HMI GENERATION

OBJECTIVE

After working through this instruction module, students are able to design and implement a graphic interface for efficient process management. To this end, they are acquainted with the objectives of process management. They understand the basic concepts of representation and are acquainted with different representation techniques. This enables the students to generate an efficient graphic interface that is suitable for being applied in ***PCS7***.

THEORY IN SHORT

A modern process control system like the ***PCS7*** provides the operators with different display screen-supported windows to the process that can be used to handle all process management tasks. Because of the large volume of information from the technical process that the operator has to take in and attend to, it is advisable to structure the information. Also, certain rules have to be adhered to for navigation and representation in order to establish an interface to the technical process that can be conveniently operated and supports the operator to the greatest extent possible in his manifold process management tasks.

PCS7 supports the design process of the process displays for control and monitoring in multiple ways. On the one hand, operator symbols and operator panels are defined for many of the elementary blocks and individual drive functions we use in basic automation -symbols and panels that allow for the project-wide uniform interaction with similar technical equipment. On the other hand, the plant hierarchy can be used to favorably structure the representation of the information.

Through this structure, very many elements of the operator communication system that have to be handled manually in other systems can be generated automatically and faultlessly through a generation run. The two remaining essential tasks when designing the process displays are, on the one hand, the representation of the static process structure (containers, piping, etc.) for better orientation, and on the other hand the insertion of elements for navigation along process streams on a plant hierarchy level.

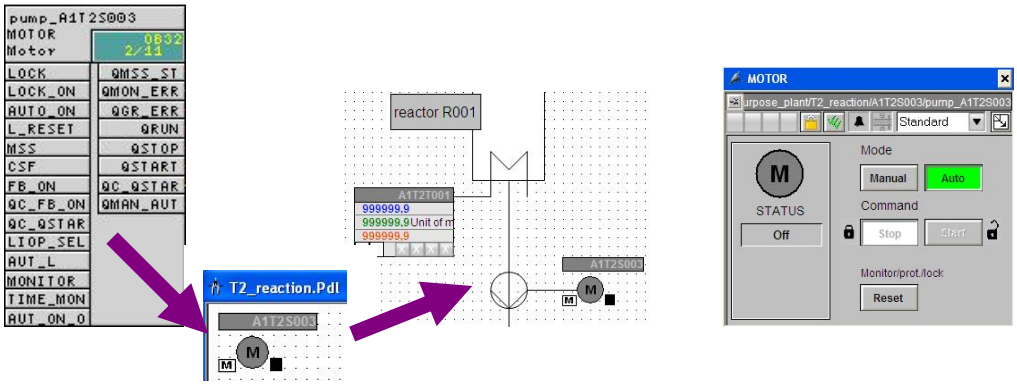


Figure 1: From individual drive function to operator controls and displays

By means of the generation run, icons are set up for all operator-controllable blocks of a hierarchical level. They subsequently only have to be moved and supplemented with static elements to get a complete operator display for the hierarchical level (refer to Figure 1).

THEORY

OBJECTIVES OF PROCESS MANAGEMENT

In a process engineering plants, it is the operator's task to carry out the intended operation of this plant economically and environmentally compatible. This task is called process management. The operator has to ensure a constant, high product quality and product volume (yield) and keep the amount of waste as low as possible. He also has to balance disturbance factors such as fluctuating properties of the raw materials used, plant disturbances or fluctuating throughput. He has to make sure that the availability and life of the plant is maximized. In addition, he has to make sure that emission values are adhered to and energy and material consumption are minimized [1].

To meet these objectives, the operator has to be continuously in a position to monitor the plant, to diagnose faults and intervene in the running process to eliminate the faults. The operator's workplace is the operator station of a control desk. This operator station is provided with all displays and intervention options that the operator needs to perform his job. The control system makes an operator interface available for this which he uses to carry out his tasks according to his capabilities, proficiencies and needs [1].

CONCEPTS OF REPRESENTATION

The representation of data and information on the graphic user interface has a decisive influence in the operator's performance. For that reason, it has to be 'tuned' to his capabilities, proficiencies and needs. The following questions have to be answered in this regard:

1. For whom is the representation, and what purpose does it serve?
2. What is to be represented?
3. How is it to be represented?

These questions depend on the plant that has to be configured, and have to be answered for the respective project. However, the following aspects have to always be considered:

Organizing what is to be represented

The information and data to be represented has to be organized in a way that is suitable for the representation. We have to specify how the existing elements are subdivided and arranged, how they relate and how the operator can navigate among the representations. To this end, it has to be determined how much information and data is to be represented overall (***quantitative aspect***), and it has to be specified which information and data is to be represented simultaneously and jointly (***qualitative aspect***).

The ratio has to be decided of what is new (information, dynamic part of the display) to what is known (data, static part of the display). The aim is to make the share of information as high as possible but with sufficient data to allow for the pertinent and purposeful interpretation of the information.

The result is the distribution of information and data to the different operator displays. We still have to define how the operator gets from one operator display to the next (***Navigation***).

Filling Degree

Depending on the user interface, only a limited area is available to simultaneously represent data and information. To ensure that the information and data can be read and differentiated under all operating situations, only a certain part of this area is to be assigned characters. This part if called the ***filling degree*** of the representation.

The recommended filling degree depends on the type of characters and picture elements as well as on the necessary arrangement of these elements. It thus depends on the representation technique used. For example, the filling degree of a process engineering

flow chart should not amount to more than 50%; for an alarm page, on the other hand, it may amount to up to 80% [1].

Coding

Coding specifies how certain information is represented. Information can be coded by color, shape, form, expansion, direction (angle), position and dynamics (flashing). Uniform coding facilitates taking in the information and evaluating it.

Good coding is clear, distinguishable and does not contradict existing conventions. For example, the color green should never be used for STOP. If, instead, a red flashing signal is used as coding for STOP, this coding should be used consistently for the entire operator interface. Also, this coding should not be used for any other information, to rule out any confusion. In addition, good coding should be apparent, so that it is easily learned and remembered by the operator.

Conspicuousness

A central task of the user interface is calling the operator's attention to important information. Since usually different kinds of information are displayed in an operator display, it is advisable to design this information -depending on its relevance and priority- with different conspicuousness. The more conspicuous an item of information is represented, the sooner it is detected. In addition, based on its conspicuousness, the operator recognizes which informational item requires his primary attention at the moment. Based on some examples, Table 1 shows the stepped increase of conspicuousness of the representation for different information.

Table 1: Application of stepped conspicuousness according to [1]

Increasing conspicuousness of means

Increasing conspicuousness of visualization

Combination of means				Application
Contrast	Color	Flashing	Acoustical signal	
high	X	X	X	Alarm
high	X	X	-	Change of state (requiring acknowledgement)
high	X	-	-	Change of state (not requiring acknowledgement)
high	X	-	-	Curves
low	X	-	-	Material and energy flows
high	-	-	-	Text of message line, explanatory texts
medium	-	-	-	selctable and operator-controllable object (keys)
low	-	-	-	currently not selctable and operator-controllable object

Consistency

Often, certain information occurs in several representations simultaneously. In this case, it is important that this information is represented ***consistently*** in the entire user interface. That means that the information looks identical in all representations, and behaves identically. Always the same terms and symbols are to be used. The operator input sequence should always be the same and the response by the system to the operator input sequence should be similar with respect to time as well as to content.

REPRESENTATION TECHNIQUES

Basic Structure of the Display Fields

In principle, the display field should be structured the same for all types of representation. It makes orientation, information absorption, and thus process management easier for the operator.

The basic structure according to VDI 3699 [1] recommended for this is shown in Figure 2. In the upper area, the message line is located where the current messages are displayed as group messages. Below it, an overview area is located where the available representations (such as process displays in ***PCS7***) are listed. From here, any representation can be opened. The work area occupies the largest part of the display area. Here, the representation that is currently selected is displayed. The bottom area contains the keys for activating general functions. In the working area, windows can be opened containing supplementary information (for example, different views of the ***PCS7*** blocks). All areas except for the workimg area are reserved and are permanently displayed.

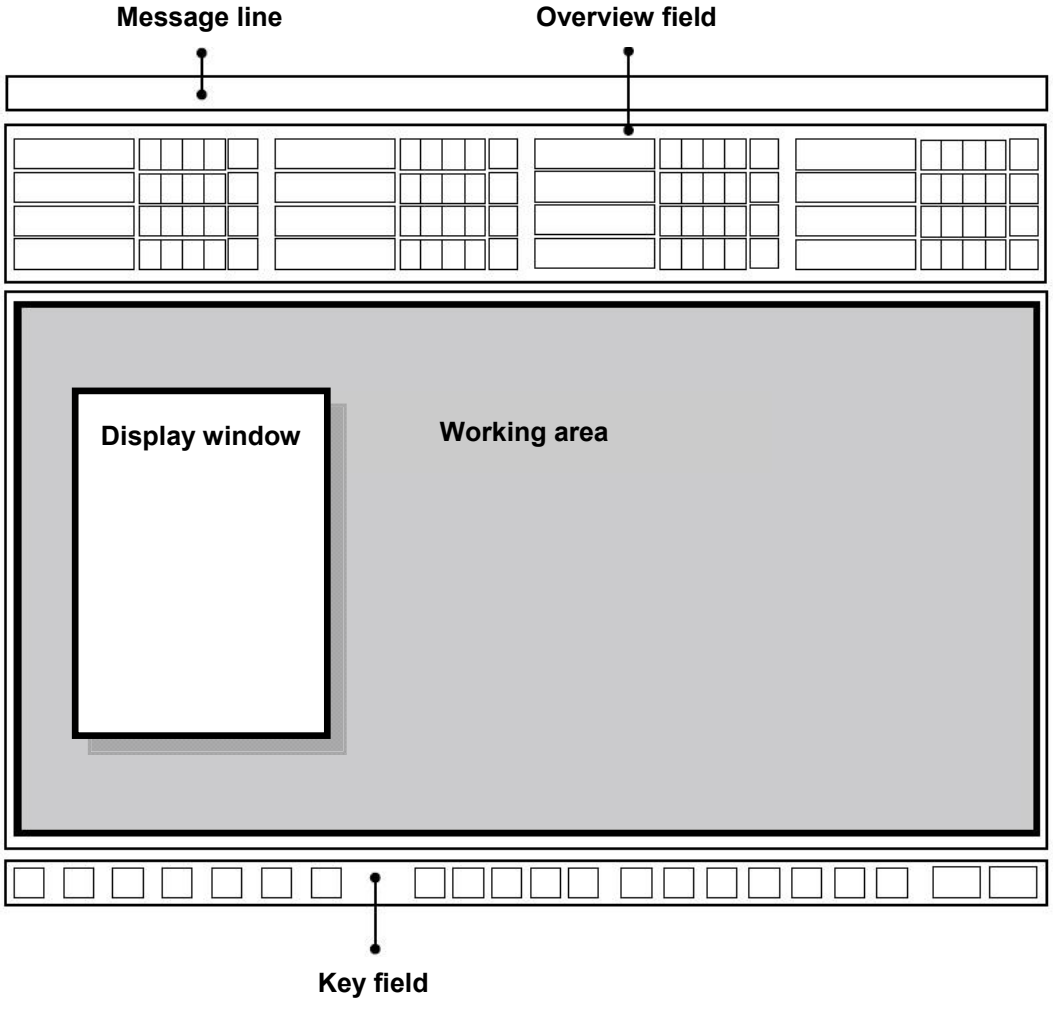


Figure 2: Basic structure of display fields

Flow Diagrams

A flow diagram is the "schematic representation of components, including their connections, through (flow) lines to represent relationships in a process engineering plant and in control engineering“ [1]. It renders the structure of the plant in a simplified manner and provides information regarding the paths of material, energy and signal streams between the different plant parts. With flow diagrams, process engineering and control engineering information is represented, and process intervention is made possible.

Flow diagrams consist of static and dynamic elements. The static display elements are represented by the ***basic display***. This basic display contains the dynamic display elements that are continuously updated.

The static basic display provides the context for the dynamic display elements; i.e. it renders the meaning of the represented objects and their relationship to each other. The basic display represents all data that remains unchanged during the display. These are the display background, the headings and the labeling as well as the plant parts and the devices (inasmuch as their representation is not to change).

The dynamic display elements render the information for process management. ***Display elements*** represent the changes, characteristics and relationships of the process values. They render the state of the plant, of the control system or of the process. ***Selection and operator elements*** allow for operator intervention regarding process control. In addition, often additional information such as function charts or curves can be displayed as windows in the flow diagram.

Flow diagrams are classified as follows:

- ***Control system flow diagrams*** only represent components of control engineering such as controllers, actuators and open loop controls as symbols. They are connected by means of signal flow lines.
- ***Process system flow diagrams*** graphically represent, in a simplified manner, plant segments by means of symbols. Here also, three different types are distinguished:
  - A ***basic flow diagram*** represents plants, units or plant segments only in the form of rectangles. They are connected to each other by means of flow lines for materials, energy or energy carriers.
  - A ***process flow diagram*** represents a process using (simplified) graphic symbols. The symbols represent the corresponding plant segments and are connected to each other with flow lines.
  - A ***piping and instrument flow diagram*** (P&I flow diagram) represents the technical equipment of the plant using graphic symbols. In addition, measuring points, control system blocks and actuators are represented. The symbols are connected to each other with lines for piping and signal paths.

In the ***PCS***, flow diagrams are referred to as ***process displays***. Within the scope of the step by step instructions, different process displays are generated for the configured plant.

HMI GENERATION IN THE PCS7

The ***PCS7*** provides an extensive control and monitoring system that consists of the following subsystems [2]:

- A ***graphics system*** for displaying process information and operating the process
- A ***curve system*** to represent and analyze the time sequences of stored process values
- A ***message system*** for diagnosing the process
- A ***log system*** for documenting the process
- An ***archive system*** for saving and storing process values, messages and logs

In this chapter, the ***PCS7*** graphics system is discussed. The message system will be discussed in detail in the next chapter 'Alarm Engineering'.

The graphic system represents the plant in a plant overview, displays process images in the working area of the service interface, provides operator process communication/ operator system communication and indicates alarm states. The corresponding service interface is generated on the system's operator station (OS). Thus, the OS is the central station for controlling and monitoring a ***PCS7system*** [2].

Configuring the Service Interface in PCS 7

The selected plant hierarchy of the project is the basis for organizing the service interface. The plants and units set up are mapped to the service interface through corresponding process displays. Display and directory names of the plant hierarchy are incorporated automatically. In the process mode, the available process displays are represented in the overview area corresponding to the plant hierarchy.

Initially, the process displays are set up at the corresponding location in the plant hierarchy and assigned to an OS. Then, the OS has to be compiled. After that, the process displays can be configured in the ***Graphics Designer*** of the ***WinCC Explorer***. The ***Graphics Designer*** is an editor where static and dynamic display elements are inserted, arranged, wired and connected to each other.

Configuring the Process Displays in PCS7

Technological blocks from the ***PCS7*** libraries that can be controlled and monitored are already provided with corresponding graphic representations, so-called ***block icons***. They are inserted automatically in the corresponding display when the process displays are configured. In the process display, block icons represent -in a kind of overview- the most important information about the represented block.

By means of the block icons, different preconfigured ***picture blocks*** (***face plates***) can be called that open as windows in the working area. Picture blocks are dynamic display elements that are connected to the parameters of the represented block and are updated automatically. They make it possible for the operator to comprehensively monitor and operate the associated technological block. Depending on the block type, different ***views*** exist for the associated picture blocks. These views allow for accessing parameters for very specific tasks. For example, in addition to the standard view there often is also a parameter view to assign parameters, an alarm view for diagnosis or a limit view for setting operator limits of the setpoint. It depends on the represented technological block which views are offered.

The ***Graphics Designer*** makes additional dynamic standard objects available which can be inserted manually. These objects can be wired directly to the connections of the blocks in the CFCs and SFCs, and thus implement the desired dynamic behavior. Examples of standard objects are input and output fields for entering and displaying values, status displays to display the binary states of an object, as well as bars for the relative representation of values.

In addition, the ***Graphics Designer*** includes different libraries with preassembled graphics elements -such as piping or valves- that can be used to generate the static basic display. As an alternative, it is possible to generate and use one's own graphics.

In the step by step instructions below, additional features and capabilities of the ***Graphics Designer*** are discussed. Furthermore, some additional important ***WinCC*** tools are introduced.

LITERATURE

[1] VDI 3699 (Status 2005-05): Process Management with Screens.

[2] SIEMENS (2009): Process Control System PCS 7: OS Process Management (V7.1).

STEP BY STEP INSTRUCTIONS

TASK

This task requires setting up the operator station (OS) after a few presettings in the ***SIMATIC Manager***.

An overview display of the multi-purpose plant and a display respectively for educt tank B003, reactor R001 and product tank B001 are created.

OBJECTIVE

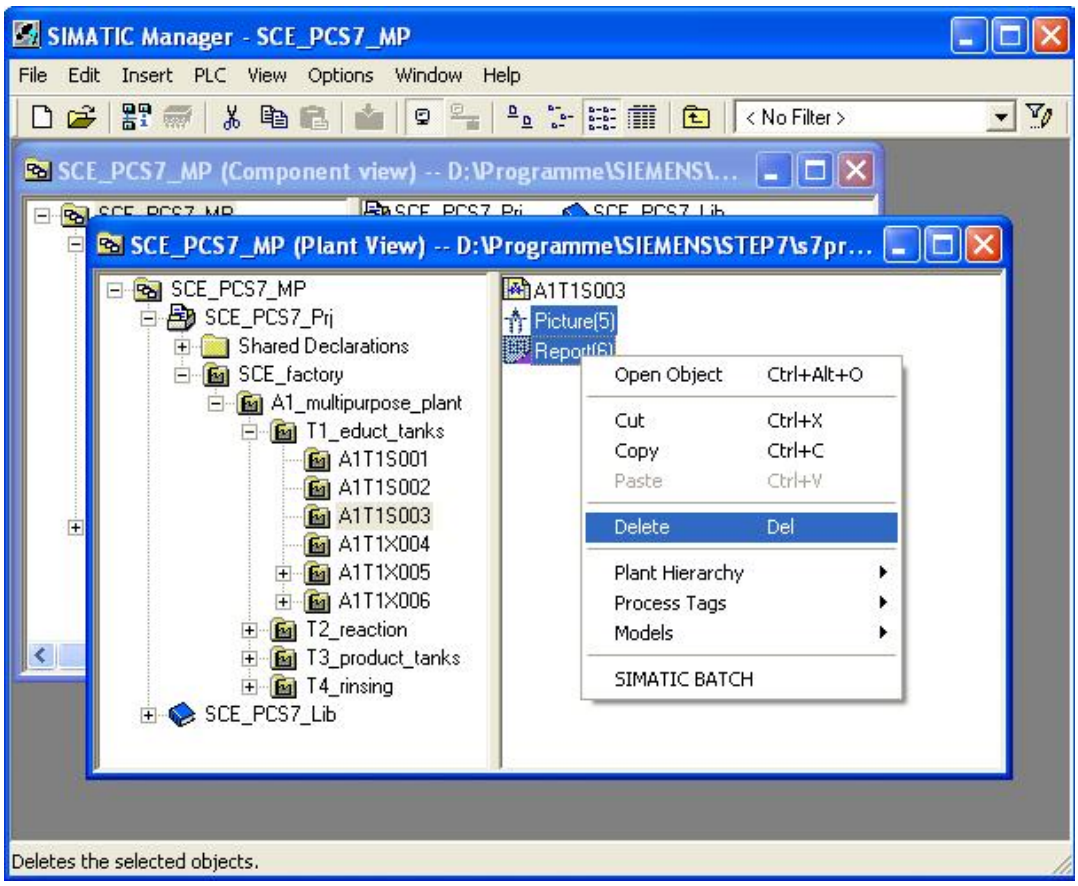
In this chapter, the student

- Learns to generate the operator station (OS) in the ***SIMATIC Manager***
- Is acquainted with the configuration environment ***WinCC***
- Learns to create displays with the ***Graphics Designer***

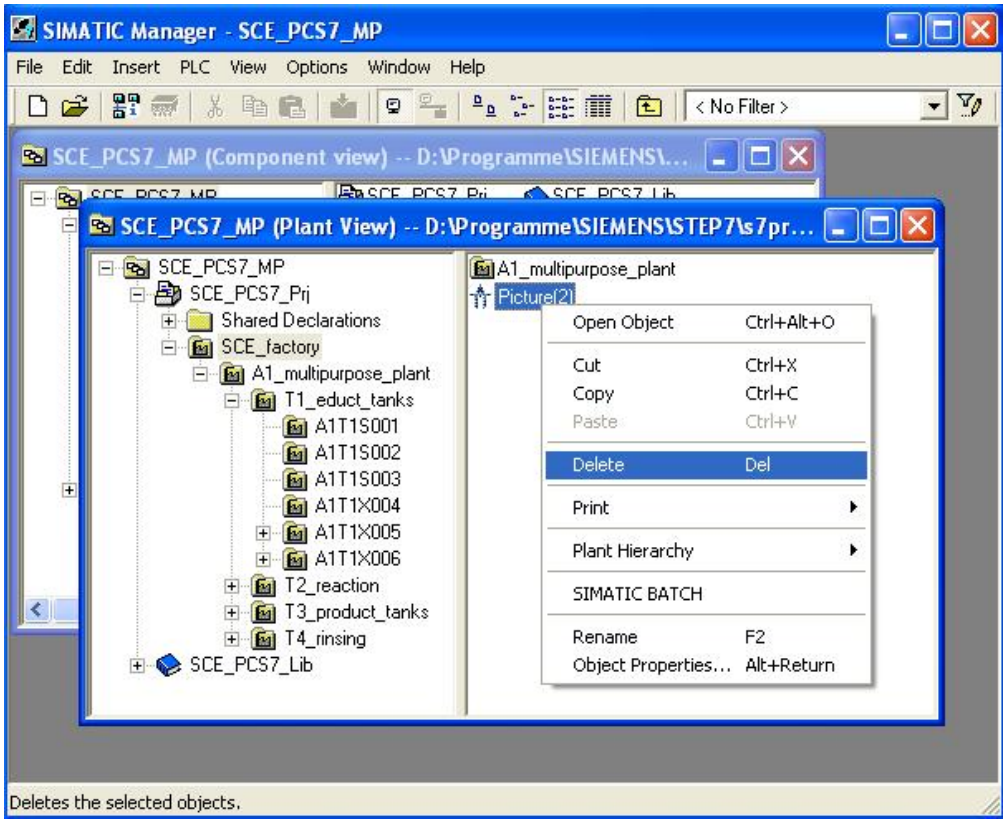
PROGRAMMING

1. Before we start inserting displays, we remove from Level 4 in the plant view the displays inserted by the wizard at the very beginning of project generation. In our project, we only want to use displays in hierarchy level 2 and 3.

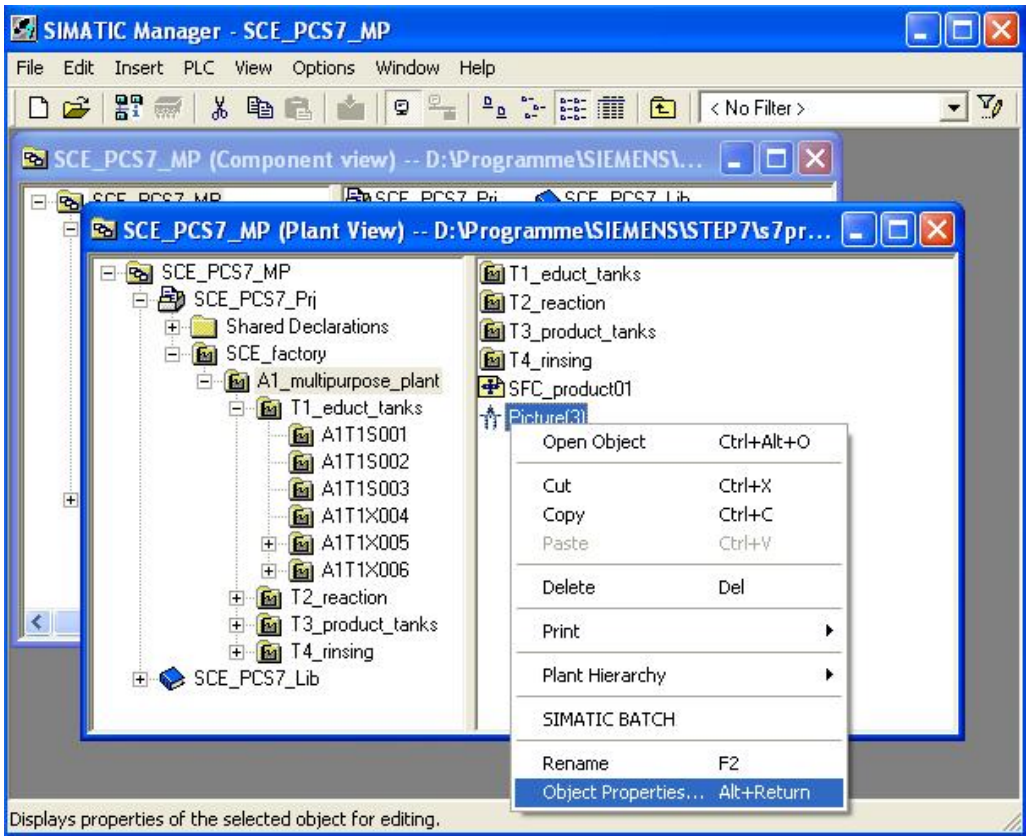
   (→ Display(6) → Delete)



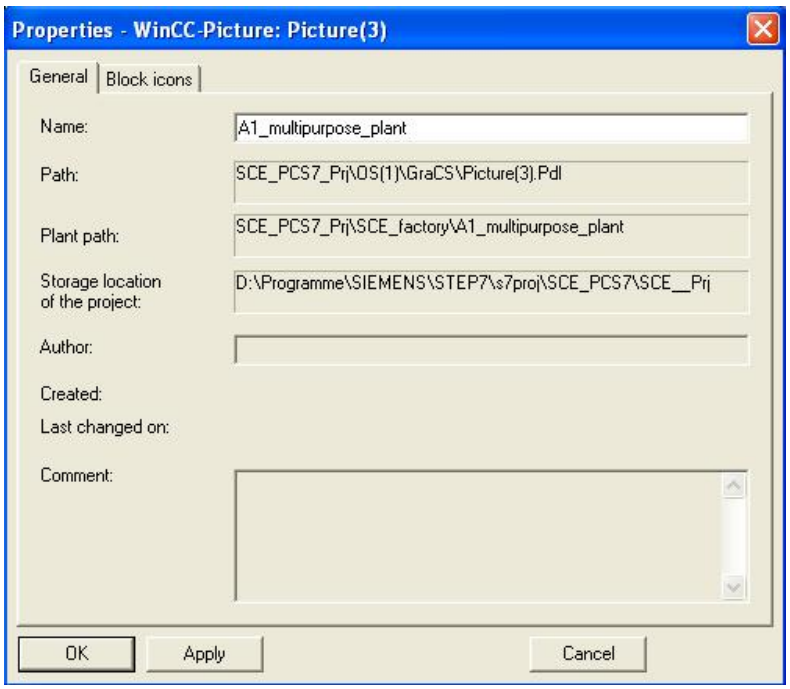
2. Likewise, we delete the display in Level 1. (→ Display(3) → Delete)



3. Now we can edit 'Display(4)' in folder 'A1_multipurpose_plant'. (→ Display(4) → Object Properties)

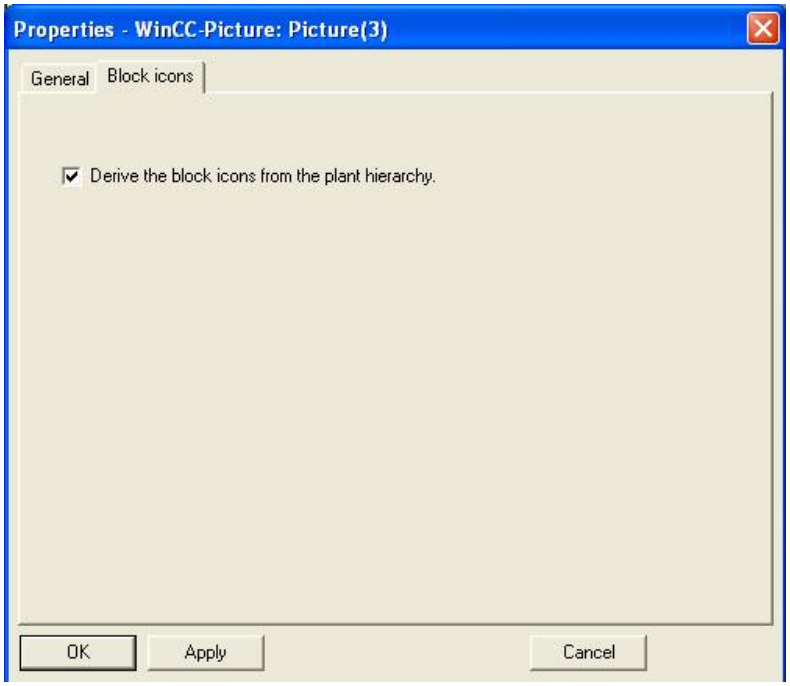


4. At the general properties, we change the name of the display to the name of the folder. (→ General → A1_multipurpose_plant)

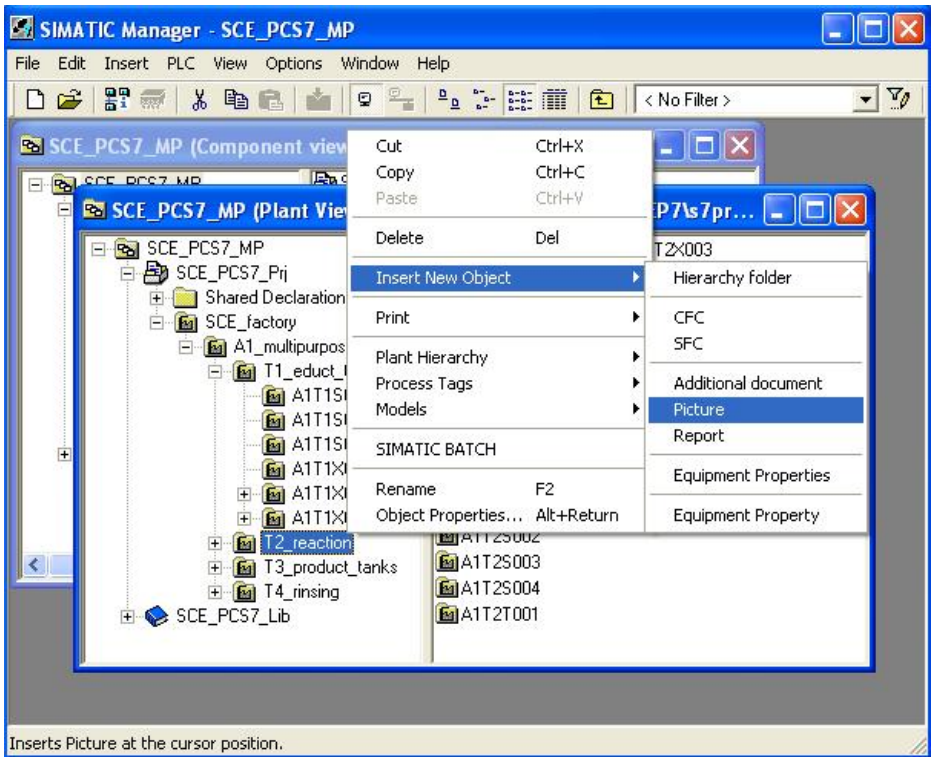


5. In the displays, block icons can be generated automatically in the CFCs. By clicking on such a block icon in runtime, the face plate for this block can then be opened. Here, we set that the block icons are set up by taking the hierarchy into account.

   (→ Block icons → Derive the block icons from the plant hierarchy → OK)

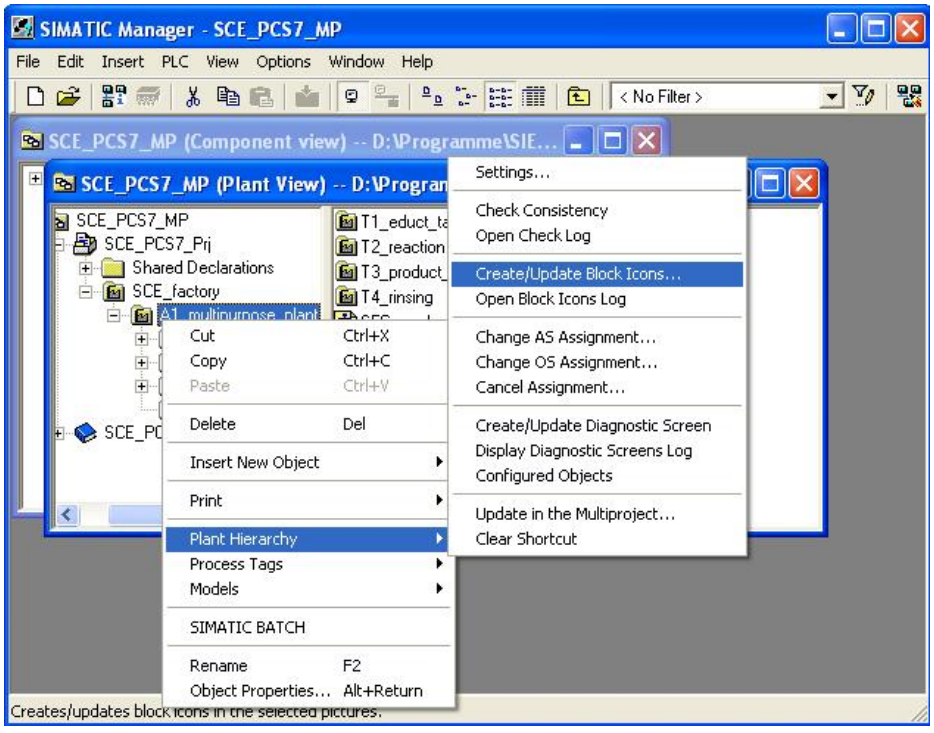


6. Now the missing pictures are inserted in Level 3, and then renamed accordingly. (→ T2_reaction → Insert New Object → Picture)



7. As a result we have -in the folders 'A1_multipurpose_plant', 'T1_educt_tanks', 'T2_reaction', 'T3_product_tanks' and in 'T4_rinsing'- one picture respectively with the name of the folder. Now, the block icons are generated in all these pictures.

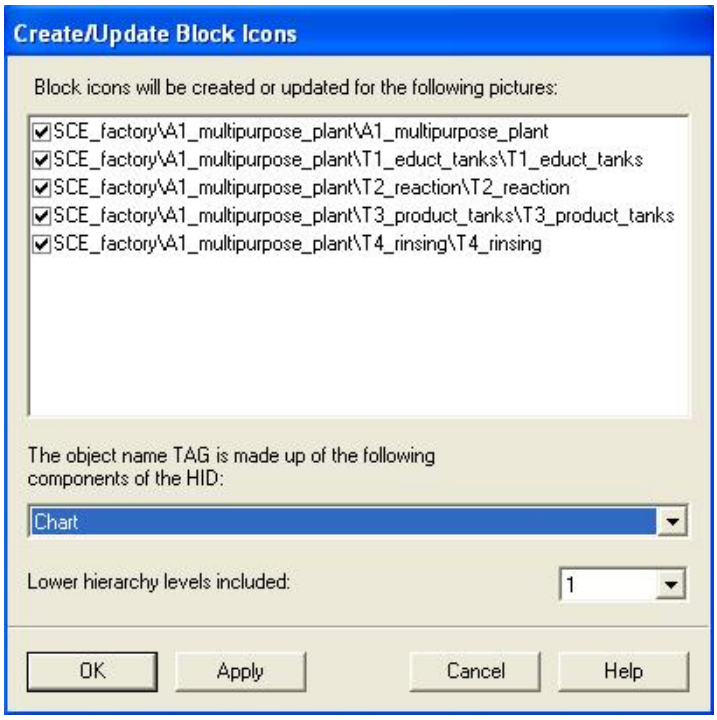
   (→ A1_multipurpose_plant → Plant Hierarchy → Create/Update Block Icons)

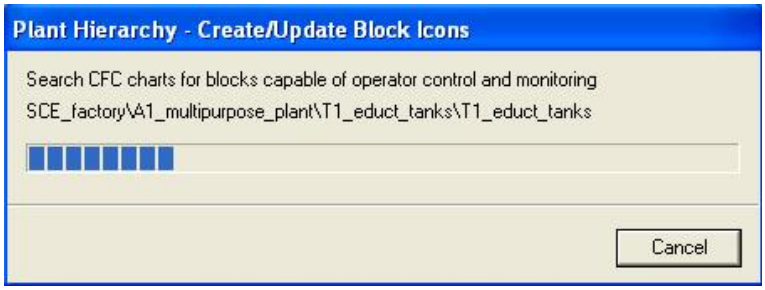


8. In the following dialog, we leave all pictures checked and select the name parts of the plant designation for the object designation of the tags. Here, only the name of the chart is always entered.

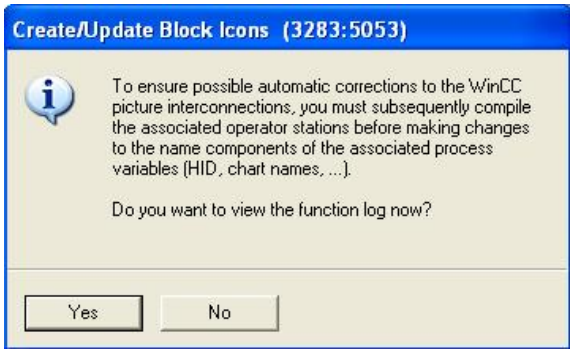
   The number of the included lower hierarchy levels is to be 1. The status for generating block icons is then displayed.

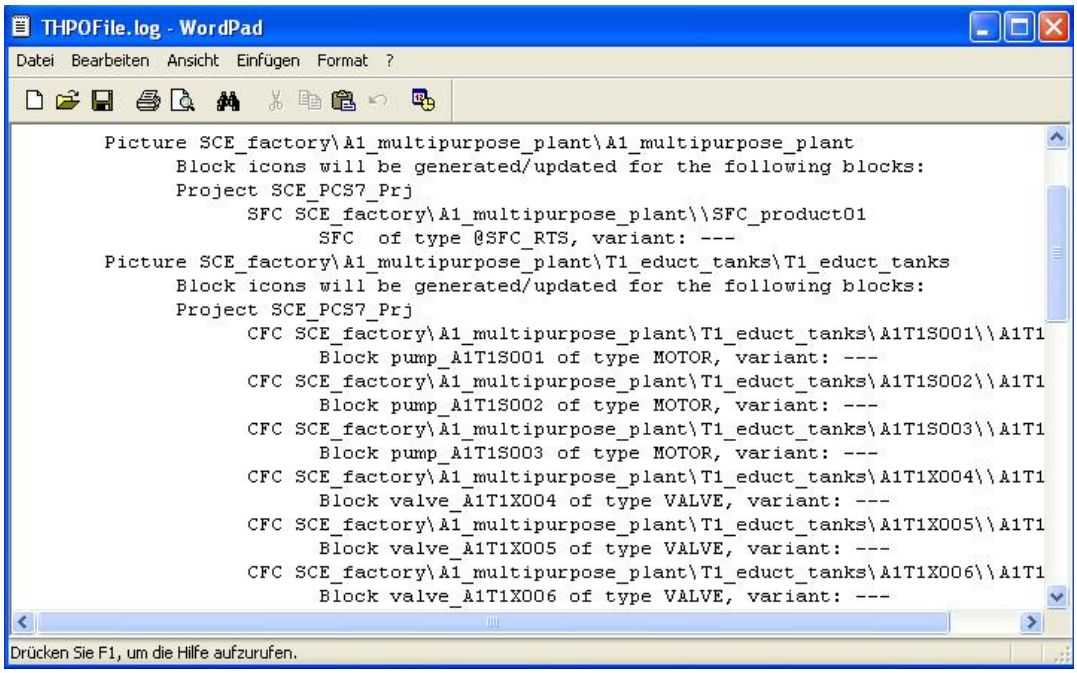
   (→ Chart → Lower hierarchy levels included 1 → OK)





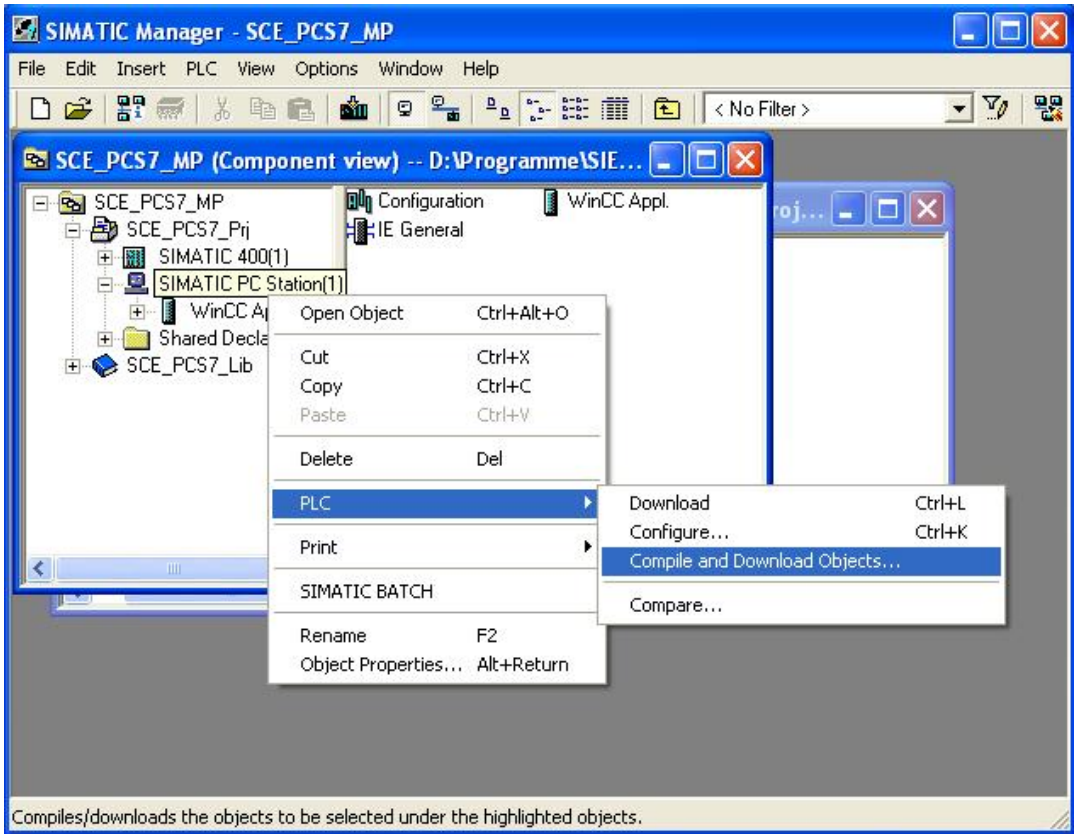
9. After we did that, the operator station (OS) has to be recompiled. The log shows which picture block was set up in which picture. (→ Yes)





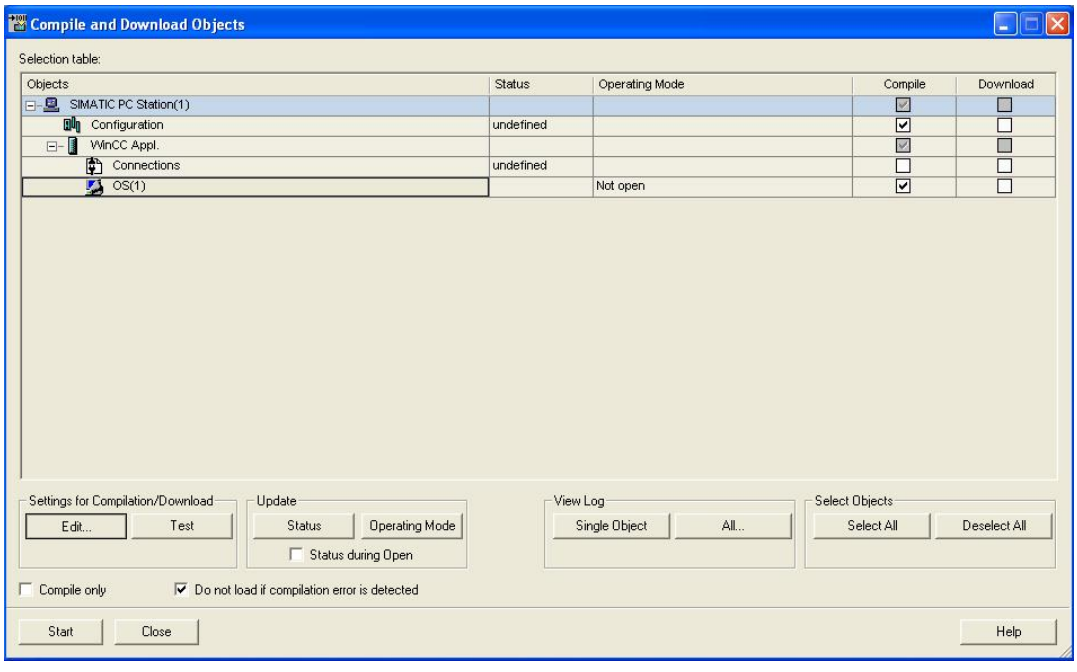
10. After the pictures were set up in the plant view and the block icons were set up there, we can start compiling the operator station (OS) in the component view.

    (→ Component view → SIMATIC PC Station(1) → PLC → Compile and Download Objects)

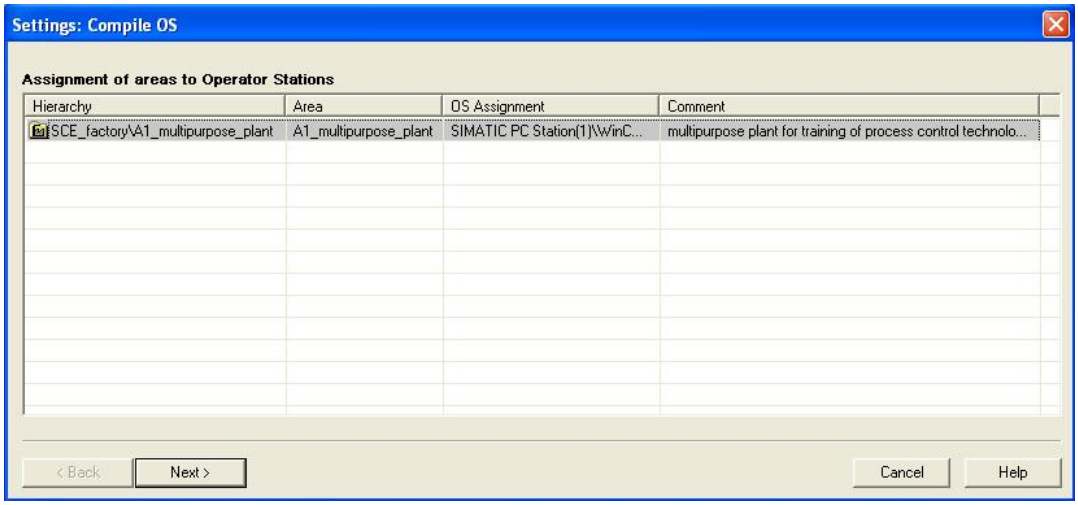


11. Prior to the compilation run, we have to select the settings for the OS.

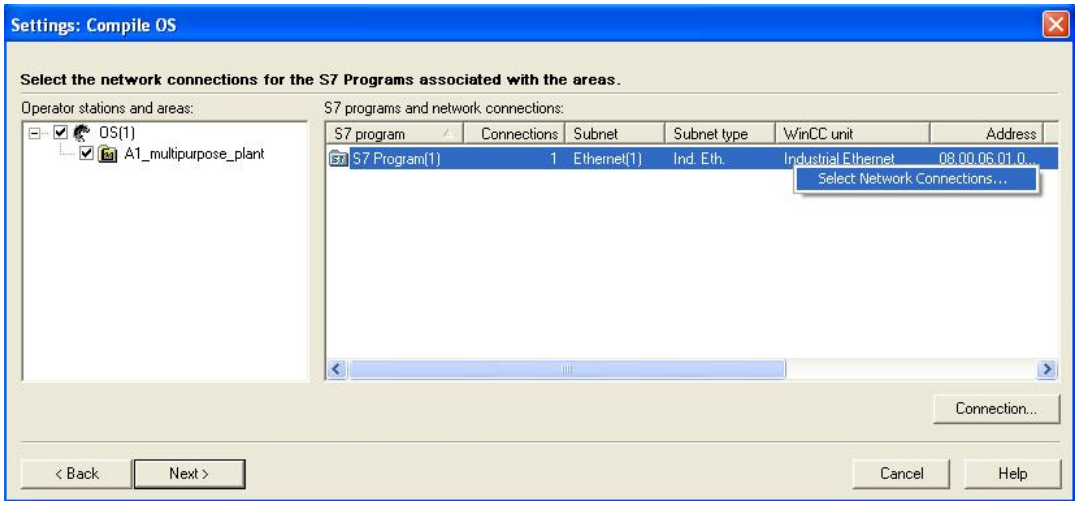
    (→ OS(2) → Edit)



12. Regarding the assignment of areas to operator stations: we don't have to set anything in our case (with only 1 OS). (→ Next)

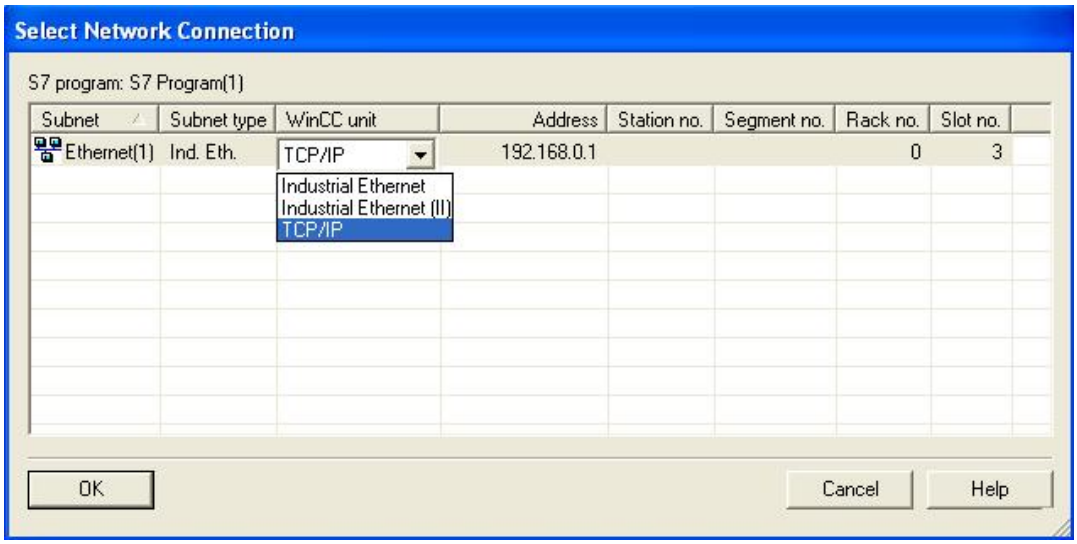


13. In the next window, we select the dialog in order to select a network connection. (→ Select Network Connections)

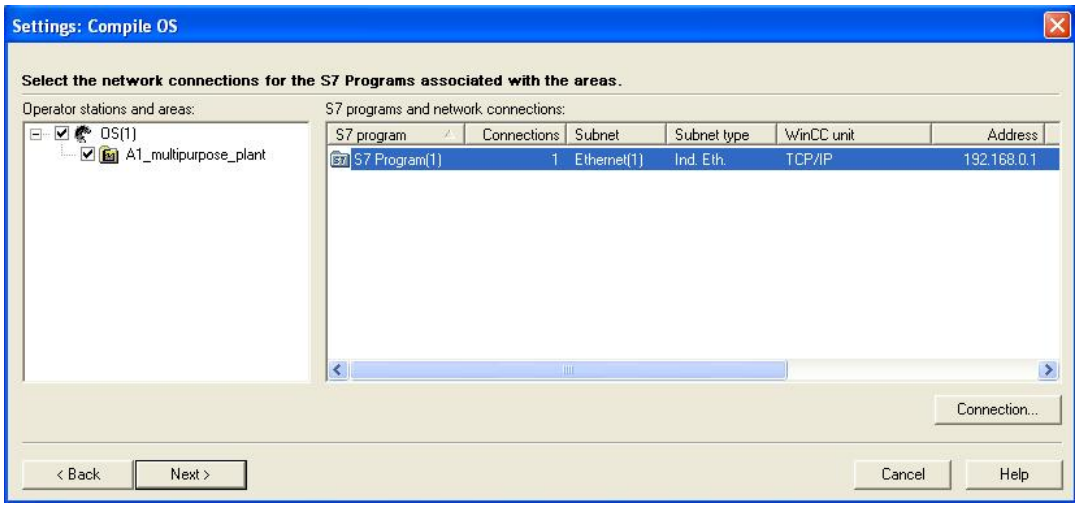


14. Here, we select as WinCC unit TCP/IP. The address is then correctly included automatically from the hardware configuration.

    (→ WinCC unit → TCP/IP → OK)

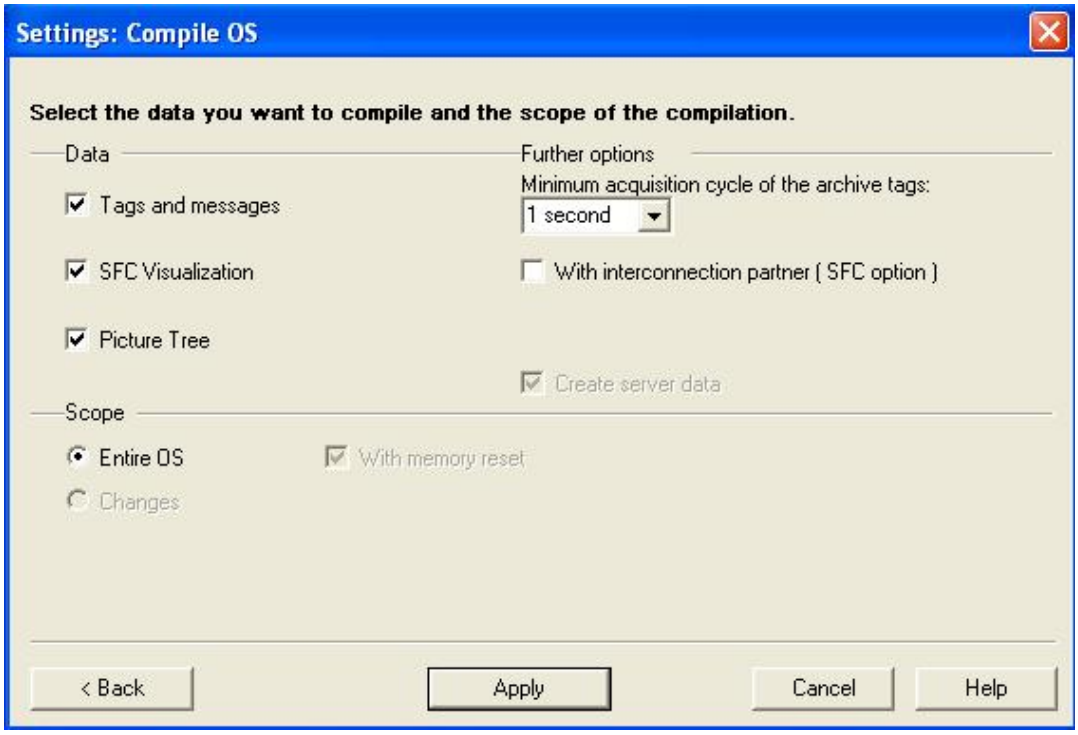


15. After the area assignment, the OS assignment and the network connection were checked once more, we continue. (→ Next)

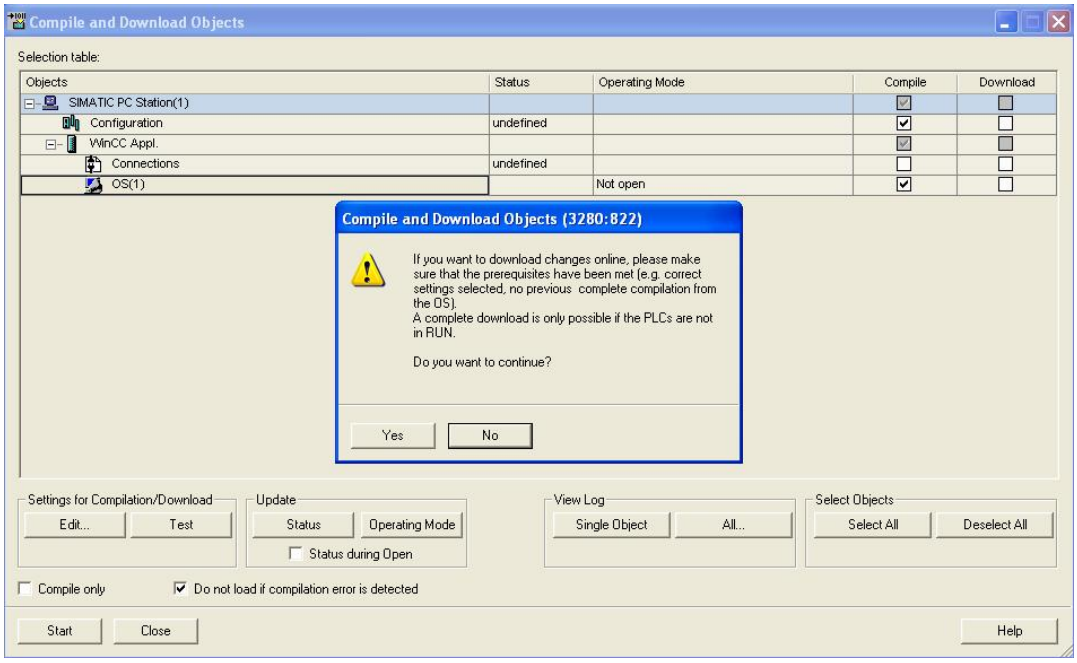


16. In the selection that follows, we can select the scope of the compilation.

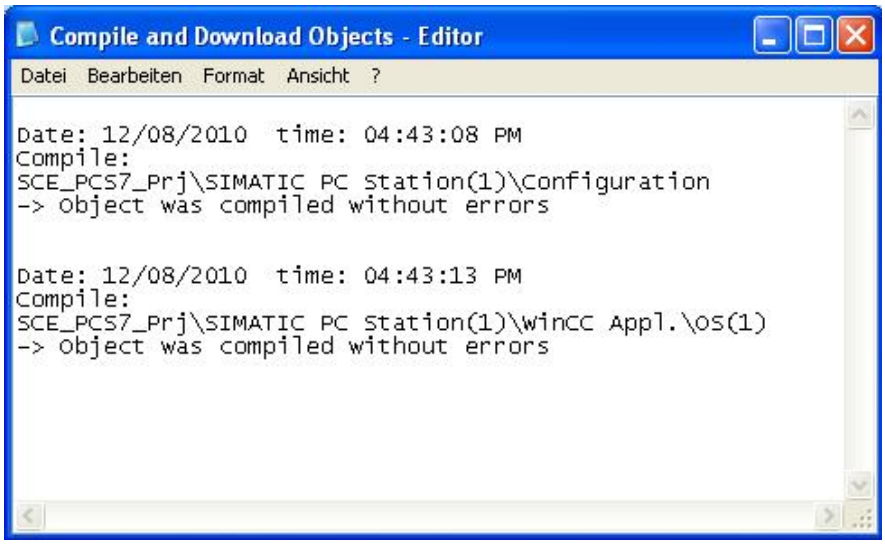
    (→ Tags and messages → SFC Visualization → Picture Tree → Entire OS → With memory reset → Apply)



17. Since in our plant, the operator station (OS) is started on the engineering system (ES), we are selecting only Compile and not Download. After the compilation run is started, the prompt is confirmed with Yes. (→ Start → Yes)

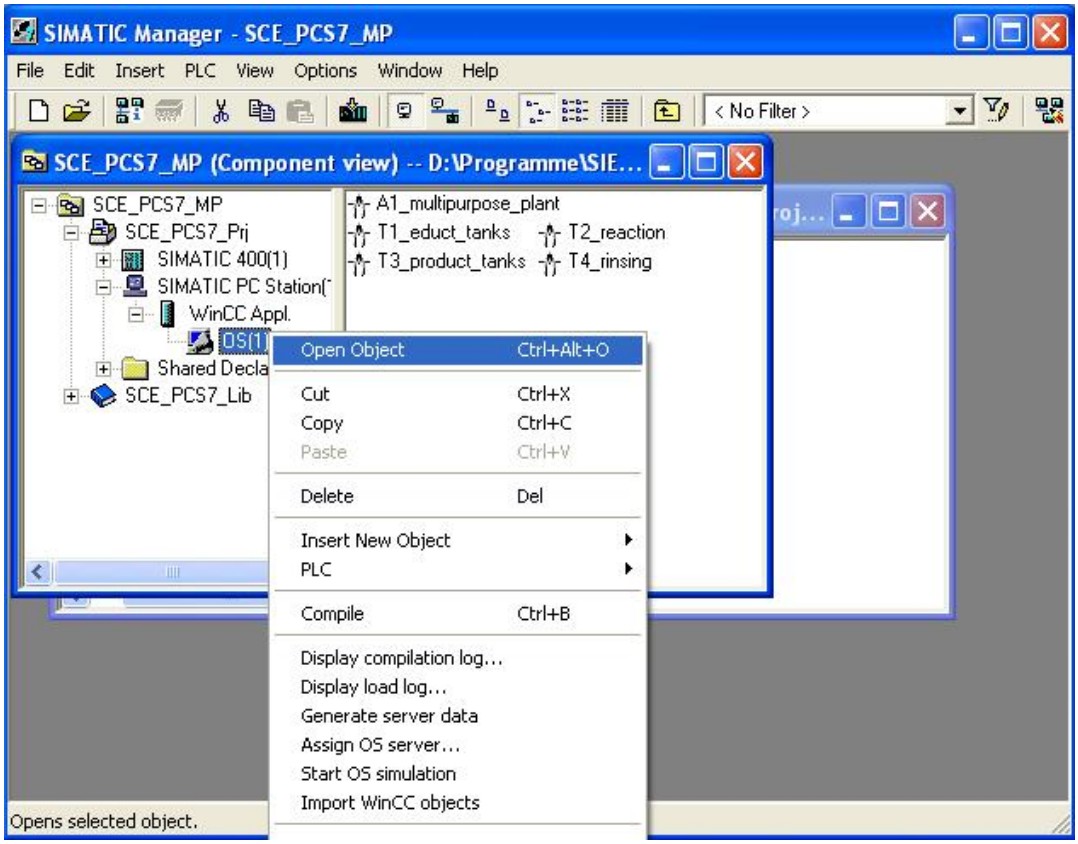


18. At the end, a log displays errors and warnings. We close the window. (→)



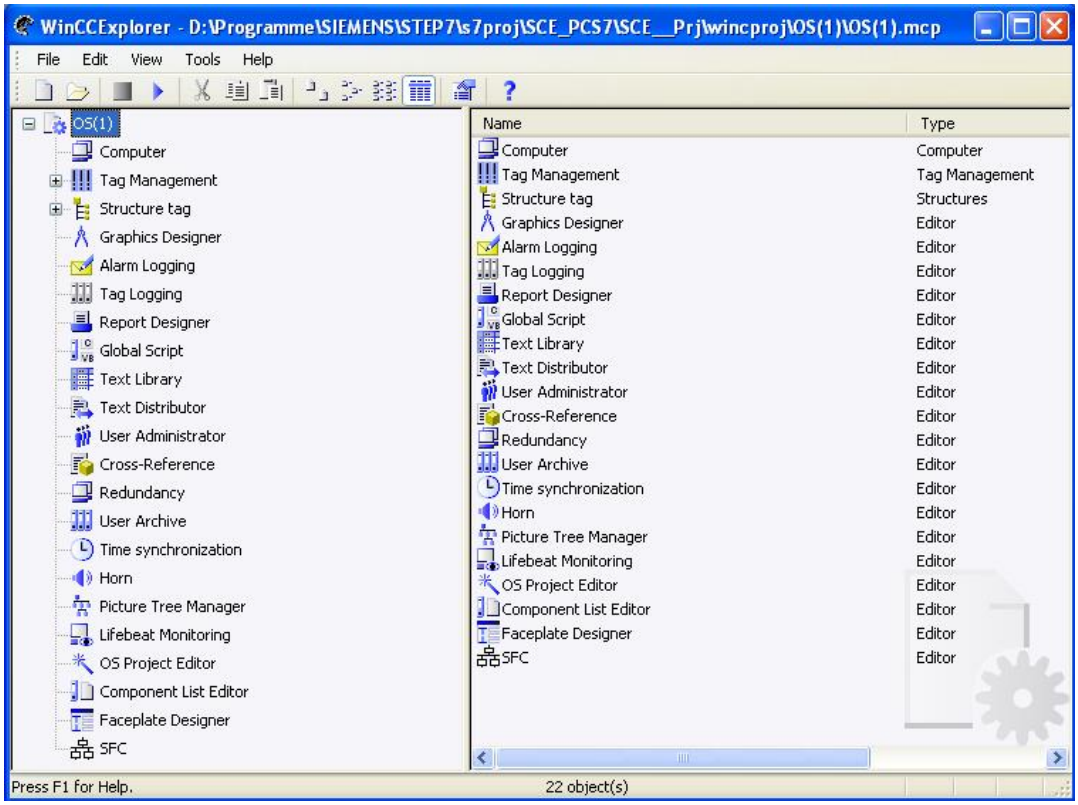
19. We then open the OS from the component view for further editing.

(→ OS(2) → Open Object)



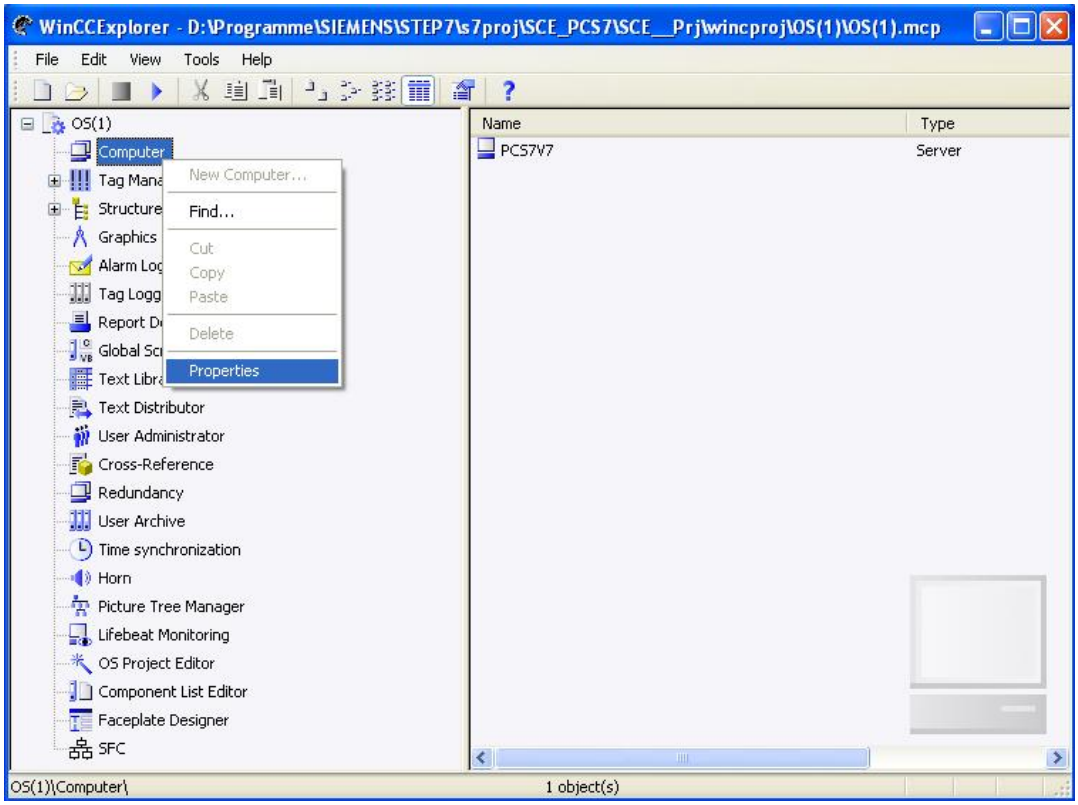
20. The operator station (OS) is edited and represented In the ***WinCC Explorer***. Here, you will see the editors and functions of the ***WinCC*** version installed here. In the next steps, we will become acquainted with some of these editors and functions. Additional information is provided in Help or the manuals.

    (→ F1)

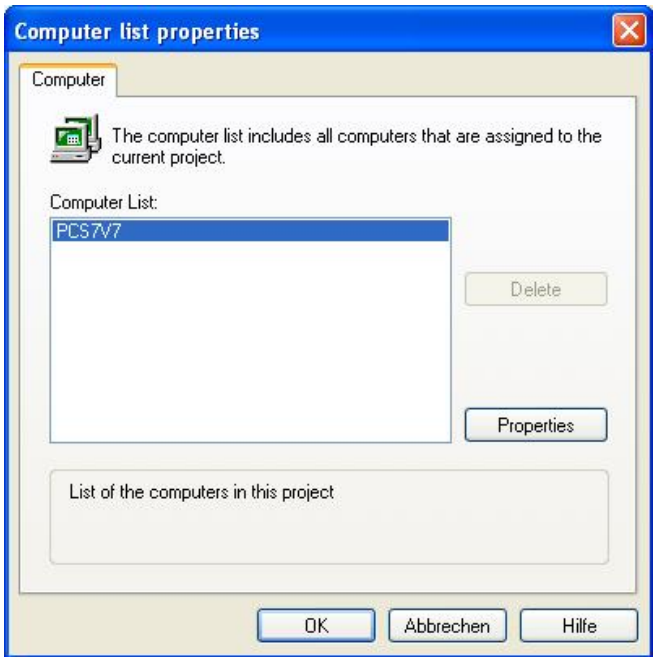


21. We now have to specify the properties of the computer where the project is to be started later. In this case, it is the configuration computer. Right click on the computer and then select Properties.

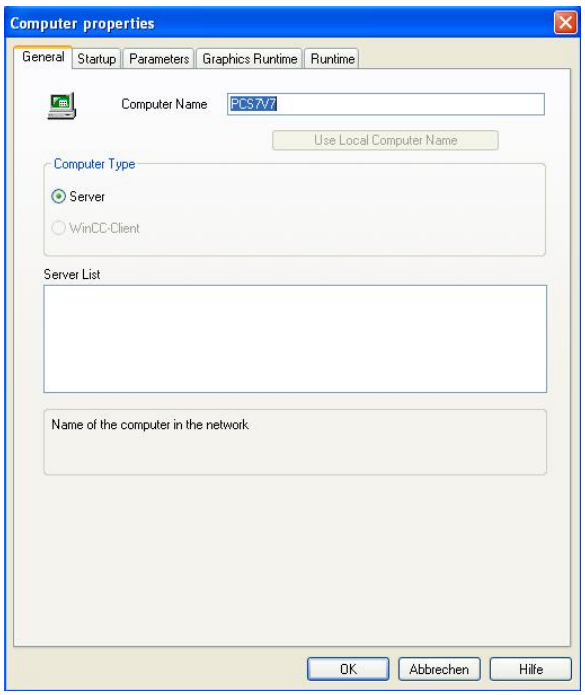
    (→ Computer → Properties)



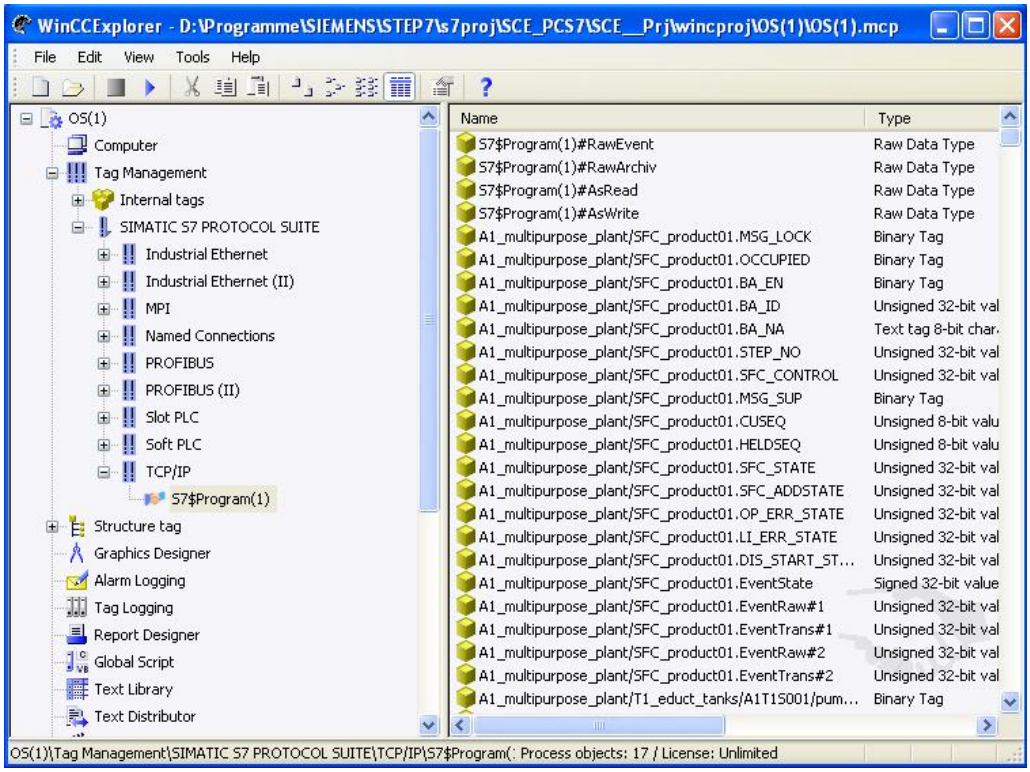
22. In the picture that follows, the name of the configuration computer is displayed that was specified previously in Windows. With the button 'Properties', we can specify the properties for the computer. (→ Properties)



23. Now, a screen form is displayed where different parameters can be set. Here, the computer type of the computer on which WinCC runtime is to be started (in our case, with the local computer name 'PCS7V7') is specified as server. (→ PCS7V7 → Server → OK → OK)

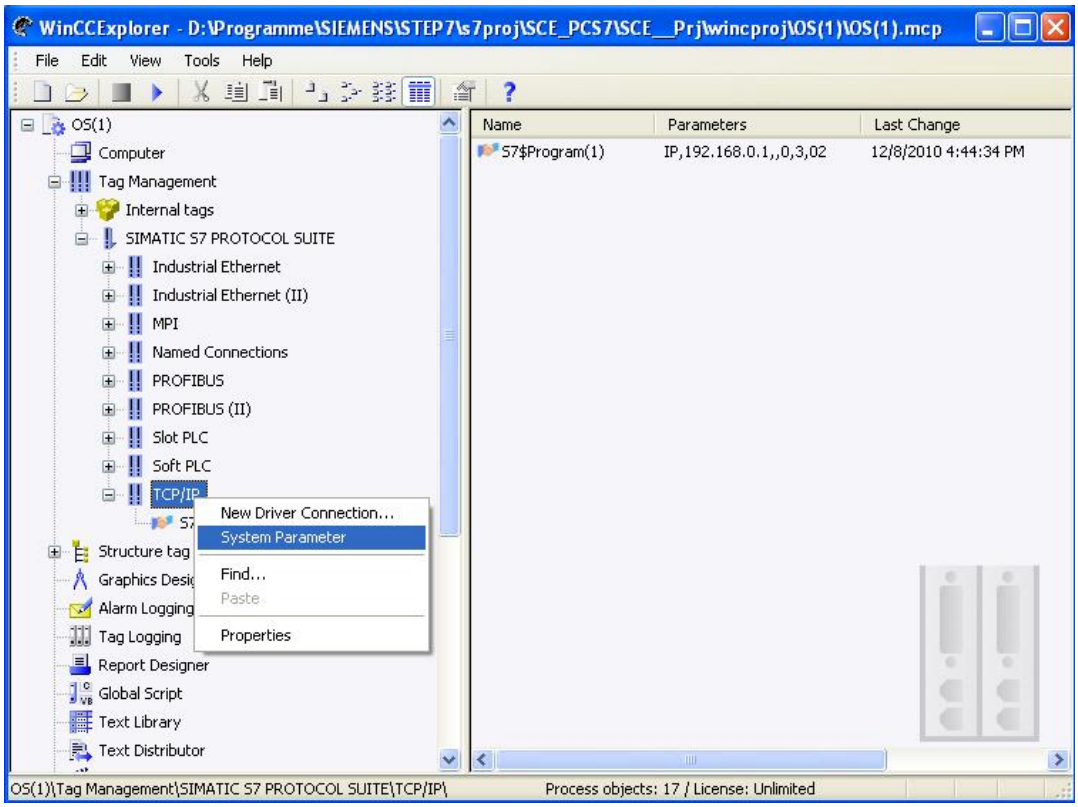


24. In Tag Management, under 'S7-Programm(1)', the tags for the corresponding CFC blocks and SFC step sequences that were automatically set up during the compilation run are located. They are used to visualize the face plates, but can also be utilized individually for visualization within the displays. (→ Tag Management → SIMATIC S7 PROTOCOL SUITE → TCP/IP → S7 Program(1))

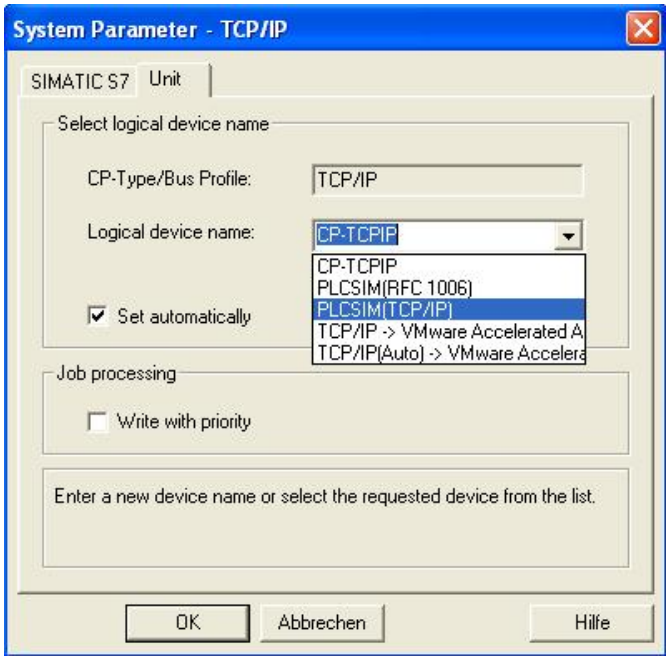


25. To ensure that communication is working, we still have to select the system parameters of the TCP/IP connection.

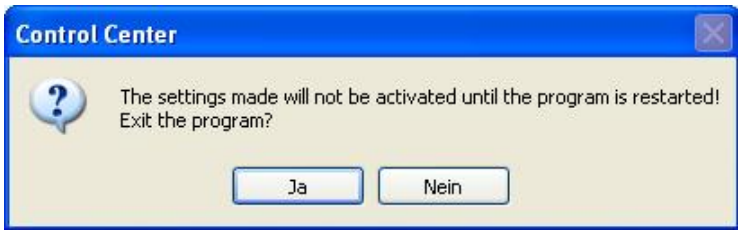
    (→ TCP/IP → System Parameters)



26. Here, we can now select any TCP/IP communication interface that was installed in STEP7 under PG-PC interface. We link this setting with the setting in the ***SIMATIC Manager***. (→ Unit → PLCSIM(TCP/IP) → OK)

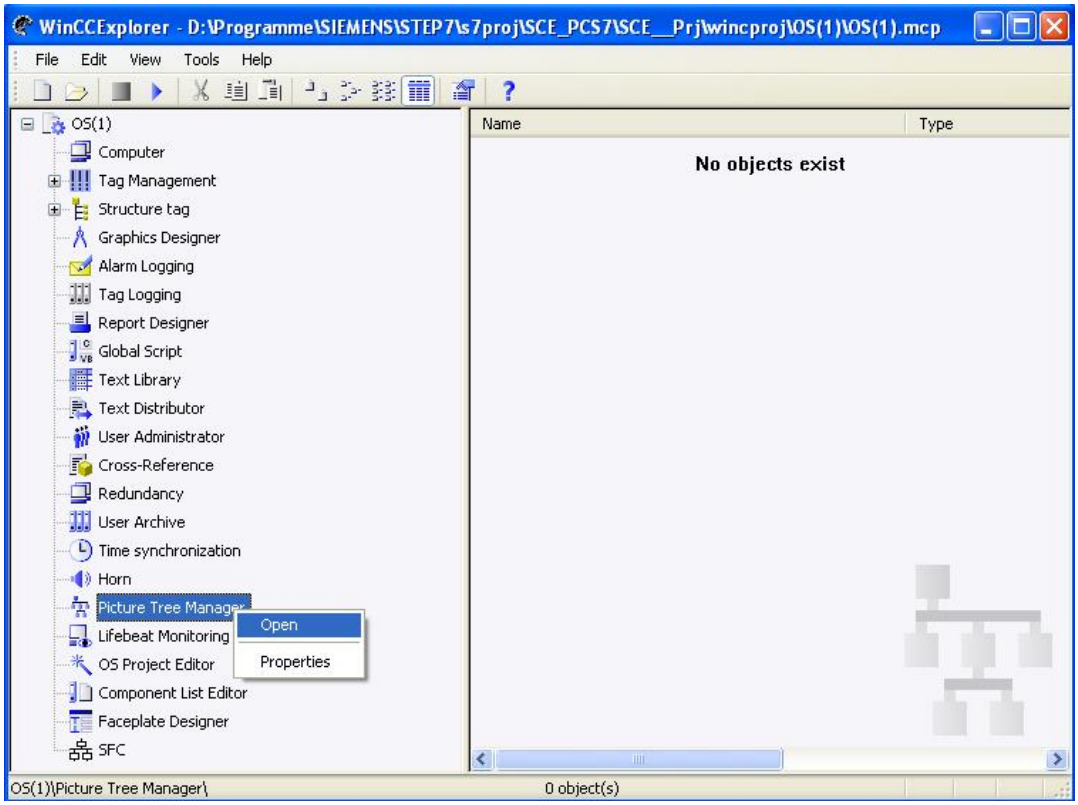


27. After system parameters are changed, ***WinCC*** has to be exited and restarted. (→ Yes → OK)

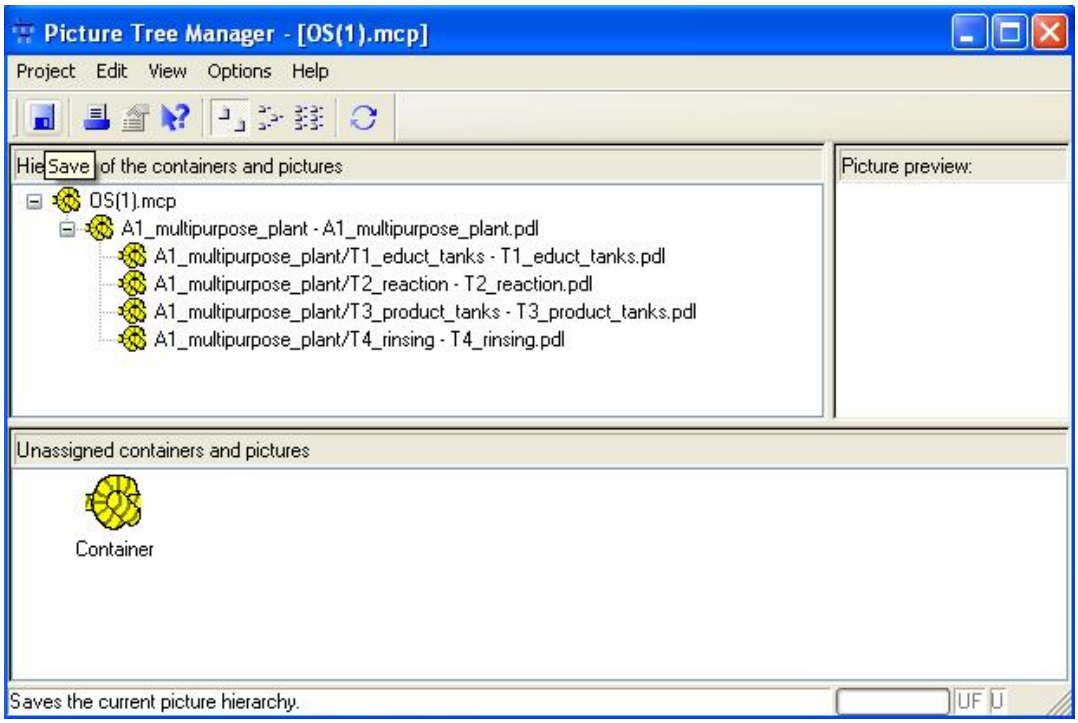


28. After we restarted WinCC, let's take a look at the ***Picture Tree Manager***.

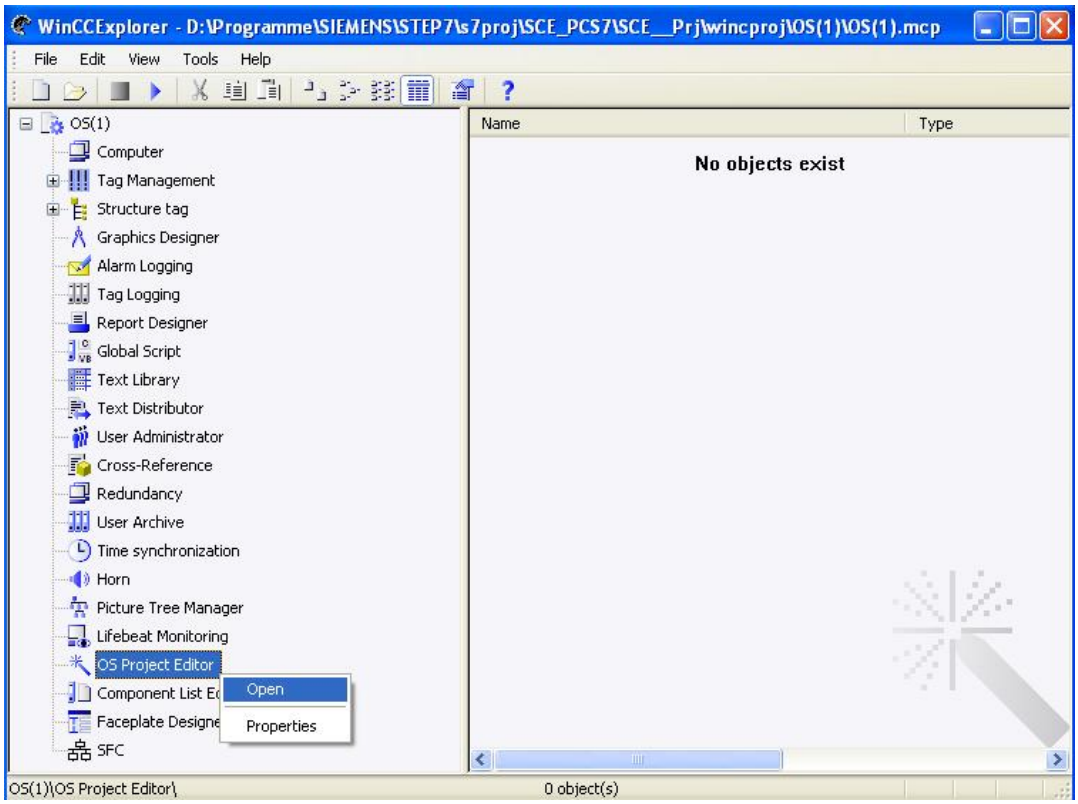
(→ Picture Tree Manager → Open)



29. In the ***Picture Tree Manager*** we specify the sequence in which our pictures can be called later. We keep the structure, make a save and exit the editor again. (→  → )

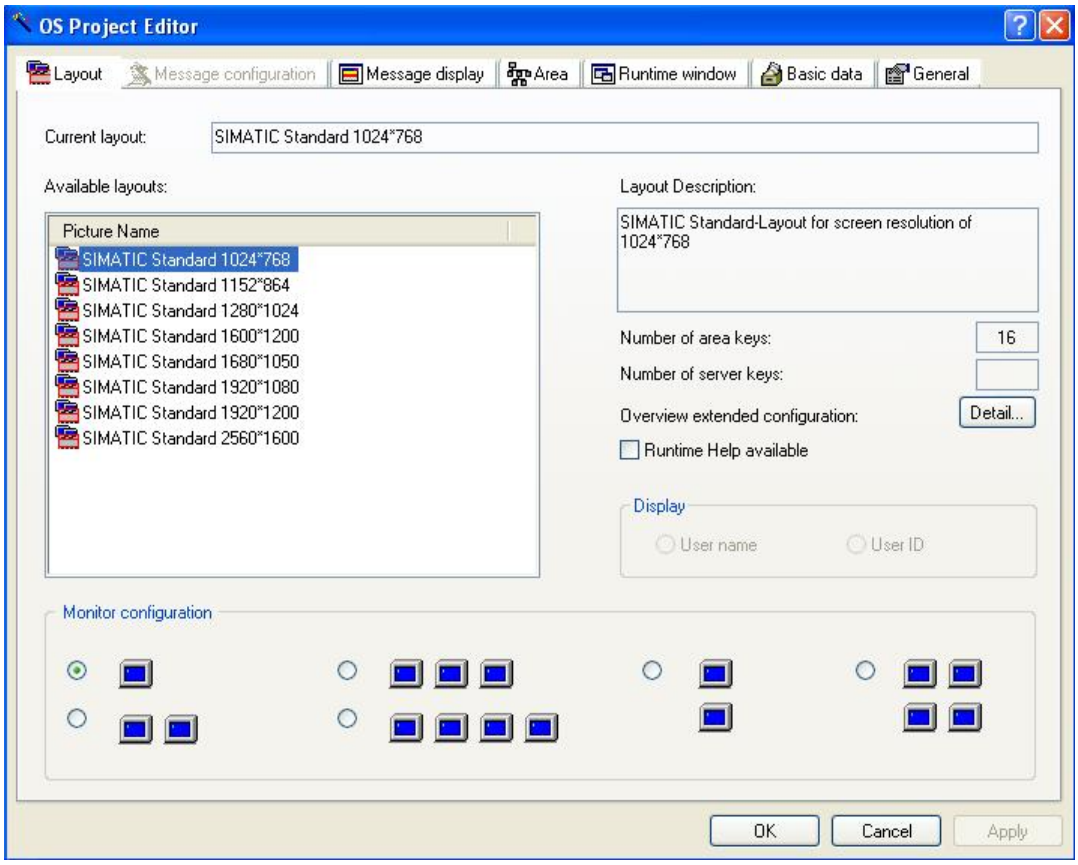


30. Now let's look at the ***OS Project Editor***. (→ OS Project editor → Open)



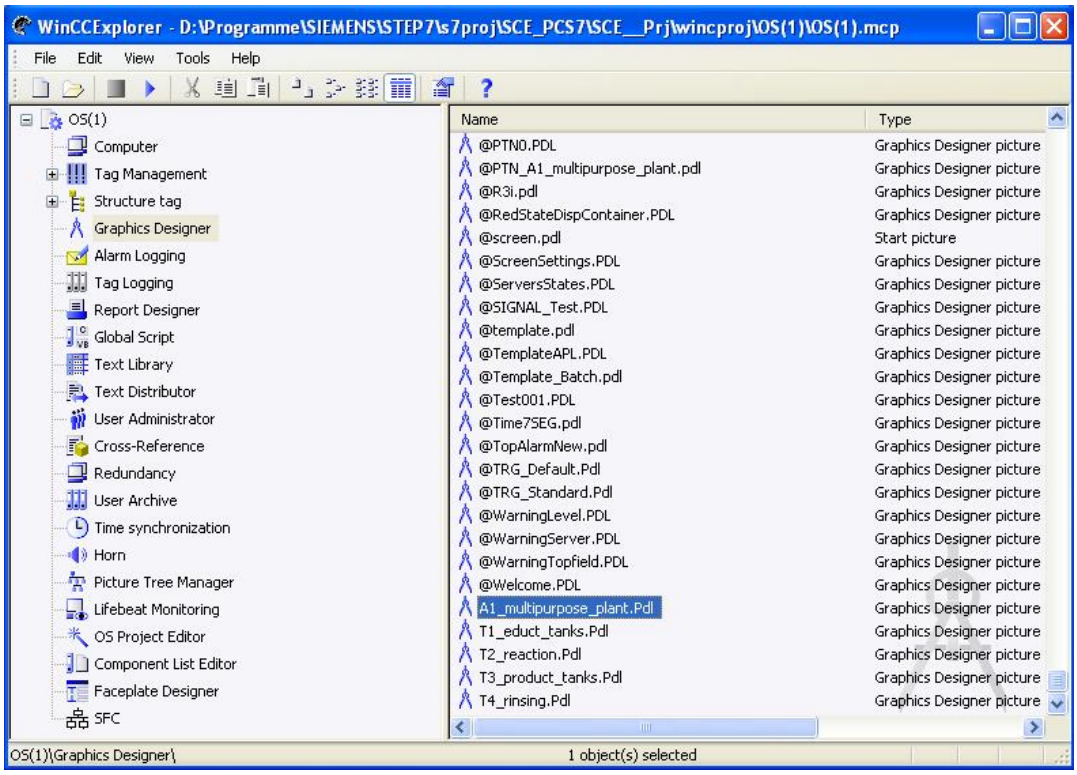
31. In the ***OS Project Editor***, under 'Layout', we can select the monitor-configuration and the screen resolution.

    In addition, there are settings for message display, the visible areas, the window arrangement in the runtime window and other general basic settings. We accept these settings without changing them. (→ OK →)

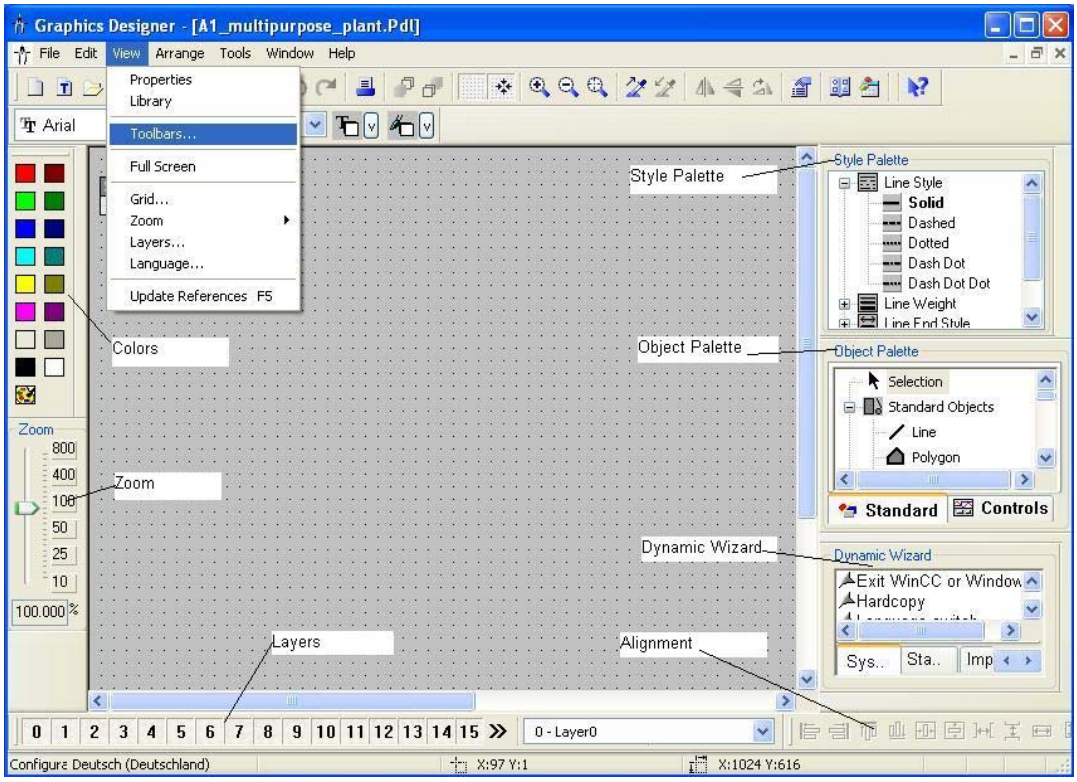


32. User friendly and clearly laid out operator displays are generated in the ***Graphics Designer***. Barely any limits exist as to the programmer's creativity. In addition, it is possible to import graphics files or video sequences that were generated with other graphics programs. It is best to open individual pictures by double clicking on the names in the right window.

    (→ A1_multipurpose_plant)



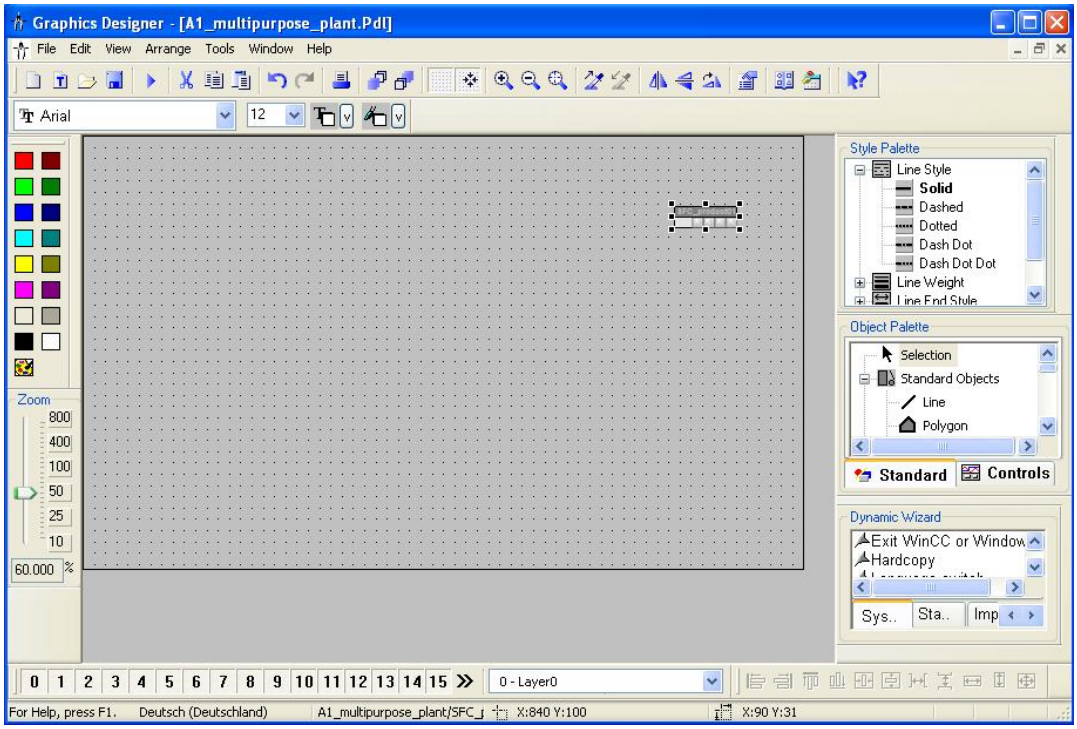
33. The ***Graphics Designer*** provides the most varied functions for generating process displays. They can be hidden or shown in the menu with View/Toolbars. (→ View → Toolbars)



These toolbars have the following functions:

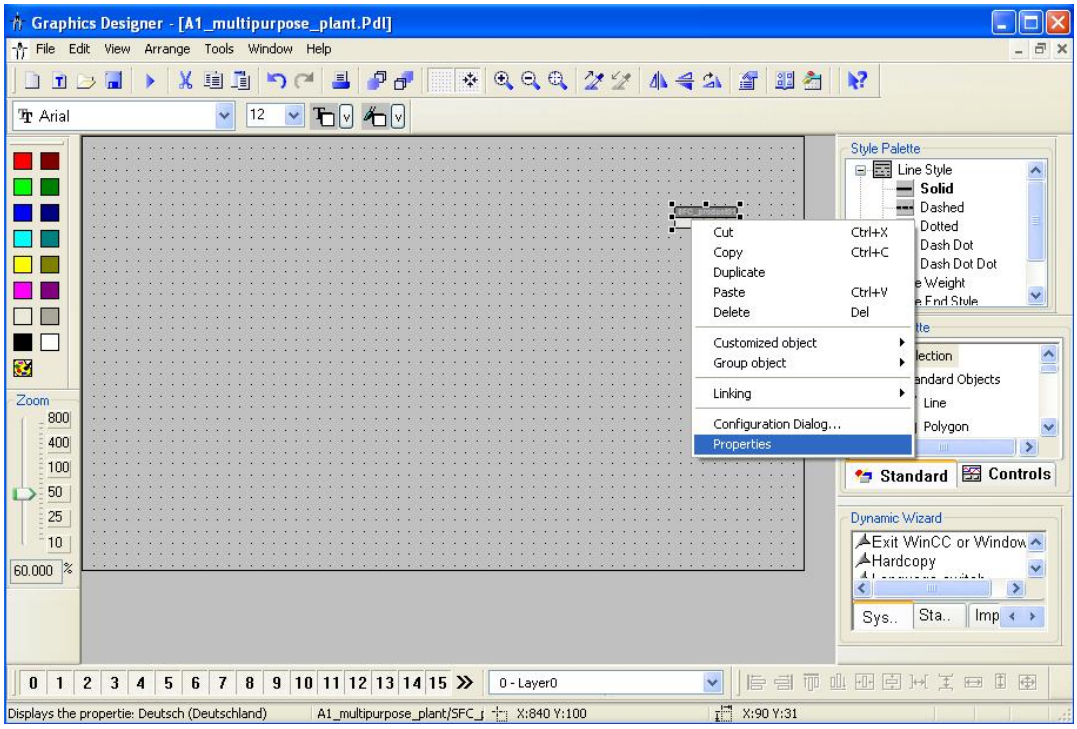
- ***Standard Toolbar***: Contains symbols and buttons to execute frequent commands quickly.
- ***Colors***: Provides for assigning colors to selected objects (one of 16 standard colors and one user defined color).
- ***Zoom***: Sets the zoom factor (in percent) for the active window.
- ***Style Palette***: Changes the look of a selected object. Depending on the object, we can change the line/frame style, the line/frame width, the line end style or the infill pattern.
- ***Object Palette***: Contains the standard objects (polygon, ellipse, rectangle, etc), smart objects (OLE Control, OLE Element, IO field etc.) and Windows objects (button, check box etc.).
- ***Dynamic Wizard***: Makes numerous frequently needed functions available. They can be generated using a dialog that guides the operator and also provides help.
- ***Layers***: Selects which of the 32 layers (Layer 0 to 31) is visible. Layer 0 is selected as a matter of standard.
- ***Alignment***: Lets you change the absolute position of one or several objects, change the position of objects relative to each other, or make the height and width of several objects uniform.

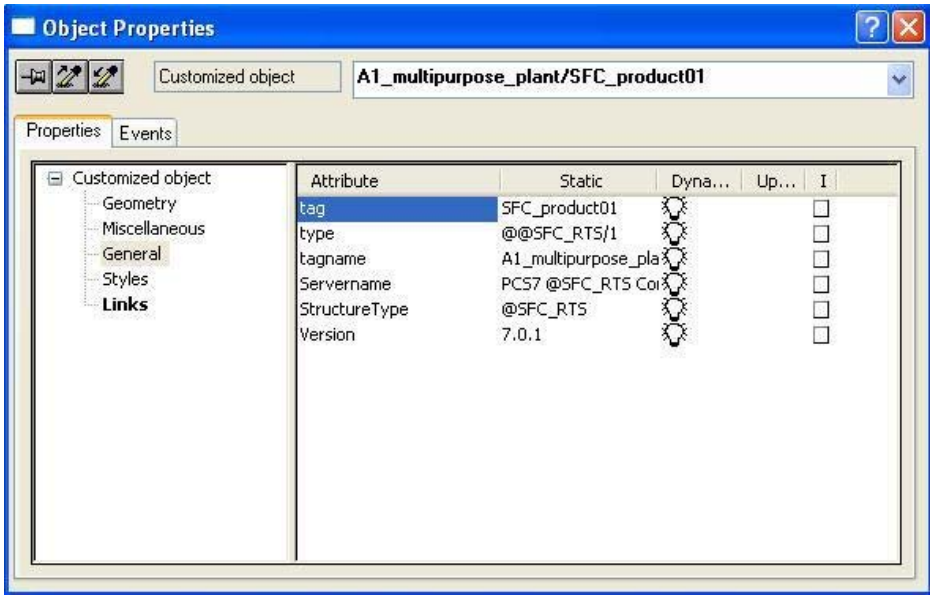
34. In our displays, the block icons are already included that can we can position as desired within the displays; as was done here for the SFC step sequence 'SFC_product1'. (→ SFC_product1)



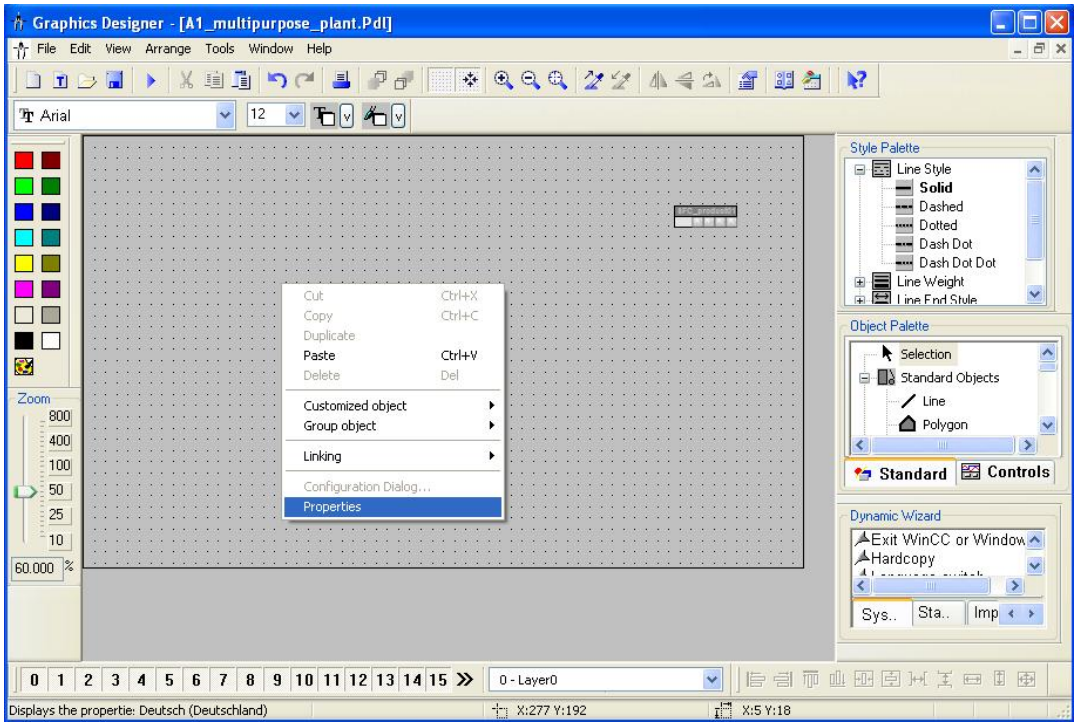
35. In the properties of the block icons, we can also specify the displayed name. Otherwise, a very long name is displayed where the path is included also.

    (→ Properties → Properties → General → tag → SFC_product01)



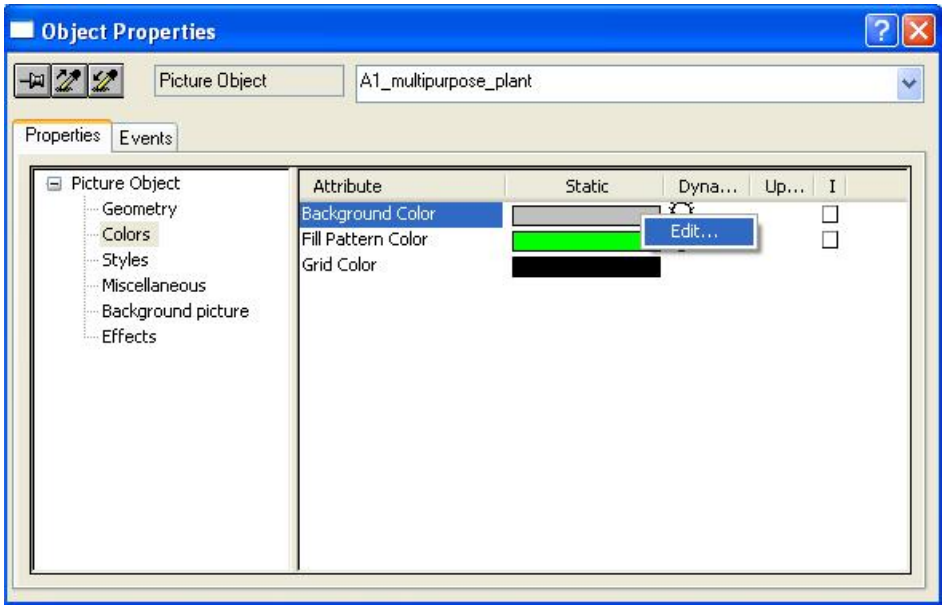


36. Next, we want to change the picture's background color to white. To do this, we right click on the picture's background and then select Properties. (→ Properties)

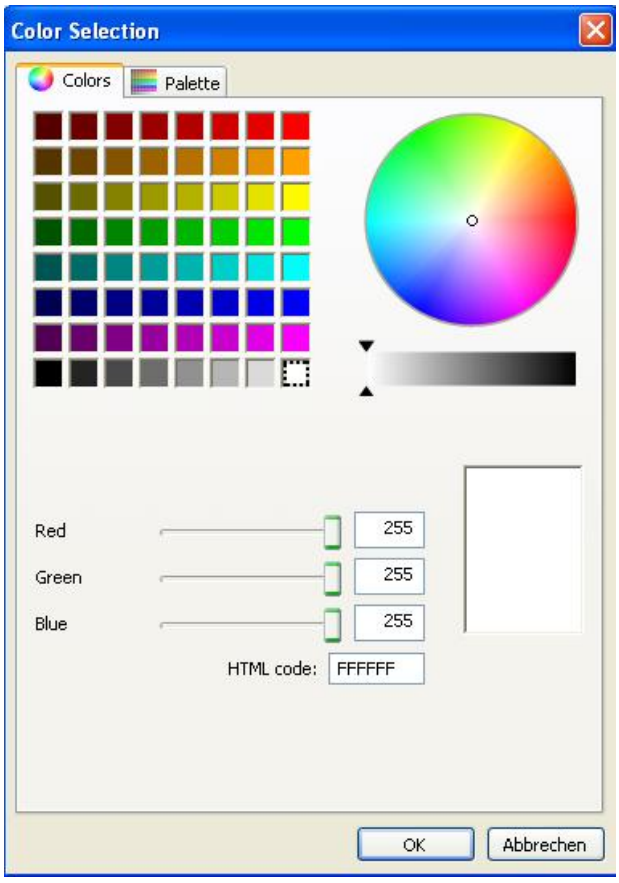


37. For each object as well as for the picture there are numerous properties that can be changed statically or dynamically (for example, interfaced with process variables). Here we are editing the background color.

    (→ Properties → Colors → Background Color → Edit)

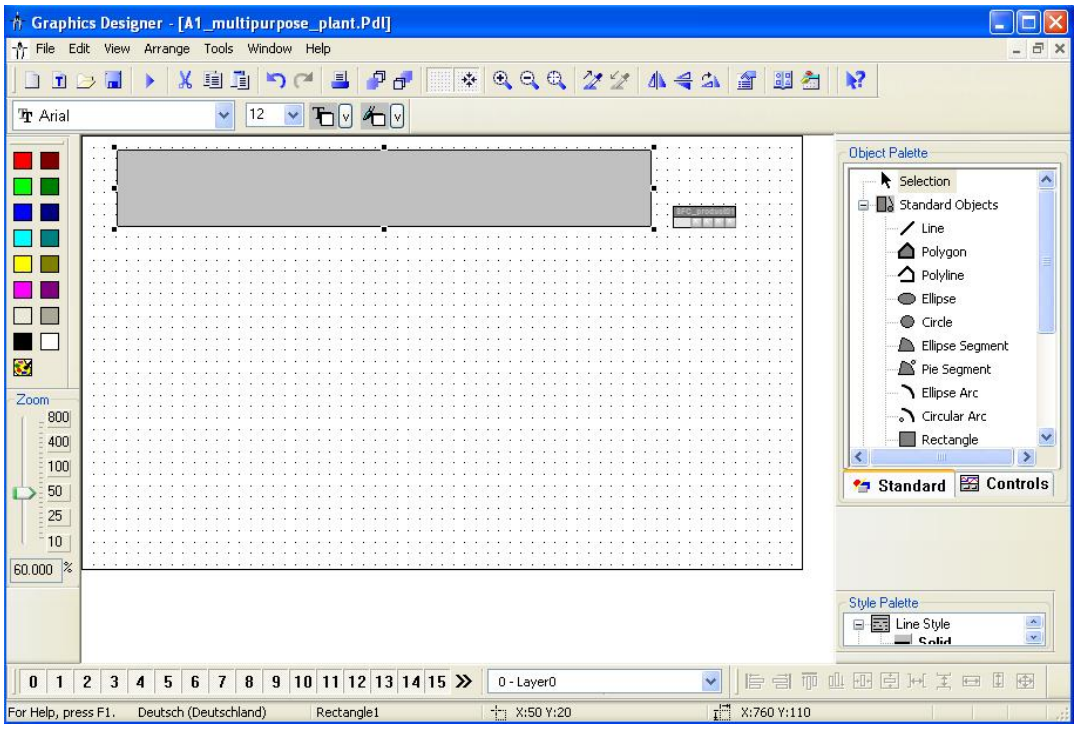


38. For our color we select white. (→ OK)

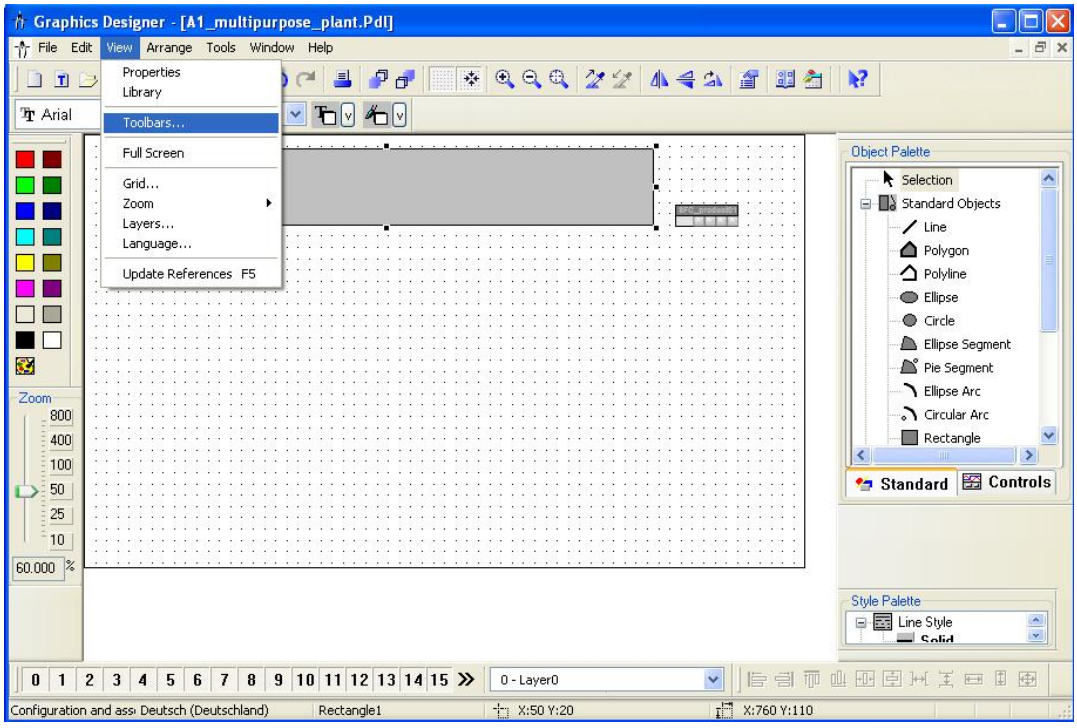


39. Next, we click in the Object Palette on Rectangle and then draw a large rectangle in our picture.

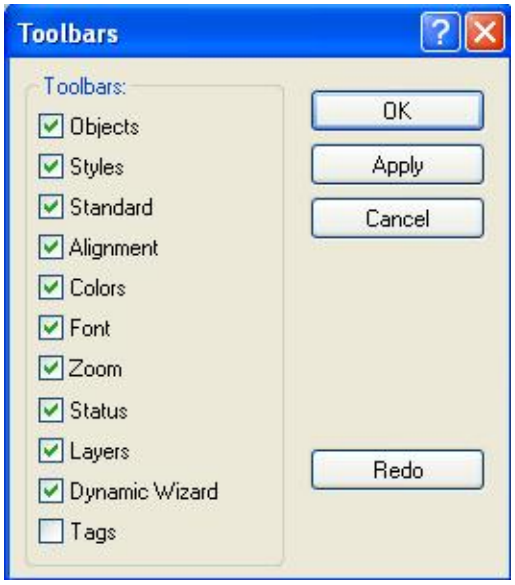
    (→ Object Palette → Standard Objects → Rectangle)



40. Now we open the selection for the toolbars if the ***Dynamic Wizard*** is not yet displayed. (→ View→ Toolbars)

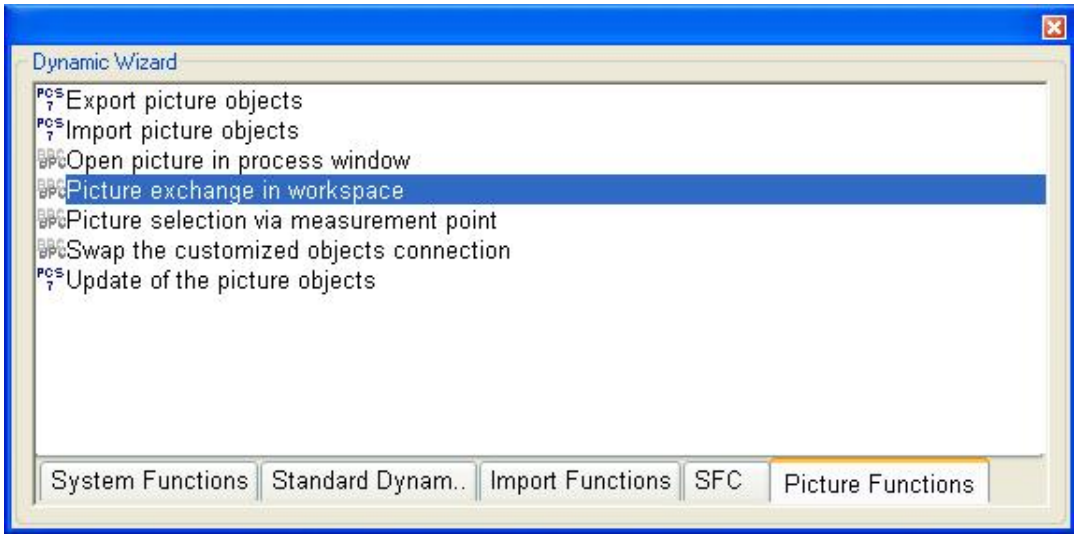


41. In the selection, we click on 'Dynamic-Wizard'. (→ Dynamic Wizard → OK)

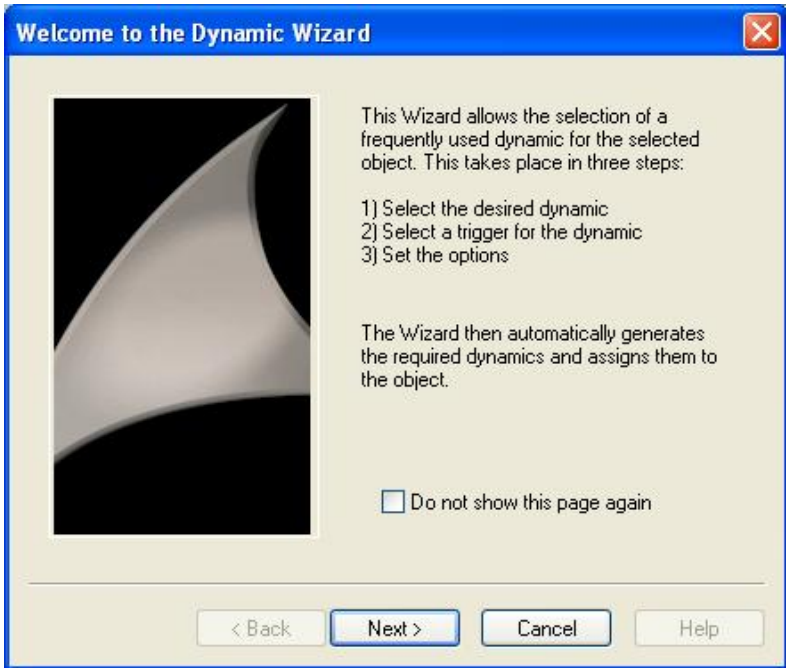


42. When the toolbar for the ***Dynamic Wizard*** is displayed, we select in 'Picture Functions' the 'Picture exchange in workspace'.

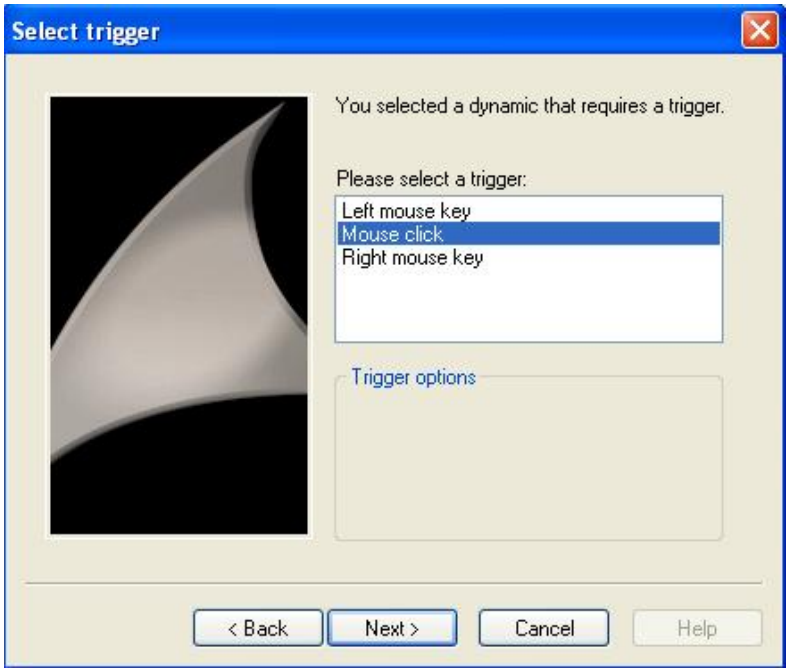
    (→ Picture Functions → Picture exchange in work space)



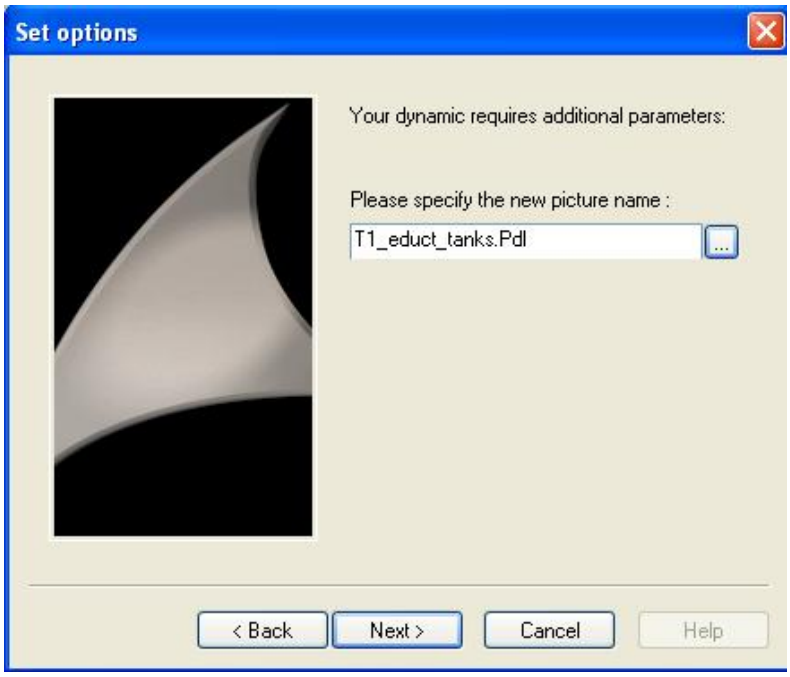
43. We read the explanation and continue. (→ Next)

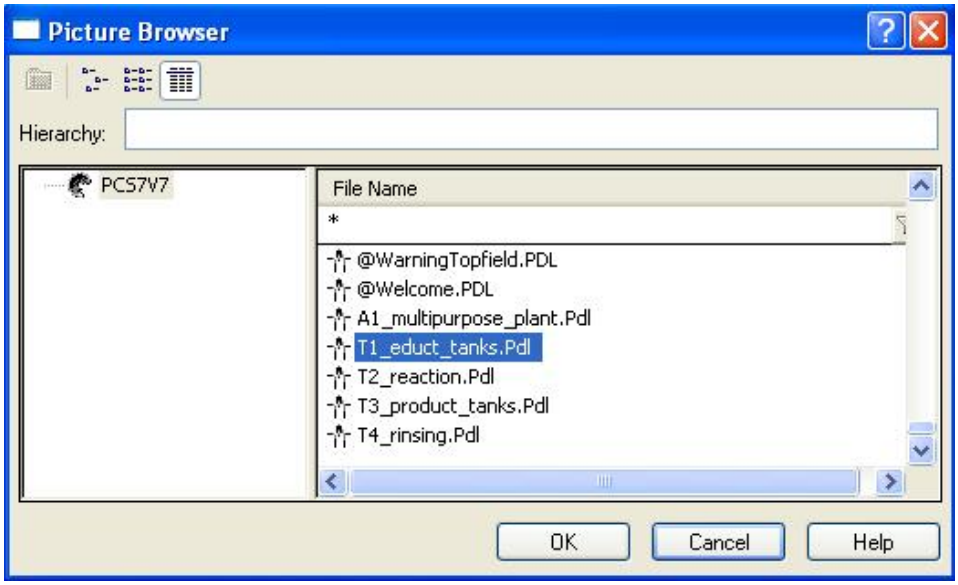


44. As trigger we select 'Mouse click'. (→ Mouse click → Next)

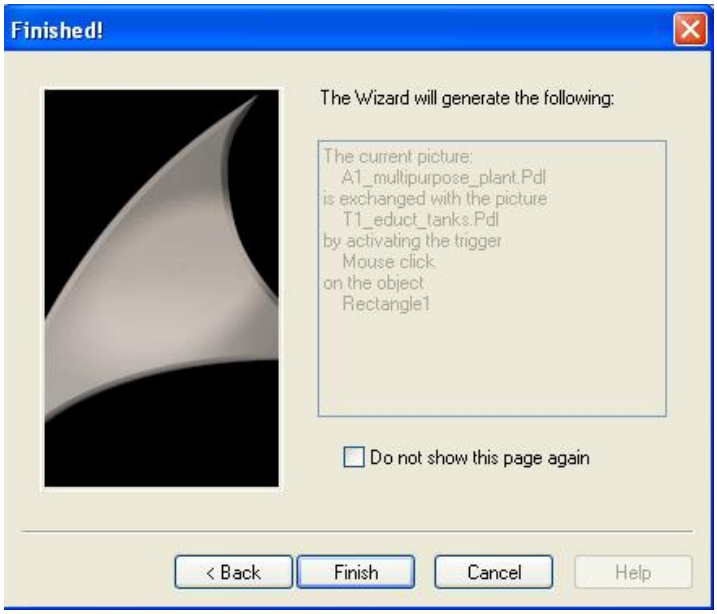


45. Here we select the picture that is to be displayed after operating the button. (→  → T1_educt_tanks.Pdl → OK → Next)



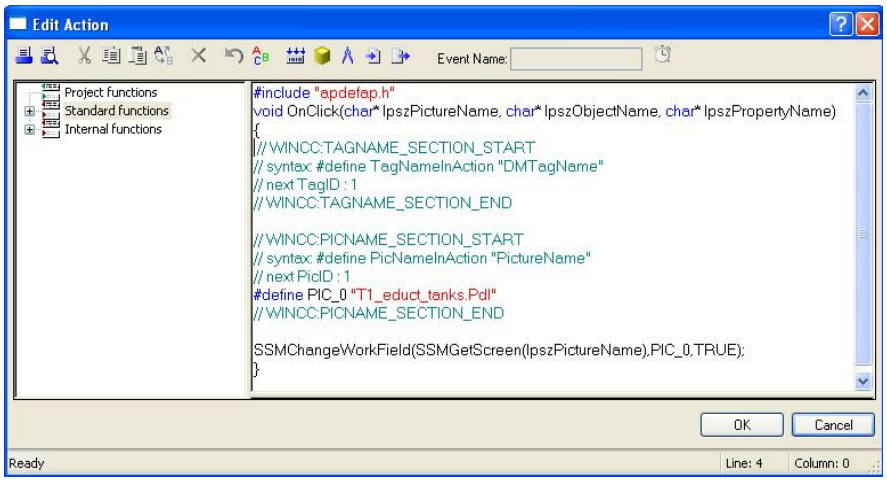


46. In the last window of the dialog, the complete selection is listed. (→ Finish)

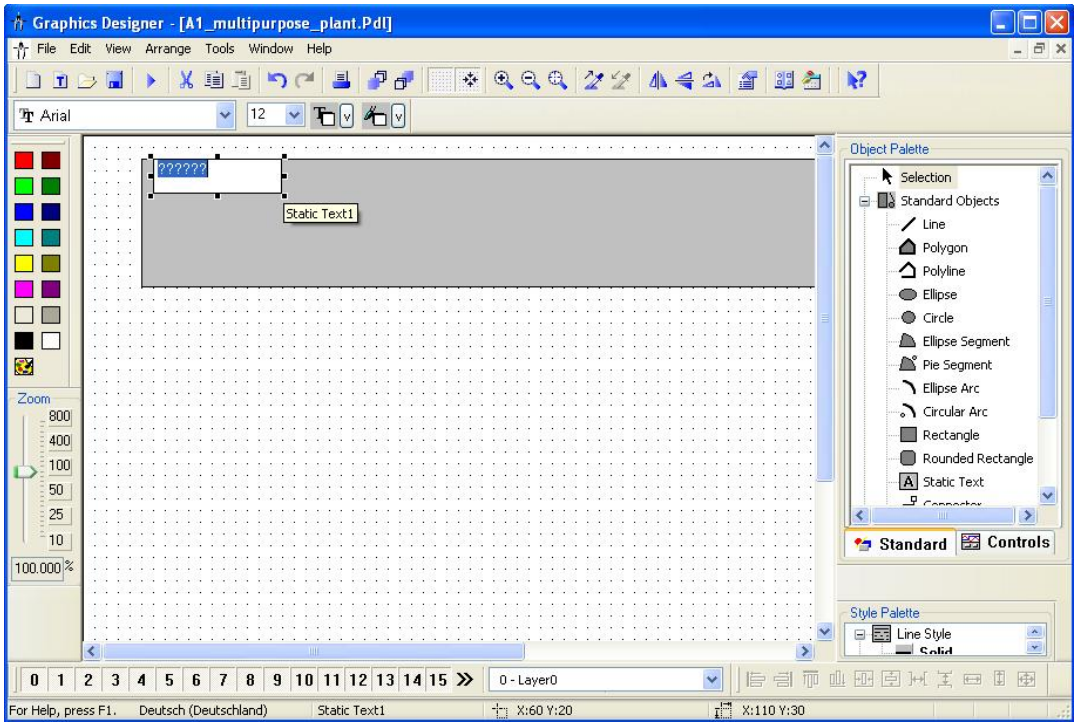


47. If you want to view the result, the mouse and the mouse action are located in Properties under 'Events'. By double clicking on the symbol , you can view the C-script that was generated.

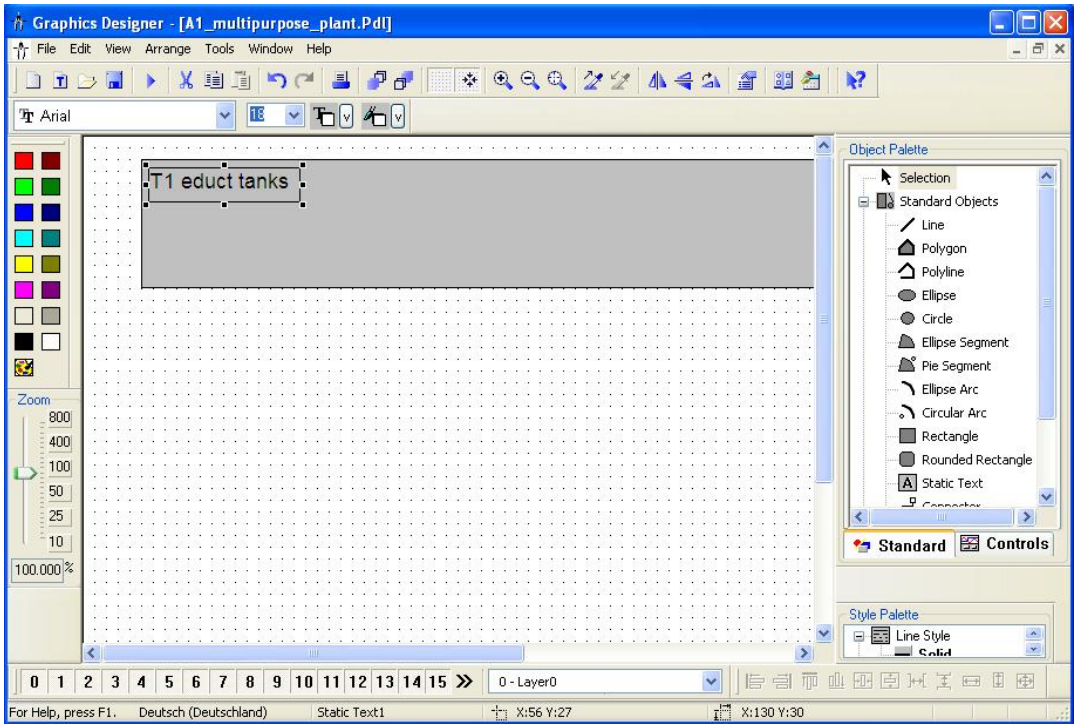
(→ Properties → Events → Mouse → Mouse Action → OK →)



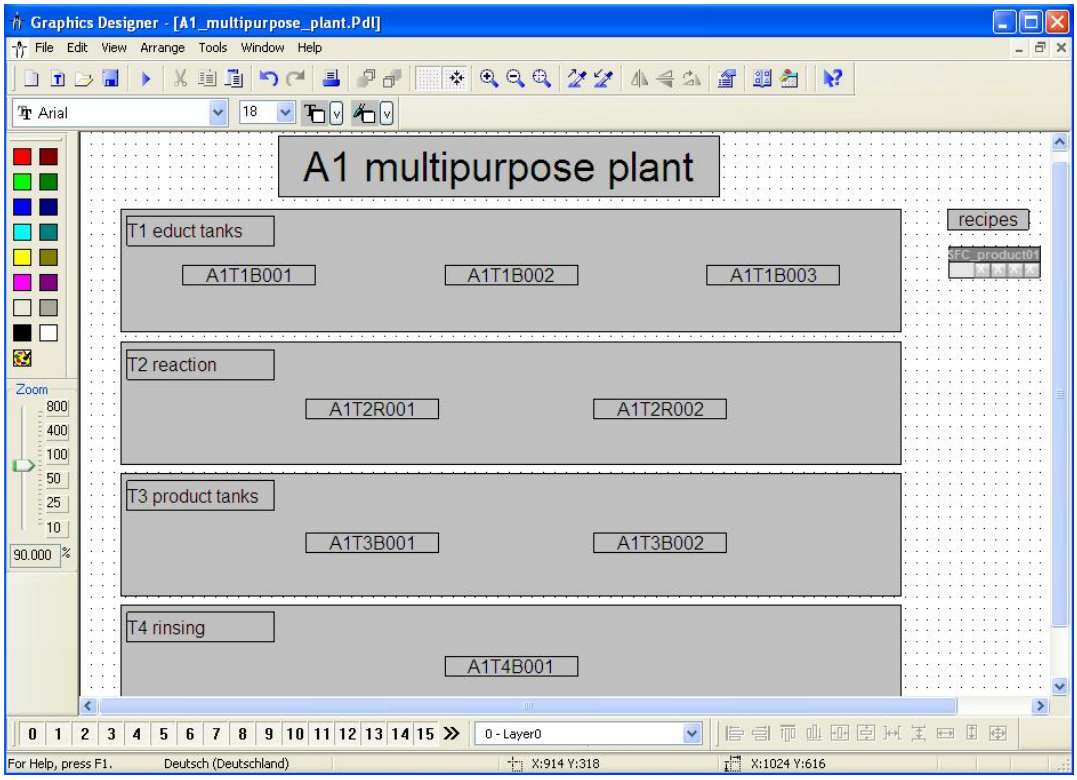
48. Now we add the texts to the area for T1_educt_tanks. To this end, we select from the object palette the static text. (→ Static Text)



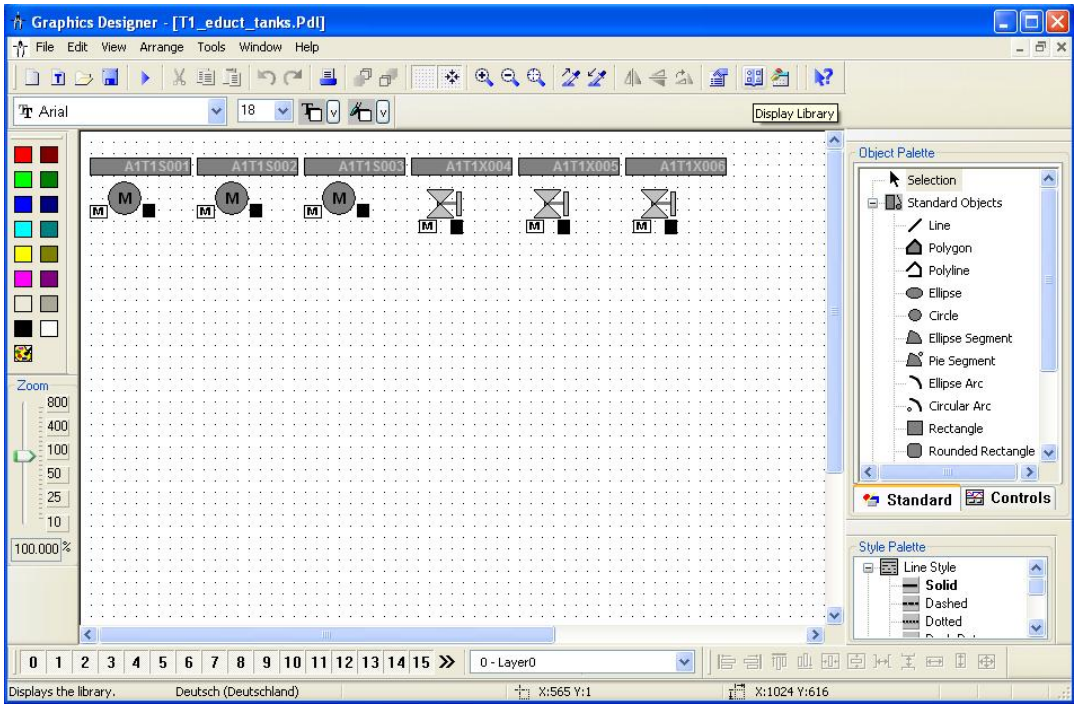
49. Enter the text and change the font and the color. Using the menu bar is the simplest way of doing this. (→ T1 educt_tanks)



50. Now, complete the picture 'T1_educt tanks' with the texts and buttons shown here for changing to T2, T3 and T4.

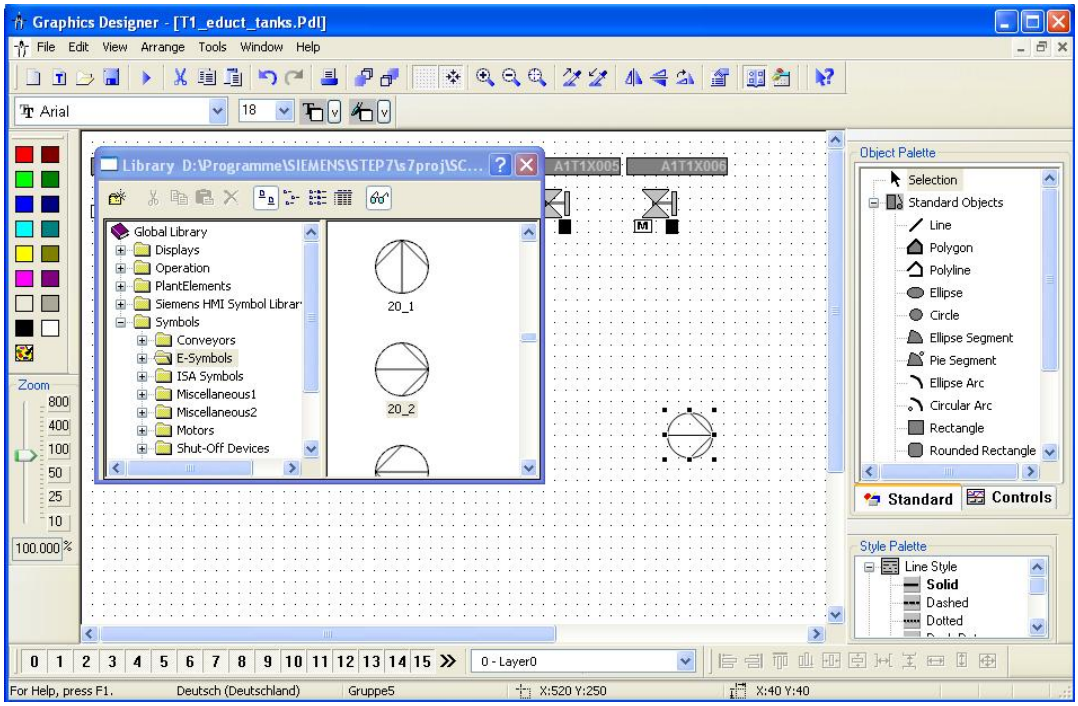


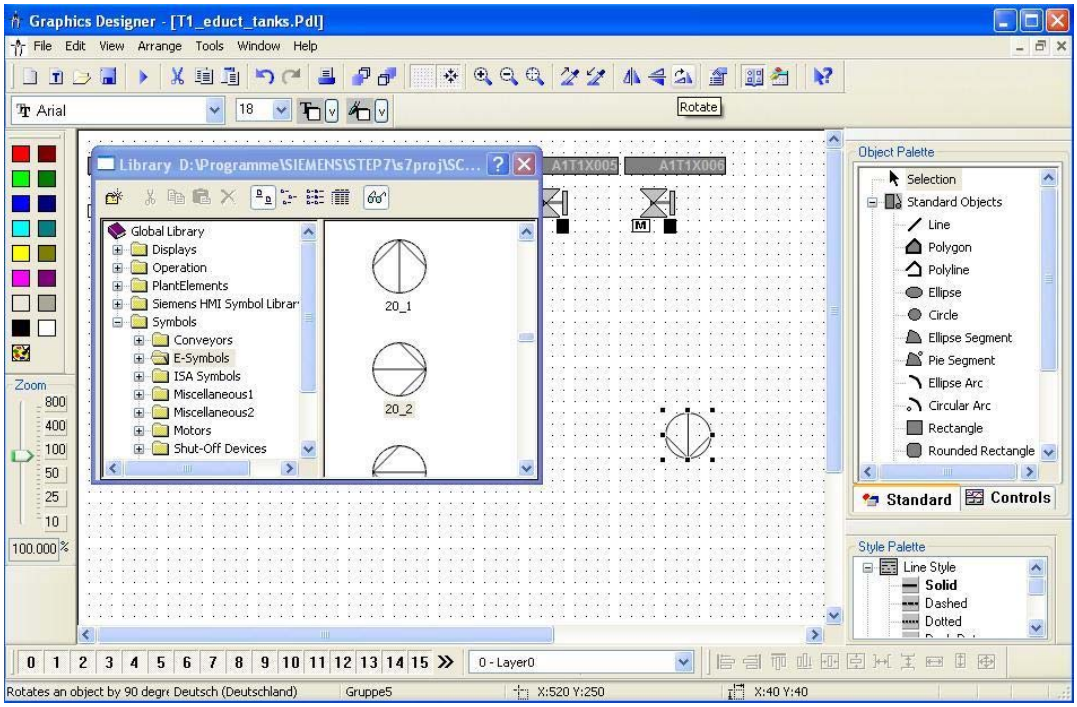
51. As the next picture we open 'T1_educt_tanks' from the ***WinCC Explorer***. There, after having changed the background color to white, we want to have the library displayed by clicking on  . (→ )



52. From the library, we now first drag a symbol for the pump into the work display and then rotate it in downward direction.

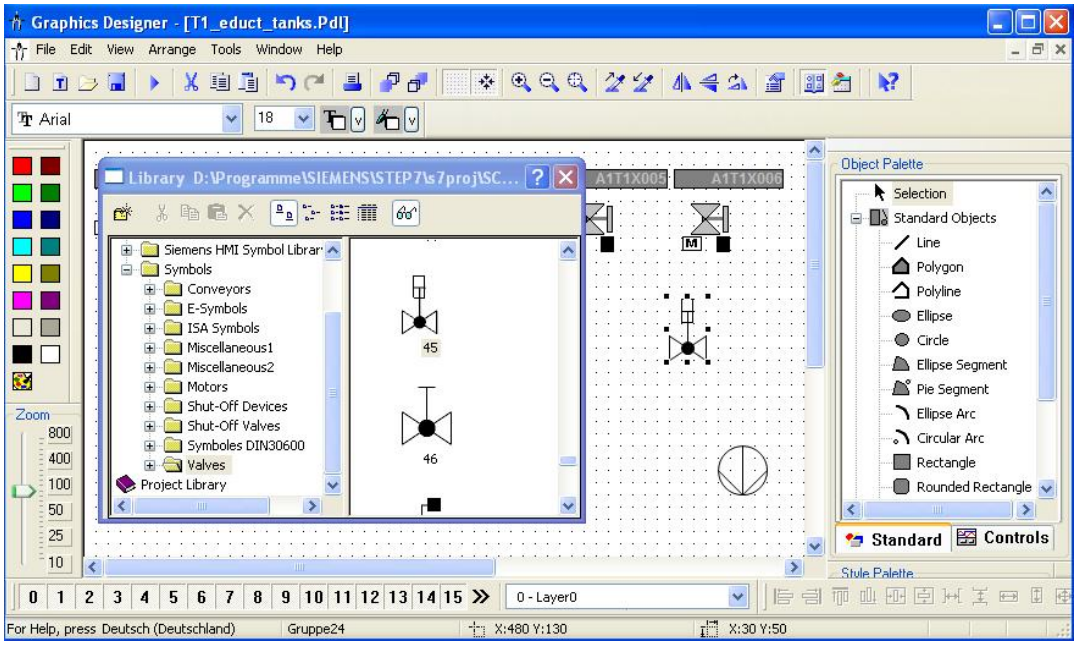
(→ Global Library → Symbols → E-Symbols → 20_2 → )



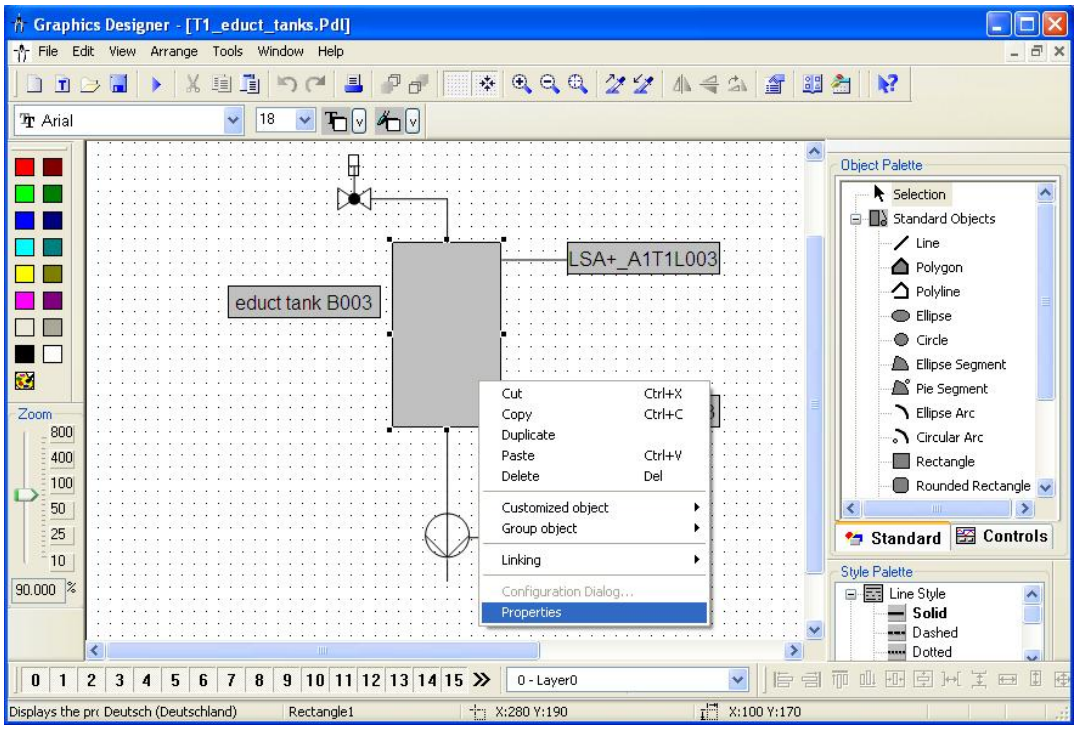


53. Then, we drag a symbol for the valve from the library into the work display.

    (→ Global Library → Symbols → Valves → 45)

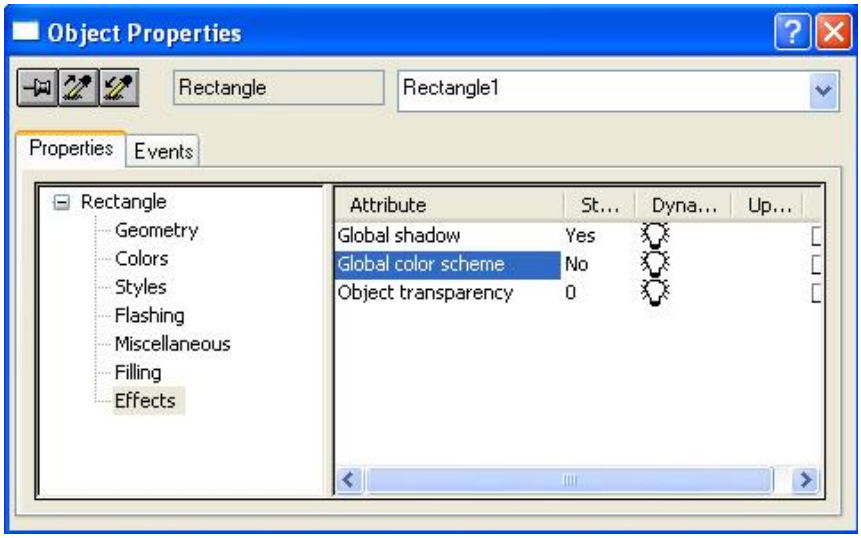


54. After we have inserted -as shown here- additional lines and text fields, we are positioning a rectangle to represent the tank, and select its properties. (→ Rectangle → Properties)



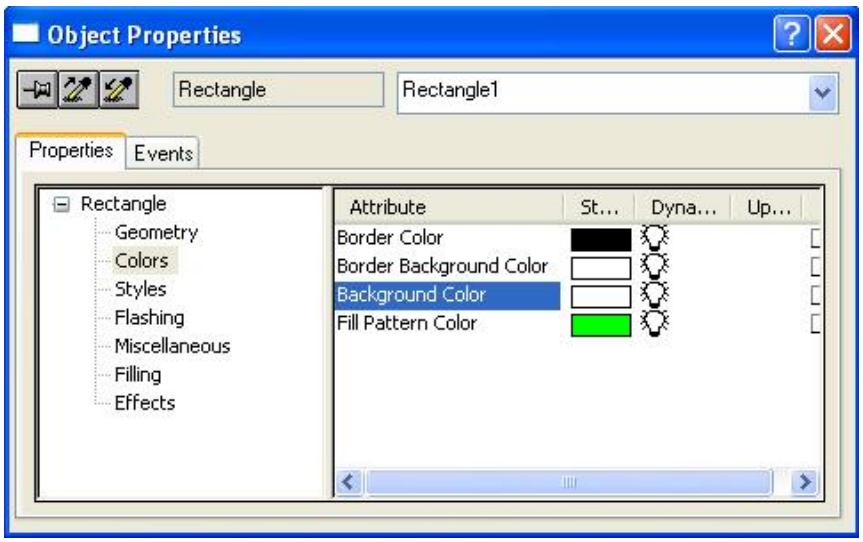
55. To change the color we deactivate the global color scheme.

    (→ Properties → Representation → Global color scheme → no)



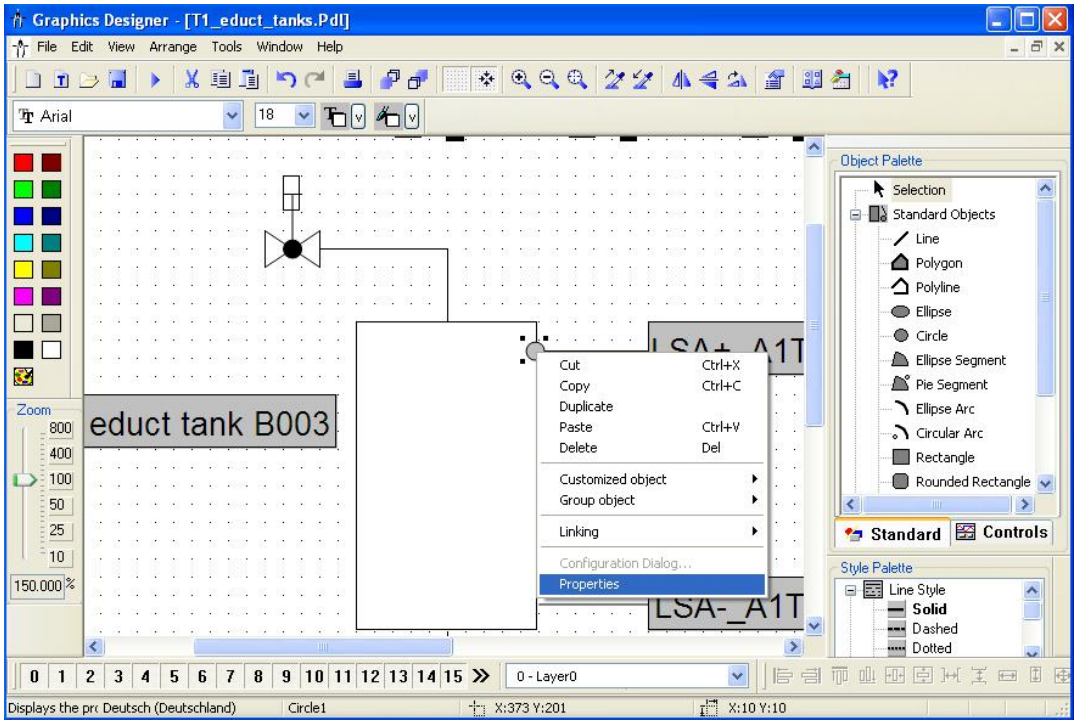
56. Then we change the background color to white.

    (→ Properties → Colors → Background Color)

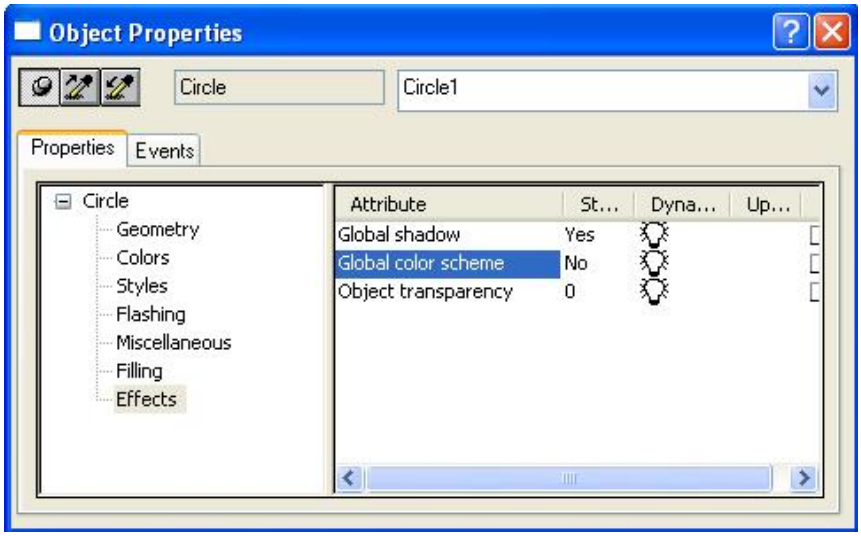


57. Now we are configuring a display of the level encoders. To this end, we are drawing -as shown here- a circle into our display. Then we select its properties.

    (→ Circle → Properties)

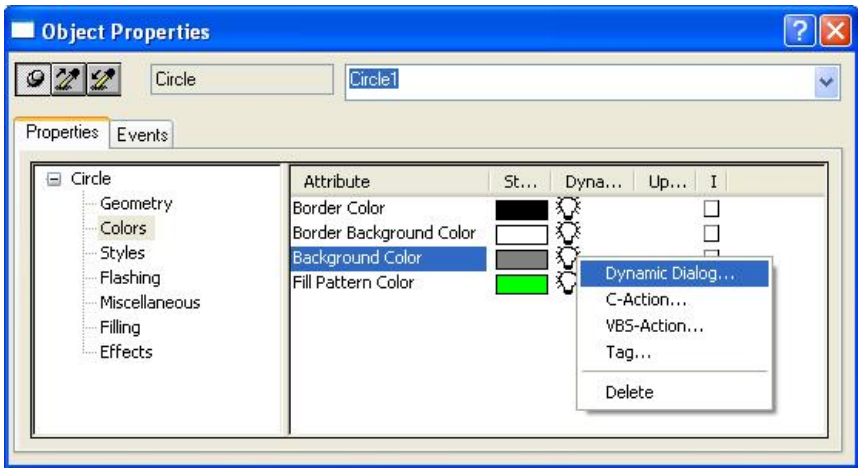


58. To have the color displayed dynamically, we deactivate the global color scheme. (→ Properties → Representation → Global Color Scheme → no)



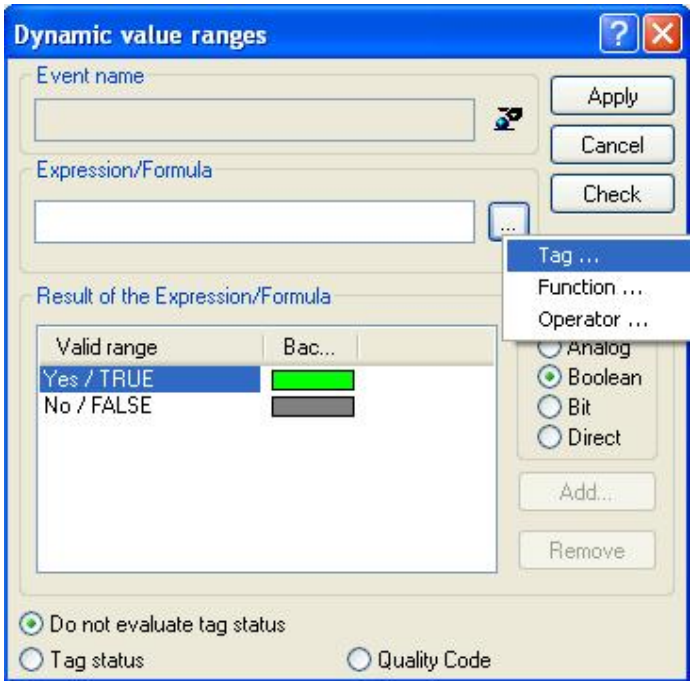
59. To implement a dynamic display, we select the background color with the right mouse key, and then the dynamic dialog.

    (→ Properties → Colors → Background Color → Dynamic Dialog)



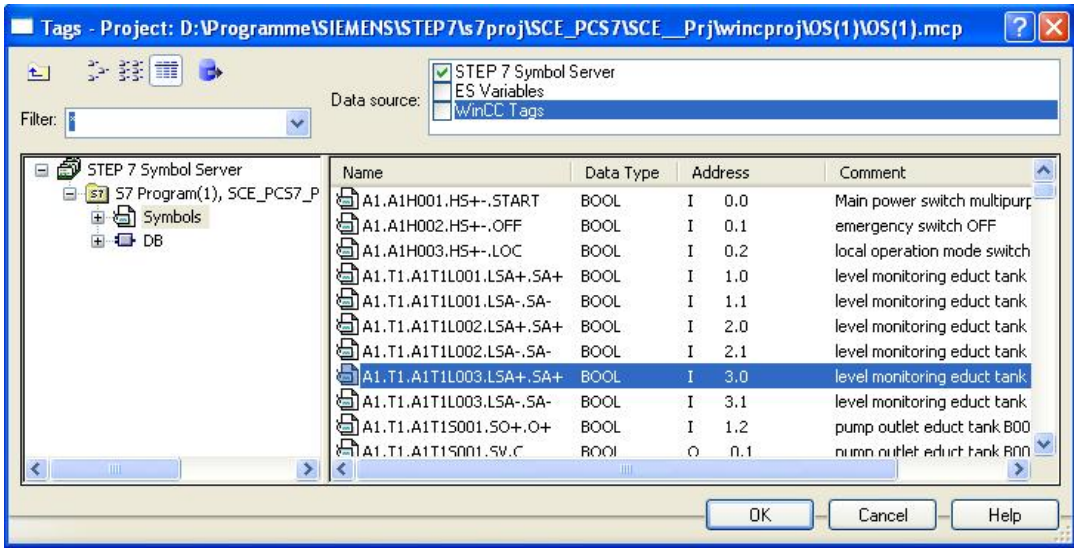
60. In the following dialog, we first select the data type Bool, and then we change the color at Yes/TRUE to green. Finally, we select 'Tag' to make it dynamic.

    (→ Bool → Yes/TRUE →  →  → Tag)



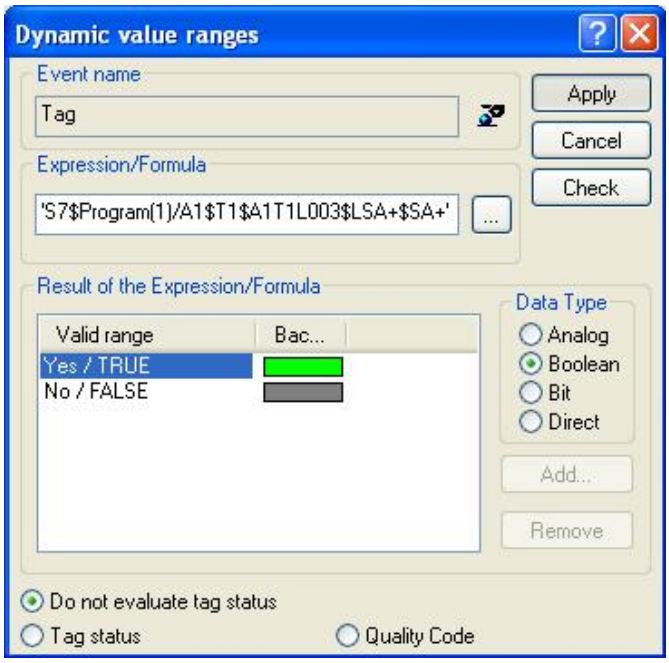
61. At Tags, we select as data source 'STEP 7 Symbol Server' and there, at the symbols, Input I3.0 for 'level monitoring educt tank B003 Operating Point H'.

    (→ Data source → STEP 7 Symbol Server → A1.T1.A1T1L003.LSA+.SA+ / I3.0 / level monitoring educt tank B003 Operating Point H → OK)



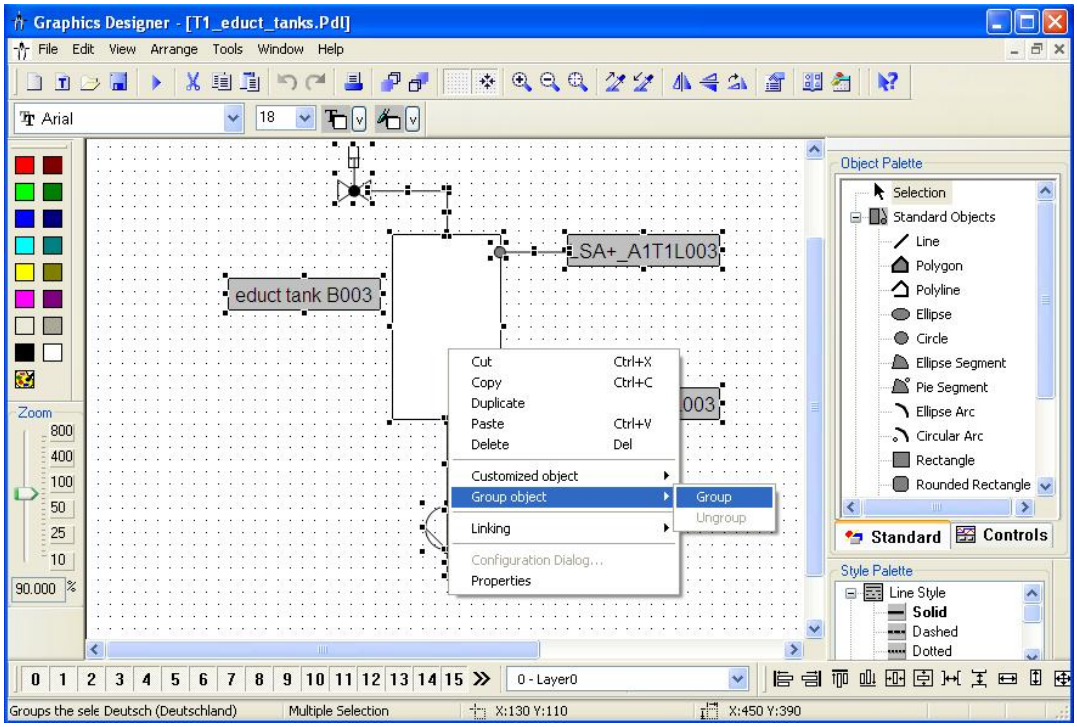
62. Then, we accept the settings in the dynamic dialog.

    (→ Apply → )

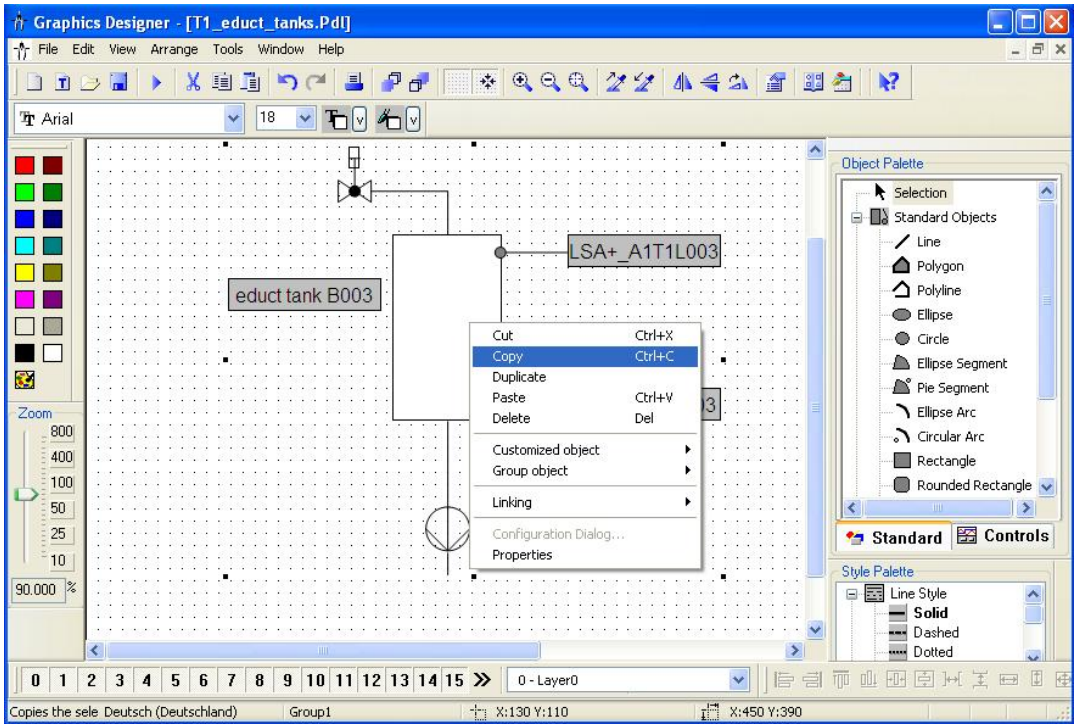


63. The steps shown above are also performed for encoder 'A1.T1.A1T1L003.LSA+.SA- / I3.1 / level monitoring educt tank B003 Operating Point L'. Then, the elements shown here are highlighted together and grouped.

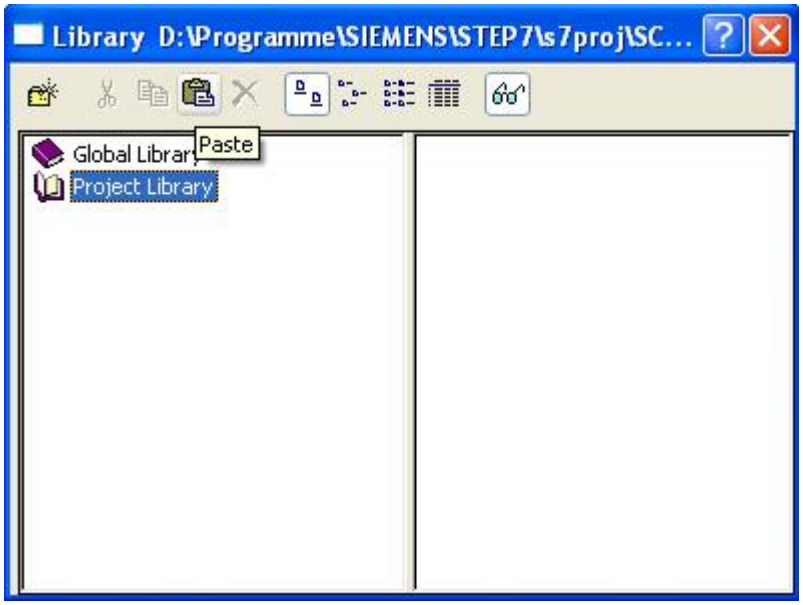
    (→ A1.T1.A1T1L003.LSA+.SA- / E3.1 / level monitoring educt tank B003 Operating Point L → Group Object → group)



64. Then, we copy the group. (→ Copy)

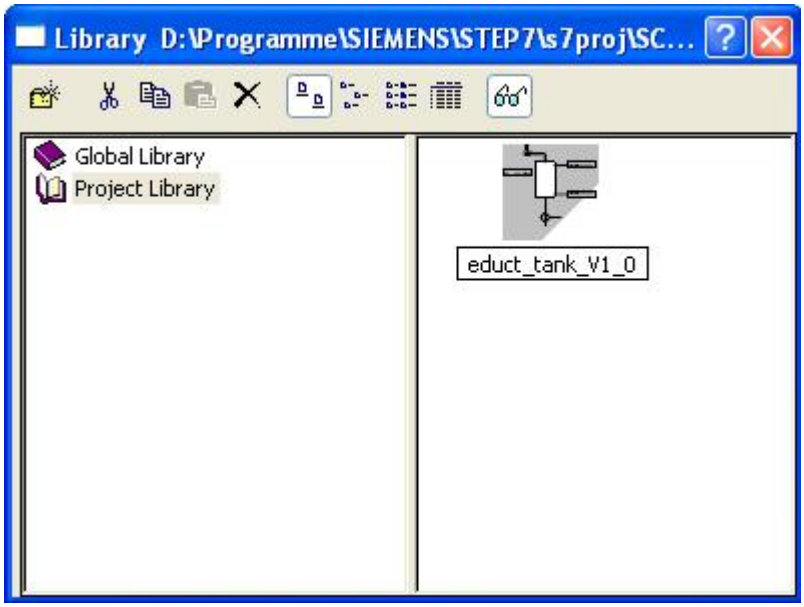


65. Next, we open the library and enter the group in the project library. (→  → Project Library → )

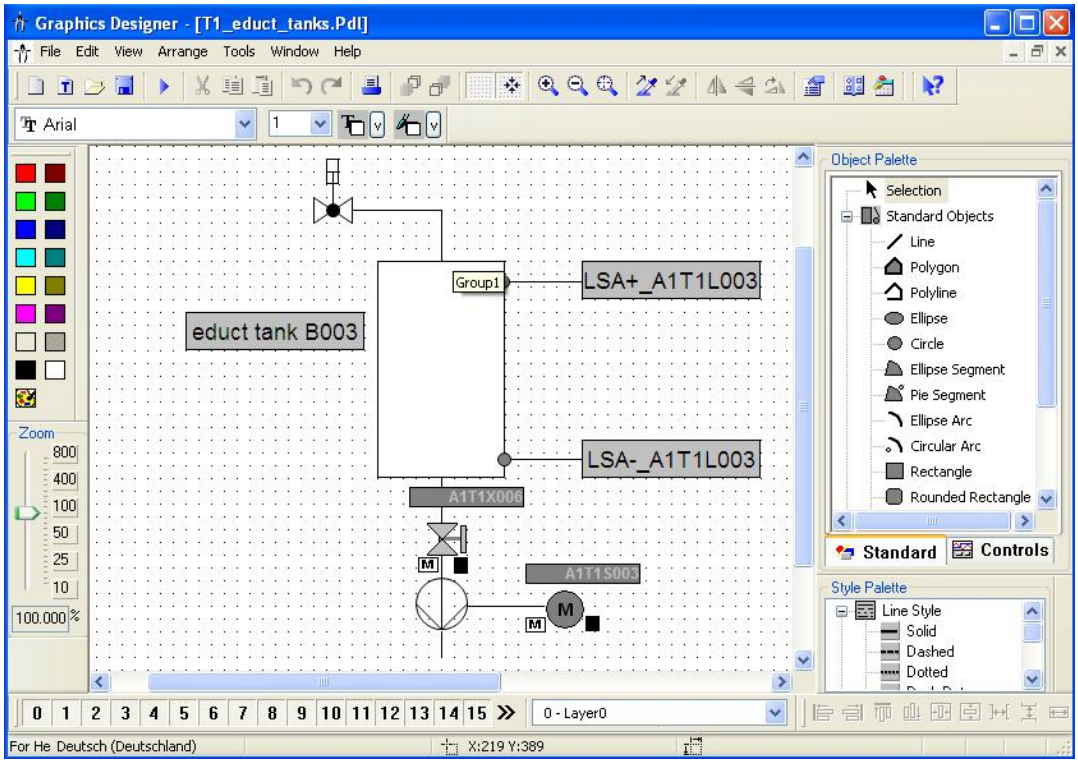


66. Our group in the library is then renamed 'educt_tank_V1_0'.

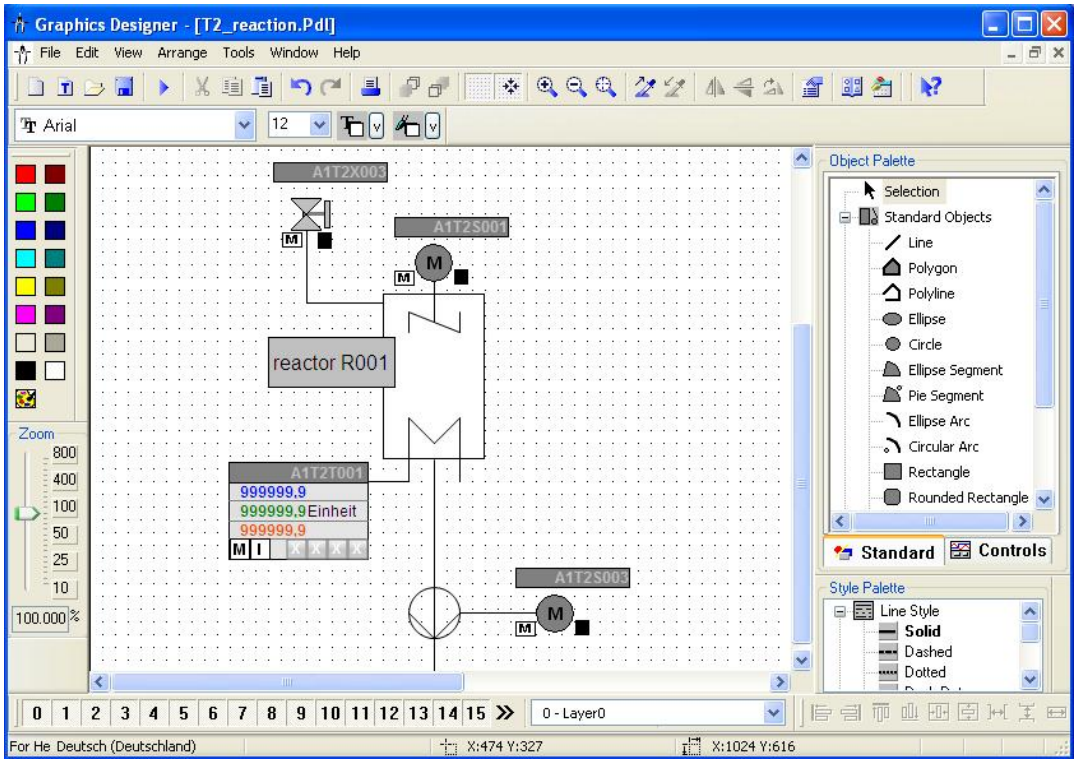
(→ educt_tank_V1_0 → )



67. Now, in picture 'T1_educt_tanks.Pdl', the picture blocks (face plates) for valve 'A1T1X006' and pump 'A1T1S003' are positioned as shown here.

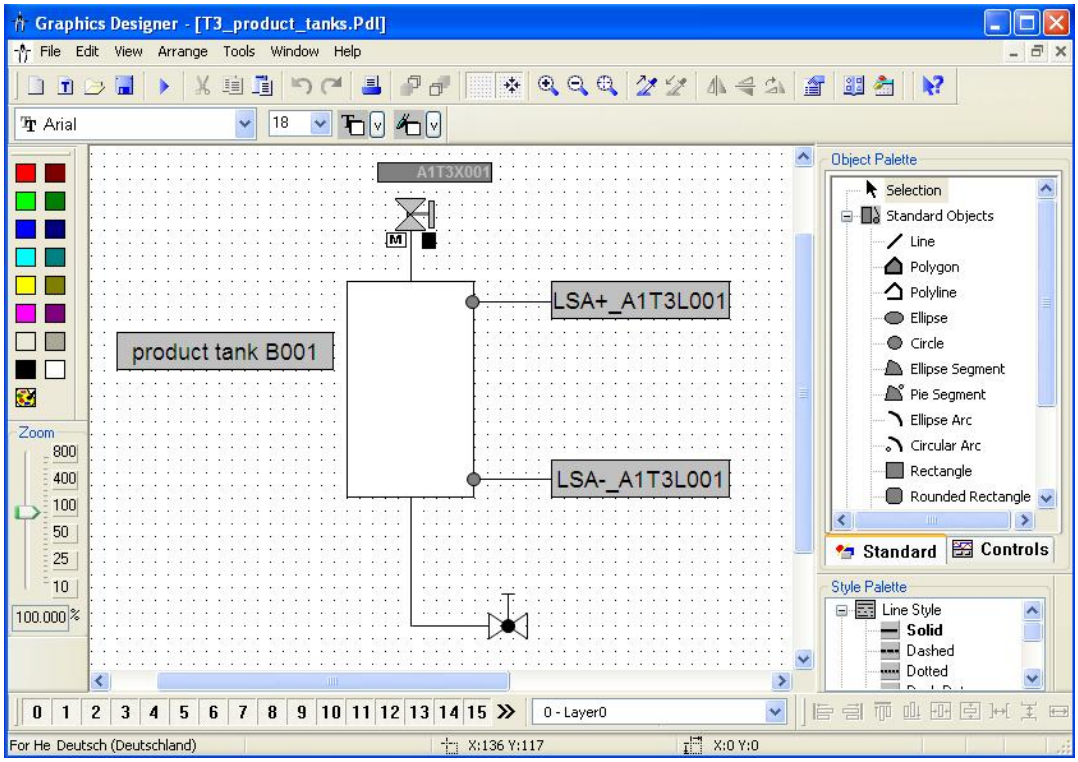


68. Picture 'T2_reaction.Pdl' is set up just as picture 'T1_educt_tanks_Pdl'. Here also, a group 'reaction_V1_0' is stored in the project library. The stirrer can be taken from the library as the picture is generated. (→ Global Library → Symbols → E-Symbols → 11)

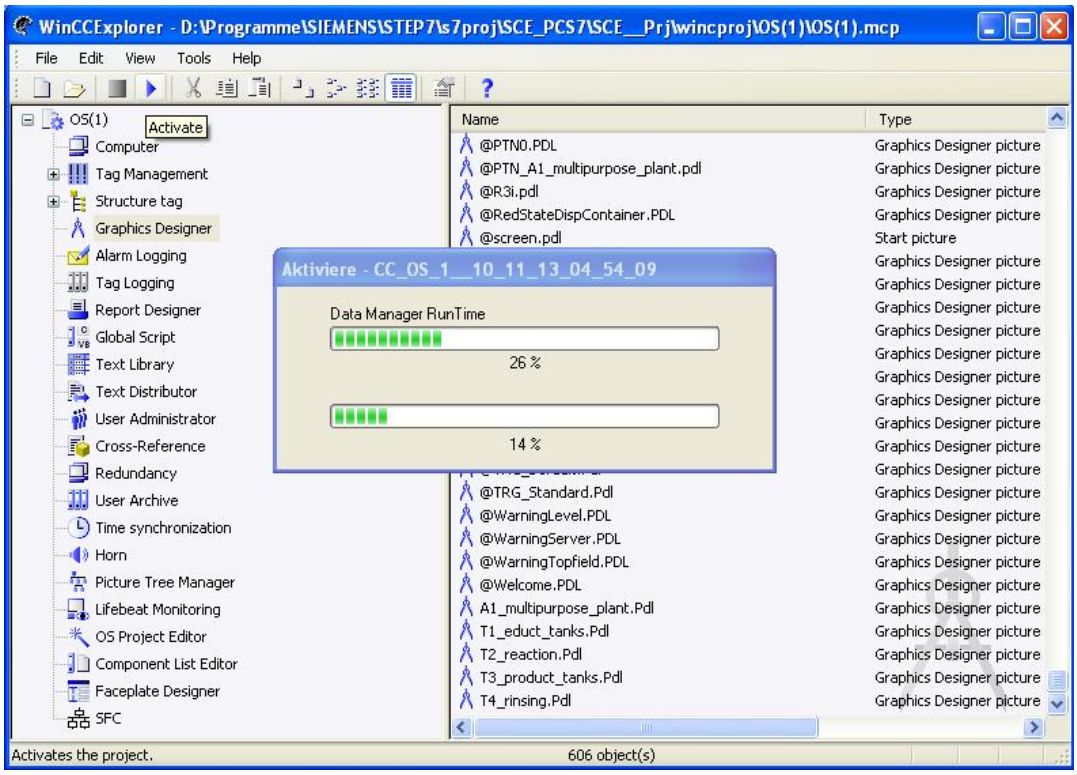


69. The picture 'T3_product_tanks.Pdl' is set up just as the picture'T1_educt_tanks.Pdl'. Here, the product tank is stored in the project library as the group 'product_tanks_V1_0'. The valve can be taken from the library as the picture is generated.

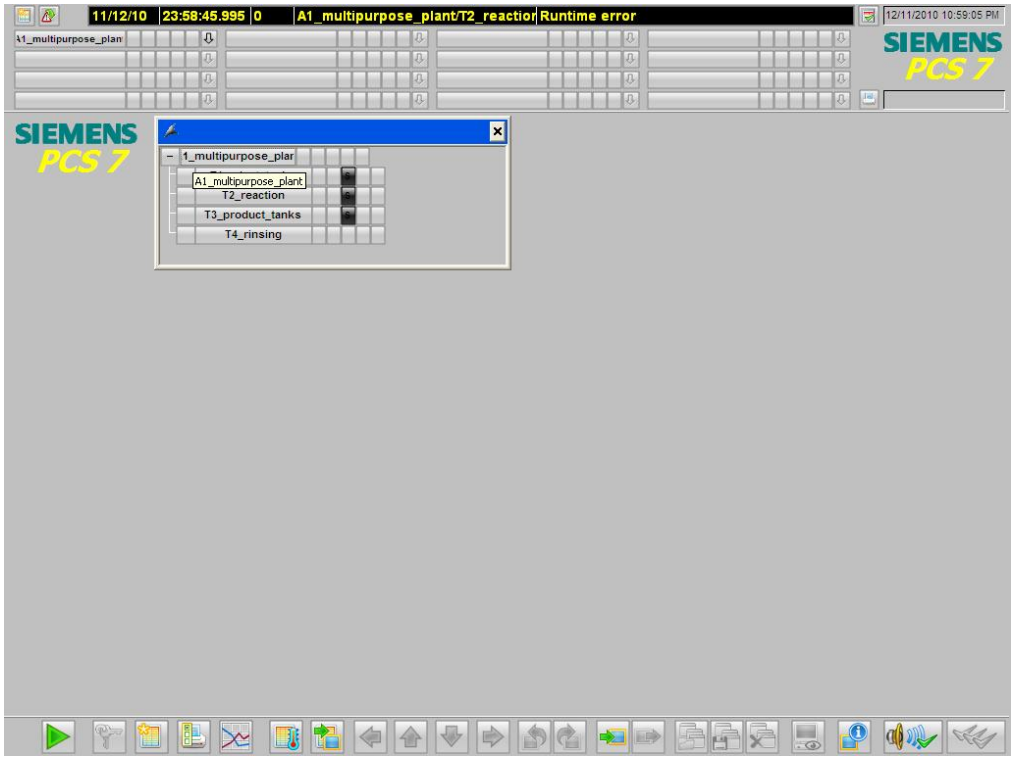
    (→ Global Library → Symbols → Valves → 46)



70. We now start the OS in runtime in the ***WinCC Explorer***. (→ Activate)

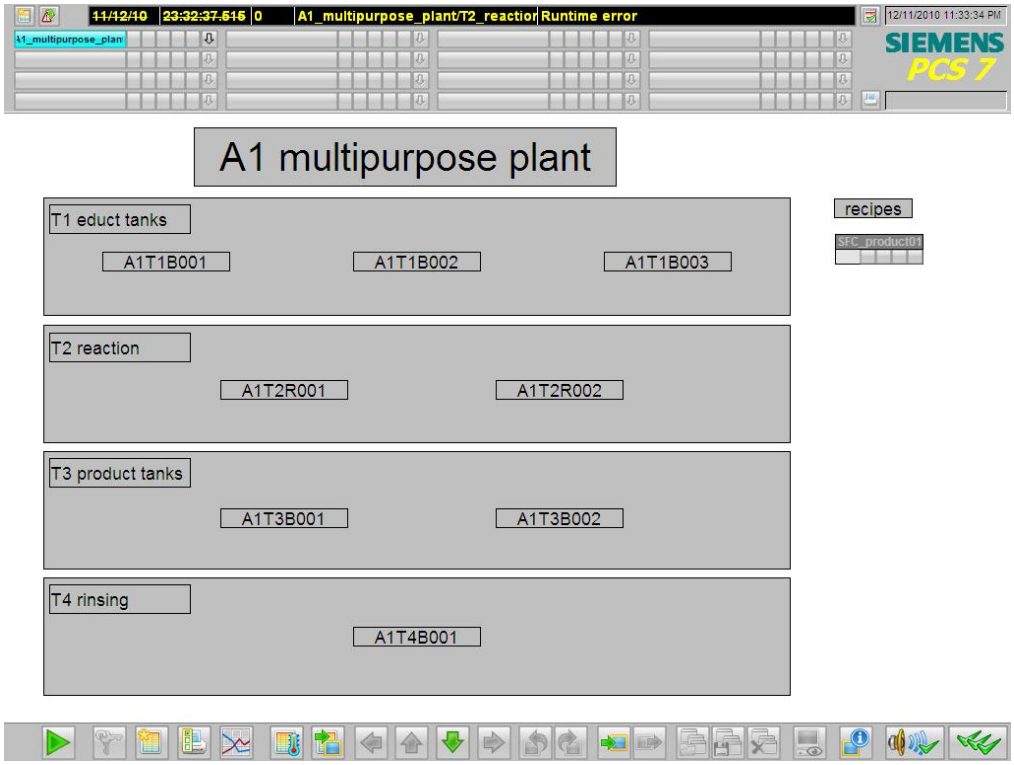


71. To open a certain picture, we click on 'A1_multipurpose_plant' in the overview area, for example. A window appears with the pictures listed. (→ A1_multipurpose_plant)

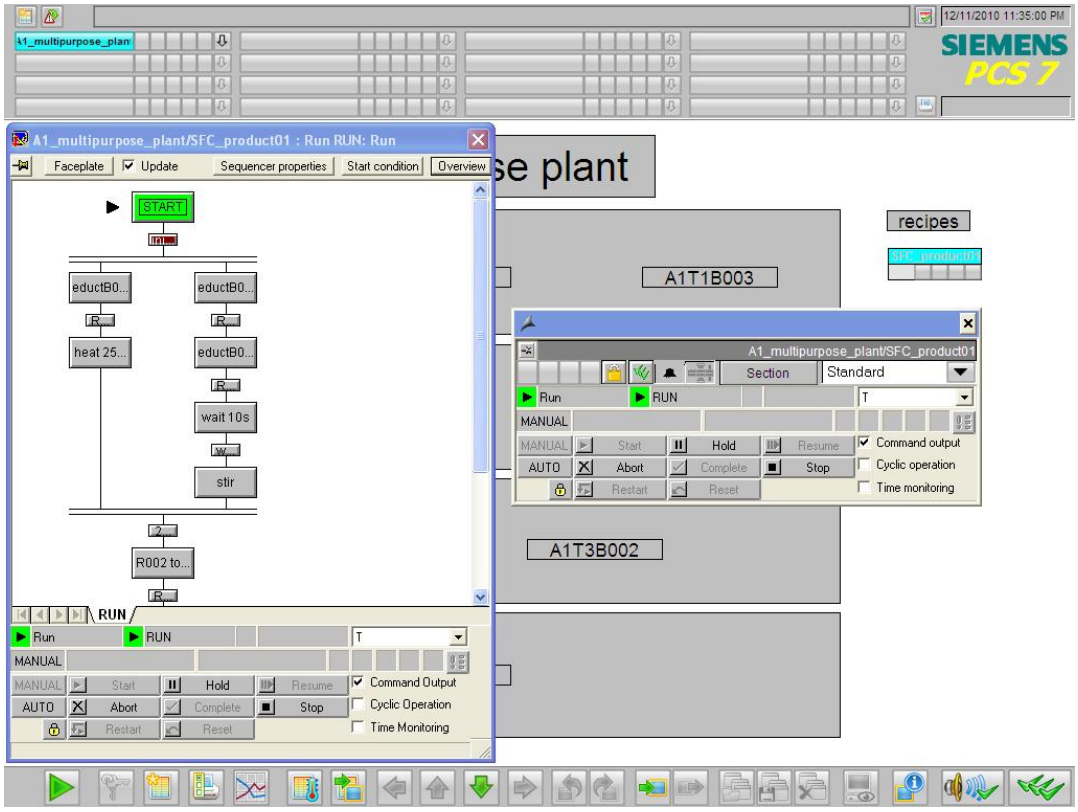


72. In our picture 'A1 multipurpose_plant', an overview of all areas and the block icon for the step sequence are displayed. By clicking on the block icon, we can open and operate the SFC_Product01.

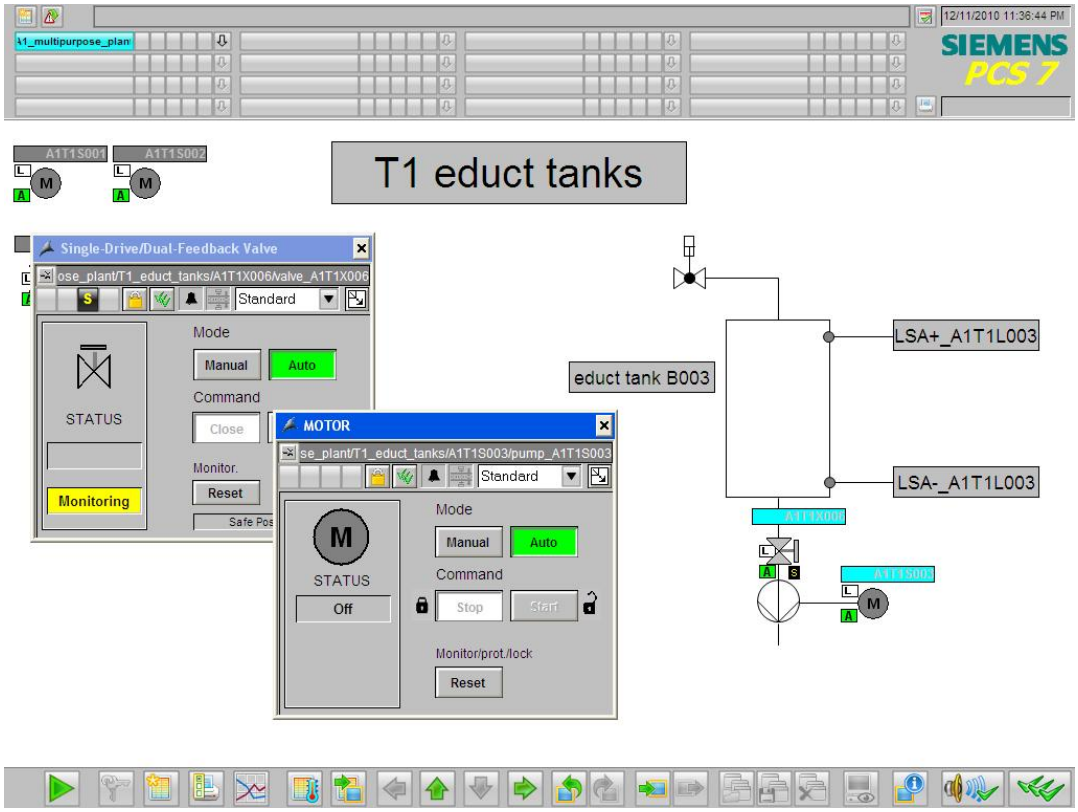
(→ SFC_Product01)



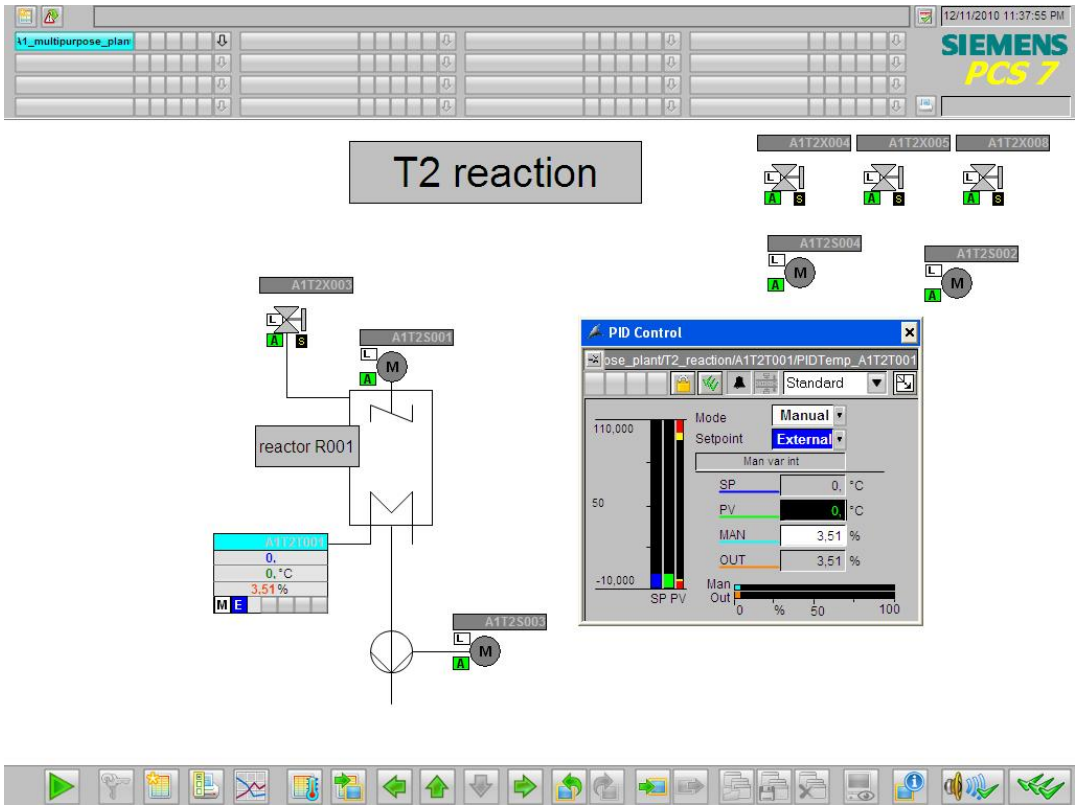
73. Here, the step sequence can be started, watched and operated, just as directly as in the SFC Editor.



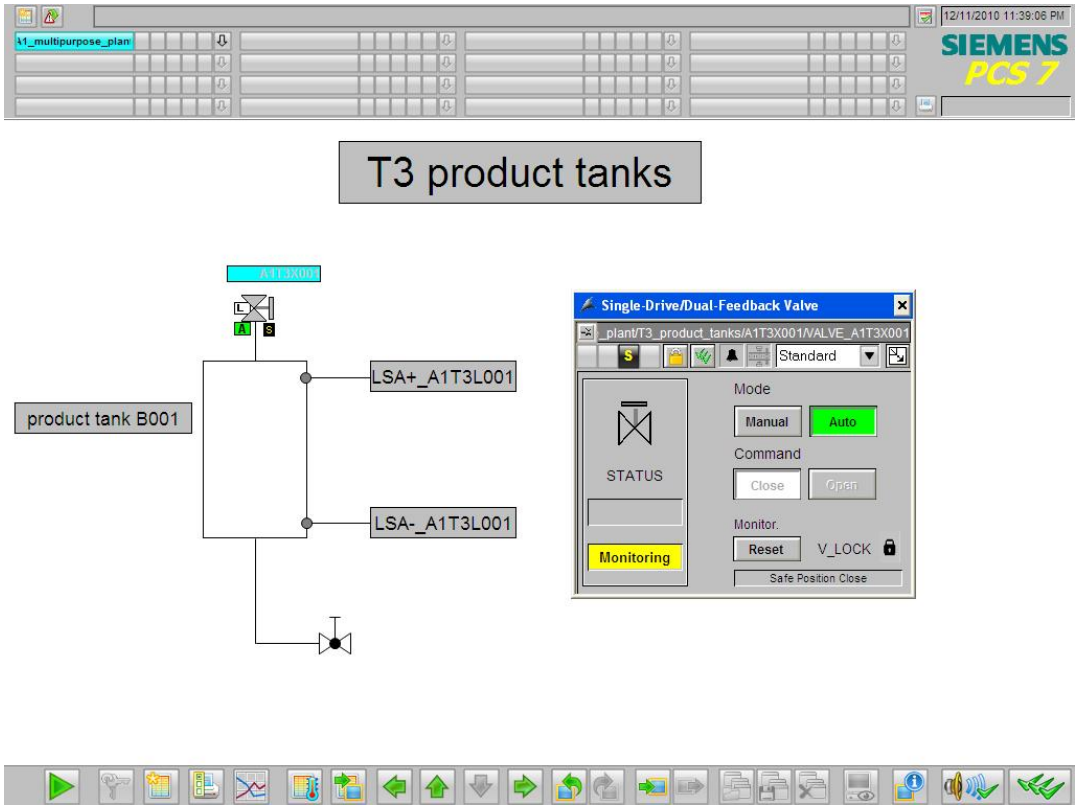
74. Here, we can see the operating screen for 'T1_educt tanks', with the face plates for 'Valve_A1T1X006' and 'Pump_A1T1S003'. Within the face plates, their operating modes can be selected and then operated in that mode.



75. Here, the operating screen for 'T2_Reaction' is displayed with the faceplate for 'PIDTemp_A1T2T001', the PID controller for the temperature in the reactor. The controller parameters can be changed in this faceplate.

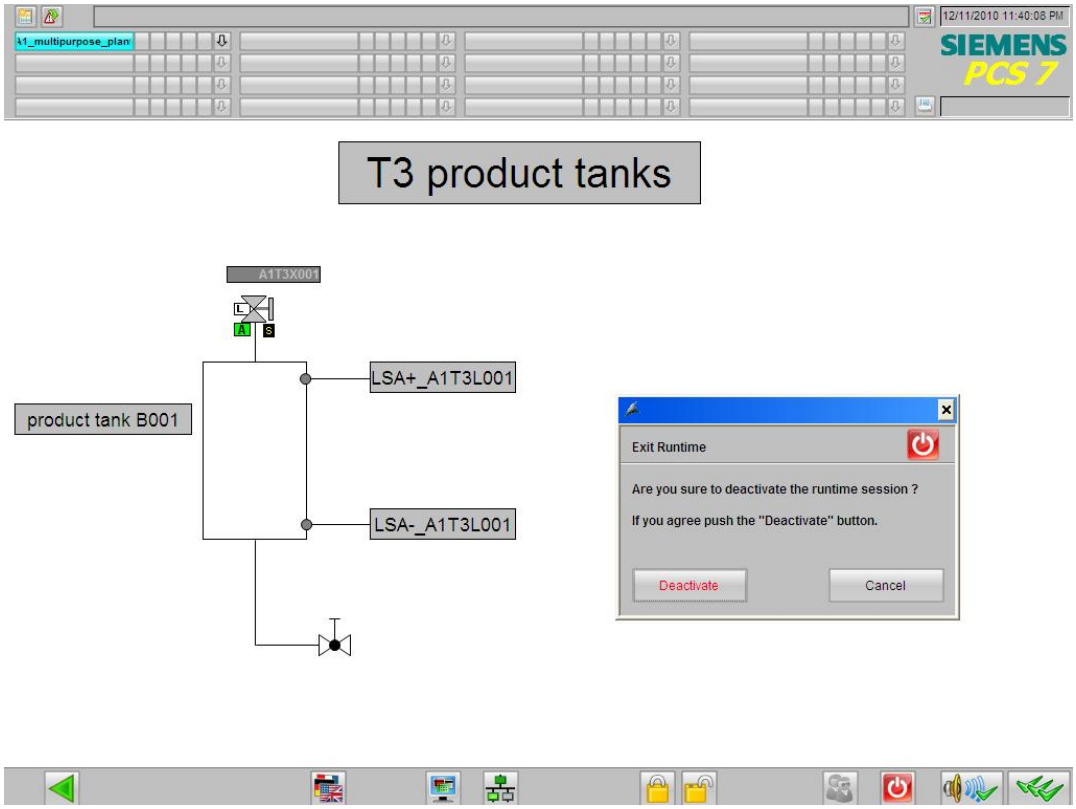


76. Here, the operating screen for 'T3_product_tanks' is displayed with the faceplate for 'Valve_A1T3X001'.



77. To exit runtime, press the button  .

(→  → Deactivate)



EXERCISES

We are going to apply to the exercises what we learned in the theory chapters and the step by step instructions. To this end, we are utilizing the already existing multi-project in the step by step instructions (PCS7_SCE_0201_R1009.zip) and expand it.

In the step by step instructions, again only one plant line was implemented. For that reason, the objective of the exercise is to implement the missing line. The tasks are designed to be an aid for generating all needed plant parts in the configuring tool ***WinCC***.

TASKS

The tasks below are based on the step by step instructions. For each exercise, the corresponding steps in the instructions can be utilized as an aid. Regarding the arrangement, the VDI3699 [1] rules have to be noted.

1. Educt containers A1T1B001 and A1T1B002 are to be implemented in addition to container A1T1B003 in the picture 'T1_educt_tanks.pdl'. The pumps and valves needed for this are already available and can be used directly for the new plant parts. The containers should be taken from the library so that their representation does not differ from the container generated initially.
2. Reactor R002 is to be implemented in addition to Reactor R001 that already exists in the picture 'T2_Reaction.pdl'. Also, the connection from the outlet of Reactor R002 to the inlet of Reactor R001 is to be displayed.