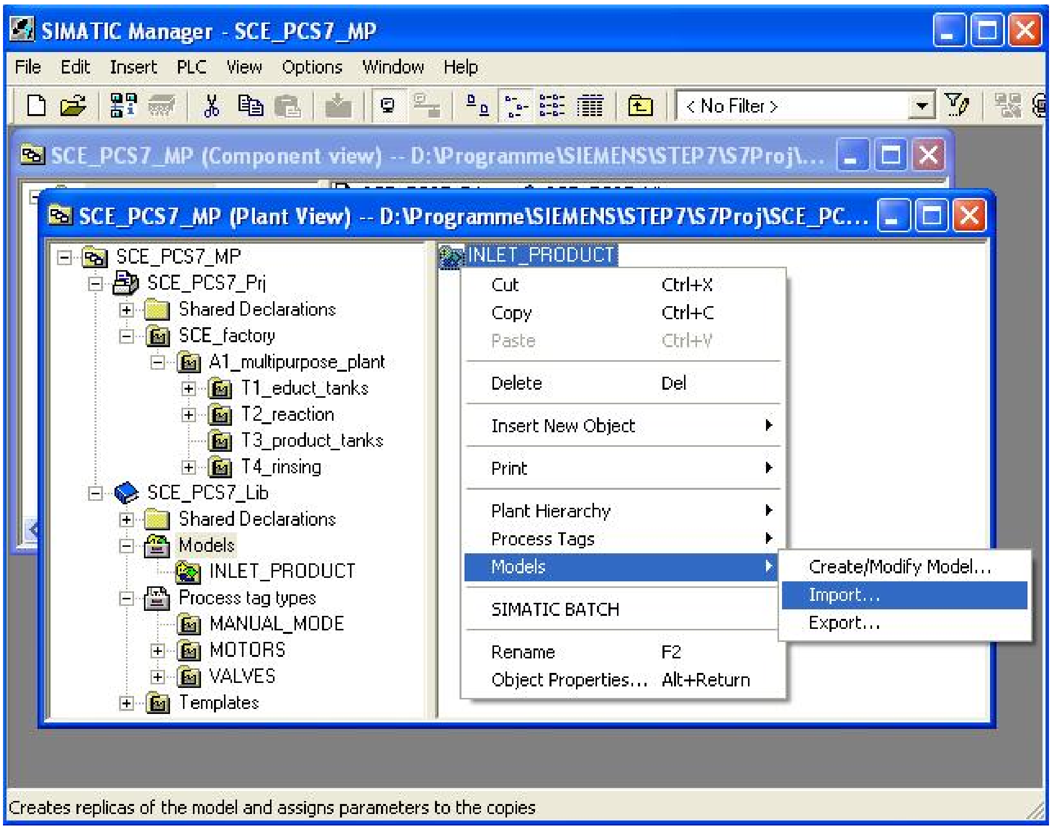




This procedure cannot be undone!
Do you really want to delete the selected objects
'A1T3X001' ?

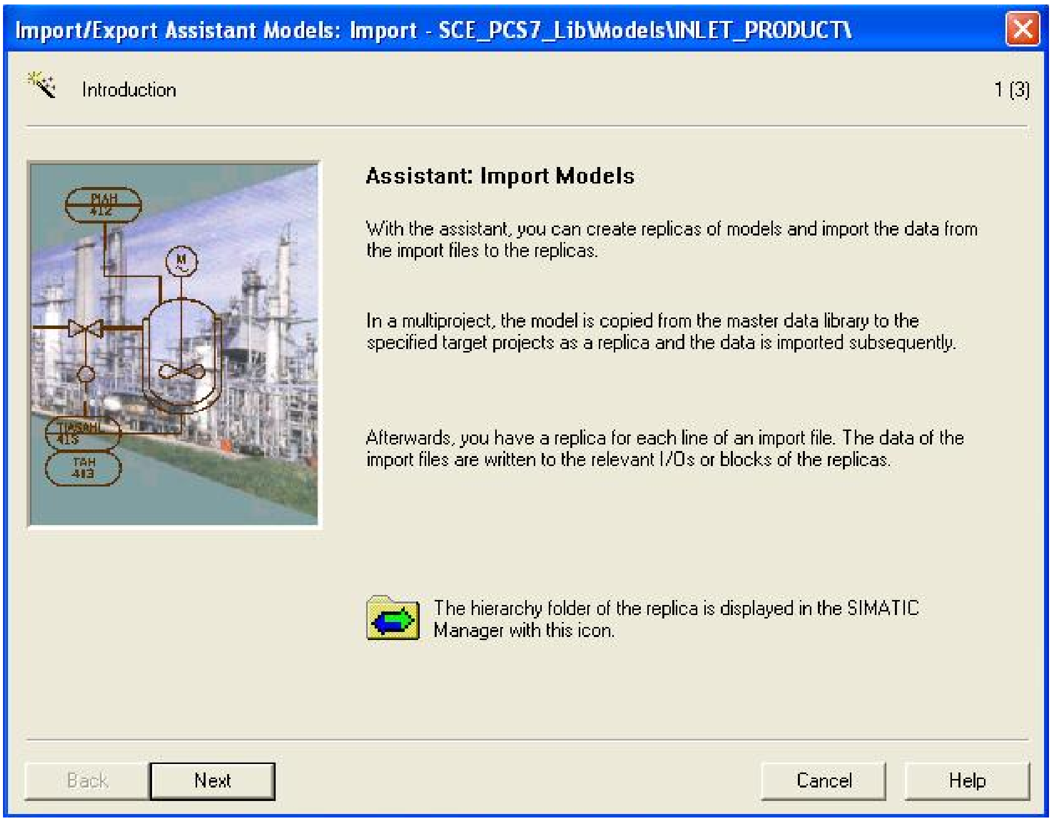
50. Now we are starting importing with the import file we set up.

(→ INLET_PRODUCT → Models → Import)



Creates replicas of the model and assigns parameters to the copies

51. Then, the information is displayed for the assistant. (→ Next)


Import/Export Assistant Models: Import - SCE_PCS7_Lib\Models\INLET_PRODUCT\

Introduction 1 (3)

Assistant: Import Models

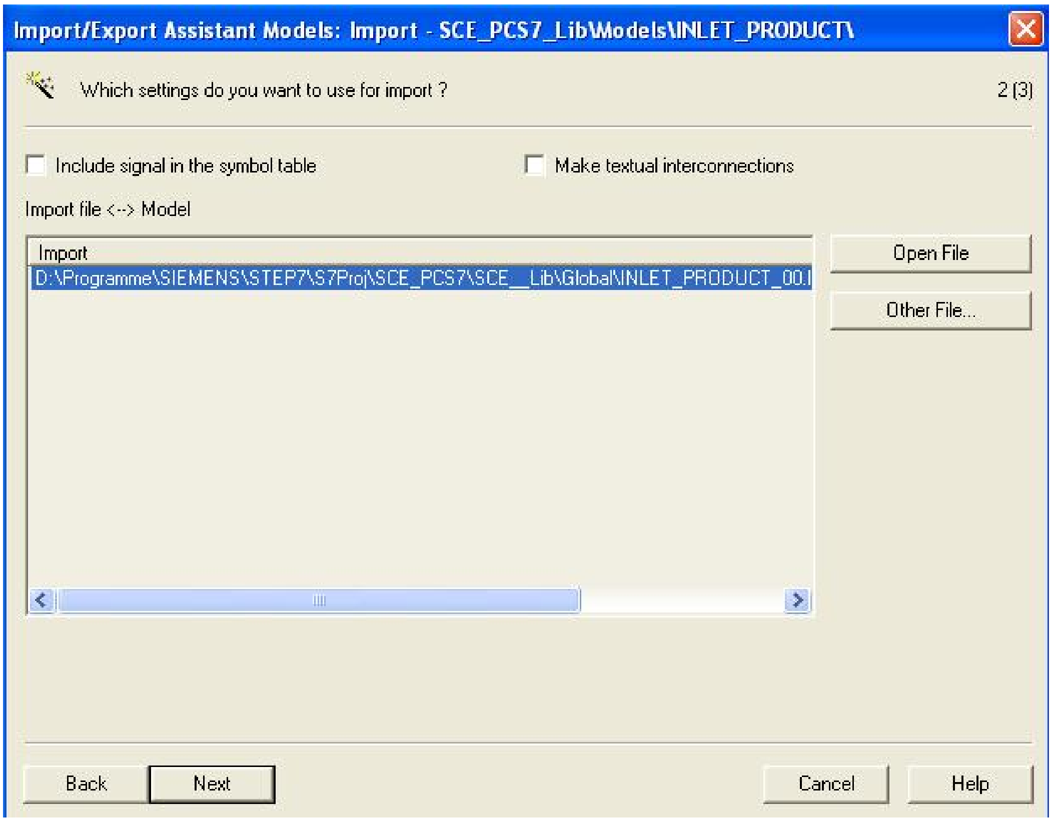
With the assistant, you can create replicas of models and import the data from the import files to the replicas.

In a multiproject, the model is copied from the master data library to the specified target projects as a replica and the data is imported subsequently.

Afterwards, you have a replica for each line of an import file. The data of the import files are written to the relevant I/Os or blocks of the replicas.

The hierarchy folder of the replica is displayed in the SIMATIC Manager with this icon.

52. We select the previously established import file and the option ’Make textual interconnections’. (→ Make textual interconnections → Next)



Import/Export Assistant Models: Import - SCE_PCS7_Lib\Models\INLET_PRODUCT\

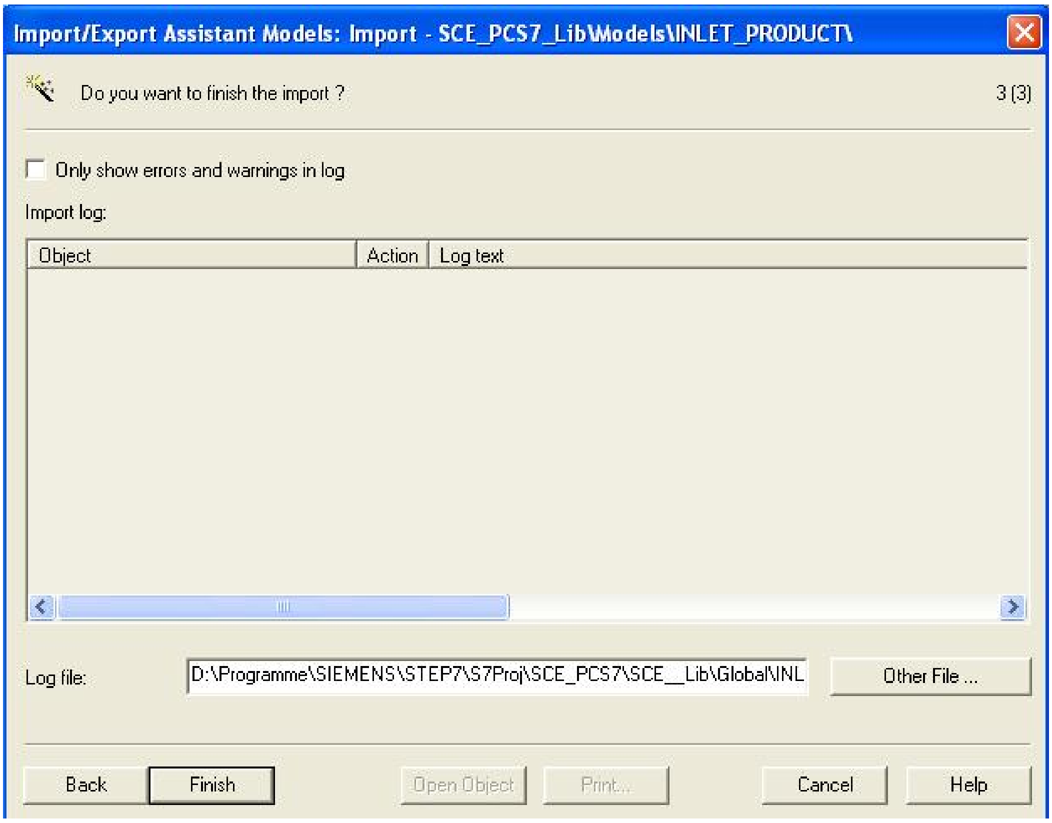
Which settings do you want to use for import ? 2 (3)

□ Include signal in the symbol table □ Make textual interconnections

Import file <--> Model

Import
D:\Programme\SIEMENS\STEP7\S7Proj\SCE_PCS7\SCE__Lib\Global\INLET_PRODUCT_00.I

53. We then start importing. (→ Finish)



Import/Export Assistant Models: Import - SCE_PCS7_Lib\Models\INLET_PRODUCT\

Do you want to finish the import ? 3 (3)

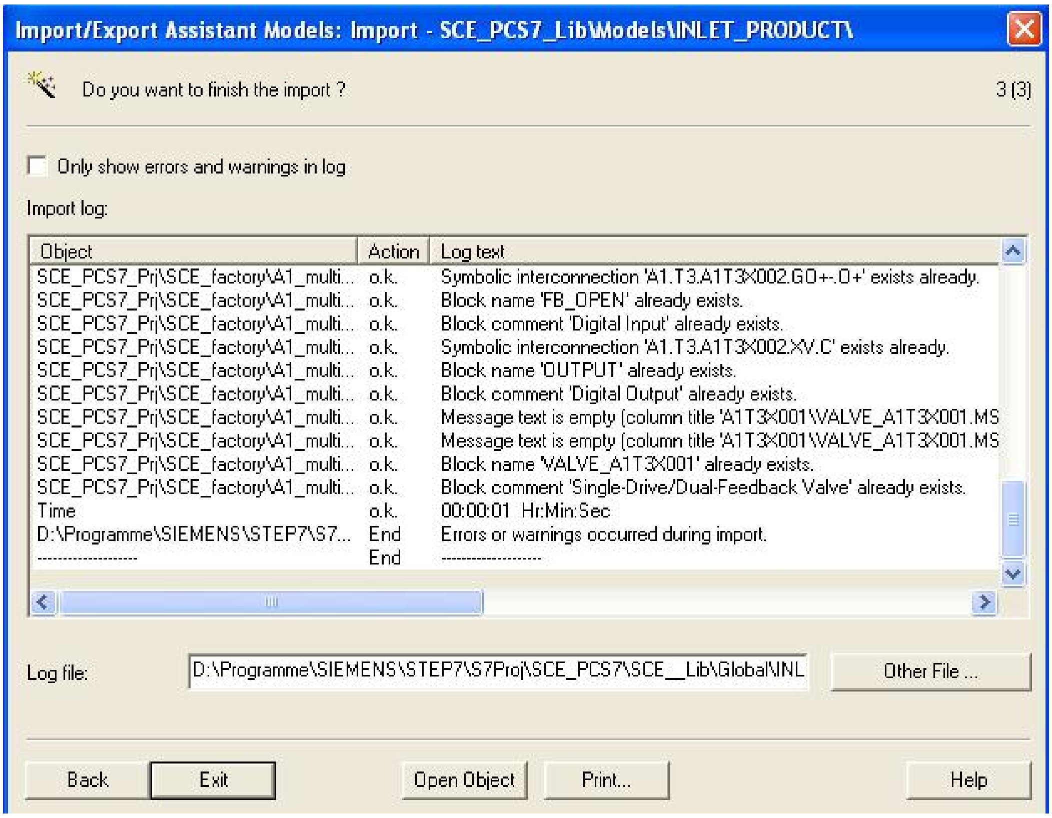
□ Only show errors and warnings in log

Import log:

Object	Action	Log text

Log file: D:\Programme\SIEMENS\STEP7\S7Proj\SCE_PCS7\SCE__Lib\Global\INL

54. Then, an import log is displayed. (→ Exit)



Import/Export Assistant Models: Import - SCE_PCS7_Lib\Models\INLET_PRODUCT\

Do you want to finish the import ? 3 (3)

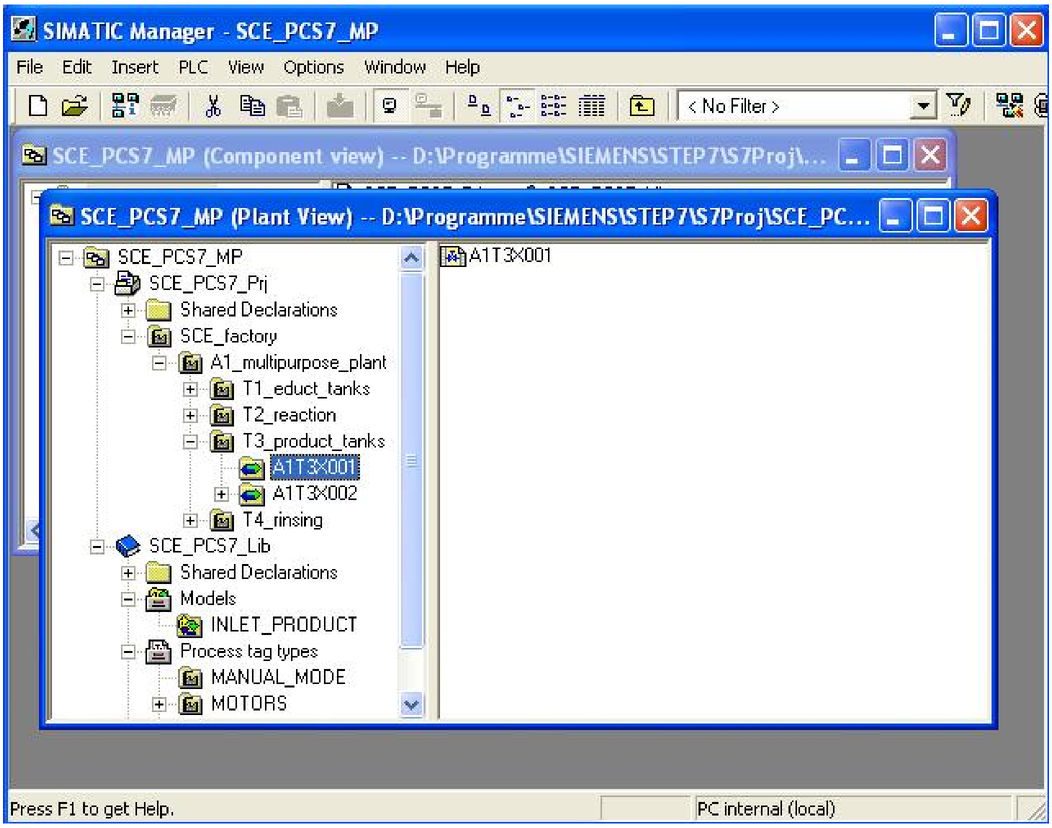
□ Only show errors and warnings in log

Import log:

Object	Action	Log text
SCE_PCS7_Prj\SCE_factory\A1_multi...	o.k.	Symbolic interconnection 'A1.T3.A1T3X002.GO+-.O+' exists already.
SCE_PCS7_Prj\SCE_factory\A1_multi...	o.k.	Block name 'FB_OPEN' already exists.
SCE_PCS7_Prj\SCE_factory\A1_multi...	o.k.	Block comment 'Digital Input' already exists.
SCE_PCS7_Prj\SCE_factory\A1_multi...	o.k.	Symbolic interconnection 'A1.T3.A1T3X002.XV.C' exists already.
SCE_PCS7_Prj\SCE_factory\A1_multi...	o.k.	Block name 'OUTPUT' already exists.
SCE_PCS7_Prj\SCE_factory\A1_multi...	o.k.	Block comment 'Digital Output' already exists.
SCE_PCS7_Prj\SCE_factory\A1_multi...	o.k.	Message text is empty (column title 'A1T3X001\VALVE_A1T3X001.MS
SCE_PCS7_Prj\SCE_factory\A1_multi...	o.k.	Message text is empty (column title 'A1T3X001\VALVE_A1T3X001.MS
SCE_PCS7_Prj\SCE_factory\A1_multi...	o.k.	Block name 'VALVE_A1T3X001' already exists.
SCE_PCS7_Prj\SCE_factory\A1_multi...	o.k.	Block comment 'Single-Drive/Dual-Feedback Valve' already exists.
Time	o.k.	00:00:01 Hr:Min:Sec
D:\Programme\SIEMENS\STEP7\S7...	End	Errors or warnings occurred during import.
--------------------	End	--------------------

Log file: D:\Programme\SIEMENS\STEP7\S7Proj\SCE_PCS7\SCE__Lib\Global\INL

55. In the ***SIMATIC Manager***, the hierarchy folder of the model is shown with a new symbol .



EXERCISES

We are going to apply what we learned in the theory chapters and the step by step instructions to the exercises. To this end, we are using and expanding the already existing multi-project in the step by step instructions (PCS7_SCE_0203_R1009.zip).

The tasks of this last exercise are to be considered comprehensive tasks. All objects not implemented so far are to be implemented.

TASKS

The following exercises are based on the step by step instructions. For each exercise, the corresponding steps in the instructions can be used as an aid.

1. Analyze the different CFCs and determine the structures that recur.
2. Then, create the matching process tag types and models from the existing CFCs for these structures. To do this, you may have to clear assignments that already exist.
3. Generate the import file for each process tag type and each model. Add the missing information there, and import the file.
4. Then, create the graphics and position the symbols that were generated automatically.
5. Test your completed plant!